

若为自由故

自由软件之父理查德·斯托曼传

FREE AS IN FREEDOM RICHARD STALLMAN'S CRUSADE FOR FREE SOFTWARE



[美]SAM WILLIAMS 著

邓楠 李凡希 译

 人民邮电出版社
POSTS & TELECOM PRESS

若为自由故 自由软件之父理查德·斯托曼传

理查德是第一位奋起反抗的人。如今，他的抗争演变成了更大规模的行动。他指出了软件专利背后的荒谬逻辑：哪怕只有一行的代码，仅仅因为这行代码的作者是第一个想到这代码的人，就可以宣称对这种方法的完全所有权。当年，理查德曾孤一人为我们战斗。他曾警告过我们，软件业现有的知识产权模式不仅不会帮助从业者，反而会让开发人员倍受折磨。如今，软件专利肆虐，已不再是推动行业进步的动力，而这一切，早就被理查德言中了。

——蒂姆·伯纳斯-李，万维网创始人，万维网协会执行官

理查德发展了一套严密的哲学体系。这套哲学让我们不得不重新思考很多事情：何为软件开发，何为知识产权，软件开发社区又应该代表谁的利益？

——埃德·舍恩伯格，纽约大学计算机系教授

有谁会没有任何报酬的情况下来做这些专业工作？什么样的爱好者可以为他的产品投入三年年的精力去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？

——劳伦斯·莱西希，斯坦福大学法律系教授，《Code and Other Laws of Cyberspace》一书的作者在理查德那里，我有生以来第一次看到了那种典型的长发络腮胡的黑客形象。这样的人在赫尔辛基并不多见。

——林纳斯·托瓦兹，Linux内核的创始人，1990年第一次见到斯托曼时对他的评价

这是一本期待已久的传记，它要描述的是一位传奇人物。他凭借一人之力，改变了整个世界对技术的看法。

——罗伯特·杨，红帽公司联合创始人

斯托曼坚持自由，但其实一点不反对商业。他关注的核心是技术背后的潜规则是否合理公正，他开创了自由软件的体系，直接导致开源软件的兴起，成为互联网蓬勃发展的基石。

——蒋涛，CSDN/极客邦基金创始人

RMS为软件行业树立了道德基础：鼓励人们学习、交流、分享和互助；提供了技术方案：GNU操作系统；法律框架：copyleft，GNU GPL/AGPL/LGPL/FDL。这一系列开创性的工作是互联网/移动互联网创新、捍卫信息时代人类自由的基石之一。而这一切始于20世纪70年代，感谢RMS的远见和坚持！

——徐继哲，北京自造星球科技有限公司创始人&首席执行官

封面设计：Edie Freedman, Hanna Dyer, Emma Colby, 张健



O'Reilly Media, Inc. 授权人民邮电出版社出版

此简体中文版仅限于中国大陆（不包含中国香港、澳门特别行政区和中国台湾地区）销售发行
This Authorized Edition for sale only in the territory of People's Republic of China
(excluding Hong Kong, Macao and Taiwan)

分类建议：计算机/IT人文

人民邮电出版社网址：www.ptpress.com.cn

ISBN 978-7-11



ISBN 978-7-115-36951-2

定价：49.00 元

O'REILLY®

若为自由故——自由软件之父 理查德·斯托曼传

[美] Sam Williams 著
邓楠 李凡希 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

若为自由故 : 自由软件之父理查德·斯托曼传 /
(美) 威廉姆斯 (Williams, S.) 著 ; 邓楠, 李凡希译
— 北京 : 人民邮电出版社, 2015. 4
ISBN 978-7-115-36951-2

I. ①若… II. ①威… ②邓… ③李… III. ①斯托曼
, R. M. — 传记②软件开发 IV. ①K837.126.1
②TP311.52

中国版本图书馆CIP数据核字 (2015) 第002143号

版权声明

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2015. Authorized translation of the English edition, 2002 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Copyright© 2002 by O'Reilly Media, Inc.

All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 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."

本书中文简体版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。

版权所有, 侵权必究。

-
- ◆ 著 [美] Sam Williams
 - 译 邓楠 李凡希
 - 责任编辑 陈冀康
 - 责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷
 - ◆ 开本: 720×960 1/16
印张: 16.25
字数: 231 千字 2015 年 4 月第 1 版
印数: 1-3 000 册 2015 年 4 月河北第 1 次印刷
著作权合同登记号 图字: 01-2013-3998 号

定价: 49.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315

内 容 提 要

理查德·斯托曼是自由软件之父，他是自由软件运动的精神领袖、GNU 计划以及自由软件基金会的创立者。

本书用叙事体的风格，讲述了理查德·斯托曼的成长经历和传奇人生。全书共 14 章，记录了斯托曼从一名黑客成为自由软件的倡导者，并发起自由软件运动的成长经历。作者通过接触斯托曼本人、他的家庭、追随他的程序高手以及 IT 产业界，以全新的视角和丰富的笔触，塑造了这位软件产业界的“罗宾汉”的形象由此也帮助读者了解他所倡导的自由软件运动的缘起、理念和发展历史。

本书是了解黑客文化、自由软件运动以及斯托曼的一本轻松读物，适合任何 IT 行业相关人士、程序员以及 Linux 和 UNIX 爱好者阅读。

译者序一 若为自由故

“自由”这个词，有人崇尚，有人厌恶，还有人打心眼里对它咬牙切齿。而还有那么一小撮人，绞尽脑汁地要诉说着什么是自由，该如何保卫自由。而理查德·斯托曼（Richard Stallman）是这样一个另类：他定义了“软件用户的自由”，并且身体力行地为自己的自由理念宣传，在计算机技术领域掀起了一场“自由”运动。

瞎折腾——恐怕不少人会如此评价斯托曼。

如今，计算机与网络的普及改变了人们的生活方式。然而，技术的便利却没能带来意想中的地球村。用户面前不是一个信息流畅的互联网络，而是军阀割据的战乱天地。在利益与专制面前，自由再一次沦陷。费劲周折，人们重拾起自由这面大旗。此时才发现，斯托曼早已在这个战场上为大家奋战多年。

在我看来，理查德·斯托曼注定会成为一位俄狄浦斯式的悲剧英雄。他固执，有原则，拒绝妥协。而这也就注定了他难以如愿。他追求着一份乌托邦式的理想，也恰恰是这个理想本身铸就了他的几分睿智、几分悲凉。作为自由软件运动的发起人，他大可从容地把同根通源的“开源运动”视作晚辈后生，在当今众多公司投身开源的热潮里，摘取本该属于他的果实。然而，他却认为开源运动强调的是软件开发模式，忽略了软件用户的自由。尽管绝大部分开源软件许可证也是自

由软件许可证，尽管没几个人真的在乎“自由软件”和“开源软件”究竟有什么区别，他还是拒绝与开源运动有任何瓜葛。人们诉说着自由软件的成果，分享着 GNU/Linux、Apache 基金会的成功故事，斯托曼则在旁边不断泼冷水：自由软件运动远没有结束，我们离目标依旧很远。

正如科学永远要与误解相伴一样，自由也总有被私利与强权侵蚀的可能。追求自由如同追求真理，永无止境。

这本斯托曼的传记为人们描绘了一位计算机界的奇才。然而，与那些硅谷创业故事不同，这里没有一夜暴富的上市传奇，没有商场战场的尔虞我诈，更不会看到主人公的圆满谢幕。如今的理查德·斯托曼，依旧参与着自由软件的开发，依旧在世界各地演讲，宣传他的自由软件运动；也依旧会遭受人们的误解与不屑，被视作异类。斯托曼的执着与固执令他难见成功，也成就了他的传奇。

本书写成于 21 世纪初期，书中描写的一些环境当下早已不再：如今，自由/开源社区蓬勃发展，甚至硅谷的各大公司也纷纷抛来橄榄枝。自由软件与开源软件早已不是商业公司的敌人，反而是众多企业的得力助手，甚至是利润来源。哪怕是曾经自由软件运动的宿敌——微软公司，也投身到自由/开源软件的开发中，甚至公开了自家公司产品的代码。这一切的发展，都始于斯托曼当年的一份宣言。而其中原委，只有读过之后才能品味。

本书原题为 *Free as in Freedom*。本意是强调 Free 一词中的“自由”意项。对应的中文翻译则让我们犯了难。为此，我们曾在 GNU 官方网站的中文翻译组征求意见，最终选定了半句匈牙利诗人裴多菲·山多尔的“若为自由顾”来作为本书的标题。裴多菲曾领导了 1848 年爆发的匈牙利革命，并为此献出了自己 26 岁的生命。尽管斯托曼不一定会赞同裴多菲的政治理念，但那份为了争取自由而拼进全力的

执着，却是两位领袖的共通之处。

非常荣幸能与凡希合译本书。他翻译了第9章以后的内容，并完成了绝大部分的校对工作。是他的热情，让我这个习惯拖延的人追上进度。感谢人民邮电出版社陈冀康编辑的帮助，是他的耐心才让这本书最终得以出版。也感谢哲思自由软件社区，特别是徐继哲先生对本书的支持。

与很多翻译作品一样，本书也难免存在疏漏。但作为一本“自由”图书，读者可以参与改进，我们也会根据读者的建议更新本书的电子版。

邓楠

2014年5月于山景城

译者序二 自由自在

1999年，我第一次尝试安装 GNU/Linux，这也应该是我第一次接触“自由软件”。这次尝试的结果就是让我损失了硬盘上积累的自从1994年第一次接触电脑以来所有的资料。虽然遭遇了这样的不幸，却仍然磨灭不了我的好奇心。1999年到2006年期间，我屡战屡败地安装过各种各样的 GNU/Linux 发行版，但最终还是选择了 GNU/Linux 作为我日常使用的操作系统。

2004年，我第一次接触 Emacs 编辑器。Emacs 实在是个难以上手的家伙，好不容易通过 Emacs 内置的教程学到一点皮毛后，我就决定尝试把这份教程翻译成中文，造福后人。我向 Emacs 的邮件列表发了一封热情洋溢的邮件，咨询是否可以由我来完成翻译工作。很快收到了两封回信，说 CVS 中已经有人完成了教程的中文翻译，会随着下一个版本的 Emacs 发布，并欢迎我对现有的译文进行审阅。

在继续学习 Emacs 的过程中，我终于认识了 GNU 工程，也才真正明白了什么是“自由软件”。后来我加入了 GNU 官方网站的中文翻译小组，在翻译 GNU 网站的各种页面和文章的过程中，对自由软件、GPL 以及自由软件运动发起人理查德·斯托曼有了进一步的认识。

2008年5月24日，我参加了在上海复旦大学举行的“哲思自由软件峰会”，第一次与斯托曼有了面对面的交流。除了他所使用的

OLPC XO-1 笔记本电脑让我垂涎欲滴外，他在现场所表现出来的强烈个性更是给我留下了深刻的印象。会后，我请他在某本图书上签名的请求被他毫不留情地拒绝了，“这不是一本自由图书，我不能在上面签名。”斯托曼的解释非常干脆。

2011 年的某一天，我在无意中发现，七年前回复我有关 Emacs 教程翻译问题的人之一就是斯托曼本人。我与斯托曼的这两次直接交流，得到了截然不同的反馈：对于向自由软件社区贡献自己力量的行为，斯托曼会给予热情的支持和鼓励；对于非自由的软件和图书作品，斯托曼的抵触毫不留情。

在 GNU 官方网站的中文翻译小组中，我和邓楠一度是比较活跃的两成员。作为翻译小组的协调员，邓楠需要负责所有翻译成果的发布工作。我们的合作非常愉快，在 2010 年左右完成了 GNU 网站大部分二级页面的翻译工作。

所以，当邓楠邀请我跟他一起完成本书的翻译工作时，我马上就答应了。虽然这只是一本薄薄的小册子，但是对于我这样习惯于技术言语的人来说，要把这么多词句组织得不是那么干涩，还是有着不小的挑战。中间的曲折故事暂且不表，经过近一年的努力，我们终于还是完成了这本书的翻译工作。

交稿前夕，正好斯托曼要来中国进行几场巡回演讲，我有幸在杭州与他进行了一次更深层次的交流。谈到本书的出版工作，斯托曼本人表现出了极大的兴趣，甚至希望我们能立刻把这本书印出来在他接下来的巡回演讲中进行分发。同时，他也迫切地希望我们能够继续翻译出版他自己修订过的本书第二版的中文版。

由于本书英文版出版时间较早，书中所引用的很多网址如今都已经失效。不过在翻译过程中我们还是保留了很多失效的网址，因为即使对应的网页现在已经无法正常访问，但从网址本身也许仍可以窥探

出一些有用的信息。

感谢人民邮电出版社陈冀康编辑在本书出版过程给予的帮助，正如本书作者在跋中所写的那样，在传统出版领域出版这样一本“自由”图书还是面临着不少的挑战。陈冀康编辑的耐心和执着，最终促成了本书。感谢好友贾征主动帮忙审阅了全书的内容，也感谢家人对我翻译工作的支持。

尤其值得一提的是，这本书使用的是 GFDL 许可证，也就是说，这是市面上极为少有的一本“自由”图书。在 GFDL 条款的保护下，读者可以自由地复制、分发和修改本书。我们已经把全书的 LaTeX 源文件公开发布到网上，希望通过我们的努力，抛砖引玉，让更多的人可以参与到了了解、完善和传播本书内容的活动中。请访问 <http://www.freemindworld.com/faiif> 获取本书的电子版本，并欢迎读者参与到本书第二版的翻译工作中来。通过读者与社区的共同努力，希望能在第一版的基础上进一步精雕细琢，完成一部至臻完美的作品。今后如果你有机会见到斯托曼本人，不妨带上本书，他一定会非常乐意在这本记录他人人生故事的“自由图书”上签名的。

李凡希

2014 年 5 月于杭州

前 言

理查德·M·斯托曼 (Richard M. Stallman) 的作品本身就为自己代言。从源代码中的文档到公开发表的论文和演讲，很少人能像斯托曼那样把自己的想法和作品毫无保留地公开给大众。

这样的“公开”让人耳目一新——请原谅我用一个与斯托曼特质格格不入的词来形容他的话。不论如何，我们仍然生活在一种把信息尤其是个人信息当成是一种无价的物品的社会中。人们不禁要问，为什么有人会愿意公开如此多的信息，却并不求什么回报？

我们会在后续的章节中看到，斯托曼并不是完全无私地公开他的言论和作品。他发表的每一个程序、演讲或有记录的格言都有标价，即使不是一般人所熟悉的那种需要用货币来支付的价格。

把丑话说在前面不是作为一种警告，而是一种坦诚。作为一个花费了去年一年时间发掘有关斯托曼个人历史的人，对他的作品做出这样的评价必须得拿出一些勇气。马克·吐温 (Mark Twain) 有一句格言：不要为了不实报道跟媒体人争吵。如果用在斯托曼身上，就是永远不要给一个把自己一切想法都公诸于众的人写权威传记。

对于那些愿意花几个小时来阅读本书的读者，我会很有信心地说，本书中有些史实和引文是你们在 Slashdot 或 Google 搜索中无法

找到的。当然，你需要为获得这些信息支付一定的费用。对于纸质书的读者，你的购买行为就已经支付了这些费用。对于电子书的版本，你可以用支持自由软件相似的方式来“付费”。感谢 O'Reilly & Associates 公司的同仁，得以让本书以 GNU 自由文档许可证 (GFDL) 的形式发行，这意味着你可以修改本书，或者发布一个你自己改编过的版本，并以相同的许可证来发布。

如果你正在阅读的是电子版，并愿意以支持自由软件的形式对本书做出贡献，也就是说，如果你愿意对本书进行改进或把它推荐给更多的读者，我会非常感激。从 2002 年 6 月开始，我会在在线商店中发布一个本书的纯 HTML 的版本——<http://www.faifzilla.org>。我的目标是定期对它进行更新，并随着时间的流逝把 Free as in Freedom 的故事继续延续下去。如果你选择后一种形式，请阅读本书的附录 C，这是一份完整的 GNU 自由文档许可证，里面描述了你所享有的权利。

对于只是想坐下来安安静静阅读本书的人，即使你选择的是在线电版本而不是纸质书，我仍把你们对本书的关注看成与付费购买纸质书一样是对本书的支持。对此，你无须感到惊讶，因为你是采用了一种不同的方式来表达了你对本书的美好祝愿。

最后，我想说明，这只是一本人物传记，但它也是一本技术文档。在撰写和编辑本书的过程中，我和编辑权衡了很多这个故事中的关联人物所提供建议和素材，包括理查德·斯托曼本人。我们意识到在这个故事中有很多的技术细节，需要由更多素材的支撑才能表达得更为清晰。由于本书以 GFDL 发布，因此我们也欢迎读者向给自由软件打补丁那样为本书打补丁。所有被我们接受的改动都会整合到本书的电子版中并最终可能会整合到本书以后的印刷版本中。如果你愿意为本书提供改进意见，请发邮件到 sam@inow.com。

建议与问题

请把你的建议和问题寄送给本书的出版商：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)
奥莱利技术咨询 (北京) 有限公司

我们为本书提供了一个网站，上面有本书的勘误表、示例和其他相关的资料。这个页面上还提供了与作者和其他读者进行交流的网站链接。你可以通过下面的网址来访问这个网站：

<http://www.oreilly.com/catalog/freedom/>

如果有任何的建议或技术问题需要咨询，请发电子邮件到：

bookquestions@oreilly.com

有关 O'Reilly 相关的图书、会议、资源及 O'Reilly Network 的相关信息，请访问 O'Reilly 的网站：

<http://www.oreilly.com>

致谢

由衷感谢 Henning Gutmann 对本书的关注，由衷感谢 Aaron Oas 在创作初期给 Tracy 的建议。感谢 Laurie Petrycki、Jeffrey Holcomb 和其他 O'Reilly & Associates 的同仁。感谢 Tim O'Reilly 对本书的支持。感谢所有参与审阅本书初稿的朋友：Bruce Perens、Eric Raymond、Eric Allman、Jon Orwant、Julie 和 Gerald Jay Sussman、Hal Abelson

和 Guy Steele。希望你们能喜欢这本没有错别字的书。感谢 Alice Lippman 提供访谈内容、花絮和照片。感谢我的家庭成员 Steve、Jane、Tish 和 Dave。最后，感谢理查德·斯托曼的勇气和忍耐力，让我们去了解有关他的一切。

——Sam Williams

1000 Graystone Highway West

Sausalito, CA 94965

电话：

旧金山湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

<http://www.williams.com/eric/williams>

湾区最好的书店：湾区最好的书店：湾区最好的书店

eric@williams.com

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

<http://www.williams.com>

电话：

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

湾区最好的书店：湾区最好的书店：湾区最好的书店

目 录

第 1 章	从一台打印机说起	1
第 2 章	黑客路漫漫	13
第 3 章	黑客正年少	25
第 4 章	逆天行道, 弹劾上帝	38
第 5 章	自由一隅	62
第 6 章	Emacs 公社	82
第 7 章	道分左右, 义无旁支	97
第 8 章	ST. IGNUCIUS	119
第 9 章	GNU 通用公共许可证	134
第 10 章	GNU/Linux	157
第 11 章	开源	171
第 12 章	开往黑客地狱的短暂旅途	187
第 13 章	斗争仍在继续	193
第 14 章	跋: 粉碎孤独	204
附录 A	术语	222
附录 B	黑客的三层含义	224
附录 C	GNU Free Documentation License	231

第 1 章

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）刚刚发现了这问题，这可苦了他了。他当年 27 岁，是麻省理工学院人工智能实验室的一名程序员。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，之后就忙着干活去了。现在他正站在那台打印机前面，眼巴巴地看着它被纸卡住：总共才打印了的四页纸，还没有一页是自己的。换句话说，斯托曼那份 50 页的文件，外加别人的半份文档，全被一

张纸卡住了，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。不过斯托曼还是能苦中作乐。可再怎么讲，等着机器干活，和在它跟前盯着它干活毕竟是两码事。就说现在吧，斯托曼正盯着打印机，看它一页一页慢吞吞地往外吐纸。这情景对他来说可不陌生。斯托曼好歹也是个程序员，成天都在改进机器和程序，提高它们的效率。当下，他恨不得掀开这台打印机，仔细查查究竟是哪里出的毛病。

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些时间想点别的：能不能用别的办法，绕道解决这个卡纸的问题？

这台打印机来人工智能实验室有多久了？斯托曼努力回忆着。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文件，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。平均大约一秒打印一页纸。以前要花二十多分钟打印的东西，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，它绝不会打出个鸡蛋；让它画条直线，它也绝不会拐着弯瞎划拉。

怎么看，这打印机都是份难以拒绝的礼物。

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诱因。这台打印机是在复印机的基础上设计的。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。用工程师的行话讲，这系统需要用户参与。所以，对于复印机来说，卡纸问题也就没人关注了。设计复印机的时候，施乐公司的工程师们都没把卡纸的事情放在心上，专心去对付其他问题了。可把类似的机械部件用在打印机上，问题就来了。

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改变。现在，在同一时刻，这机器不再只服务一个用户。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地站在机器前面，而是坐在网络的另一端，隔着老远，向这台机器发布打印命令，指望它能按要求完成任务。等这些用户回过神，想起去取打印的文件时才会发现，大半的内容都被一页纸卡住出不来了。

在这台打印机之前，斯托曼也遇到过卡纸问题，他也是实验室里第一个提出解决方案的人。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。不过他能修改控制程序，让它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。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治标不治本，但也算讨巧。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 10~15 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，那些顶尖程序员不屑于用“程序员”这词。他们更喜欢用圈子里的行话，自称“黑客”。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，有这样一条铁律：从头开发一个软件只是小儿科；改进一个程序才显真本事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，以及配套的软件捐赠到黑客聚集的地方。要是这些黑客们改进了其中的软件，这些公司就可以把这些修改拿过来，化为己用，投入市场。用商业术语讲，这叫优势社会资产。公司花小钱，就能有个附属研发部门。

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0和1。除非他们有闲功夫，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。

尽管斯托曼可谓是精通计算机，但还是没那份闲心和时间去解释这些二进制文件。不过作为一名黑客，他总是有法子。他知道，黑客这个圈子里，分享信息是至关重要的。过不了多久，就会有黑客，从大学实验室或者哪个公司的机房里，拿出这激光打印机控制软件的源代码，分享给圈子里的同胞。

经历了几次卡纸事件之后，斯托曼还耐着性子安慰自己。不管怎么说，当初也遇到过类似情况的。几年前，实验室需要一个网络程序，让那台PDP-11和PDP-10之间协调工作。实验室里这群黑客自然能胜任这项任务。不过，作为一名哈佛毕业的校友，斯托曼记起了当初在哈佛计算机系，有个黑客也写过类似的程序。就运行在哈佛的一台PDP-10上。那台PDP-10，和当下人工智能实验室这台一样，只是运行了不同的操作系统。哈佛的机房里有规定，要求所有安装在PDP-10上的程序，都必须发布源代码。

作为校友，斯托曼还可以进哈佛的机房。他就大模大样地走进去，复制出源代码，带回了麻省理工的人工智能实验室。紧接着，他改写了程序，让它能在人工实验室的机器上运行。不费吹灰之力，人工智能实验室的软件设施又进步了一截。斯托曼甚至还在哈佛的软件基础上，增加了点新功能。“后来这个程序我们用了好几年。”斯托曼回忆道。

对于20世纪70年代的程序员来说，把软件复制来复制去，就好比从邻居家借碗酱油那样稀松平常。所不同的是，你从邻居那拿了多少酱油，邻居就少了多少。而从别人那里复制一个程序出来，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复制斯托曼改进的程序。这也算礼尚往来了。在哈佛那边，后来确实有个从 Bolt, Beranek & Newman 这家公司来的程序员，复制走了斯托曼的程序。之后还又加了点功能，斯托曼则又把改进后的程序复制了回来，用回到人工智能实验室里。

“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，也许会感慨这块从风格看应该是 20 世纪 60 年代早期的；之后的部分估计是 70 年代中期完成的。”

圈子里的这种氛围简单朴实，它让知识逐渐积累。这种积累为以后的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，都得益于这个氛围。比如在美国的西海岸，一群加州大学伯克利分校的计算机科学家和 AT&T 的工程师合作，遵循着如此积累知识的理念，创造了一整套操作系统。他们把这操作系统叫作 UNIX。UNIX 这个名字来源于另一个操作系统的名字——Multics，一个早期的学术研究项目。倘若是当初的程序员，只要愿意支付成本价和运输费，就可以买到一份 UNIX 系统的副本。整套系统会被复制到一个全新的磁带上，完整地寄到购买者手里，谁都能买，童叟无欺。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，绝大多数人都有着斯托曼那般的情结：要是一个程序或者一个补丁可以解决自己的问题，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。

一开始，施乐公司不分享源代码的行为倒也没惹到谁。斯托曼在找源代码的过程中，甚至都没想过去联系施乐公司：“人家都送咱打印机了，何必还非得兴师动众地管他们要代码呢？”

可是眼巴巴地等着源代码，如今还没找到，倒是让斯托曼起了疑心，勾起了他几年前一段不快的回忆。当初，一位在卡耐基梅隆大学的计算机博士，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。这软件可以让用户通过网络传递文档并且还支持自定义的字体，这在当时可是首例。之后这种功能的普及则带来了 HTML——如今 WWW 网络上的通用语言。1979 年，布莱恩决定把 Scribe 出售给一家在匹兹堡的公司，名叫 Unilogic。他当时刚好博士毕业，正打算把开发 Scribe 的任务转手给别人，免得它流落到公有领域里。为了让这份订单更吸引人，布莱恩在 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。

对于布莱恩来说，这个买卖是个双赢策略。一方面，正如布莱恩所期望的那样，Scribe 的代码的确没有流落到公有领域；另一方面，Unilogic 也有了一个向用户收费的途径。可这对斯托曼来说，这份交易完全是背叛了程序员的良心。布莱恩非但没遵循分享再传播的精神，而且竟然还助纣为虐，帮助公司去强制限制用户自由。

一周过去了，依然没有施乐激光打印机源代码的一丁点消息。斯托曼觉得这好像又是一笔索财要钱的勾当。不过还没等他给这事定性，好消息就来了：一个在卡耐基梅隆大学，计算机系的教授刚刚从施乐公司离开。这位教授还在激光打印机部分工作，而且从收到的八卦消息看，他在卡耐基梅隆大学还在继续激光打印机方面的研究。

于是，斯托曼这才抛开之前的胡思乱想，打算下次去卡耐基梅隆大学的时候，去会见这位教授。

也没多久，他就得到了一次访问卡耐基梅隆大学的机会。和斯托曼所在的麻省理工学院一样，它们那里也有一个人工智能实验室。这才几个月不到，斯托曼就因为工作需要去卡耐基梅隆大学出差访

问。访问中，他在计算机系请人帮忙找到了那位曾在施乐公司领导激光打印机项目的教授。他正好就在办公室。

工程师之间的讨论总是开门见山的。几句简单寒暄过后，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，以便可以修改后用到麻省理工学院的 PDP-11 计算机上。可令斯托曼吃惊的是，这位教授竟然拒绝了他的要求。

“他跟我说他答应了公司不能泄露代码。”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的。他都不记得当初为了什么事去的卡耐基梅隆大学，连具体哪年去的都给忘了。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。20 世纪 70 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。他在 80 年代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。

“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，“我想斯托曼拜访斯布鲁那会，他刚刚到卡耐基梅隆大学还不到一个月。”

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：“我确实不记得有这么一件事。”

当事双方谁也不记得事情的细节，甚至都无法确定是否真的发生过。至于斯托曼说的，斯布鲁硬生生地拒绝了他的这件事，也就无从考证了。在之后的各种演讲中，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，

签署保密协议已经非常普遍。不过在当时，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，这种保密协议完全是另外一回事。那时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。

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，斯托曼也没把重点集中在施乐公司那边，而是怪罪到了个人性格上。作为一个标准技术宅男，斯托曼和人沟通起来难免有点障碍。他这次亲自跑去拜访一位程序员同行，已经是一种充满诚意的行为了。而到头来却被当头一棒。让他实在觉得对方有点鲁莽。“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。”

二十多年过去了，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，本能地对绝对权威存有戒心。在经历了这一系列事件后，他变成了一位斗士，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。

“它让我思考了一些脑海中由来已久的问题，”斯托曼说，“我以前有过一些初步的想法，认为软件本该共享。可当时还不知道怎么表

述。那时的想法还没有清晰到可以用简单几句话给别人介绍。”

尽管之前也有过类似的不快经历，可这次的打印机事件彻底让斯托曼意识到，这一系列的事件，正悄悄地侵蚀自己所珍视的文化——那个神圣不可侵犯的小圈子。作为一名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忍受的程序和机器。这些程序很少能影响到人工智能实验室，斯托曼和实验室的成员之前一直都可以自由地重写软件，添加功能。直到有一天，人工智能实验室把一台计算机的操作系统从不相容分时系统（Incompatible Time Sharing Operating System）换成了商业的 TOPS 20 系统，这些就再也不能做了。

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了变化。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，这台打印机的出现是个信号，预示着软件是公司的重要财产。谁也不会再想发布软件源代码，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，这台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站得一角。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。它不仅给了斯托曼足够的怒火去对以后的各种礼品心存戒备，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，他究竟会不会拒绝复制代码呢？

“这是我第一次碰上这种保密协议，它很快让我明白，保密协议面前，总会有无辜的受害者”，斯托曼坚定地说，“在打印机事件中，我和整个人工智能实验室扮演了受害者的角色。”

斯托曼带着这种态度，经历了动荡的20世纪80年代。在这期间，麻省理工学院的同事们纷纷离开实验室，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。而保证软件在早期的开发阶段不被泄露，则可以保证让各位黑客朋友们可以进入顶尖项目中工作。这些借口，在斯托曼看来，则是迈向深渊的第一步。

“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，并告诉他‘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，但会更圆滑地对它加以评判。正因为如此，斯托曼被誉为“最后的黑客”。这使他与专有软件市场渐行渐远。拒绝提供源代码，在斯托曼看来，不仅违背了第二次世界大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，倘若没有这次的事件，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，更不会去解决别人从未想过的各种问题。最重要的，他心中也不再有那份不平，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。”

尾注

1. 要深入了解“黑客”这个词，参阅附录 B。

第2章

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：一个安检处拦在了整座大楼的唯一一道入口前面。

走过安检口，气氛似乎缓和了些。可是，依旧能看到各处林立的警示牌指向消防出口。一路下来，可能会有这样的感受：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。

当继续向前，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。他们斜躺在椅子上，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）格雷斯·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，他也是人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（Craig Mundie）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——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：用户可以自由地使用、学习、修改和再发布软件。如今，GPL 已经俨然成为了自由软件社区的核心工具。简单来说，GPL 是一个软

件使用许可证，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，因为它像病毒一样，“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。

在黑客这个圈子里，这个操作系统也被当作战略武器，用来对抗微软公司。这个总部坐落于华盛顿州雷德蒙德的软件公司，从 20 世纪 80 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。在 Windows 中，几乎每行代码都是专有的。至少按照法律条款看，它归微软所有。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地复制、修改这个系统，并销售修改后的版本，这可能会瞬间颠覆微软公司龙头老大的地位。

对软件业来说，二十年的时间可是个不短的年头。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，则是之后的事情了。而当今人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。如今的很多计算机巨头，在当时也一样还没建立。包括 AOL、Sun、亚马逊、康柏、戴尔等。

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²。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专有化，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）³，公司也将失去继续创新的动力。正如微软副总裁蒙迪于 5 月 3 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。”⁴

在过去的十年中，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专有软件哪个更为成功，而是哪个更道德。不管怎么说，这场自由软件与专有软件的战役在软件工业发展史上是一次重要的事件。哪怕像微软这样强大的企业，也需要第三方软件开发者。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⁴。

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 恤和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 15:1。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。



图一 理查德·M·斯托曼，大约摄于 2000 年。“我决定要开发一套自由的操作系统，哪怕这要花费我一生的时间。”照片来自 <http://www.stallman.org>。

人潮涌动，斯托曼则离开了演讲台，坐在第一排的听众席上，把一直运行着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（Mike Uretsky）。

“大学之地，争辩之所；争辩之处，兴趣所在。”乌列茨基说，“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（Edmond Schonberg）。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。”

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，斯托曼的表现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。复制软件则要求更少，只要轻点鼠标，费点电。而分享之后，起码有两份收获：增进了友谊；同时，下次需要帮忙的时候，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。在这样的背景之下，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，斯托曼轻车熟路。他好似律师在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。”

在这二十年里，理查德·马修·斯托曼，从当年一个不太得志的学者，成为了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观和远见。这之中，也当然包含了斯托曼编写的高质量代码，字里行间都凝结了斯托曼的心血，将他的经历铸成了计算机界的传奇。传奇之上，更能看到 GPL 不可遏制的蓬勃生命力。而作为法律界的一大创新，GPL 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。斯托曼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⁵。面对这一项目，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，虽然 GNU 工程有过几次跳票延期，但在大多数时候还能按时交付，完成一个又一个发布计划。踏过了十八个寒暑，GNU 工程也日渐成熟，在软件市场中赢得一席之地。而这近二十年里，自由软件的理想被斯托曼精心呵护，渐渐传遍大江南北。

要想知道这潮流背后的缘由，就要兼听八方言论。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。’”

尾注

1. 事实上，GPL 的没有这么夸张的“威力”。把程序与另一个 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. 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/articles/ghosh-2000-01-p1.html>
3. 其实杀手级应用并不一定需要是专有化的。比如说，传奇的 Mosaic 浏览器，它的许可证就允许在一定的限制条件下对该程序进行非商业化的修改。读者应该这样来理解：软件市场就像买彩票一样，预期的收益越高，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>
4. 参考微软资深副总裁克雷格·蒙迪的 *The Commercial Software Model* 一文，节选自他 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/05-03sharesource.asp>
5. 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。我们也有一种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”）。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，因此它也一点也不‘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>

第3章

黑客正年少

理查德·斯托曼的母亲，爱丽丝·李普曼（Alice Lippman）始终还记得她意识到斯托曼天赋的那一刻。

“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，做了单身妈妈。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》。李普曼一个教艺术的代课老师，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待着的儿子，决定开始挑战本周的思考题。

“其实，这些题目我做得不是很好，”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。”

可她刚一开始做，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。在餐厅的橱柜里，摆放着一张8英寸×10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·斯托曼这么一个孩子，李普曼总是有很多话可说。每当别人说起她儿子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，”她说着，把双臂铺开，以示强调，“曾经就在这张桌子前，我们俩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？’”

李普曼 1948 年和理查德的父亲丹尼尔·斯托曼（Daniel Stallman）结婚，于 1958 年离异。李普曼成了单身妈妈，抚养和监护这孩子的重担几乎都压在了她的身上。一路走来，她见证了她这个儿子蔑视权威的个性，更见证了他对知识的如饥似渴。这两股力量在她儿子身上几经纠缠，也让李普曼和理查德·斯托曼有过不少摩擦冲突。

“他好像一直都不想吃饭一样，”李普曼回忆着儿子八岁到 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。很多原则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入学早了两年。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，老师就把妈妈约去学校，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在四年级的时候。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10~11岁那年，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20世纪60年代早期。那时，李普曼已经离异两年，也从皇后区搬了出来。斯托曼当时七岁，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，她觉得这个兴趣就和玩数学游戏一样。可有一次

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！’就好像真有点资料要给 NASA 看似的。”

这个小事倒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（Johnny Unitas）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¹。”

虽然斯托曼总比同龄人显得古怪很多，但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”²，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。

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³。在自闭症这样一个统称之下，理查德坦言，要是出生在当下，很可能就被拉去做检查了。当然，他的那群在计算机世界中的朋友，也一样跑不了。

“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。”

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此事。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应，“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。”

这几年，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，希望不要对他产生太大影响。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·斯托曼的父亲曾是参与过第二次世界大战的一名退伍老兵，于2001年去世。每当说起他，理查德·斯托曼总是对他又尊敬又恼火。一方面，这是一位有担当的士兵，他曾为了更好地帮助盟友，特地学

习法语；另一方面，他又是个总会把事情变得残酷不堪的父亲⁴。

“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：“我们俩之间可是战事频发。我常带着几分悲伤说‘我想家了’，其实我的意思是，所谓的‘家’对我来说就是个幻想中的空中楼阁。”

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。可在他 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的去世列为第二个沉重打击斯托曼的家庭变故。“他当时悲痛欲绝。他和几位老人关系特别好，老人们去世之前，理查德特别开朗外向，都属于远近一带的‘孩子王’的角色。他们一走，理查德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

和与众不同。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，斯托曼则钟情于古典乐。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。其中一位是他在 12 岁那年的科学夏令营的顾问。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⁵。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 20 世纪 60 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⁶。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。

之后不久，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中的学生，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 84 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课期间打断他讲课。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，‘对是对的，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 名学生中成绩排名第四。

STUDENT INFORMATION										TRANSCRIPT OF HIGH SCHOOL RECORD										SCHOOL INFORMATION									
NAME: WILLIAM RICHARD N...					ACADEMIC COUNSEL					SCHOOL: BRONXES D BRANCIS HS					ADDRESS: 145 WEST 24TH ST.					CITY: NEW YORK									
CLASS: 11					GRADE: 11					SCHOOL YEAR: 1968-69					SCHOOL ID: 237625					STATE: NY									
COURSES										COURSES										COURSES									
ENGLISH I										ENGLISH II										ENGLISH III									
MATH I										MATH II										MATH III									
SCIENCE I										SCIENCE II										SCIENCE III									
HISTORY I										HISTORY II										HISTORY III									
PHYSICS										CHEMISTRY										BIOLOGY									
FRENCH I										FRENCH II										FRENCH III									
SPANISH I										SPANISH II										SPANISH III									
ART I										ART II										ART III									
MUSIC I										MUSIC II										MUSIC III									
PE I										PE II										PE III									
GARDENING										CRAFTS										OTHER									
TOTAL CREDITS: 46										COURSES TAKEN: 46										COURSES PASSED: 46									
EXAMINATIONS: 96										EXAMINATIONS: 96										EXAMINATIONS: 96									
GPA: 3.4										GPA: 3.4										GPA: 3.4									
CLASS RANK: 4										CLASS RANK: 4										CLASS RANK: 4									
SAT: 1100										SAT: 1100										SAT: 1100									
ACT: 28										ACT: 28										ACT: 28									
AP: 1										AP: 1										AP: 1									
IB: 0										IB: 0										IB: 0									
TOTAL: 11										TOTAL: 11										TOTAL: 11									

图二 到了 1969 年 11 月，当斯托曼即将从布兰德斯高中毕业时，他的英语成绩已经相当出色了。李普曼回忆说：“他虽然很不情愿写作文，但他还是出色地完成了学业。”

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南战争抗议者的队伍，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。’”

尾注

1. 参见迈克尔·格罗斯（Michael Gross）1999年的访谈：*Richard Stallman: High School Misfit, Symbol of Free Software, MacArthur-certified Genius*，这篇访谈是我迄今为止看到过的有关斯托曼的最为中立的作品，向广大读者强烈推荐。
<http://www.mgross.com/interviews/stallman1.html>
2. 朱迪·丝蒂特（Judy Steed），《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（Glenn Gould）非常相似，他们都是非常聪明、善于表达的人，但是同时也都孤独。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他人打交道非常困难。
3. 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

4. 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·斯托曼已经去世了。
5. 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。参见本章尾注1中格劳斯在1999年所作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。”
6. 译注：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）。

indian-writings/11-People-to-be-not-warned-by-viewers/12011

第 4 章
逆天行道，弹劾上帝

第 4 章
逆天行道，弹劾上帝

第 4 章

逆天行道，弹劾上帝

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，20世纪50年代，斯托曼一家试图重拾第二次世界大战以前的那份宁静、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·斯托曼的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，20世纪50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，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，却发现关于离婚的只有一本。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。

重回童年居住过的纽约市曼哈顿上西城，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。为了建高楼，几个街区成了一个政治决斗场。人们分成了两派：一派人，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；另一派人，主要是来自爱尔兰或波多黎各的房客，他们都不富裕，所幸已经在这里找到了便宜舒适的住房。他们不想请来白领租户，因为那样会抬高当地房价和租金。

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身母亲，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，她希望为儿子找个日托所，以便在自己工作的时候，有人照顾他。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，是个爱干净的人。护士学校的环境和这个帮助中心的反差太大了。我们看了一眼帮助中心里日托房的房间，就立马走人了——环境实在太让人恶心。”

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，大家之所以看不起穷人，甚至仇视他们，党内腐败也许才是一切症结所在。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作斗争²。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的会议上，以此，她取得了更多的话语权。

李普曼说：“我们的主要目标是对抗以卡米思·德·萨皮奥（Carmine DeSapio）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 20 世纪 60 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（William Fitts Ryan），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南战争的议员。

不久之后，李普曼也开始公开反对美国涉足越南战争问题。“自从肯尼迪政府把军队送去越南，我就一直反对越南战争。从一开始，我就阅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这个家庭里。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，越南战争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（Andrew

Lippman)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20世纪70年代，理查德·斯托曼虽然年纪还小，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·斯托曼来说，越南战争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。斯托曼如此地厌恶权威，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。”斯托曼如是说。1971年，政府最终取消了学生延期入伍的政策。抽签征兵⁴的结果让他很是担心：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。”

斯托曼说，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。斯托曼很受父亲情绪的鼓舞，他说：“我很崇拜他这么做，”斯托曼坦言，“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。”

然而，斯托曼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，斯托曼每周末都会看到聚集在哥伦比亚大学的抗议群众⁶。最终，斯托曼形容说，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，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·斯托曼，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。可这也并不意味着就有会有个姑娘愿意与他一起共度春宵。

“我并不喜欢这种反抗文化，”斯托曼坦言，“我不喜欢他们的音乐，不喜欢毒品。我非常害怕毒品。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南战争，就意味着支持越南北方政权。他们就天真到想不出另外一种可能。”

这样的评论也许是在为斯托曼当时的怯懦找到些借口，但也标志着斯托曼自己政治思想的逐步成熟。对于他来说，政治上的信心和自信心成正比。到了1970年，斯托曼已经在数学和科学以外的一些领域里树立起了自信。当然，有在数学方面的信心和基础，让斯托曼可以从纯逻辑的角度分析越南战争。这种分析最终带来的结论让斯托曼觉得，尽管反对越南战争，但不可否认，战争在维护自由和捍卫正义方面起到了积极作用。然而，斯托曼也不想让自己在同龄人里过分特立独行，于是他关于战争的这些分析，也就一直留在自己肚子里了。

1970年，斯托曼离开了家，离开了每晚家中餐桌上的越南战争和政治讨论，去了哈佛大学。他回忆，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斯托曼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·斯托曼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，斯托曼轻松通过数学 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。“我们是数学党的，”柴斯笑道，“跟科学之星比，哈佛不在话下。”

可要修成正果，斯托曼、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。只有真正肯下工夫的人，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）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，他回忆道，“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，他记得当时斯托曼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。斯托曼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，斯托曼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，斯托曼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，他回忆道：“在一堂实分析的课上，斯托曼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。我第一次看到有人能自己想出来，并用这么漂亮的方法解决了这个问题。”

柴斯坦言：眼看着斯托曼在黑板上写下解答，他感到了前所未有的压力。中学时候，柴斯也被认为是班上的数学明星。而如今见识了斯托曼之后，让他觉得自己竟然如此渺小。几年之后，柴斯逐渐接受了这个现实：自己在数学方面虽有过人之处，但也算不得顶尖高手。回想起来，在斯托曼大二时，柴斯就已经从斯托曼的那次证明中隐约意识到自己的位置。

“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。”

我算不上一流的数学家。如果斯托曼当初选择做数学家的话，他肯定会是一流的。”

有得就必然有失，斯托曼在课堂上的得意，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，而斯托曼则自己一个人应付作业题。住宿问题上斯托曼也坚持如此。在哈佛大学住宿申请的表格上，斯托曼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，竟然接受了斯托曼的要求，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斯托曼同住一个宿舍楼的。他回忆，斯托曼当时的确慢慢地学着如何和别人沟通交流。他记得，当时其他几个宿舍的学生，都被斯托曼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。斯托曼总能比大家更早看出各种决策的优劣。”

斯托曼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。我很惊讶，不少人觉得长生不老并非什么好事情。我们要能看到 200 年之后，世界是什么样子，那会如何？”

尽管众人已经公认，斯托曼会是个一流数学家，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，斯托曼大一的时候，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，帕特南竞赛也是个残酷的竞技场。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 多分。就这分数，我还能排上第 101 名。”

可让人吃惊的是，斯托曼——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，几个人把斯托曼围在墙角，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写下了几个帕特南竞赛的题目，扔给斯托曼，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。”

斯托曼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，斯托曼承认，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。斯托曼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。

斯托曼说：“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。”

究竟是不是因为这种恐惧，才让斯托曼没能成为数学家，我们不得而知。但无论如何，大一结束的时候，斯托曼已经有了新的兴趣：编程。这兴趣从斯托曼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选修艺术类和历史课来放松，斯托曼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，诱使着斯托曼去了解更多。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，斯托曼开始仔细阅读，比较各个机器之间的差别和联系。并且，他写了一个测试程序，这样可以在把手册还回去的同时上机测试一下。在那个时候，尽管很多实验室并不允许随便让一个陌生的小孩子在他们的电脑上操作，不过对于斯托曼这样才华横溢的孩子，他们还是同意让他尝试运行自己的程序。

第一年快结束的时候，斯托曼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，里面有各种高级计算机和软件。

斯托曼被这一切吸引住了，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，斯托曼才发现，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，斯托曼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。

传言中说，人工智能实验室里的计算机对外开放，随便谁都可以坐到终端前摆弄计算机。不过，斯托曼还是觉得按照原计划行事比较妥当。斯托曼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。斯托曼回忆：“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。”

在对方的帮助下，斯托曼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⁸。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，斯托曼觉得，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。这实在太愚蠢了。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”，斯托曼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，斯托曼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，可每间办公室的房门都是紧锁的，却只能在那里按部就班地等着用那几台公共终端。在斯托曼看来，这实在暴殄天物。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。”

斯托曼很快了解到，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，都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。而这个小团体中的有些人，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⁹；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¹⁰。

这批小团体里的人把自己称作“黑客”。当然，在不久的将来，他们也会把斯托曼归类为“黑客”。在成为“黑客”的过程中，斯托曼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。斯托曼也逐渐学到，

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，三是自私自利的行为。

斯托曼在这里学到的东西里，还包括黑客们解决问题的各种新颖技巧和相关的轶事。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事。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。斯托曼说：“我当初还看到过一架手推车，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¹¹。”

这些方法明目张胆，它们在明确地传达着黑客的理念。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。实验室计算机上的有些硬件就是他们设计的；而几乎所有的软件，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。它被称作 ITS，全称是“不相容分时系统 (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，可最终还是归了人工智能的研究人员¹²。

尽管 ITS 会吓跑一些门外汉，但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。斯托曼来之前的那段时间，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¹²。

斯托曼回忆：“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。”

通过使用这个功能，斯托曼得以深入研究其他黑客所编写的程序是如何在计算机上工作的。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。

斯托曼说：“如果说，‘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某个状态给修改了。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。”

到了 1970 年年底，斯托曼每周都会定期跑去人工智能实验室。周一到周四，斯托曼会待在哈佛大学。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。

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，确实让斯托曼甚是陶醉。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，斯托曼回想当年，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，你会有种由内而外的舒适和满足感¹³。”

斯托曼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，斯托曼会毫不留情地当面指出。在麻省理工学院的第一年，斯托曼也曾想办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，斯托曼

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。

当斯托曼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，斯托曼开始另寻他法：他开始的想法是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。斯托曼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？

斯托曼自己开始尝试。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，把磁条有黏性的部分朝上。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，斯托曼又为黑客开门技巧做了贡献。

“有时候，门锁打开后你还得踹两脚才能把门弄开，”斯托曼回忆起这个方法的一点瑕疵，“拉磁条的时候你得小心点，一般是站在一把椅子上，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斯托曼期望表达自己的政治诉求。人工智能实验室的实干精神，让斯托曼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，斯托曼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。”

问起这条蛇的故事，斯托曼澄清了几点事实。根据斯托曼的回忆，那条蛇实际是拿去参加了卡瑞尔楼的学生干部竞选。那是斯托曼住的宿舍楼，并非全校范围内的学生竞选。斯托曼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，”斯托曼说，“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，我们好几个人竭尽全力也没能找到它。”

除了让蛇参加选举，斯托曼在竞选方面还开过很多玩笑。在另一次选举中，斯托曼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，斯托曼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，据说伍德斯托克在马路上被一辆车压死了，”斯托曼如此回忆道，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¹⁴。”

在人工智能实验室，斯托曼的政治活动更加积极，产生的矛盾也更尖锐。在20世纪70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。到了70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。

斯托曼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。

“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，”斯托曼事后解释，“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¹⁵。”

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录。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，破解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。这封邮件的内容大概是这样的：

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¹⁶。

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账号。这和以前没有密码的日子没什么分别。这就是斯托曼和黑客们要表达的：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。他们认为学院的计算机，甚至包括计算机里的文件都应该是公有的，而不是属于任何个人的财产。斯托曼在一次为 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¹⁷。

斯托曼的这种反密码抗议，以及各种拒绝安全保护的行为，最终还是要失败的。20 世纪 80 年代起，哪怕是人工智能实验室，也开始支持密码登录系统。可即便如此，斯托曼的行为标志着他个人政治思维成熟的里程碑。斯托曼迈出了一大步。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，他成为了一名活跃人士，把各种嬉笑怒骂当作家常便饭。

反对计算机安全这事体现了斯托曼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：对集体的责任感和信任；还有遇到问题，直接行动的黑客精神。用软件界的行话来说，反密码抗议行动体现的正是斯托曼 1.0 版。虽然有些地方还很不完善，但大部分已经成熟。

斯托曼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。只是我在这方面比较活跃。”

不过，斯托曼依旧很感激人工智能实验室氛围唤醒了作为政治活动家的潜质。曾经还是少年的时候，斯托曼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事，可花了很长的时间后，我才觉得这是一种道德责任。”

斯托曼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，斯托曼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。斯托曼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。这也给了他不少和女孩子搭讪的机会，可惜在大学期间，每次搭讪都没能换来单独的约会。跳舞让斯托曼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20世纪80年代早期，斯托曼走得更远了：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使他的社交能力提高多少，可刚到哈佛时给人的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。斯托曼很兴奋，他当场制作了一个徽章，上面写着“弹劾上帝”。

在斯托曼看来，所谓“弹劾上帝”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，斯托曼把“弹劾上帝”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。‘弹劾上帝’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，“弹劾上帝”也影射了 20 世纪 70 年代的水门事件。它把尼克松比作暴虐的天神。水门事件对斯托曼的影响很深。斯托曼从小就厌恶权威。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，斯托曼自信满满。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，从天而来，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。尔等世人，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¹⁸。”

如果只是觉得这段话是对水门事件的恶搞，就只听懂了一半。另外一半，恐怕只有那群黑客的同类能听懂。百年之前，阿克顿男爵（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：权力本身导致腐败。斯托曼并没有直接列举各种腐败案例，他直指过分信任权力，从而酿成腐败的整个社会。

斯托曼回忆起徽章的事情，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。”

尾注

1. 参见迈克尔·格劳斯的采访：《理查德·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。20世纪30年代因操纵选票丑闻而一蹶不振。但在50年代又有起色，60年代左右逐渐退出政治舞台。
3. 卡米思·德·萨皮奥曾是坦慕尼协会的第一个意大利裔主席。关于他和纽约第二次世界大战后的政治格局，参见约翰·达文波特（John Davenport）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年）：3：1。
4. 译注：抽签征兵，1969年~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. 柴斯（Chess）是哥伦比亚科学之星计划的另一个学生。他把当时的抗议游行形容为“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。因此，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. 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。
9. 译注：休伯特·德莱弗斯，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*。
10. 参见史蒂芬·李维 (Steven Levy), 《黑客》(1984 年, 美国企鹅出版社), 第 144 页。史蒂芬用了五页内容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 (John Conway) 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。
11. 杰拉尔德·萨斯曼, 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, 斯托曼还没来。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。
12. 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·加芬史尔 (Simon Garfinkel) 的书《信息社会的建筑师: 麻省理工学院计算机实验室的 35 年》(麻省理工学院出版社, 1999 年)。
13. 参见理查德·斯托曼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>。
14. 本书最后一审期间, 斯托曼发来邮件说, 他从哈佛大学也得到了很多政治上的鼓舞。斯托曼说: “我在大一的时候, 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。”

15. 参见上述的斯托曼在瑞典皇家技术研究所的演讲。
16. 参见史蒂芬·李维，《黑客》（1984年，美国企鹅出版社），第417页。在这里，我对邮件内容作了一些改动，在原书中李维也是从别处摘录了这段文字，用于证明斯托曼所写程序可以真正解密系统中的密码。李维在摘录时用了“某某”来代替实际的密码。
17. 参见史蒂芬·李维，《黑客》（1984年，美国企鹅出版社），第417页。
18. 译注：此处翻译以意译为主。原文为：“My name is Jehovah,...I have a secret plan to end injustice and suffering, but due to heavenly security reasons I can't tell you the workings of my plan. I see the big picture and you don't, and you know I'm good because I told you so. So put your faith in me and obey me without question. If you don't obey, that means you're evil, so I'll put you on my enemies list and throw you in a pit where the Infernal Revenue Service will audit your taxes every year for all eternity.” 斯托曼所说的地狱税务局（Infernal Revenue Service）和美国国家税务局（Internal Revenue Service）有同样的首字母缩写，IRS。属于常见的文字游戏。

第5章

自由一隅

如果谁和斯托曼在一起待上超过一分钟，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，是斯托曼凝视你的眼神。透过他绿色的眼睛，你能看到一种真实的信仰。

要说斯托曼眼神犀利，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，很多描写斯托曼的文章都会从信仰的角度开始写起¹。但是，这些报道都忽视了斯托曼人格中脆弱的一面。盯着

斯托曼的眼睛再久些，就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。正像斯托曼本人所怀疑的那样，如果他的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一个人的时候，那个人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，眼中传出几分绝望。

我第一次见识到传说中的斯托曼的目光，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斯托曼。斯托曼决心在这次见面会上简要回顾 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 的媒体发布会上被斯托曼亲自上了一课。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作 Linux，”斯托曼回应，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。这些软件并不都是林纳斯·托瓦兹开发的。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这个操作系统。如果不提及这些开发人员的工作，那么既不礼貌，也不尊重历史。所以我坚持把这个操作系统称作 GNU/Linux，也希望你这么称呼它。”

我把这段话记在采访本上，整个会场异常地安静。等到我抬起头来，才看到斯托曼的眼睛眨都不眨地盯着我，让我心生一丝胆怯。这

时候，直到另外一名记者正确地使用了 GNU/Linux 的名字提问，才打破僵局。GNOME 项目的组长，米格尔·德·伊卡萨（Miguel de Icaza）回答了这名记者的问题。伊卡萨回答到一半，斯托曼才把目光从我身上移开。这时，我才觉得些许放松。等到斯托曼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。

从斯托曼的角度看，这种面对面的时刻是有意义的。这次大会结束的时候，各路记者在斯托曼面前已经改用 GNU/Linux 这个名字。而《连线》的记者则开始忙着写一篇通讯。这篇通讯中，斯托曼被形容为被黑客和各大公司遗忘的英雄。甚至整个 GNU 工程也被忽略，因为这些黑客和公司不喜欢 GNU 工程中的政治目标。其他类似的报道也相继出现，尽管没有在报道中使用 GNU/Linux 的名字，但它们大多都提及了斯托曼的贡献，说到他十五年前发起的开发自由操作系统的项目。

再次见到斯托曼，是在 17 个月后。在这 17 个月中间，斯托曼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，但斯托曼此次还是有目的的。他代表自由软件基金会，领取了林纳斯·托瓦兹社区贡献奖。斯托曼接受奖杯的时候，打趣道：“这次代表自由软件基金会领取林纳斯·托瓦兹社区贡献奖，感觉就像唐僧代表师徒四人去领取悟空奖²。”

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³。这次上市，印证了大多数记者的猜测：“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，大家再也没看到斯托曼的身影。

我第二次再和斯托曼见面，是在第三届LinuxWorld大会之后不久。当时我听说斯托曼又要来硅谷，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。这个地点选得很有意思。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。而斯托曼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，斯托曼会作何反应。带着好奇心，我离开奥克兰市，驱车前往帕罗奥多。

按照斯托曼给我的地址，我来到了Art.net的总部。这是一家汇集各路“网上艺术家”的非营利组织。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，斯托曼混在硅谷，似乎也不是个很怪的想法。

我看到斯托曼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。互相寒暄后，他又把目光聚焦回笔记本的屏幕上。

“我刚刚写了一篇关于黑客精神的文章，”斯托曼一边打字，一边说，“过来看看？”

我走过去开始读文章。房间阴暗，屏幕上编辑软件的界面，又被斯托曼设置成了黑底绿字。我读了几个字，眼睛才逐渐适应。文章开始回忆了斯托曼有次去一家韩国餐馆的经历。当时服务员摆桌的时候，在斯托曼面前放了三双筷子。一般人恐怕都会把多余的两双筷子拿开。可斯托曼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。斯托曼以这个事情来作为整个文章的引子。

我读的时候，感觉斯托曼一直在盯着我。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里。我夸奖了一下他的文章，勉强让他的眉毛抬了抬。

“稍等一下，我们马上就出发。”他说。

斯托曼坐回座位，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，斯托曼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。斯托曼的一双大手，则在这个小键盘上飞舞。80年代后期，斯托曼每周都工作 70~80 小时，编写 GNU 工程中最基础的那些自由软件，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。

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

斯托曼用力敲了几下键盘，合上笔记本电脑，拔掉电源，结束了写作。

“吃午饭去吧？”斯托曼问。

斯托曼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，让他这么一个民族舞爱好者从此告别任何舞蹈活动。斯托曼叹道：“我可喜欢跳舞了，不能跳舞对我来说简直是个悲剧。”

这悲剧从斯托曼的身材上就体现出来了。由于缺乏锻炼，斯托曼开始发福。他的脸颊越来越圆，啤酒肚也凸出来了。你能明显看出，

他的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，斯托曼停下脚步，在一簇玫瑰花前俯下身子。剥开花瓣，斯托曼把鼻子凑到花蕊前，深吸一口气，然后起身长叹，甚是满足。

“嗯，*rhinophytophilia*。”他揉揉腰说⁴。

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，我让斯托曼选择去哪家餐馆。尽管很多报道把斯托曼描述成苦行僧一般，可事实上，斯托曼在饮食方面可谓是个美食家。作为一位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。斯托曼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，斯托曼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，一部分是因为斯托曼刚刚去过中国，期间在香港做了短暂停留，这才想当下吃点中餐。斯托曼本人并不喜欢太辣的东西，所以他也就没选川菜和湘菜。“我对辣的东西不感冒。”斯托曼坦言。

快十一点的时候，我们赶到餐馆。结果发现餐馆的门前居然已经排起了长队，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，生怕斯托曼大动肝火。可出乎意料，斯托曼竟然坦然地接受了这个事实。

斯托曼对我说：“就只有我们俩来了，真可惜。人多了吃起来才热闹。”

等待期间，斯托曼开始走起舞步。他步步小心，可还是能看出他有点儿功底。我们开始聊起时事。斯托曼说，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，它很好地诠释了斯托曼的观点：所谓自由软件，并不与自由市场和自由经济相冲突。然而，斯托曼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。”斯托曼说。

这种冷漠恰恰和地球另一边的热闹形成对比。尤其是亚洲各国，甚是热情。看看斯托曼 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，斯托曼只在美国停留了 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的语言，进入各国的文化。

“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，”斯托曼说，“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，但听众都不太理解。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备受争议的软件，使得音乐爱好者之间可以互相复制音乐文件。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斯托曼的想法：

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，分享行为作为人类的一种本能，就很难被阻止。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（Marilyn Patel）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⁵。美国唱片协会发言人希拉里·罗森（Hillary Rosen）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⁶。”

对于斯托曼这样的黑客来说，Napster 的商业模型内包含着复杂的信息。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，该公司又出售着基于专有软件的服务。这种复杂性，让斯托曼对这一事件少言寡语，因为他知道，自己已经花了很大工夫去传播自己的思想，而对这种部分和自己思想相同，但很大程度上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。可斯托曼依旧承认，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可以了，”斯托曼说，“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，身旁的墙壁是一整面镜子。

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，斯托曼就已经开始在自己的菜单上点菜了。“香

煎鲜虾腐皮卷，”斯托曼读着菜单，“我喜欢腐皮的口感，点一个。”

这一下子又引来了斯托曼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，”斯托曼大声说，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。当时我又突然想起来，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，斯托曼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。这其中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（Bill Gates）。斯托曼的照片这次上了首尔的热销报纸的头版，比尔·盖茨的照片竟然被挤到了下面。除了这件事情让斯托曼非常满意之外，当地的美食也让他欢欣鼓舞。他回忆道：“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”一词在斯托曼嘴里应该是很高的评价了。因为正当斯托曼介绍完冷面，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。”斯托曼说。

我赶紧回头，瞄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，正在结账。当两人起身离开的时候，我发现了斯托曼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。”斯托曼抱怨道。

一声叹息，斯托曼心神回位。这倒给了我一个机会，问问斯托曼

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，斯托曼和女性见面，都会以吻手背的方式来打招呼⁷。在 Salon.com 上，有一份 2000 年 5 月 26 日的采访，把斯托曼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⁸。

我提到，在 1999 年，一本名为《开源软件文集：开源革命之声》的书中，斯托曼在一篇文章里提到，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，所以斯托曼这么做对她来说更是一个惊喜。不过，最后内核的主要开发人员还是把内核名字改成了 Hurd⁹。虽然最后斯托曼与她分手了，这个故事还是给我们留下了一个问题：虽然大部分的媒体上都把斯托曼描写成是一个狂热的激进分子，但斯托曼真的就是一个堂吉诃德式的古怪人物吗？他是否只知道去挑战风车，把这看成是一种浪漫，并以此来吸引他理想中的爱人？

整个上午，斯托曼脸上都挂着笑容。我问起关于吻手背的事情，斯托曼说：“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。”

亲切感是斯托曼生命中永恒的主题。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。斯托曼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，都被斯托曼以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，”斯托曼说，“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。我和斯托曼都花了些力气，才把这大块的萝卜糕切开。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。

“我觉得这样的确对我有好处，”斯托曼嚼着一个饺子，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。”

斯托曼以音乐喜好为例，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，而斯托曼则喜欢听古典乐。他回忆，在 1964 年，自己中学生时候，披头士在《埃德·沙利文秀场》栏目出现¹⁰，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，斯托曼决心抵制披头士音乐。

斯托曼说：“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。”

斯托曼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。

“我想把这乐队叫作‘东京花’，或者‘日本披头士’。”

我知道斯托曼喜欢世界各地的民族乐，我就问他对民谣歌手鲍勃·迪伦（Bob Dylan）什么看法。斯托曼摇摇头：“我倒是更喜欢彼得、保罗和玛丽。他们让我想起了伟大的讲客（filk）音乐。”

我问他什么是讲客音乐，斯托曼解释说，这是专指为现成歌曲重新填词后创作的歌曲，这样的活动在黑客和科幻圈子里颇为流行。经典的讲客乐有《意大利面条之上》，这是借用《旧思墨客山之上》的曲调¹¹。还有由讲客乐大师，“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。

斯托曼问我是不是想听他唱讲客歌。我说想听。他很快开始唱起来，声音出奇的清晰。用的是《随风飘散》（*Blowin' in the Wind*）的曲调¹²：

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，斯托曼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¹³。犹豫了几秒，我也冲斯托曼笑了笑。

“这个粟米球你还吃吗？”斯托曼眨着眼睛问到。我还没回答，斯托曼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。斯托曼端起茶杯，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，”斯托曼说，“说起歌曲音乐和程序，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，斯托曼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，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，斯托曼觉得对此类作品，读者不能有权修改。第三类则是“艺术类”，包含了艺术和娱乐作品。

这三类作品中，对于“功能类”的作品，用户应该具备无条件地

修改它们的权利；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。斯托曼坚信，无论哪类作品，用户出于非商业目的复制与分享行为都应该被允许。也许有人会说，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，然后把副本发送给几百个陌生人。但斯托曼认为，这些复制行为也应该被允许。“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，”斯托曼说，“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，也必须要允许用户，出于非商业目的，把这些作品的副本分享给公众。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。斯托曼打断了我：“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。如今，法庭对于这些现有法律的解释往往会非常严苛，因为这些法律当初都是被出版商们极度拥护才被制定出来的。”

这段话反映出了斯托曼的政治哲学：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，禁止朋友之间分享复制品的法律，和种族歧视法一样，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？如 20 世纪 50~60 年代的民权运动，斯托曼的政治思想也是基于类似的普世价值：自由，正义，公平竞争。

斯托曼这会儿被一撮纠缠在一起的头发分了心，没注意到我的问题。之后我把这个问题更具体化，把斯托曼和马丁·路德·金（Martin Luther King）做比较。这时，斯托曼刚刚弄断头发分叉处的死结，把死结扔进了嘴里。他打断了我的话。“我们不是一个团体，但我们做着同样的事情。”斯托曼一边嚼着东西一边说。

接着，我提起另外一位民权活动家，马尔科姆·艾克斯（Malcolm X），把他的观点和斯托曼的做类比。和他一样，斯托曼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。

斯托曼嚼完嘴里的东西，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。”

尽管如果，对于政治盟友始终抱以怀疑的态度，可以算得上是斯托曼的一个性格特征。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，他的那番类比，更加深了人们对他的这一印象。斯托曼头脑守旧，不肯在政治大趋势和市场面前做出半点让步。

红帽公司总裁罗伯特·杨（Robert Young）总结理查德·斯托曼看似矛盾的政治行为时，说：“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。”

如果从自由软件运动以外的领域来看，斯托曼并非是因为不关注其他政治人物而形成了这种独行侠的作风。在他的麻省理工学院的办

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。打开他的个人网站，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。

鉴于斯托曼这种活动家的倾向，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？

斯托曼整理了一下头发，思考了一阵。

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。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。”

斯托曼之后的一席话，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，但若真的投入其中，则要犹豫再三，要权衡自己的时间和精力，并且避免让自己陷入到并不熟悉的领域中。

“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，斯托曼要结账。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。这张纸币一钱不值，服务员来的时候，斯托曼拽着他的袖子：“我又给你加了零美元的小费。”斯托曼说着，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。”斯托曼说。

尾注

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>。另见《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>。把斯托曼比作宗教信仰徒的文章还有很多。Linux的作者林纳斯·托瓦兹曾用更极端的语言把斯托曼比作宗教信仰徒。参见他2001年发表的自传《乐者为王：一场无意间发起的革命》（*Just For Fun: The Story of an Accidental Revolutionary*, Harper Collins Publishers, Inc., 2001），第58页。托瓦兹说：“斯托曼简直就是自由软件之神。”
2. 译注：原句为“Giving the Linus Torvalds Award to the Free Software Foundation is a bit like giving the Han Solo Award to the Rebel Alliance.”里面借用了《星球大战》中的人物关系。为了读者方便理解，翻译时做了意译。
3. 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。

4. 开始，我还以为斯托曼说的是这花的学名。几个月后，我才发现斯托曼所谓的 *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在亲热。更多斯托曼关于花的玩笑，参见：<http://www.stallman.org/texas.html>。译注：*rhinophytophilia* 是斯托曼自创的一个希腊词汇。*rhino* 意为“鼻子”；*phyto* 意为“花”；*philia* 意为“爱”。
5. 参见 News.com 2000 年 7 月 28 日 Cecily Barnes 与 Scott Ard 所著的文章 *Court Grants Stay of Napster Injunction*。<http://news.cnet.com/2100-1023-243817.html>
6. 参见美国唱片协会于 2001 年 2 月 13 日发表的新闻稿 *Clear Victory for Recording Industry in Napster Case*。<http://www.riaa.com/print.php?id=EA742466-C6DB-1D28-CF94-411172C649C7>。
7. 参见 Mae Ling Mak 于 1998 年 12 月 17 日发表的 *Mae Ling's Story*。<http://www.crackmonkey.org/pipermail/crackmonkey/1998q4/003006.htm>。Mak 是目前为止唯一一个愿意和我聊这事，并且公开姓名的人。除了她以外，还有几个女性跟我提到过斯托曼的这种行为。Mak 一开始对于斯托曼的行为比较反感。不过在 1999 年的 LinuxWorld 上，Mak 也抛弃了这种偏见，还和斯托曼一起跳了段舞。http://www.linux.com/interact/potd.phtml?potd_id=44。
8. 参见 Annalee Newitz 于 2000 年 5 月 26 日在 Salon.com 上发表的文章 *If Code is Free Why Not Me?*。http://www.salon.com/2000/05/26/free_love/。
9. 参考理查德·斯托曼，《GNU 操作系统与自由软件运动》，《开源软件文集》，(O'Reilly & Associates, Inc., 1999)，第 65 页。

10. 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》(*Ed Sullivan Show*)这一节目。从此以后披头士在美国也开始火爆。
11. 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。
12. 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。此处的歌词翻译为意译。原文为：

How much wood could a woodchuck chuck,

If a woodchuck could chuck wood?

How many poles could a polak lock,

If a polak could lock poles?

How many knees could a negro grow,

If a negro could grow knees?

The answer, my dear,

is stick it in your ear.

The answer is to stick it in your ear.

整个歌词并没有什么特殊的意思，主要是一个文字游戏：将一个单词按照发音拆成两个，然后用拆出来的单词组成一些看似有些意义的话。例如，*negro*（黑人，黑奴）按照发音被拆成了 *knee*（膝盖）和 *grow*（生长）两个词，所以 *negro* 就变成了 *knee grow*。本歌词的翻译也基本采用类似原则。

13. 斯托曼自己也创作过讲客歌词，可以在他的个人网站中找到：<http://www.stallman.org/doggerel.html>。想听斯托曼唱歌的话，可以在 GNU 网站上找到他唱的自由软件之歌：<http://www.gnu.org/music/free-software-song.html>。

第 6 章

Emacs 公社

20 世纪 70 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。里面有尖端的项目和顶级的研究人员，为整个实验室赢得了美名。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。对于像斯托曼一样的黑客来说，那里更像是家。

“家”这个字，在理查德·斯托曼的心中，有着特别的分量。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。有一次，他提起自己的大学生涯，说到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。这才发现斯托曼早有准备：“哈佛有个规矩，你修了太多的课程，就必须得毕业了。”斯托曼道。

离开了宿舍，又不想回纽约。斯托曼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，斯托曼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，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,而斯托曼听到“电脑流浪汉”这个词依然心存不快。一说起这事,斯托曼就用一般现在时来形容,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。斯托曼回忆,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(William Shockley);1939年毕业的校友,著名的理论物理学家理查德·费曼(Richard P. Feynman);1951年毕业的校友,夸克之父默里·盖尔曼(Murray Gell-Mann)。除了这些名人以外,吸引斯托曼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。斯托曼说:“我那时越来越喜欢编程,但当时觉得,没准我能两者兼顾,编程、物理两不耽误。”

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,斯托曼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,跑去民族舞俱乐部,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，从此不能再跳舞了。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，斯托曼开始担心。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·斯托曼丧失了所有的社交机会。斯托曼发现自己与其他黑客圈子以外的人，尤其是与女性的交际能力明显下降了，就好像宇航员处在了失重的状态下那样倍感无助。在人工智能实验室待了四个月以后，他在哈福大学四年中所积累起来的自信在不经意间就消失得无影无踪了。

“我当时就觉得自己像个泄了气的气球，失去了做各种事的动力，彻底绝望了。”

这之后，斯托曼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10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，斯托曼觉得，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。而在软件领域中，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·斯托曼一人能感受到。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，他们和斯托曼一样，都在学业中途转职成了全职黑客。其中的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。虽然斯托曼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。斯托曼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，放到斯托曼面前却变得轻而易举。

杰拉尔德·萨斯曼是一位麻省理工学院的教授，也曾经是人工智能实验室的一名成员。回忆起斯托曼，他赞叹：“他太特别了。思维清晰，设计流畅。”1975年年初，萨斯曼曾邀请了斯托曼加入一个研究项目。这个项目非常复杂，需要通过编写程序，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，还要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的一种计算语言。

这些正式项目之外，斯托曼也会花时间维护自己的个人项目。黑客们都喜欢改进实验室的各种基础软件。斯托曼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。

20世纪70年代，斯托曼在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，也值得一书。20世纪50~60年代，计算机刚刚进入大学校园，所谓的编程还是个很抽象的概念。那时的程序员，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，通常被称为“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。这个概念之前曾叫作“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，多个程序可以有效地利用计算机。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。

20世纪60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，用来显示各种输出。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20世纪70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，每敲进一个字母，都会在屏幕上显示出来。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，正如斯托曼所说，这种技能是需要“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，以及事先规划。

在当时，除了 TECO，还有几个其他的全屏编辑器。1976 年，斯托曼去了一趟斯坦福的人工智能实验室。在那里，他见到了一个叫作 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 20 世纪 70 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，即 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。

斯托曼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 上有个功能叫作“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。斯托曼对这个功能还做了个不起眼但影响深远的修改：他允许用户把一连串 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。斯托曼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，他回忆起斯托曼的这个改动，说：“这在当时是个突破⁶。”

斯托曼回忆，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，并且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。”

越来越多的人开始使用宏，并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。斯托曼说：“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⁶。

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。每个人的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。这就导致人们只要坐到一台电脑前,第一件事情就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斯托曼的注意。

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,并没怎么和斯托曼打过交道。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·斯托曼,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,斯托曼把剩下的都做了。”

这个名字被称为 Emacs,是斯托曼建议的名字。Emacs 是 Editing macros 的缩写,意思是“宏编辑器”。它标志着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,斯托曼注意到,在 ITS

系统上，还没有哪个软件名是以 E 开头的。把它叫作 Emacs，用户就可以自己在设置中把它叫作 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⁷。

为了可以开发一套标准的宏，斯托曼和斯蒂尔需要非常小心地处理一些政治上的问题。因为开发一个所谓的“标准”程序，明显是与黑客精神相违背的，黑客们所崇尚的应该是一种“去中心化”的设计理念。同时，TECO 上现有的五花八门的创新设计也为这项工作增加了难度。

盖·斯蒂尔说：“一方面，我们试图创造一个统一的命令集；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。”

为了解决这个问题，斯托曼、斯蒂尔、大卫·穆恩（David Moon）和丹·温里布（Dan Weinreib）决定把进行标准化的工作范围仅局限在控制在屏幕上显示文本的那些所见即所得相关的命令上。Emacs 其余的部分仍然会保留像积木玩具那样的可扩展性。

这个时候，斯托曼又遇到了一个难题：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，斯托曼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。斯托曼把这称作“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。对于斯托曼来说，这更是个社会契约。在早年的一份备忘录上，斯托曼在 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。”

并不是所有的人都乐意接受这个契约。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。这些编辑器中，有些在名字上就提及了斯托曼的 Emacs 编辑器。比如：Sine，

全称 Sine is not Emacs; Eine, 全称 Eine is not Emacs; Zwei, 全称 Zwei was Eine initially。作为一个资深的黑客，斯托曼没有理由要去通过法律的手段来阻止这些创新。但是，总是会有一些不自觉的人，总想着从社区的宝库中偷出一些东西，起个新名字就开始在市场上抛头露面。

如此粗鲁的行为与黑客圈子中其他开发者的行为格格不入。正如第 1 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斯托曼看来不是个好兆头。布莱恩回忆起斯托曼对此的看法，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，可每次提起斯托曼的指责，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⁸。”

虽然斯托曼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，斯托曼开始利用他的力量发起一轮政治影响。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⁹”：如果实验室的成员不反对系统登录密码，斯托曼就不给他们 Emacs 用。这次的抗议活动并没有进一步提升斯托曼在社区中的偏激形象，不过，它传达了一个明确的观点：社区的成员必须努力弘扬黑客的价值观。

在接受《黑客》一书的作者史蒂芬·李维采访时，斯托曼回忆说：“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，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⁹。”

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。如今，GNU Emacs 的用户遍布全球，大家已经把它扩展得异常强大。用户可以把它当作电子表格、计算器、数据库、网络浏览器等。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。斯托曼说：“这就是我们想要传达的。Emacs 里包含的东西既有用，又有趣。”

和斯托曼同时代的人工智能实验室成员更是感谢斯托曼的贡献。哈尔·埃布尔森（Hal Abelson）曾是麻省理工学院的一名博士。20 世纪 70 年代，他曾在萨斯曼教授手下做过研究。之后，帮助斯托曼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，又不会影响整个系统。斯托曼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¹⁰。”

盖·斯蒂尔对此也表示了钦佩。如今，盖·斯蒂尔已经是 Sun 公司的一名科研人员。回忆起斯托曼，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。”尽管他和斯托曼性格不太合得来，可他们俩还是合作了一段时间。这段时间里，盖·斯蒂尔对斯托曼的那种暴风闪电般的编程风格印象深刻。他记得在 20 世纪 70 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斯托曼的注意，接着斯蒂尔和斯托曼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斯托曼的编程风格了。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。跟斯托曼一起工作，他必须要集中精力、心无旁骛。如今回忆起来，斯蒂尔说，斯托曼的这份才智和精力，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。”

尾注

1. 参见乔希·麦克休 (Josh McHugh) 的文章《黑客之爱》(*For the Love of Hacking*)，《福布斯》1998 年 8 月 10 日。
<http://www.forbes.com/forbes/1998/0810/6203094a.html>。
2. 参见理查德·斯托曼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>。
3. 参见约瑟夫·魏岑鲍姆的书，《计算机的能力与人类的理性》(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/87-1-emacs.txt>。
7. 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>。
8. 理查德·斯托曼在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>。

译注:布莱恩此处对理查德·斯托曼的理念理解有误。斯托曼所说的是,所有软件必须是自由(free)的,而非免费的。然而英文中,自由和免费对应的单词都是 free。因此很多人都把斯托曼的思想误解为“软件不能收费”。实际上,斯托曼自己也曾出售过 Emacs 的副本。关于自由软件可以出售的观点,参见 <http://www.gnu.org/philosophy/selling.html>。

9. 参见史蒂芬·李维,《黑客》(1984年,美国企鹅出版社),第419页。
10. 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。如果了解更多技术层面内容,参见理查德·斯托曼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斯托曼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。”

第 7 章

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的，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，叫作 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，以及“一些其他的东西”¹。除此以外，它还提供很多原本 UNIX 并不具备的功能，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。这些都非常吸引眼球。

“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，EMACS 编辑器的作者。如今，有很多人都在效仿这个编辑器。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，以及非兼容分时系统（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¹。

似乎是命中注定一般，斯托曼的 GNU 系统没能赶在感恩节假期结束前发布。不过，斯托曼依旧赶在 1984 年 1 月的时候做出了不小的成果，他自己也完全融入到了 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 20 世纪 80 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的硬件环境。因此，面对硬件更替，UNIX 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的。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。这些运行着 UNIX 的计算机有着更小的体积，却和当年的 PDP-10 性能相当。

为了增强可移植能力，负责 UNIX 开发的程序员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 PDP-10 上的 ITS 系统不同，UNIX 并不使用针对某个硬件平台所特有的系统资源进行开发，而是采用了一种更为通用和现成的实现方式。于是，开发者们可以放眼全局，集中设计各个组件之间的协调机制，设计组件的接口标准，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，可以轻易地被移植到各种计算机上。如果某个用户对哪个组件不满意，他们可以按照定义好的接口标准，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，抛开美学上的意义不谈，UNIX 的设计具备极大的灵活性，这也促进了计算机这个市场的发展，更为 UNIX 带来了无限生机²。

斯托曼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，也伴随着 ITS 的消亡，渐渐分崩离析。这个改变对斯托曼来说，可谓打击重大。施乐公司打印机事件也从另一个方面证明了，人工智能实验室的黑客文化正渐渐迷失在实验室外的商业世界里。

ITS 的消亡过程历经了很长时间。从 20 世纪 70 年代初开始，很多原本可以用于计算机科学研究的国防经费也开始捉襟见肘。为了能够寻求到更多的资金，实验室和大学都开始转向私人领域。就拿人工智能实验室来说，要从一些私人投资者那里争取到资助是非常容易的。作为战后时期很多重要项目的发源地，人工智能实验室很快就成为各种高新技术的孵化器。事实上，到了 1980 年，包括很多黑客在内的实验室大部分成员都同时参与研究项目与商业项目。

一开始，这看上去是个双赢策略：黑客们可以在最优秀的项目里工作，还可以为实验室带来最新的计算机技术。然而，这种行为的弊端也逐渐显露。黑客们投入大量时间去开发前沿的商业软件，而无暇维护实验室的软件系统。很快，各个公司开始把黑客们一个个雇走。随着黑客们的离去，实验室的程序和机器要么没人修，要么也要等上好人才有人打理。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³。”

该来的终归躲不过。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, DEC) 出品的 16 位计算机。在 1982 年，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。当下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。由于黑客们在人数上处于劣势，因此只好将就着用 Twenex 系统。

几年之后，斯托曼回忆起那时的情景：“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。”³

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⁴。斯托曼则由此把 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，把它叫作 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-11⁵。”

黑客们当年第一次看到 Decsystem 的时候也许还会借机嘲笑两

句，可是，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，黑客人数上不占优势，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。

见得此状，斯托曼破解了密码，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，斯托曼向全体人工智能实验室的人发出了警告信³。

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斯托曼挡在幕后。到了 1982 年，斯托曼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，使用斯托曼的登录账号，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 20 世纪 80 年代是加利福尼亚州的一名程序员，他当年就曾利用斯托曼的账号访问人工智能实验室的计算机。他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⁶。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，斯托曼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，管理员很快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斯托曼公开用户名和密码的行为。对此，斯托曼并没有放弃使用 Oz 系统，而是在回应中强调了自己的底线³。

斯托曼之后说：“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。因此，我也坚决不会帮助他们维护一个充满安全设施的计算机系统³。”

20 世纪 80 年代早期，斯托曼的这次抗议标志着黑客和系统管理员之间的矛盾进一步升级。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 KL-10 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。一派主要围绕一家名为 Symbolics 的商业软件公司，另一派则围绕 Symbolics 的对手公司 Lisp Machines Inc. (LMI)。这两家公司在 Lisp 机的市场竞争中不分上下，这种 Lisp 机是以 Lisp 编程语言为基础构建的。

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。20 世纪 50 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级研究计划局（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·格林布拉特领导。在70年代末期，他们设计出了专门用来高效地执行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，由Lisp机项目产生了两家商业公司：由人工智能实验室的前主管，罗素·诺夫斯科（Russell Noftsker）创立的Symbolics公司；和理查德·格林布拉特创立的“Lisp机公司”（Lisp Machines Incorporated），即LMI。

Lisp机的软件系统本身是黑客创造的。它的版权归麻省理工学院所有。但是，按照黑客传统，任何人都可以获得源代码、复制或修改。然而，这种自由分享的方式却让商业公司很是烦恼：公司的产品无法在市场中显得独一无二，也就少了商机。为了获取优势、吸引用户，两家公司开始从人工智能实验室里挖走黑客。它们把黑客们安排在自己公司里，让他们的代码不再流入人工智能实验室。

在Symbolics和LMI两家公司里，Symbolics挖人的行为相比更加激进。20世纪80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，除了斯托曼以外，则都在LMI公司就职⁷。

一开始，斯托曼倒也接受了Lisp机商业化的行为——这无非就意味着他得多承担些实验室的活，不过斯托曼也没抱怨什么。LMI和Symbolics的Lisp机操作系统都是从麻省理工学院得到的授权，从人工智能实验室的Lisp机操作系统上衍生而来。斯托曼当年的任务是继续开发实验室的Lisp机系统，让它能与最新的研究成果同步。Symbolics公司允许斯托曼查看该公司Lisp操作系统的源代码，但是禁止复制。斯托曼说，当年Symbolics和人工智能实验室有个“君子协定”：他们允许实验室在自己的操作系统中，实现Symbolics公司的

操作系统中的类似功能，并且允许实验室继续按照黑客的传统发布实验室开发的代码。

1982年3月16日，斯托曼清晰地记得这一天，那是他的生日。在这一天，Symbolics的主管们决定不再遵循当初的那份君子协议。LMI是Symbolics的直接竞争对手，它们使用的操作系统也是来自于人工智能实验室，而实验室的操作系统则是随时更新的，且包含Symbolics系统最新特性的。Symbolics于是决心彻查人工智能实验室Lisp操作系统的代码，并且严格执行许可证上的协定。如果人工智能实验室还想继续使用自己的操作系统，并且还希望和Symbolics公司系统保持同步，那么最稳妥的解决方案恐怕就是转用Symbolics的系统，并且切断与LMI代码共享。

斯托曼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——格林布拉特的指点帮助。当下，Symbolics的做法可是惹恼了斯托曼。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。斯托曼也发起了反攻，他切断了连接实验室和Symbolics公司的微波通信信道，发誓再也不会再在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，以此帮助LMI，打击Symbolics。斯托曼回忆说：“当初我觉得，人工智能实验室是个中立国，就好像第二次世界大战时的比利时一样。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。”

这样，一场“Symbolics之战”就打响了。而关于此事的实际情况，怕是无从考证，只能凭借当事人回忆来重现历史。Symbolics公司的主管们逐渐意识到，自己公司的操作系统的最新特性，总会依旧

出现在麻省理工学院的 Lisp 机系统上，甚至是竞争对手 LMI 的 Lisp 机系统中。于是，他们在斯托曼的计算机上安装了间谍程序。斯托曼说，他为实验室的系统添加的特性都是自己从头开始写的。虽然许可证上允许他阅读 Symbolics 系统的源代码，但出于版权法的考虑，他必须要想办法把代码写得和 Symbolics 公司的代码有所不同。而 Symbolics 公司的主管们则声称斯托曼抄袭了他们的代码，并把这件事情告到麻省理工学院的管理层。根据 1994 年出版的哈维·纽奎斯特 (Harvey Newquist) 所著的《大脑创作者》(*The Brain Makers: Genius, Ego, and Greed, and the Quest for Machines That Think*) 一书中的描述，麻省理工学院曾警告过斯托曼，让他“远离 Lisp 机项目”⁸。而斯托曼则说麻省理工学院一直是支持他的：“他们没给过我警告。不过这件事倒是确实提醒了我。为了以防万一，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。”

无论当初事实究竟是什么样，这事最终还是平息了。斯托曼不再读 Symbolics 的代码，他按照自己的想法去开发，也吸引了人工智能实验室的成员来为他报告 Bug。同时，他也确保了 LMI 公司可以访问到他最新的代码。斯托曼说：“我当初就觉得，我要是这辈子只能再做一件事，那就是要惩罚 Symbolics。”

这话说得甚是坦白。显然，斯托曼不是个和平主义者。随着矛盾的升级，给双方带来的情感变化也逐渐加大。根据哈维·纽奎斯特的说辞，斯托曼当年甚至发了一封邮件，扬言要“浑身绑满炸弹冲进 Symbolics 的办公室”⁹。斯托曼则否认发过这样的邮件，并且说这完全是“恶毒的谣言”。不过，他倒是承认有过这种极端想法：“我倒的确幻想过要自杀式袭击毁掉他们的办公楼”⁵。

这样的绝望感，让斯托曼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，斯托曼把自己形容为 Ishi——加利福尼亚州最后一位 Yahi

族印第安人。这样的说法倒是为斯托曼的经历平添了几分史诗色彩。现实中，斯托曼与实验室中的其他黑客的关系则越来越紧张。很多斯托曼当年的同事并非觉得 Symbolics 好像要上灭下绝。他们反倒认为 Symbolics 的行为是迟到的正义。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斯托曼是在捍卫什么，而是觉得他只是代表着一种无政府主义的想法而已。

斯托曼倒也没有否认当年同事们的这种观点。这种私下的过结也促成了 Symbolics 下达“最后通牒”的决定。斯托曼说，Symbolics 公司雇走大量黑客之前，斯托曼就觉得很多黑客都在有意躲避自己。这些黑客之后大部分都加入了 Symbolics 公司。斯托曼回忆：“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，当年他被人告知，不许他跟我说大家一起出去吃饭故意不带我这个事。”

虽然斯托曼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 公司不再给用户提来源代码的时候，斯托曼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。

这也为斯托曼在黑客圈子里平添了几分名气。斯托曼早就因为 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。斯托曼的这

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”；并把斯托曼比作现代的约翰·亨利（John Henry）¹⁰。史蒂芬·李维在书中说，很多 Symbolics 公司工作的黑客们都钦佩斯托曼的能力。他引述了曾在 Symbolics 的帕罗奥多分部工作的比尔·高斯伯的一席话，对于斯托曼那段时期的产出，比尔·高斯伯甚是惊奇：

“我读过斯托曼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¹¹！”

对于斯托曼来说，这段时期和 Symbolics 的竞争，既让他感到骄傲，又让他感到了深深的难过。斯托曼是个彻头彻尾的自由主义者，自己的父亲也曾在第二次世界大战期间为自由而战。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。斯托曼曾被这文化滋润，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，静静地躺放在那里，魂归西天。

“我当时泪如泉涌，”斯托曼说，“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，留给斯托曼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，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斯托曼晚两年入学哈佛，算是斯托曼的学弟。不过，斯托曼当年并不认识他。1983 年 9 月 27 日，斯托曼在 net.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，曾有很多个人计算机用户私下复制微软的软件。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¹²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¹³。”

比尔·盖茨 1976 年 2 月发表了这封公开信，那时还没有几个人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 20 世纪 70 年代走到 20 世纪 80 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 20 世纪 60~70 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。

斯托曼十几岁的时候，人们就觉得他少年老成。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，禁止用户随意复制或修改的专有软件，斯托曼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，斯托曼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。”

要让斯托曼放弃他所热爱的计算机和编程，放弃从搬到剑桥市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，斯托曼可绝对不能答应。他没有退却，决定主动出击。

作为一个无神论者，斯托曼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。做出这个选择，对于他来说是再自然不过了。毕竟，凭借斯托曼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，斯托曼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¹⁴？

在外演讲时，斯托曼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，究竟可以做些什么来改变现状？这个问题稍加思考，很快就可以看出来，解决这个问题需要的恰恰是一个操作系统开发者。”

斯托曼说，一旦想通了这点，其他的就“顺理成章了”。斯托曼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，帮助自己和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斯托曼的工作，让他得以继续使用实验室的设备。凭借斯托曼的能力，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，斯托曼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们的支持，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（对于斯托曼的项目）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，这可确实是个超级大的工程。凭着黑客做事风格，斯托曼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，斯托曼联系 VUCK 的作者，询问它们这个软件是不是自由的。作者告诉斯托曼，所谓“自由大学”，指的是阿姆斯特丹自由大学 (Vrije Universiteit in 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。这番回复令斯托曼非常沮丧。

斯托曼回忆：“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。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³。”

放弃了 VUCK，斯托曼转向另一款名为 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 的作者告诉斯托曼，这个编译器可以被自由传播和修改。遗憾的是，这款编译器占用内存太多，以至于在很多平台上无法运行。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，普遍没有这么大的内存。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，斯托曼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，斯托曼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，斯托曼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 的其他组件。

在 1984 年 9 月，斯托曼开始开发 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 (Bill Joy) 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 (Ken Thompson) 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。为 UNIX 系统提供一个 Emacs 编辑器，斯托曼可以更好地展示他的才能，也能让更多 Emacs 用户参与进来。

如今回忆起来，斯托曼说，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，作为一名黑客，斯托曼对于低效的敏感，帮助他避免了重新制造轮子。在开发 UNIX 版的 Emacs 的过程中，斯托曼很快找到了卡耐基·梅隆大学毕业的一位博士生詹姆斯·高斯林 (James Gosling)¹⁵。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。因为斯托曼也想在类似的 Lisp 平台上开发 GNU Emacs，所以他希望能大量地参考高斯林的代码。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斯托曼，并担保可以自由使用。仅管如此，UniPress 还是发现了斯托曼在他的项目中使用了 Gosmacs 的代码，并要求他能尊重版权。于是，斯托曼只好把从 Gosmacs 中参考的代码都从项目中去除，从头开发相关的功能。

通过对高斯林的解释器进行反向工程，斯托曼完成了一个完整的 Lisp 解释器，GNU Emacs 不再依赖高斯林开发的解释器。但是，开发者出售软件版权的行为还是让斯托曼怨恨不已，即便开发者完全有权这么做。在 1986 年，斯托曼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。

斯托曼曾说：“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成光盘，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。不过要是那样，可能很多人都不敢要我送出的光盘³。”

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，对于斯托曼和整个自由软件运动，这个做法不失为明智之举。它让斯托曼必须去直面 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。它也催促着斯托曼更加明确自由软件运动的政治目标。在 1985 年，斯托曼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，斯托曼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？”

斯托曼对此的回应是：“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¹⁶。”

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，斯托曼拉来了一些挚友，组建了自由软件基金会（Free Software Foundation, FSF）。这是一个非营利组织，旨在促进 GNU 工程完成目标。理查德·斯托曼担任主席，斯托曼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 Chassel）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。”

除了提供一个门面以外，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，斯托曼的 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 界的人士曾一度把斯托曼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。

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，去开发和完善一个软件，可到头来却被别人占为己有，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有一个重要原因，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，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。从此，它就与世长辞，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，他好像感受到了当年斯托曼心中的那份怒火。“对我来说，有一件事情越来越清楚：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。”

尾注

1. 参见理查德·斯托曼，《GNU 计划初始宣言》，1983 年 9 月：
<http://www.gnu.ai.mit.edu/gnu/initial-announcement.html>。
2. 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。
3. 参见斯托曼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>。
4. 参见《黑话词典 3.0.0》<http://catb.org/jargon/html/T/TWENEX.html>。
5. 参见 http://www.as.cmu.edu/~geek/humor/See_Figure_1.txt。
6. 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/touristpolicy.html>。
7. 参见 H. P. 纽奎斯特 (H. P. Newquist) 所著的《大脑创作家》第 172 页，Sams Publishing, 1984。
8. 同上，196 页。
9. 参见哈维·纽奎斯特所著的《大脑创作家》(*The Brain Maker*)。原文中提到：“斯托曼的这一言论在 Symbolics 员工那里倒是引起了点注意。不过，最后还是没人把他的话说得太当真。”
10. 译注：约翰·亨利，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第二次世界大战期间美国的宣传画册上。

11. 参见史蒂芬·李维，《黑客》（1984年，美国企鹅出版社），第426页。
12. 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。
13. 参见比尔·盖茨1976年2月3日发表的《致爱好者的公开信》。
<http://www.blinkenlights.com/classiccmp/gateswhine.html>。
14. 参见理查德·斯托曼的文章《GNU工程》，<http://www.gnu.org/gnu/the-gnu-project.html>。斯托曼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：斯托曼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。
15. 译注：詹姆斯·高斯林就是后来的Java之父。
16. 参见斯托曼所著的《GNU宣言》<http://www.gnu.org/gnu/manifesto.html>。

第8章

ST. 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沾不上一点边。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，盯着外面如画的风景，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，暴晒着这位黑客的苍白的皮肤，他也在所不辞。

我赶到现场的时候，会议室里早已人满为患，大家都在等待着斯托曼的演讲。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。

斯托曼身穿纯蓝色 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，倒是让人没太注意斯托曼缺少阳光滋润的皮肤。不过，他的一头长发和络腮胡须却依

旧能引人注目。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。演讲开始前，斯托曼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 20 世纪 70 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，斯托曼的演讲总会被记录拍摄下来，留作档案。作为当地的著名自由软件组织，茂宜 FreeBSD 用户组不能让同行失望。他们也要录音和录像，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。

斯托曼的这些演讲记录，就像感恩而死乐队（Grateful Dead）的现场演出一样，都是被追捧者现场录下、私下传递的。把斯托曼和感恩而死乐队做类比，可是有说头的。每当斯托曼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和录像，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¹。

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，20 世纪在 90 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斯托曼的号召。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，斯托曼创立

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²。”

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斯托曼的讲座。

和那次在纽约的讲座不一样，这次没有人为他做开场白，斯托曼自己也没做自我介绍。FreeBSD 用户组的人把设备一架好，斯托曼就自己踏上前来，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，”斯托曼开门见山，“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 20 世纪 70 年代曾加入了一个大家庭，在那里大家都会互相分享软件。正因为如此，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？也正因为如此，我得到了完全不一样的答案。”

斯托曼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。

“有些人会和我讲，‘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，我们必须要尽一切努力，让用户可以自由地使用计算机和软件³。”

斯托曼最后说：“我们要走的路还很长。”

对于一些听众来说，斯托曼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，斯托曼停下来，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，”斯托曼说着，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。可是，如果把它用在信息世界中，就越来越显得古怪了。软件版权和软件专利这两个概念看似差不多，实则差距甚远。创作者可以借助版权禁止他人复制自己的代码，但却无法阻止别人利用不同的代码实现相似的功能或软件。换言之，一个开发者可以从头克隆一个现有软件的所有功能。只要他不去复制别人的源代码，就不会触犯版权法。这种克隆想法或功能的行为非常普遍。

可软件专利就不同了。按照美国专利局的说法，个人或公司的算法方面有创新想法，就可以提交专利申请，由公众来审核。理论上讲，专利拥有者需要将自己的发明公诸于众，而作为交换，专利拥有者获得法律保护，可以在二十年内禁止他人实现这个想法，计时从专利提交起开始。当年专利法设立的初衷是为了避免优秀的想法被当作商业机密封锁在公司内部，不被大众所用。但是实际上，到了软件领域，

这个事情就变了样。软件专利的创新想法并不能让公众受益。因为大多数软件专利中提到想法往往是不言自明的。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。简单地说，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，如图像处理领域、加密领域等。

对于斯托曼来说，软件专利更督促着黑客要时刻警觉，而且还平添了几分戏剧色彩。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，用户的自由比其他东西更重要。斯托曼说，软件专利所可能造成市场的垄断，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。

斯托曼说：“不是因为我们技不如人，而是因为软件专利禁止我们实现很多想法，不让我们去创造优秀的软件。遇到这个问题用户会作何感想？这要看这个用户是被哪种观念吸引进自由软件圈子里来了。一种是‘开源软件’的观念，它把软件用户的自由看作是一种工具，赋予用户这些自由，可以让软件更加强大，功能更加完善。如果用户是被‘开源’的理念打动而使用自由/开源软件的，那么当他发现某些自由软件由于专利原因，无法实现某种功能的时候，

他可能会说：“当初你们承诺软件会很强大很完善，可如今你们却没能实现这些功能，你们欺骗了我！”与“开源”的理念不同，“自由软件”强调软件用户的自由本身的重要性，它是我们的目的，而不是一种手段。当用户了解到“自由软件”的观点时，面对同样情况，他可能会说：“那些软件专利怎么能如此禁止人们实现这些功能？它们怎么能这样剥夺我们的自由？”倘若用户都能如后者一般，那么面对如今和未来的大量软件专利，我们才可能生存下去。”

斯托曼的这番评论确实有些拗口。其实，大多数开源软件的支持者其实也一样反对软件专利。但关于开源软件和自由软件的区别，斯托曼的描述却也不错：开源软件的观念更偏向于实用主义；而自由软件的观念则更强调用户自由。开源的支持者不怎么喜欢强调软件用户的自由，他们更愿意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，而不必在道义上过多解读。埃里克·雷蒙德（Eric Raymond）是开放源代码促进会（Open Source Initiative, 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，用以表明他对遵循斯托曼的政治路线的失望。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斯托曼的一些意见不同，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，斯托曼的言论对他们无效。面对这些，我们开始迷惑，甚至怀疑⁴。

雷蒙德写道：那另外95%的人中，包括很多公司经理、投资人，以及普通的计算机用户，他们不是黑客，他们会倾向于跟从专有软件的大市场。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。

RMS 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⁴。

看着斯托曼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？斯托曼引述自己的信念，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。”

斯托曼的这个回答，倒是更让我觉得他有点宗教人士的感觉。就好像是犹太人只吃洁食的行为，摩门教徒不喝酒。斯托曼仅仅使用自由软件，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，在听过斯托曼的演讲之后，都会明白斯托曼所说的正确的软件之路。

演讲最后，像个总结一样，斯托曼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·IGNUcius⁵，来自 Emacs 教堂。”斯托曼举起右手，好似在为众人祈福：“我保佑你的计算机，孩子们。”



图三 斯托曼打扮成圣·IGNUcius。照片由 Wouter van Oortmerssen 拍摄

大家的笑声在几秒内变成了一阵掌声。大家在鼓掌的时候，斯托曼头上的那个硬盘碟片反射来一道强光，一下看起来，好像斯托曼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。

斯托曼接着解释道：“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。我们把他们称作 Emacs 信徒。”

很多人以为斯托曼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，斯托曼好似在说：“你们尽情笑吧，我知道我这样很搞怪。”

之后，我和斯托曼聊起圣·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。斯托曼说，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，斯托曼虽然固执，但还不至于疯狂到不苟言笑。不过这之后，一些人借着这个舞台形象，把斯托曼形容为一个软件界的空想家。埃里克·雷蒙德就在 1999 年的一次与 Linux.com 的采访中，说道：

“我说过，RMS 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⁶。”

斯托曼显然不同意埃里克·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。”

斯托曼也承认，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的感觉。”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。斯托曼也曾向别人建议过 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，我和斯托曼聊天的时候，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。斯托曼立即回复道：“不，但是我在很多方面都很敬仰格鲁桥·马克思，我说过的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。”

斯托曼一直以来对俏皮话、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。要说斯托曼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，甚至反而会觉得斯托曼正在内心中笑话你。

不过你别担心，斯托曼可没笑话你。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，但斯托曼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。”斯托曼继续对听众说道：“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。斯托曼在摘掉自己头顶的“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!”

一个简短的提问环节过后，听众们围在斯托曼周围，有些人向他索要签名。斯托曼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，斯托曼的行为会始终如一，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，会一直体现在斯托曼的每个言行之上。在演讲之后的晚餐上，一位程序员向斯托曼提起了自己参与“开源”软件的经历，斯托曼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，斯托曼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，用斯托曼的话说，就是好比很多第三世界国家建立的城市。人们怀揣梦想从四面八方涌入城市，希望能让自己的生活变得更加富裕，或者至少能融入到这个充满活力和开放文化的社会，但是那些真正有权有势的人则通过制定各种新的方针政策来玩弄老百姓。软件专利就是一个很好的例子。“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。斯托曼说：“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，斯托曼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？”

斯托曼答：“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。”

斯托曼立即反应道：“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。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20世纪80年代——绝大部分程序员都没想到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，”斯托曼说，“在版权局能查到的公开文章就是告诉你，从现在开始往后二十年，这个想法任何人都不能随便用。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

你不能用。”

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。斯托曼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，斯托曼又重复了一遍那句口头语。

“Happy Hacking。”斯托曼说。

尾注

1. 参见《1985 年~1995 年北美演出收入排行榜》，<http://www.dead101.com/1197.htm>。
2. 参见埃文·列伊博维奇在文章《谁会害怕大灰狼》(Who's Afraid of Big Bad Wolves)，2000 年 12 月 15 日，ZDNet Tech Update。<http://www.zdnet.com/news/whos-afraid-of-the-big-bad-wolves/298394>。
3. 为了叙述方便，我（原作者）一直没有机会详细介绍斯托曼所说的自由软件。在 GNU 项目的网站上，定义了软件用户的 4 个基本自由：
 - 出于任何目的而运行软件的自由（自由度 0）；
 - 阅读、学习并且修改软件源代码的自由（自由度 1）；
 - 将软件对外发布的自由（自由度 2）；
 - 将修改后的软件对外发布的自由（自由度 3）。

想要了解更多信息，参见 GNU 项目网站上的《自由软件定义》：<http://www.gnu.org/philosophy/free-sw.html>。

4. 参见埃里克·雷蒙德于1999年6月28日发表在网络上的短文《少说话，秀代码》。
5. 译注：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。斯托曼的圣·IGNUcius本身并没有任何宗教意义 (斯托曼本人是无神论者)，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。
6. 参见1998年5月18日刊登于Linux.com的文章 *Guest Interview: Eric S.Raymond*: <http://www.linux.com/interviews/19990518/8/>。

第 9 章

GNU 通用公共许可证

1985 年新春伊始，理查德·斯托曼完成了 GNU 工程中第一个完整的里程碑式的软件：用 Lisp 语言编写的 Emacs 编辑器。为了能让别的用户可以真正自由地使用这个软件，他面临着两大挑战。首先，他需要改写出一个跨平台的 Emacs 版本；其次，他需要基于早期的“Emacs 公社”探索一种全新的发布方式。

与 UniPress 之间的纠纷暴露出了 Emacs 公社契约中的一些不足：这些条款都过度依赖斯托曼个人在这个社区中的地位。对于那些斯托曼所没有触及到的领域，比如 1984 年以前的 Unix 系统，其他个人和公司都可以制定他们自己的规则。

在 GOSMACS 之前，修改软件的自由与软件作者的特权之间就已经形成了一种矛盾的关系。1976 年修订的美国版权法案把法案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“表现方式”声明版权，但不能对程序中的“实际处理过程和方法”声明版权¹。换句话说，程序员与软件公司可以像对待一个故事或一首歌那样来对待一个软件，可以从一个已有的软件中获取灵感，但是在没有得到原作者允许的前提下，不能直接复制代码或在它的基础上直接衍生出新的版本。尽管这部新的法律规定了软件中即使没有包含版权声明也同样受到法律保护，程序员们还是纷纷开始在他们的程序中添加上了版权声明，声张自己的权力。

刚开始，斯托曼把这些版权声明看成是一个不好的兆头。那时候，大部分程序都会从其他以前的程序中借鉴一些代码，然而，总统的大笔一挥，议会就给予了程序员和软件公司把社区开发的软件变为专有软件的权力。这就使得原来游走在灰色地带的事情变得公开合法化。即使黑客们有能力去追溯一段代码最初的来源——也许没有办法找到几十年前的源头，但至少可以往前查几年，但是为了证明这些原本属于社区的代码不应当被专有化而对每一段代码进行这种追查所要花费的时间和金钱都远远超出了大部分黑客所能接受的范围。简单地说，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们来说占据了天然的不利地位。

然而，软件版权的支持者也有他们相反的观点：如果没有版权的保护，程序员的作品可能会在不经意间进入公有领域。同时，在自己的作品中加入版权声明常常也象征着对软件质量的负责，程序员或软件公司在代码中加入了署名的版权声明也就意味着把自己的信誉与这份代码绑定在一起。另一方面，版权声明也可以看成是一种合约或是所有权的声明。软件作者可以把版权声明用作一种具有弹性的许可

证，通过版权声明放弃部分权利，换取用户方某种形式的行为。比如，软件的作者可以允许人们随意复制软件，但是不得把这些副本用于商业用途。

这些观点最终使斯托曼反对软件版权声明的态度慢慢发生了变化。回顾 GNU 工程的历史，斯托曼说他直到差不多 Emacs 15.0 发布的时候才逐渐意识到使用版权来保护软件的好处，Emacs 15.0 是 GNU 诞生前的最后一个 Emacs 大版本。斯托曼回忆道：“那时我看到有些人在电子邮件中也开始使用版权声明，还加上了一行‘允许原文复制’的许可声明，这触发了我的灵感。”

从 Emacs 15 开始，斯托曼在软件中加入了一项版权声明，允许用户随意复制并分发软件。他还允许软件的用户修改软件，但不允许用户把修改过的软件作为自己的作品并享有排他特权，就跟 GOMACS 一样。

斯托曼认为，Emacs 15 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”，不适合用作 GNU 工程的软件许可证。在开始编写 GNU 版本的 Emacs 时，斯托曼开始与自由软件基金会的成员一起探讨应该如何来完善许可证的文本。他还咨询了帮助他创建了自由软件基金会的律师。

马克·费雪（Mark Fischer）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。他有两条原则，第一条是要让软件变得尽可能的开放，第二条是尽可能鼓励其他人接受和使用类似的许可证协议。”

鼓励其他人接受和使用类似的许可证就意味着不再允许专有的 Emacs 版本出现。为了实现这一目标，斯托曼与他的自由软件同僚们找了一种解决方案：用户们可以随意修改 GNU Emacs，只要求他们

把他们的改动都公布出来；并且，这些衍生出来的软件作品需要与 GNU Emacs 采用相同的许可证。

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它。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笔交易，也是相当于给 GNU Emacs 打上了“价格标签”。不过，这里的价格却不以金钱度量，它允许用户自由使用软件，但如果对软件进行修改，则必须把修改后的代码贡献出来。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。

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，斯托曼收到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（John Gilmore）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（Sun Microsystems）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 公司开发的 Unix 发行版。在移植的过程中，吉尔摩按照 GNU Emacs 许可证的要求公开了他所作的修改。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（Larry Wall）的作品，他后来还开发了 Perl 语言和 Unix 程序员用于给软件代码打补丁的 Patch 工具。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。

如此的声明在映射着黑客的精神和气质的同时, 也映射出要把这种松散、非正式的精神转换为严谨的软件版权法律语言的困难之处。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, 进而把这个软件变成专有软件的漏洞。他以一种律师和黑客都可以理解的方式, 向世人展现了黑客的精神。

吉尔摩回忆说, 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他软件作品上。1986 年 11 月的一天, 吉尔摩受到 Usenet 上一个帖子的启发, 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”或其他类似的名词。在不久的将来, Emacs 也许就不再是 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 许可证使它可以适用于 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 Emacs 字眼并把它转变为一个保护所有 GNU 软件版权的许可证。GNU 通用公共许可证就这样诞生了, 缩写是 GPL。

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, 也就是斯托曼打入 Unix 阵营的第二个重要作品 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利，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权利。

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。

与此同时，斯托曼也不得不对原来那个不太正式的“Emacs 公社”协议进行修改。在“Emacs 公社”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以相同协议公开衍生作品的代码即可。也就是说，程序员修改了 Emacs 并仅供自己个人使用时，就无须把代码发给斯托曼了。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。

回首往事，斯托曼说，GPL 中所做出的妥协很大程度上是因为他个人对原先“Emacs 公社”中存在的“老大哥”角色感到不满。当他对黑客社区了解的越深入，他越感觉到未来的源代码维护人员可以用他们的权力迫使他让 GPL 做出妥协。

“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。”

历史证明，GPL 是斯托曼最伟大的创造之一。它在现有的版权法律的框架中创造了一种平等的社区系统。更重要的是，它向世人展现

了法律文书与软件代码之间的相似性。GPL 的序言其实暗示了一个信息：与其用怀疑的眼光看待版权法，不如用黑客的精神渗入到这个系统中。

“GPL 的设计过程与其他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。社区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，这样才能更容易地为大众所接受。”另一位曾帮助斯托曼一起设计 GPL 的律师杰里·科恩（Jerry Cohen）说，“这样的设计方式非常有效，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。我们利用了软件版权法本身来对抗整个现有的版权系统⁵。”几年后，斯托曼开始使用一些不那么敌对的条款来描述 GPL。“我在考虑那些有关道德意识、政治意识和法律意识相关的问题，”他说，“我需要设法让它可以融入现有的法律系统。从本质上来说，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。我需要设法在现有法律框架上去设计它，这在以前的设计中从来没有过。”

正当斯托曼在思考自由软件有关道德、政治和法律上的问题时，一位来自加州名叫堂·霍普金斯的黑客给他寄了一份 68000 微处理器的手册。霍普金斯是一名 Unix 黑客，同时也是一名科幻小说爱好者，这本手册是他早些时候向斯托曼借阅的。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，有一张贴纸上的文字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”一词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

GPL 在自由软件基金会的官方标识⁶。

德国著名社会学家马克斯·韦伯（Max Web）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、政治化和道德化的东西，这样这种宗教的精神才可能长时间在不同的文化之间进行传播。

尽管 GNU GPL 不是家教，但是它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的那套理论，也会很乐意去接受 GPL 表面上所呈现出来的普适价值。一部分人把 GPL 看成是对自己软件作品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，但它在业界的影响力依然不可低估。

基思·博斯蒂克（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，他是加州大学伯克利分校（University of California, Berkeley）的一名雇员。博斯蒂克所就职的计算机系统研究小组（Computer System Research Group, CSRG），从 20 世纪 70 年代就开始从事 Unix 系统的研究，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 网络协议就是出自这个小组。到了 80 年代，Unix 商标所有者 AT&T 开始专注于 Unix 的商业化，他们选择了由博斯蒂克和他在伯克利的同事们一起开发的“伯克利软件套件”（Berkeley Software Distribution, BSD）这个 Unix 的学术分支作为未来商用 Unix 的主要技术来源。

虽然伯克利的 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，但是要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 的专有代码，所以，伯克利套件只能在那些已经向 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，这种看上去很无辜的授权协议就越来越成为一种负担。

1986年，博斯蒂克开始进行一个把BSD移植到DEC公司(Digital Equipment Corporation)的PDP-11计算机上的个人项目。博斯蒂克回忆说，在这段时间里，斯托曼有时会来西海岸，所以他有与斯托曼进行一些密切的接触。“我直到现在还清晰得记得当时的情景：斯托曼坐在我们研究小组一台借来的工作站上与我争论有关软件版权的问题。争论了一段时间，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拥有版权的那部分代码中剥离出来，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⁷。与GPL不同，在加州大学许可证中，对代码的商业化使用是被允许的。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，但它只能与其他专有代码放到一起才能真正运行起来。

在接下来的几年中，博斯蒂克和其他加州大学的同事们一起，重新编写BSD中所缺少的组件，把BSD变为一个完整的可以自由分发的操作系统。虽然这项工作因为AT&T下属的Unix系统实验室(Unix System Laboratories)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20世纪90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就已经开始进入斯托曼的GNU工程中。

博斯蒂克回忆说：“如果没有GNU的影响，我想我们不可能如此坚定地走得那么远。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 20 世纪 80 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，就会更直接地表明自己作为自由软件身份。布鲁斯·佩伦斯（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。”佩伦斯是一个知名的 Unix 工具软件 Electric Fence 的开发者，也是 Debian GNU/Linux 开发团队的领头人。在 GPL 发布后的几年，佩伦斯说他决定废弃 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。”

作为一名程序员，李奇·莫林对斯托曼最初的 GNU 宣言在一定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聚集到了 GPL 的大伞之下。作为一名 SunOS 用户组的领头人，莫林在 80 年代期间的一项重要工作就是分发一些载有优质免费软件和自由软件工具的磁带。为了完成这项工作，他需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，以及他们是否愿意把这个软件放入公有领域。莫林说，到了 1989 年左右，很多出色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，就知道我不用担心授权的问题了。”莫林回忆说。

为了补偿在 Sun 用户组中制作分发磁带的成本，莫林会向订阅者收取少量的费用。现在，这些程序都改用 GPL 了，莫林收集软件并制作磁带只需要花以前一半的时间，从此就有了一点小小的盈利。莫林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，并创办了 Prime Time Freeware 公司。

这样的商业探索完全是在自由软件推广计划框架之中的。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⁸”。到了 20 世纪 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’理解成是‘免费啤酒’中的‘免费’，它是‘自由言论’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。在商业市场中如果不考虑软件的所有权，就算是一个很小的软件公司，也有机会与 IBM 和 DEC 这样的大公司自由竞争。

迈克尔·蒂曼（Michael Tiemann）是最初少数几个抓住了这个理念的企业家之一，他是一名程序员，同时也是斯坦福大学的一名研究生。在 20 世纪 80 年代，蒂曼曾经一度像一名爵士乐发烧友追随一名他所喜爱的音乐家一样追随着 GNU 工程。1987 年 GNU C 编译器的发布，让他开始真正认识到自由软件内在的巨大潜力。GCC 的诞生可以看成是一个爆炸性的新闻，蒂曼觉得，GCC 的出现进一步巩固了斯托曼作为一个传奇程序员的地位。

“就像所有的作家都梦想要写出一部伟大的小说一样，20 世纪 80 年代时，每个程序员都在梦想要写出一个伟大的编译器，”蒂曼回忆说，“斯托曼以迅雷不及掩耳之势完成了这项工作，这实在是让其他的程序员蒙羞。”

“这就是所谓的单点故障，GCC 就是一个很好的例子，”博斯蒂克回应说，“那时候没有别的编译器，GCC 就是唯一的一个。”

在蒂曼看来，与其与斯托曼正面竞争，不如在他的作品基础上进行二次开发。GCC 最早的版本大概有 110000 行代码，但是在蒂曼眼里，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些代码，并且只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

半导体公司（National Semiconductor）的 32032 微处理器上。次年，蒂曼就开始进一步修改 GCC 的源代码，并开发了一个原生的 C++ 语言的编译器。有一天，蒂曼在贝尔实验室做一个关于 GCC 的报告，他发现有些 AT&T 的开发者还在努力地做着相同的事情。

“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，”蒂曼回忆说，“主持人说具体的开发人员的人数是保密的，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。”

蒂曼继续回忆说，不久以后，他就灵光一闪，有了个想法：“我已经在这个项目上投入了半年的时间。我开始思考，能如此高效的开发出新的编译器，这到底是我自己的功劳，还是这份代码的功劳？谁更应该在自由市场中获得嘉奖？”

蒂曼从《GNU 宣言》中获得了更多灵感，这虽然会让一部分软件开发商的利益受损，但是它鼓励其他软件开发商更多从消费者的角度出发，更好地利用自由软件。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，而这些其实是软件市场中利润最高的几个领域之一。

在 1999 年的一篇文章中，蒂曼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呐喊，但是它们还是大不相同的。我在里面发现了一个隐蔽的商业计划⁹。”

约翰·吉尔摩也是一位 GNU 工程的粉丝，蒂曼和他一起成立了一个软件咨询服务公司，专注于提供对 GNU 程序的定制化服务，这就是后来的 Cygnus Support 公司。1990 年 2 月，Cygnus Support 与客户签订了第一份软件开发合同。同年年底，该公司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 工程的最初五年时间中开发的唯一软件。截至 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 公司的莫林提到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。莫林说：“hello 软件包其实就是一个只有五行代码的‘hello world’C 语言程序，但这个程序按照 GNU 发行包的标准进行了打包。里面包含了标准的 Texinfo 的文档和 configuration 配置脚本。这个软件包为其他的软件开发者提供了一个很好的示例，展示了 GNU 工程是如何让软件包可以平滑地移植到不同的环境中的。这件事非常重要，这不但影响到所有斯托曼的软件作品，也影响到其他所有的 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：“对于某个软件作品来说，我可以去改进它。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。从某种程度上来说，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产生积极的影响¹⁰。”

不管怎么说，GNU 的工具在 80 年代后期发挥出了它的优势，斯托曼所在的人工智能实验室也以能设计出严密的程序而在整个软件开发社区中建立了良好的声誉。

杰瑞米·埃里森 (Jeremy Allison) 在 20 世纪 80 年代曾是一名

Sun 的用户。90 年代，他开发了 Samba 这个著名的软件。谈笑间，他说起了斯托曼当年的声望。80 年代后，埃里森开始使用 Emacs，并且对它的社区开发模式产生了兴趣。他曾满怀希望地给 Emacs 贡献过代码，结果他的补丁却没能被采用。

埃里森说：“这就像是美国《洋葱报》(*The Onion*) 的头条新闻，孩子们在祈祷，但是上帝回答说‘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，但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 20 世纪 80 年代末，GNU 工程因为没有自主开发的内核引发了越来越多的抱怨。跟其他 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工作。根据 1987 年 1 月的“GNUsletter”所描述的，那时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。

回顾 80 年代末“GNUsletter”中的新闻，GNU 工程在软件开发管理上的压力在那个时候已经渐渐显现出来。1987 年 1 月，斯托曼宣布 GNU 工程正在尝试修改 TRIX 这个由 MIT 开发的 Unix 内核。一年以后，1988 年 2 月，GNU 工程又宣布他们已经把注意力转向 Mach，一个由卡内基美隆大学开发的“微内核”。这也就意味着，在 1990 年以前，GNU 工程并没有真正开始官方的内核¹¹。

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，莲花公司 (Lotus Development Corporation) 起诉了它的一家竞争对手——Paperback Software International 公司，控告该公司抄袭了莲花公司的 1-2-3 电子表格程序中的菜单设计。莲花公司的

官司，加上苹果和微软之间的“视觉体验”的斗争，给 GNU 工程带来了颇多麻烦。虽然这两场官司跟 GNU 工程都没有直接的关系，因为他们都主要是与个人电脑上的操作系统和应用软件相关，而不是关注于类 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”，他们在莲花公司门口以及波士顿审理莲花公司这起案子的法庭外发起了抗议。

这次抗议非常引人注目¹²，他们的行动在软件工业发展史上留下了浓重的一笔。应用软件正悄悄地取代操作系统成为软件公司竞争的主要战场。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”的诉讼，但这在一些观察家眼中，则更显得 GNU 工程正在变得更加过时。

1990 年，麦克阿瑟基金会授予了斯托曼“麦克阿瑟院士”的称号，嘉奖他出众的能力，并授予他“杰出人才奖金”。这份 24 万美元的奖金用于奖励他发起了 GNU 工程和他在宣传自由软件哲学上所取得的成就，这帮助 GNU 工程在短期内缓解了经济上的困难。最重要的一点就是，它让斯托曼这个在自由软件基金会不拿薪水而仅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¹³。

麦克阿瑟奖还戏剧性地赋予了斯托曼选举权。在获奖前不久，发生在斯托曼公寓里的一场大火几乎让他损失了一切财产。在获奖的时候，斯托曼还只是在技术广场 545 号大楼存生的一名流浪汉¹⁴。“选举中心不同意我把那里作为我的住址，”斯托曼后来回忆说，“不过报纸上那篇有关麦克阿瑟奖的新闻帮我证实了这个现状，于是他们就让我注册了¹⁵。”

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

有意思的是，GNU 工程和自由软件运动最终的胜利差不多就是源于这些旅行。1990 年，斯托曼去芬兰的赫尔辛基大学访问。在听众中，有一名叫林纳斯·托瓦兹的学生，那时他只有 21 岁。后来托瓦兹发明了 Linux 内核，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辛基大学上学，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。在赫尔辛基很少能见到这样的人。”托瓦兹在 2001 年的自传《乐者为王》中回忆到¹⁶。

虽然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在内在逻辑：没有人可以写出没有错误的代码。通过共享软件代码，黑客们把不断改进程序作为终极的目标，甚至超越了个人野心。

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，而是对那些五花八门配置的家用电脑更感兴趣。作为一名大学生，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大型机的不方便之处有着一种不同的视角。对于斯托曼来说，主要的不方便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录他的 Unix 系统，托瓦兹开始设法寻找一种可以在他温暖的校外公寓中远程访问学校的 Unix 系统的方法。

不久，托瓦兹就发现了 Minix 操作系统，Minix 操作系统是一名

荷兰的大学教授安德鲁·塔嫩鲍姆（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，这正好是托瓦兹能买得起的最强大的机器，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，有了这项功能才能使托瓦兹足不出户模拟出一个学校电脑的终端并登录到学校的 MicroVAX 计算机上。

1991 年的夏天，托瓦兹把 Minix 从头重写了一遍，并加了上了一些他所开发的新特性。到了夏天快要过去的时候，他完成了这个名为“GNU/Emacs of terminal emulation programs”的程序¹⁷。他对这个作品很有信心。他开始在一个 Minix 新闻组上寻找 POSIX 标准，POSIX 标准定义了与 Unix 兼容的程序应有的行为。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 工程时的所发的帖子的消息：

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）¹⁸。

在不到一个月的时间内，这个帖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，但他和其他开发人员都意识到，大部分重要的工具软件其实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中的一个，有了它才让使用 C 语言进行开发成为可能。

整合 GCC 大大提升了 Linux 的可用性，但也带来了一些问题。仅管 GPL 的“传染性”并不影响到 Linux 内核，但托瓦兹在他的自由软件操作系统中使用了 GCC，也造成其他用户产生了从 Linux 借用更多东西的愿望。事后托瓦兹说，“我需要站在巨人们的肩膀上¹⁹”。因此，他开始思考，当别人开始利用 Linux 时并需要更多依赖 Linux 时事情会变成什么样子。在做出这个最终决定的十年后，托瓦兹回应了自由软件基金会的罗伯特·查瑟尔，把他的想法总结如下：

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取暴利。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些什么，我希望我一直可以看到那些别人所改动过的源代码，这样的话，我也能有机会对它继续进行改进²⁰。

Linux 的 0.12 版本中第一次完整的集成了 GCC，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 内核原有的许可证，替换为 GPL。这个决定带来了一场移植软件的狂欢，托瓦兹和其他 Linux 爱好者寻找了各种 GNU 程序并把它们移植到了 Linux 中。短短不到三年的时间，Linux 开发者发布了第一个可以用于生产环境的 Linux 1.0 版本。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 年，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，很多观察家因此认为托瓦兹在把 Linux 内核许可证切换到 GPL 后的最初几个月中，并没有把程序真正的交给社区。在第一期《Linux 通讯》杂志中，杂志编辑罗伯特·杨对托瓦兹进行了一次面对面的采访。当杨向这名芬兰程序员提问是否后悔放弃了对 Linux 源代码的专有权时，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²¹。”

这个决策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 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，绝非是一个像 Control-R 这样的小小的软件技巧²²，而是一个让类 Unix 系统可以在个人计算机上运行的新奇事物。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那样，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。这个毫无政治目的的原始想法对于自由软件基金会而言是个打击。

托瓦兹自己觉得，他只是跟很多小朋友一样，喜欢把玩具拆开再组装起来。但事实上，他利用闲暇时光在一块废弃的硬盘上创造出一个成功的项目，他放弃了对 Linux 的控制权，接受 GPL 的恩泽，这一切都足以证明年轻的托瓦兹所拥有的过人智慧。

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 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更为深刻的东西²³。”

尾注

1. 参见哈尔·艾贝尔森、麦克·费雪和乔安妮·科斯特洛（Joanne Costello）的“软件与版权法”1998年修订版。
<http://www.swiss.ai.mit.edu/6805/articles/int-prop/software-copyright.html>。
2. 参见 Trn 软件包的 README 文档。<http://stuff.mit.edu/afs/sipb/project/trn/src/trn-3.6/README>。
3. 节选自约翰·吉尔摩寄给作者的电子邮件。
4. 参见理查德·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. 参考大卫·贝茨（David Betz）和乔恩·爱德华兹（Jon Edwards）的“理查德·斯托曼与 BYTE 编辑关于他的公有领域 Unix 兼容软件系统的访谈”一文，Byte 杂志，1996年7月（GNU 工程网站上转载的地址：<http://www.gnu.org/gnu/byte-interview.html>）。

这篇访谈从一种有趣、直率的视角展示了斯托曼在 GNU 工程早期的一些政治态度，对于研究斯托曼推广 GNU 理念时采用的各种方式的演化过程具有很大的意义。

对于创造 GPL 的目的，斯托表示：“从宏观角度来说，我希望能改变人们获取知识和信息的方式。试图独自占有知识、控制其他人使用知识或是禁止其他人分享知识的行为，都是野蛮的行为。”

2000年，斯托曼在与本书作者的一次对话中表示：“我希望你在思考问题时不要再使用‘知识产权’这个词语。这个词语具有很大的误导性，因为它混淆了版权、专利和商标这些不同的概念。这些概念之间存在着天壤之别，把它们混为一谈是一种愚蠢的行

为。如果你听到有人在谈论知识产权，而且还不是引述其他人的观点，那就你不应该与这个人进行讨论，因为他根本就没有把问题想清楚。”

6. Copyleft 是黑客们对英语中“版权”这个词汇玩的一个构词游戏。“版权”在英文中对应的单词是 Copyright，其中的“right”如果以一个独立的单词来看，意思是“右”。为了表达 GPL 与一般意义上的版权法的不同，把“右”改成“左”，也就是 left，形成了 Copyleft 一词。

7. 不过，加州大学这种类似于广告的条款让人生厌，并且在后来也被证明是存在问题的。有些人为了寻找一种比 GPL 宽松的许可证，直接把加州大学许可证中的“加州大学”一词换成自己所在的机构。这造成的结果就是很多自由软件因为整合了很多不同机构开发的组件，就不得不把这些机构的许可证一一列出，看上去就像一堆杂乱的广告。1999 年，在斯托曼坚持十年的不断的游说下，加州大学同意放弃这个条款。

参考“BSD 许可证所存在的问题”，<http://www.gnu.org/philosophy/bsd.html>。

8. 译注：自由软件的英文是 Free Software，Free 这个词在英文中既有“自由”的意思，也有“免费”的意思。

9. 参考 O'Reilly & Associates, Inc. 于 1999 年出版的《开源软件文集》一书中第 139 页，迈克尔·蒂曼所写的“Cygus Solutions 公司的未来”一文。<http://www.oreilly.com/catalog/opensources/book/tiemans.html>。

10. 参考 BYTE 杂志 1986 年对理查德·斯托曼的采访。

11. 参考“Hurd 历史”一文。<http://www.gnu.org/software/>

hurd/history.html。

12. 根据自由编程联盟的发布的文章 (<http://progfree.org/Links/prep.ai.mit.edu/demo.final.release>)，这次抗议很引人注目的原因之一是他们使用了一个十六进制的口号：

1-2-3-4，把律师们抛出门外

5-6-7-8，创新不应被起诉

9-A-B-C，我们不用 1-2-3

D-E-F-0，再见“视觉体验”之争

13. 在这里用“编写”这个词并不是很严密。斯托曼获得麦克阿瑟奖的时候，他的手患上一种慢性病，经常疼痛，所以那时大部分打字的工作都是由 FSF 雇佣的一名打字员来完成的。有些人认为斯托曼所得的病就是在程序员群体中很常见的重复使力伤害症 (RSI)，但他自己对此并不是很确定。他说：“这不是腕管综合症，因为我是手疼，不是手腕关节疼。”后来斯托曼开始换用一种按键更为轻松的键盘，免去了请打字员录入的麻烦。
14. 参考鲁文·勒纳 (Reuven Lerner) “斯托曼荣获 24 万美元麦克阿瑟奖”一文，麻省理工学院，The Tech 杂志，1990 年 7 月 18 日。<http://the-tech.mit.edu/V110/N30/rms.30n.html>。
15. 参考迈克尔·格罗斯“理查德·斯托曼：高中的弃儿、自由软件的象征、麦克阿瑟奖的天才获奖者”一文，1999 年。
16. 参考林纳斯·托瓦兹与大卫·戴蒙 (David Diamond) 《乐者为王：一场无意间发起的革命》一书，哈珀·柯林斯出版集团，2001 年，第 58~59 页。
17. 参考林纳斯·托瓦兹与大卫·戴蒙《乐者为王：一场无意间发起

的革命》一书，哈珀·柯林斯出版集团，2001年，78页。

18. 参考“Linux 10周年”。<http://www.linux10.org/history/>。
19. 参考林纳斯·托瓦兹与大卫·戴蒙《乐者为王：一场无意间发起的革命》一书，哈珀·柯林斯出版集团，2001年，第96~97页。
20. 参考林纳斯·托瓦兹与大卫·戴蒙《乐者为王：一场无意间发起的革命》一书，哈珀·柯林斯出版集团，2001年，94~95页。
21. 参考罗伯特·杨“Linux之父林纳斯访谈”，Linux通讯杂志，1994年3月1日。<http://www.linuxjournal.com/article.php?sid=2736>。
22. 译注：Control-R是Bash命令行上的一个快捷键，用于调阅以前曾经执行过的命令。这个快捷键在实际操作中非常实用，但不少Bash的用户都不知道它的存在。
23. 参考林纳斯·托瓦兹与大卫·戴蒙《乐者为王：一场无意间发起的革命》一书，哈珀·柯林斯出版集团，2001年，59页。

第 10 章

GNU/Linux

1993 年，自由软件运动的发展到了一个转折点。从乐观的角度讲，种种迹象都表明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在读者面前树立了自己的品牌形象。而 Internet 这个原本只在黑客和科学家群体中流行的词汇，也渐渐地融入了主流社会，就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，得到了广泛的普及。同时，黑客们所开发的软件也进入了新一代计算机用户的视野。然而，GNU 工程却还没完成它创建一个完整自由操作系统的目标，喜欢尝鲜的用户只能使用 Linux 作为一个替代系统。

不管你怎么想，这听起来仍然是个好消息，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，大众开始接纳这种文化。

事实确实如此吗？从悲观的角度来说，被大众所接受也就意味着麻烦正在滋长。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，这种疏远大众的繁荣真的算是一件好事吗？当然，白宫表明态度说 Internet 是一个很好的趋势，甚至已经在网上注册了自己的域名：`whitehouse.gov`，但是它也在召集各个公司、审查机关和执法机构一同开会，设法管理如同西部荒原般的 Internet 文化。是的，个人电脑正在变得越来越强大，但是 Intel 作为个人计算机市场上的芯片供应商，已经帮助软件公司一起，形成了一种由专有软件开发商垄断的局面。虽然有很多人通过 Linux 接触并开始使用自由软件，但与此同时，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。它没有像 GNU 那样的设计错误，也没有 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发展成这样。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（麻省理工学院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。整个体系最为精华的部分，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已有¹。”

这种所谓的惰性，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。从中可以看出，托瓦兹对于整件事并没有一个理智的规划。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·索亚粉刷篱笆一样，托瓦兹的天才能力并不表现在他的大局观上，而是在于他可以召集到一批黑客一起努力，用最快的速度完成任务。

托瓦兹和他找来一同工作的黑客们大获成功，而其他人则产生了一些疑问：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 宣言中所表述的自由软件哲学的一种表现形式呢？还是只是一个人在家用电脑上就可以土法炮制出来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，但这些版本的质量可谓是参差不齐。

“这就是红帽公司（Red Hat）或其他商业发行版出现前的情况。”伊恩·默多克（Ian Murdock）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’。这些公司有很多会在自己的发行版里添加些自己的程序和代码。然而，这些公司往往只干几年就销声匿迹了。”

作为一名 Unix 程序员，默多克还记得他第一次在家里的个人电脑上下载安装 Linux 时，被这个系统“扫地出门”的情景。“不过这仍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，参与到这个项目中的最好的的方式就是开发一个不添加

任何额外组件的 Linux 发行版。于是，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“激发人们的兴趣”，默多克在包括 comp.os.linux 这个 Usenet 新闻组在内的各种网上渠道发布了他的想法。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”的传奇人物。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 Linux 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 来完成一个操作系统。也就是说，好像斯托曼对我们所做的事很有兴趣，这和他们努力的方向是一致的。”

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，他都还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 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。”

斯托曼朋友的报告并不完全正确。Linux 是为基于 386 的机器所设计的，这就意味着 Linux 生根于这类低成本的平台。他的朋友没有发现 Linux 所内涵的一个重大优势，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，这对 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，西姆森·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。GNU 工程的成员，以及 GNU 工程的非营利性赞助商——自由软件基金会的一些成员回忆说，当时的大家的心情非常不好，甚至加芬克尔的文章对大家造成的伤害都已经显得不是那么重要了，“当时的情况很明显，至少在我看来是这样，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，“一旦错过了这个时间窗口，人们对新的系统就不会有太大的兴趣，但这却正是当时大家所面临的情况³。”

1990 年~1993 年期间，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埃里克·雷蒙德认为，造成一切问题的根源是体制。“自由软件基金会太骄傲了，”雷蒙德说，“他们把自己的目标从开发一个成熟的操作系统，转移到了进行操作系统的研究。他们认为除了自由软件基金会，没人可以对他们造成影响。”

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，”默多克说道，“微内核在 80 年代和 90 年代初期是一个非常热门的话题。很不幸的是，那正是 GNU 工程开始设计他们的内核的时间，结果就是带来了额外很多的包袱，这些包袱很难被轻易甩掉。”

斯托曼自己则寻找了一些客观因素来解释整个项目延期的原因。

莲花和苹果的官司带来了政治上的分心，加上那时斯托曼手上的病症，导致他不能全身心地投入到 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。“我们花了很多时间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，他和其他 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 Mach 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 团队遇到的问题时说，“如果有这个为基础，我们的进展会更为顺利一些。但是，事实上调试这些异步运行的多线程程序非常困难。一些时序处理上的小小问题就会导致文件损坏，这一点也不好玩。最终的结果就是，我们花了很多年的时间，还是只能开发出一个测试用的版本⁴。”

不管是一个借口还是一堆借口，当时的情况就是 Linux 内核的成功给 GNU 工程造成了一种紧张的气氛。毫无疑问，Linux 内核确实已经按 GPL 发布，但正如默多克自己所发现的那样，想要把 Linux 当成是一个纯自由软件操作系统还有很长的路要走。截至 1993 年年底，Linux 的总用户数已经从数十名 Minix 爱好者发展到 20000~100000 人⁵。曾经的个人爱好已经成长为一个快速发展的市场。就像温斯顿·丘吉尔（Winston Churchill）看到苏联军队攻入柏林的感觉一样，看到 Linux 的“胜利”，斯托曼心里像是打翻了五味瓶⁶。

虽说是赶了个晚集，但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，其他各方面的支持也开始源源不断的涌入。于是，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黛博拉（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Debian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。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化的产品，但这并不意味着Linux就比不上商业产品。如果你们有人不同意这个观点，我可以用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未来的时候了，不应该以Linux社区毁灭和它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助大家更多关注这些问题并一起想办法解决它们⁷。

宣言发布后不久，自由软件基金会就向Debian提出了一个要求。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的调查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。

斯托曼想过GNU前缀的方式去强调GNU在Linux系统中所发挥的重要作用，很多人也许根本就不会注意到这一点，默多克的观点则不尽相同。在那个时候，默多克把这个看成是消除GNU工程和Linux内核开发者之间紧张关系的一种尝试。默多克回忆说，“那时候两个社区之间出现了一些分裂，理查德对此非常重视。”

默多克认为最大的隔阂出现在glibc上。glibc是GNU C库的缩

写，用于让程序员可以直接调用操作系统内核所提供的“系统调用”。1993年~1994年间，glibc一度是Linux开发过程中最大的瓶颈。由于很多的Linux用户不断地向Linux内核添加系统调用，GNU工程的glibc的维护人员很快就被各种需求压得喘不过气来。由于GNU工程不断地延期，给人们留下了拖后腿的印象，有些Linux开发者提出要给glibc创建一个Linux专用的分支。

在黑客的世界中，分支是一个很有意思的现象。虽然黑客的行为道德中允许一个程序员对程序的代码做任何的改变，但是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，这样才能保证与其他人的程序保持最大程度上的兼容。如果在Linux开发的初期就对glibc创建分支，那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。

作为GNU工程的领导者，斯托曼早在1991年时就认识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⁸。

默多克说，当时Debian也是出于类似原因才对glibc创建了一个分支版本，这导致斯托曼坚持要在发布Debian发行版时加上GNU前缀。“这个分支最终还是逐渐与主干合并。然而，在那个时候，人们还是很担心Linux社区是否会把自己与GNU社区割裂开来，这样做会造成开发力量分散。”

斯托曼赞同默多克的看法。事实上，大部分主要的GNU组件在刚开始的时候都有多个分支版本。起初，斯托曼认为这些分支就像酸葡萄。与Linux内核开发组快速和作坊式的开发模式不同，GNU的源

代码维护者更倾向于用一种更慢更谨慎的方式去开发程序，以保证程序长期的活力。他们对于批评其他人的代码也毫不吝啬。然而，时间慢慢流逝，斯托曼在阅读 Linux 开发者的电子邮件时候，发觉到他们缺乏对 GNU 工程和它的目标的基本认识。

斯托曼说：“我们发现那些认为自己是 Linux 用户的人们并不关心 GNU 工程。他们说，‘为什么我们要花力气去做这些事？我不关心 GNU 工程。我只关心 Linux 可以正常运行，其他都与我无关。’这样的想法很让人吃惊，因为他们事实上正在使用着 GNU 系统的一个变种，而他们甚至比其他人更容易忽略这个事实。”

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。”默多克表示。

默多克很快建议，减少政治上的讨论，把更多精力集中到技术上来。虽然默多克对讨论政治问题已经习以为常，但说起设计和开发软件，他一定会坚持己见。这股默多克带来的风气很快蔓延到了别的 GNU 项目中。

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。实事求是地说，理查德是一个很难与之合作的人。”

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，佩伦斯是 Unix 下的知名软件 Electric Fence 的作者，这个软件也使用 GPL 许可证来发布。佩伦斯与默多克一样，在 GNU/Linux 刚刚展现出与 Unix 的巨大相似性时，就喜欢上了它。他也和默多克一样，虽然没有与斯托曼和自由软件基金会有过实际的接触，但他对他们的政治目标抱有极大的兴趣。

“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，”佩伦斯回忆起他在 80 年代末期第一次与斯托曼接触的情景，说道，“我写信给他，问他是如何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们工作的。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矛盾感到遗憾。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，但他仍然有能力去驾驭这样的复杂局面。“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。我个人觉得这样是可以的。我单方面做出了决定，采纳了这个名字，让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 年~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。在北卡罗来纳州，一个名叫红帽的 Unix 公司，把他们的注意力慢慢转向了 Linux。红帽公司的首席执行官是罗伯特·杨，他曾经是 Linux 通讯杂志的编辑，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，而是开始思考如果一家公司像 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。如果红帽公司对于 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（指的是托瓦兹和艾萨克·牛顿（Isaac Newton）爵士），”杨说，“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⁹。如果你是一个独立软件提供商并且想开发一个软件，比如你需要开发一个调制解调器拨号程序，但是你没有必要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，就可以立刻使用前人的成熟的果实。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 和万维网像一波外星人入侵一样深入地影响了整个市场，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，则是这一波外星人中最强大的一个。

对于试图讨好斯托曼但后来又厌恶斯托曼的微管理风格的伊恩·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。他也看到了托瓦兹在回复 Minix 作者安德鲁·塔嫩鲍姆的帖子中坦言：“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¹⁰。”像很多人一样，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，”默多克回忆道，“在同一段时间里，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 上的那些老帖子，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。”

到了 1996 年年底，所有这些“如果”都已经盖棺定论。不管是称呼它为 Linux 或是 GNU/Linux，用户们已经用行动证明了 Linux 的成功。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 的核心黑客们以外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每个人的眼前，并且充满活力。黑客们所要做的就是安静地坐下来，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。

准备好出发了吗？

尾注

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《大教堂与集市》中出现的（1997 年 5 月）。<http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>。
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. 查瑟尔所说的对于一个新的操作系统来说有 36 个月的“时间窗

口”的理论，并不仅仅适用于 GNU 工程。在 20 世纪 90 年代初，BSD 的自由版本由于他们与 Unix 系统实验室之间的官司，导致迟迟不能发布。虽然很多人认为 BSD 比 GNU/Linux 性能更好、更稳定，并坚持使用 FreeBSD 或 OpenBSD，但是跟使用 GNU/Linux 的人数相比，这依然只是一个很小的数字。

4. 参考斯托曼在茂宜高性能计算中心的演讲。
5. 很难准确统计出 GNU/Linux 的用户数，所以我在这里只是提供了一个非常宽泛的估计值。100000 这个数字来源于红帽公司的“里程碑”网站，<http://www.redhat.com/about/corporate/milestones.html>。
6. 用丘吉尔跟斯托曼作类比，是我个人的想法。后来斯托曼向我表达了他对丘吉尔的评价：第二次世界大战在我成长经历中一段不能抹去的记忆。就像丘吉尔所说的：“我们要与敌人在登陆地点作战，我们要与敌人在沙滩作战……我们决不投降！”这句话一直都能引起我的共鸣。
7. 参考伊恩·默多克的《Debian 简明发展史》，1994 年 1 月 6 日：附录 A，“Debian 宣言”，<http://www.debian.org/doc/manuals/project-history/apA.html>。
8. 杰米·泽文斯基（Jamie Zawinski）是 Mozilla 开发团队的一员，他曾经是 Lucid 公司的程序员，他维护了一个记录 Lucid/GNU Emacs 分支的网站，名为“The Lemacs/FSFmacs Schism”。<http://www.jwz.org/doc/lemacs.html>。
9. 杨在这里错误的使用了“公有领域”一词，公有领域就意味着不受版权保护。但使用 GPL 许可证的软件从根本就上保证了是受版权保护的。

10. 这句话引自在 Linux 刚刚发布以后，托瓦兹与塔嫩鲍姆之间一场有名的争论。托瓦兹为他选择移植性较差的单一内核的设计进行辩护，他说这样做是因为他想更深入的研究他的 386 电脑。“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。”参考克里斯·迪邦纳（Chris DiBona）等著的《开源软件文集》一书，O'Reilly & Associates, Inc., 1999 年，第 24 页。

第 11 章

开 源

1995 年 11 月，自由软件基金会成员彼得·萨卢斯（Peter Salus，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的剑桥召开的“自由发布软件大会”征集来自黑客们的论文。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者的参加的大会，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、NetBSD、386BSD、FreeBSD、Perl、Tcl/tk，以及其他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，自由软件和低成本软件变得越来

越重要。本次大会将召集各种可自由分发软件的开发者和通过各种渠道发布这类软件的发布者来参加。大会上会有各种教程和参考论文，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他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 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。20 世纪 80 年代，他在 GNU 工程中非常活跃，但到了 1992 年以后，他已经不再直接参与这个工程。与许多跟他经历类似的人一样，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，”雷蒙德回忆道，“这使我感觉到很沮丧，我再也不想与他合作了。”

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，雷蒙德表示了非常赞同。斯托曼是老一辈 ITS/Unix 黑客的传承人，而托瓦兹则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，尤其会受到像雷蒙德这样的年轻有抱负的黑客的欢迎。“我像是个脚踏两只船的人。”雷蒙德说。

会议召开的日子一天天临近，两个开发阵营之间的紧张关系却变得更加明显。两个团体的开发者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。在会议上，托瓦兹出人意料地证明了自己是一个出色而又平易近人的演讲者。除了一点轻微瑞典口音，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。雷蒙德说，更让人吃惊的是，托瓦兹在大会现场对其他杰出的黑客们进行了无情的抨击，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不会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这个幻灯片制作软件的发烧友。从老一辈的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宣扬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观点来看，这并不奇怪。为什么为了表明自己观点就得刻意去回避专有软件呢？作为黑客，不是为了去忍受，而是为了更好地完成工作。

“这是一个振聋发聩的观点，”雷蒙德回忆说，“然而，他可以表达这样的观点，因为在1995年和1996年，正是他在业界影响力达到巅峰的时候。”

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，只不过他记得在那次大会上，他被托瓦兹的厚脸皮刺激到了。“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，我知道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，黑客们给予托瓦兹高度的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 黑客是在专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专有软件呢？

雷蒙德曾经是一名 GNU 工程的成员，他能感觉到斯托曼与托瓦兹之间的紧张关系。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。在软件设计和人员管理方面，他以从不妥协而闻名。在 1996 年大会开如前不久，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。布莱恩·尤曼斯 (Brian Youmans)，萨卢斯招聘进入自由软件基金会的现任职员回忆到：“在那时候，彼得·萨卢斯是留在办公室中的唯一职员。”

对于雷蒙德来说，这次离职事件几乎肯定了一种猜测：最近 Hurd 项目进度的延迟和类似于 Lucid-Emacs 分裂的麻烦反映出来的是软件项目管理中的问题，而不是代码开发的问题。自由发布软件大会召开后不久，雷蒙德开始做他自己的玩具项目，一个名为 fetchmail 的邮件管理工具。根据托瓦兹提供的线索，雷蒙德在发布这个程序时附加上了一个承诺，承诺尽可能及时更新源代码。当用户们开始提交问题报告和新需求时，雷蒙德发现程序并不像他起初时想的那样陷入泥潭，而是惊人的健壮。通过分析托瓦兹成功的模式，雷蒙德很快得出了一个结论：把 Internet 当成一个“培养皿”，并且把黑客社区简单的监督看成是一个自然选择的过程，托瓦兹建立了一种革命性的模式，这种模式不依赖于集中的计划。

另一方面，雷蒙德觉得，托瓦兹找到了一种突破“布鲁克斯法则”的方式。在 IBM OS/360 项目经理佛瑞德·P·布鲁克斯 (Fred P.

Brooks) 于 1975 年出版的《人月神话》一书中, 布鲁克斯提出了“布鲁克斯法则”: 向一个软件项目中增加开发者的做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, 也做出了更好的软件³。

雷蒙德把他的发现写成了论文, 并把它设计成若干次演讲, 在宾夕法尼亚州切斯特县的一些朋友和邻居们面前宣传。在一次名为《大教堂与集市》的演讲中, 他对 GNU 工程和托瓦兹等内核黑客们所使用的管理风格进行了比较。雷蒙德说这些演讲的反响都很强烈, 不过反响最好的一次, 还得是次年春天他在德国召开的 1997 Linux Kongress 大会上的那次演讲。

“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《大教堂与集市》。这篇论文的题目是对雷蒙德的核心思想的精确概括。GNU 程序就是“教堂”, 它们都是有计划地修建而成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 个月以后, 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懒散的个性。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：“我觉得林纳斯最聪明的地方和最重要的黑客作品就是他创造出来 Linux 开发模式，而不仅仅是他所创造的 Linux 内核本身⁴。”

在总结托瓦兹的成功管理经验方面，雷蒙德自己也完成得非常出色。提姆·奥莱理（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（同时也是本书英文版的出版商）。在听完了雷蒙德在 Kongress 的演讲后，奥莱理马上就邀请雷蒙德参加下半年在加州蒙特雷举办的首届 Perl 大会，请他在那次会议上再次分享这个主题。

虽然这次会议的主题是 Perl，一种由著名的 Unix 黑客拉里·华尔设计的脚本语言。奥莱理承诺雷蒙德这次会议也会讨论其他自由软件的技术。那时商业界已经开始关注 Linux 和 Apache，Apache 是一种流行的自由 Web 服务器软件，奥莱理希望通过这次会议可以让自由软件更加流行，并用自由软件来建立一套完整的 Internet 的基础设施。从 Perl 或 Python 这些 Web 友好的开发语言到 BIND (the Berkeley Internet Naming Daemon) 这样后台的服务（用于把类似于 amazon.com 这样的容易记忆的域名转换为数字型式的 IP 地址），还有 sendmail 这样的流行的 Internet 电子邮件服务器程序，自由软件正创造着一个又一个的业界奇迹。自由软件社区存在的唯一问题就是缺少对整个社区的自我认识，总是像蚂蚁部落一样各自为营，在沙子里建造一个又一个的窝。奥莱理觉得雷蒙德的演讲可以引发人们对自由软件的认识，自由软件并不等同于 GNU 工程，它不会因为 GNU 工程的开始而诞生，同样也不会因为 GNU 工程的结束而消亡。Perl 和 Python 这样

的开发语言，BIND、sendmail 和 Apache 这样的 Internet 软件已经证明了自由软件已经无处不在，并具有强大的影响力。奥莱理还承诺雷蒙德，在这次会议上的演讲会比在 Linux Kongress 上的演讲更加轰动。

奥莱理是对的。雷蒙德笑了笑，说：“这次，我的演讲获得了听众的起身鼓掌。”

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。网景公司（Netscape）就是其中的一个。这家成立于加利福尼亚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 浏览器的市场的斗争。

出于对雷蒙德演讲的好奇和希望赢回失去的市场份额的目的，网景公司的高管们把这次演讲所传达的信息带回了公司总部。几个月以后，1998 年 1 月，网景宣布计划开放它的旗舰级产品 Navigator Web 浏览器的源代码，以此希望能得到黑客们的支持，继续完成后续的开发。

网景公司的首席执行官吉姆·巴克斯代尔（Jim Barksdale）说：雷蒙德那篇《大教堂与集市》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，向网景的高管们提供建议，并且参与了开放 Netscape Navigator 源代码的庆祝聚会。Navigator 的源代码的 code name 是“Mozilla”，这个名字一方面象征着这个程序的源代码有 30 万行的代码，同时也象征着它的尊贵血统。作为一个浏览器鼻祖 Mosaic 的商业化版本，马克·安德生（Marc Andreessen）在伊利诺伊大学开发 Navigator 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。

雷蒙德在加州的那段时间，还找到了一次去 VA Research 参观的

机会。VA Research 公司坐落于圣克拉拉市 (Santa Clara)，专职销售一些预装 GNU/Linux 操作系统的工作站。雷蒙德召集了一次小规模会议，仅仅邀请了 VA 的创始人拉里·奥古斯丁 (Larry Augustin)、少数几名 VA 的雇员以及克里斯汀·彼得森 (Christine Peterson)，他是硅谷前瞻学会 (Foresight Institute) 的主席，他们专注纳米科技。

“这次会议的议程最终归结为一点：如何来总结网景公司的决策，以便让别的公司可以去效仿？”雷蒙德不记得这次讨论的内容，但是他记得这次会议中提到的另一个问题：不管斯托曼和其他黑客们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”，这一点仍然没有赢得大众普遍的认知。很多商业管理者第一次听到这个名词时，都把它想当然地认为是“零成本”，而并不注意到其他相关的信息。除非黑客们能找到一种方法解决这个认知上的偏差，否则，即使有网景公司作为榜样，自由软件运动想要获得最终成功，还是得面临一个艰难的登顶过程。

于是，彼德森成立了一个组织，专门致力于推广自由软件，他们使用了另一个词语：开源。

彼德森回忆道，她是在与一个朋友讨论网景公司所作的决定的时候，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。

彼德森回忆，在那次会议上，这个词语得到的反馈却是截然不同的。她说：“我很犹豫要不要提出这个建议，因为我对这个组织并不了解，所以我故意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。

Mozilla 发布会前几天，奥莱理组织了一次讨论自由软件的会议。雷蒙德说，在这次会议之前，他从来不公开使用“开源”这个词语来替代“自由软件”。奥莱理把这次会议叫做“免费软件峰会”，期望把媒体和社区的注意力吸引到其他一些有价值的项目上去，因为这些项目对于网景公司决定发布 Mozilla 也起到了积极的作用。“参会的那些家伙有很多相似之处，但我很惊讶他们之前互相之间都不认识，”奥莱理说，“我同时也想让全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，是奥莱理一个深思熟虑的决定，并将产生深远的政治影响。他原本只打算邀请西海岸的软件开发者们，比如华尔、Sendmail 的作者埃里克·奥尔曼（Eric Allman），BIND 的作者保罗·维克西（Paul Vixie）。不过也有一个人是例外，那就是长居在宾夕法尼亚州的雷蒙德，因为他正好来参加 Mozilla 的发布会，所以也马上邀请了他。同时还有来自弗吉尼亚州的吉多·范罗苏姆（Guido van Rossum），他是 Python 语言的作者。“弗兰克·威利森（Frank Willison）是出版社的主编，他是 Python 的斗士，他决定邀请吉多参加，甚至都没有跟我商量，”奥莱理回忆说，“我非常高兴能邀请到他出席，不过刚开始的时候，我确实只是想组织一个本地聚会。”

从一些观察者的视角来看，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，他却认为即使邀请斯托曼也是徒劳的。有关这次会议的组织者奥莱理怠慢斯托曼的传言，在奥莱理与斯托曼公开表明在软件手册的权限问题上的分歧后，变得更能让人信服。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。而奥莱理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。奥莱理和斯托曼还为这次会议的名称发生了争论，斯托曼坚持要使用“自由软件（Free Software）”来代替不那么政治化的“免费软件（Freeware）”一词。

回头去看，奥莱理认为没有邀请斯托曼的决定并不是因为怠慢了他。奥莱理说：“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·吉尔摩和迈克尔·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，奥莱理和雷蒙德回忆说，在那次峰会上，“开源”一词成功地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，跟先前在 VA 召开的那次会议一样，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，奥莱理记得峰会的与会者托瓦兹的一段很有见地评论：“那时候，林纳斯才刚刚搬到硅谷来，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。”

Cygnus 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，”奥莱理回忆道，“直到埃里克抛出了‘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”。奥莱理说，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。奥莱理说：“我们希望能向外界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，奥莱理带着与会者们参加了一个有纽约时报、华尔街日报和其他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语言的作者拉里·华尔和Apache团队领导者布莱恩·贝伦多夫（Brian Behlendorf）的照片。开源开始向商业市场敞开怀抱。

对于像蒂曼这样的参会者来说，向外界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他的程序员和企业在面对自由软件时还是觉得困难重重。

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市场来说更加友好，也更容易引起商业市场的注意。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”这个新词汇做出回应。雷蒙德说斯托曼只是简单对这个词汇进行了一下了解，其实目的只是为了找个理由抛弃它。“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。”

1998年年底，斯托曼正式表明了立场：在讨论自由软件技术优点

时，“开源”是一个很有利于沟通的词汇，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，因此斯托曼将坚持继续使用“自由软件”这个词汇。

1999年召开的LinuxWorld博览会，托瓦兹把它看成是宣传Linux社区产出的大好机会，在这次会议上，斯托曼总结了立场，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，所以我们不需要为了去讨好公司而放弃我们的目标。如果某些公司能向我们提供帮助，那么我们可以接受。如果他们不愿意，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LinuxWorld大会之前，斯托曼已经表现出他正在慢慢疏远他的同盟。免费软件峰会召开后几个月，奥莱理主办了他的第二年度Perl大会。这一次，斯托曼参加了会议。在一次有关IBM决定在它们的商用方案中整合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，把矛头指向了与会者之一的约翰·欧斯特霍特(John Ousterhout)，他是Tcl脚本语言的作者。斯托曼把欧斯特霍特称作是自由软件社区的寄生虫，因为他通过他自己开办的名为Scriptics的创业公司销售一个商业版本的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表示⁵。

“当时的场面非常尴尬，”Prime Time Freeware公司的李奇·莫林回忆说，“约翰做出了一些很让人值得尊敬的东西——Tcl、Tk和Sprite。他是自由软件社区中一个真正的贡献者。”

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兴的人们。

斯托曼在 Perl 大会上的爆发也影响了另一位潜在的追随者——布斯·佩伦斯。1998 年，埃里克·雷蒙德打算发起开源倡议组织——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。雷蒙德请佩伦斯来起划这个标准⁷。

佩伦斯不久就从 OSI 的高管位置上辞职，他对这个组织与斯托曼和自由软件基金会対立的立场表示遗憾。回忆往事，从在自由软件基金会以外重新定义自由软件的需要来看，佩伦斯理解了为什么其他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地处理各方关系的话，他会把事情完成得更好。我觉得他应该花几个月时间远离自由软件，静下来好好思考一下。”

斯托曼的个人偏执并不足以对抗开源斗士对商业界的公关势头。1998 年 8 月，芯片厂商 Intel 购买了 GNU/Linux 供应商红帽公司的股份，纽约时报在随后的文章中评价红帽公司为“自由软件和开源”运动的产物⁸。半年后，约翰·马尔科夫（John Markoff）在苹果电脑杂志上发表文章，标题中称该公司开始使用“开源”的 Apache 服务器⁹。

这样的势头与很多公司拥抱“开源”一词的势头非常一致。到了 1999 年 8 月，红帽这家乐意把自己标榜为“开源”的公司在纳斯达克上市。同年 12 月，VA Linux 公司（曾经的 VA Research）正式 IPO。这家公司的股票价格从开盘的每股 30 美元一路彪升到 300 美元，并最后回稳在 239 美元左右。股东们很幸运地抄了底，并且享受到了 698% 的纸面收益，打破了纳斯达克的历史记录。

埃里克·雷蒙德也是这些幸运的股东们中的一员，从 Mozilla 发布开始，他就是 VA Linux 公司董事会成员，他收获了 150000 股的 VA Linux 股票。雷蒙德那篇对比斯托曼和托瓦兹的管理风格的文章给他的净资产

增值了 36 万美元，这让他自己也觉得很震惊，于是他又写了一篇文章，在这篇文章中，雷蒙德思考了黑客道德与货币财富之间的关系：

最近记者们经常问我是否认为开源社区会因为大量资金的注入而被破坏。我对他们说，我所相信的是商业公司对程序员的需求一直非常旺盛，所以，任何一个只是渴求金钱的人应该早就离开了社区。我们的社区是自我选择的结果，人们所关注的是其他的东西，比如：成就感、骄傲、艺术热情和彼此之间的关系¹⁰。

不管这些评论是否减轻了雷蒙德和其他开源拥护者的行为是以金钱为目的的嫌疑，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、托瓦兹和其他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，他认为我和林纳斯·托瓦兹串通搞阴谋绑架他的革命。理查德拒绝使用开源一词，并且故意认为我的观点是意识形态上的分歧，这是由于他自己的理想主义和领地主义的混合结果。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。但我不这样认为。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地绑在了一起，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。而雷蒙德则还没有到达这样的高度。斯托曼时常手持武器，充满攻击性。但这不是因为出于原则，而更多是出于习惯：

比如，他一开始反对 Linux 内核，现在则是作为一个自由软件运动的政治形象考虑，不愿意在自由软件理想以外的领域去冒险。

除此之外，近期关于开源的争论也表明：当斯托曼手持武器时，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：“斯托曼最重要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：“你可以坚信他永远会坚持他自己的观点，虽然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。

尾注

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）。
<http://bat8.inria.fr/~lang/hotlist/free/licence/fsf96/call-for-papers.html>。

2. 尽管林纳斯·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/>）对此做出如下解释：

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“finlandssvenskar”，并且把他们自己认定是芬兰人。他们中的很多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。

3. “布鲁克斯法则”是对他的书中如下描述的简称：

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。

参考佛瑞德·布鲁克斯，《人月神话》(Addison Wesley Publishing, 1995)。

4. 参考埃里克·雷蒙德的文章《大教堂与集市》，1997年。
5. 参考马尔科姆·麦克拉伦 (Malcolm Maclachlan) 1998年8月26日发表在 TechWeb News 网站的文章“利益驱使导致开源运动的分裂”一文。<http://www.techweb.com/article/showArticle?articleID=29102344>。
6. 译注：原文在这里使用了 libre 和 gratis 这两个词汇来表达“自由”与“免费”的含义。
7. 参考布鲁斯·佩伦斯等人的“开源定义”，来自开源倡议组织，1998年。<http://www.opensource.org/docs/definition.html>。
8. 参考艾米·哈蒙 (Amy Harmon) “大减价：自由操作系统”，纽约时报，1998年9月28日。<http://www.nytimes.com/library/tech/98/09/biztech/articles/28linux.html>。
9. 参考约翰·马尔科夫“苹果在其服务器上使用‘开源’软件”，纽约时报，1999年3月17日。<http://www.nytimes.com/library/tech/99/03/biztech/articles/17apple.html>。
10. 参考埃里克·雷蒙德“被财富所震惊”，今日 Linux，1999年12月10日。http://linuxtoday.com/news_story.php3?ltsn=1999-12-10-001-05-NW-LF。

理查德·斯托曼，这位麻省理工学院人工智能实验室的创始人，一个极具争议性的人物，在黑客世界中，他被誉为“黑客之父”，也是自由软件运动的奠基人。

理查德·斯托曼，这位麻省理工学院人工智能实验室的创始人，一个极具争议性的人物，在黑客世界中，他被誉为“黑客之父”，也是自由软件运动的奠基人。他创立了自由软件基金会，并开发了GNU操作系统。他的目标是创建一个完全自由的操作系统，让用户可以自由地使用、修改和分发软件。他的这一理念在黑客社区中得到了广泛的支持，并最终促成了Linux操作系统的诞生。

第 12 章

开往黑客地狱的短暂旅途

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的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后将在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。演讲开始前，津汇对我们的到来显得如此热情，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

缘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。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。

斯托曼叹着气失望地说：“这实在是太可怕了，我们为什么没走另一条路呢？”

终于，我们前方四分之一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离。十分钟内，这样的过程重复了好几次，直到我们最终慢慢地挪到了相邻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。

斯托曼沮丧地挥动着双手，抱怨道：“他为什么不拐弯呢？你们不觉得奇怪吗？”

我决定不回答这两个问题。我知道我正与斯托曼在茂宜岛同乘一辆车，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，我尽可能地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。不过好运气依然没有眷顾我们，汽车又一次慢慢地驶过了十字路口，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，在接下来的五分钟里，我们只能在他的后视镜里看到他的一小部分脑袋。

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幅美丽的画面。这是一副美得让人窒息的画面。我想，如果你是一个夏威夷本地人，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。我哑了哑嘴。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地到达目的地，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带来的低效才激发了斯托曼想要获取打印机源代码的想法。也正是因为要从头编写被商业公司所绑架的商用软件低效，才引发了斯托曼与 Symbolics 公司之间的斗争，从而诞生了 GNU

工程。让·保罗·萨特（Jean Paul Sartre）曾经说，如果其他人是地狱，那么黑客的地狱就是重复其他人愚蠢的错误，毫不夸张地说，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被美景环绕的城市时变得更为明显。四处都是商场、停车场和缺乏设计的信号灯，这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地变化，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），应该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，是我们向导的愚笨，他没有像一个黑客那样本能地做出最聪明的选择，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地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。斯托曼耸起的肩膀终于放松了一些，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。”

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，他摇了摇了头，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。从我们租来的车紧闭的窗户望出去，我们似乎看到了这么一句话：“嘿，这就是茂宜岛，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，“这全是你的错。如果听我的走另一条路就不会这样了。”

斯托曼加重了“听我的”一词，紧紧的抓住方向盘并两次把它拽向自己。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。这里可以容易地驶入高速公路，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。

接下来的二十分钟，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。

尾注

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。

第 13 章

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。

《大教堂与集市》一书出版后四年，雷蒙德给斯托曼留下的伤口虽已收疤，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 年在 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（Luke Skywalker）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！”斯托曼悲哀地说，“拿起武器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，所以他并不想发起战争。战争有什么好的呢！”

与此同时，对于托瓦兹来说，“拿起武器”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道德仲裁人的影响力。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。虽然 GNU/Linux 出人意料的成功让自由软件运动变得有些边缘化，但是斯托曼还是成功地找到了他的位置。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的理解存在更多弦外之音的国家。

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专有软件控制权的 Sun 公司宣布对他们的 OpenOffice 应用套件采用 LGPL 和 Sun Industry Standards Source License (SISSL) 进行双重许可。

对于这些胜利，其实斯托曼并没有做什么特别的努力。对于 Troll Tech 的案例来说，斯托曼只是扮演了自由软件教主的角色。1999 年，Troll Tech 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。Troll Tech 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 许可证，另一个使用 QPL 许可证，让开发者可以有办法绕开斯托曼所发现的许可证条款中的冲突。

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 公司的共同创始人和首席科学家比尔·乔伊为该公司的“community source”许可证辩护，这个许可证本质是放松了限制，用户无须与 Sun 公司签署协议就可以免费复制或修改 Sun 的软件。乔伊发表演讲一年后，Sun 公司的副总经理马科·博瑞斯（Marco Boerries）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 公司的决定与斯托曼没有什么直接的关系，而是更多的因为 GPL 软件蓬勃发展的势头。“现在的实际情况是，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¹。”

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（Eben Moglen）说：“他一定不是一个律师，不然怎么可能把 GPL 写成这个样子。”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·斯托曼所坚持的设计哲学。”

莫格林曾经也是一名专业程序员，在 1990 年时为斯托曼提供过无偿的法律服务，那时斯托曼在一件私人事件上请求莫格林的法律援助。莫格林后来还协助加密学专家菲利普·齐默曼（Phillip Zimmerman）与联邦政府进行相关的法律斗争，对于能够在法律事务上帮助斯托曼，他感到非常荣幸：“我跟斯托曼讲，我每天都在使用 Emacs，要还这个人情，我必须义务提供一大堆法律服务才能对得

起良心。”

从那以后，莫格林也许比任何一个人都有机会去近距离研究斯托曼的黑客哲学在法律领域中的表现形式。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往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的自由软件运动进程的人，莫格林非常理解那种不确定的同盟所带来的挫败感。莫格林说：“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，他从不尝试通过语言上的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之不可少的。”

由于自由软件基金会不愿意在一些 GNU 开发和 GPL 限制以外的问题上费太多心思，莫格林开始把他过盛的精力放在帮助电子前哨基金会（Electronic Frontier Foundation）的事务上面，这个组织向近期一些版权相关的被告提供法律援助，比如德米特里·斯柯里亚夫诺（Dmitry Sklyarov）。2000 年，莫格林同时也是一群黑客的代理律师，帮助他们处理有 DVD 的解密程序 deCSS 相关的事务。在这两个案子中，他的主要代理人没有什么明确的想法，这时莫格林才真正理解了斯托曼的倔强个性。“这些年中，我不止一次的跟理查德说：‘我们需要这么做，我们需要那么做。’这是一个战略布局。下一步我们这

么做。这是我们需要去做的。’但是理查德的回答永远都是‘我们不需要做任何事’。只需要静静地等着，那些需要完成的事都会慢慢完成的。”

“你知道么？”莫格林补充说，“总体上来说，他是正确的。”

莫格林这样的评价否定了斯托曼对自己的评价：“我不是个能运筹帷幄的人。”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基础，也就是说，‘让我们一起尽可能地打好基础吧。’”

GPL 变得越来越有名，并且继续保持它对外界的吸引力，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道德框架来树立他的新形象。其他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。如今的自由软件运动早已没有了名义上的领导，但却能自我组织，运行良好。面对这样的事实，人们也许更希望超脱于软件行业内的明争暗斗，从严肃的历史角度去审视斯托曼的一生。

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。”

毫无疑问，斯托曼的个性让他疏远了很多潜在的同盟。然而，

这也是他直率、讲究道德底线的一个体现。在必要的时候，以前的一些政治对手仍然愿意帮他说一些好话。斯托曼的理论与斯托曼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那种强烈的个性，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“一百年”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，从一百年以后往回看自由软件运动。不难发现斯托曼与以前的一些美国历史名人之间的相似之处，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，亨利·戴维·梭罗（Henry David Thoreau），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（John Muir），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（William Jennings Bryan），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，他虽然曾经很强势，但是他的光辉已经在历史的长河中渐渐地褪去。

斯托曼并不是第一个将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。基于这样的事实，看上去很值得脱离现在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，总是会迫切地想去解决它。莫格林说：“一百年以后，理查德和其他一些人也许不仅仅能在历史中留下一笔，他们还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（Theodor Holm Nelson），也叫泰德·尼尔森（Ted Nelson），他是 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的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一些不太道德的地方，”莫格林说，“这与斯托曼本人的个性非常相关，很多人都有兴趣去研究他的这种个性，但是人们常常会把它描写成理查德·斯托曼个性中的一种缺陷。”

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（Thomas Jefferson）的一样伟大：杰斐逊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。

“斯托曼另一项被人们所忽视的遗产，”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 许可证或任何其他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 年出版的《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（Lawrence Lessig）表达了类似的乐观观点。与其他法律界的学者一样，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，在过去的三十年中，带来了 Internet 的指数级的增长势头。与其他的 Internet 先驱相比，比如万尼瓦尔·布什（Vannevar Bush）、文顿·瑟夫（Vinton Cerf）和约瑟夫·利克莱德（J. C. R. Licklider）这些让大众看到计算机技术发展的人，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]把这些争论从“是不是”变为了“应不应该”。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，如果透过水晶球去看 2102 年的情景，也许会看到更为多样的信息：

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，他也看到很多的可能性：

二十年后，历史对 GNU 工程的评价，会取决于谁赢得了自由的斗争，正确地去使用公共知识。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地认识到 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，也更多地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（John Brown）。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：一百年后，尤其在 20 世纪初，历史教科书都把他描绘成一个疯子。种族隔离时期，种族间的隔阂还十分严重。那时，美国还部分地接受着南方的种族歧视主义观点。那会儿的教科书，说起美国南北战争时，依旧会提到一些种族主义观点。

这样的比较，记录了斯托曼工作自省的理性性格特点，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美国内战后重建时声名狼藉的布朗那么低，但是除了他难得使用一些言辞激烈的类比去攻击别人，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会是一样。让自由软件这个词汇变得流行，不像是一个群体运动，倒像是一系列与专有化诱惑进行的个人战斗。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。

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财富。莫格林在过去的十年中密切关注着事态的变化，他不断地提醒那些误解了斯托曼个性的人，斯托曼的个性并不是他在生活中的一种缺陷。莫格林曾说过，如果没有这样的个性，就没有这么多可以讨论的作品。作为一名曾经在最高法院工作的职员，莫格林说：

我曾经服务过的最伟大的人是瑟古德·马歇尔 (Thurgood Marshall)。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。我并不想把斯托曼与马歇尔作比较，因为他们两人并没有什么共同点。瑟古德·马歇尔是一个善于社交的人，抛开他所处的环境来看，他也只是人类社会中的一员。斯托曼与他在很多方面都并不相似，但是如果说要在茫茫人海中寻找一个可以与马歇尔相提并论的人物，斯托曼依然是其中一颗耀眼的明星。

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，很多的自由软件相关的内容在这些新闻中被提及。莫格林回忆起他与斯托曼在一起共进午餐时的情景，他们正被暴风雨般的文章和评论所包围，就像是被大风吹进台风眼中的遇难者。在接下来的几个小时中，他说，他们的对话渐渐地转向一个主题：加强 GPL 的地位。

“我们坐在那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，我简单地想了想，对于旁边路过的行人来说，我们是什么样的一个形象。这就是我们，两个留着络腮胡的无政府主义者，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扔进了汤里。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。这里就是革命开始的地方，这个人就是发起这场革命的领导者。”

莫格林说在那一刻，比其他任何时候，都准确地体现出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。’然后我还得给他付午餐钱，因为他没有钱来付。”

尾注

1. 参考作者对马科·博瑞斯的采访（2000年7月）。

第 14 章

跋：粉碎孤独

为一个还没有去世的人撰写传记有点像是在表演话剧，台前的演出与幕后的故事相比，总是显得苍白无力。

在《马尔科姆·艾克斯传记》一书中，作者亚历克斯·哈里（Alex Haley）把幕后故事搬到了台前。在这本书的后记中，哈里不再以代笔人的身份出现，而是表达了自己的心声。

我并不想把本书与《马尔科姆·艾克斯传记》进行什么比较，不过我还欠哈里一个人情：如果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，他的那篇后记被我当成是一本行事指南，指导我解决如何与一个从来不愿意与他人合作的传记人物合

作的问题。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。

正如本书中所记载的那样，幕后的故事开始于奥克兰的一个公寓，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利福尼亚州的塞瓦斯托波尔——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对理查德·M·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/>) 网站在它的“每日精彩文章”栏目中给了这篇访谈一个链接。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 Research）的一个热门的 Nerd 新闻聚合网站。短短几小时内，BeOpen 网站的访问量激增，读者们蜂拥而来。

如果只是按预期的设想，故事到这里就该结束了。采访结束 3 个月 after，当我在加利福尼亚州的蒙特雷参加 O'Reilly 的开源软件大会时，我收到一封来自特蕾西·帕蒂森 (Tracy Pattison) 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，觉得非常有趣。很久以来，我对 RMS 以及他的作品都很有兴趣，读到您的作品感觉很高兴，因为我觉得您的

作品非常出色，它准确地表述了斯托曼从事 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。

我来信的目的是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。您那里是否还有更多的信息或消息源可以把您的访谈作进一步的展开，并把它写成斯托曼的一本传记吗？可以往里面添加更多有关他个性的花絮和背景资料，这样也许能让非程序员群体对这部作品产生更大的兴趣。

特蕾西让我打电话跟她进一步讨论这个想法，我照做了。特蕾西告诉我，她的公司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 字左右的内容，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作为选题，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。

特蕾西的原话是这样的：您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。特蕾西建议我在正式签署合同之前先准备一个故事的大纲，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，起初我并没想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发给我的邮件，指责我不应该使用“Linux”一词，而应该用“GNU/Linux”。

不过，我也非常清楚，斯托曼一直期望能通过各种渠道向公众表

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。如果不能得到他的积极配合，我就只能依赖从网上找来的各种文档、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，我曾经看到过一篇斯托曼在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集合了各种软件罪恶的产物。斯托曼认为，电子书的罪孽不但在于读者们需要使用一些专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，购买电子书就像是购买了一把无法转让的钥匙，只有使用这把钥匙才能开启加密的内容。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”上其他内容，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。不再有不付钱就阅读的“漏洞”（如果从微软公司阅读器的广告内容来看，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。

不用说，这篇文章引起了我们的一些担忧。特蕾西和我都没有谈

论过有关她的公司所使用的软件或是电子书的版权协议会是什么样。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。

我很希望可以尽快投入到撰写传记的工作中去，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，斯托曼很明确地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？”

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。”斯托曼说。

对斯托曼而言，相比电子书阅读软件的问题，版权是个更大的问题。他说他可以忽略出版社或相关的第三方公司使用什么软件来发布电子书的问题，他只要出版社在版权声明中说明，允许读者自由地发布电子书的内容。斯托曼说，可以参考斯蒂芬·金（Stephen King）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表声明：他将连载的形式出版《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。截至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地原样复制的话，我就可以接受。”斯托曼说。

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双赢的协议，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，而且没有再次询问有关电子书版权协商的进展。在第一次采访后不久，我又趁热打铁地安排了第二次采

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。

正当斯托曼在塔希提岛度假时，从特蕾西那里传来了坏消息。她们公司的法务部门不愿意改变这本电子书的版权声明。如果读者想要把这本电子书转移给别人，就必须对电子书进行破解，或者把它转换成其他的开放格式，比如 HTML。而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（Henning Guttman）。这是我们的第一次会面，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，为了避免让读者逃避付费，他们并不乐意对版权条款进行改变。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，我已经采访了两次斯托曼，并且告诉他我已经承诺斯托曼我们不会以让他看起来像个“伪君子”一样去出版本书。亨宁理解我被道德所绑架的困境，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道德致敬的电子书，这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 2001 年，俄罗斯黑客德米特里·斯柯里亚诺夫（Dmitry Sklyarov）来美国参加了著名的 DefCon 大会并介绍了一些破解电子书的方法，结果却在回国前被联邦调查局逮捕入狱。这件事情后来在网上引发了大范围的讨论。不过在这事发生之前，我们也早能

猜到，破解现有的电子书只是时间问题。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，那无异于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可以使用的妥协方案。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 Sun 公司对 Open Office 这个自由桌面办公套件所做的那样。这样的话，出版社就能以正常的电子书出版方式出版一个商业版本的电子书，既可以发挥那些电子书软件的长处，又能同时再以 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，他觉得双重许可证的方式过于累赘。他之所以可以接受 Sun 的 Open Office 使用双重许可，是因为他没有能力控制这个决定的制定。对于这本书而言，他完全有能力控制最终的产出，他有权绝拒合作。

我的其他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，同意将电子书的版权条款改为“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自由的。我告诉斯托曼我不能答应这个说法，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“Happy hacking”，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从对斯托曼的道德义务中解救出来。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，也还面临着可能被其他合作伙伴起诉的风险。想要让出版社管理层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，我和亨宁一起去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，以便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。

斯托曼很粗鲁地回复道：“那么，为什么我需要因为他们的合同限制而做出妥协？”

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。

“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，在自由和商业之间做出选择。”

“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，人行道上满是步伐匆匆的行人。来到这个酒吧时，我很高兴地发现，这个地方距离那首1976年拉莫内斯（Ramones）的歌曲“53rd and 3rd”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。当这些烦

心的事情都凑到一起，我感觉到自己就像歌中所唱的那个一直遭遇挫折的家伙，精神就在崩溃的边缘了。说来也真是讽刺，这几周，我都忙着在记录被斯托曼折磨得死去活来的人。而如今，我自己却在百般努力地重复着这些人的老路，试图完成一件早已被证明是不可能完成的任务：说服斯托曼，让他妥协。

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？你是在耍我吗？你是想按他们的意志去办吗？”斯托曼咄咄逼人地问道。

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。”斯托曼冷冷地说。

“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。

“你就不能按我所交待你的去做吗？”他咆哮道。

我当时应该是一根筋地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。”

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，

当我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，自己的生活就像是电影一样。

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，只不过我还需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，这里就应该加入一个短暂、浪漫的插曲。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纽约东城，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，并且小心地避免谈论工作。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。这次交谈确实非常愉快，让我们忘记了有关斯托曼的电子书的问题。

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。一位新闻学院的老教授建议我忘掉斯托曼所谓“伪君子”的说法，把故事完整地写下来。不少同行都很了解斯托曼与媒体打交道的精明能力，他们对我的处境表达了同情，但他们一致说：决定权在你。

我决定把这本书的出版事宜先放到一边。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。不过，现在我有了不预先通过亨宁而直接与特蕾西沟通的机会。到了这年的圣诞节，我和特蕾西扯平了相互访问的次数，她飞到西海岸来了一次，我也第2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。2月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。我和特蕾西的婚礼定在5月11日。虽然这传记成了死胎，我们却收获了一份美满的姻缘。

整个夏天，我开始考虑把我的采访笔记改编成杂志的文章。从道德上来说，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。6月，在斯托曼在纽约大学发表演讲后一周，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信心。

7月，那是收到特蕾西的第一封电子邮件后的整整一年，我接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，同时它也是为埃里克·雷蒙德出版《大教堂与集市》一书的公司。作为一个记者，我以前也重度依赖O'Reilly出版的《开源软件文集》一书来作为重要的史料。我也知道该公司出版的图书中，有不少的章节，包括斯托曼曾经写过的一章，都是以允许随意分布的版权条款来出版的。如果这次再与出版

社在电子出版的问题上有分歧，那这些背景信息就会很有用。

果不其然，电子出版还是遇到了一些困难。亨宁告诉我说，O'Reilly 打算把这本传记同时以纸质书和新的 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²，但是 O'Reilly 同意允许让用户在任意媒介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：出版开放许可证（Open Publication License, OPL）或 GNU 自由文档许可证（GFDL）。

我研究了这两种许可证的内容和背景。出版开放许可证（OPL）³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⁴。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，则需要封面和封底上提供一些必要的许可证相关的信息。

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”⁵ 文章页面上找到了一些斯托曼对于出版开放许可证的批评。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，这还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，看着一本书不断地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⁶？

当 O'Reilly 的编辑劳里·帕瑞其 (Laurie Petrycki) 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，电子书已经成了一个几乎无人关心的话题。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电子书的推广。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的方式，那为什么还非得出电子书？

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。虽然去年发生过种种不愉快的事情，我还是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，他也算是促成我们联姻的半个媒人。”

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，我和特蕾西差不多可以听清与斯托曼对话的人的声音。她是莎拉，一个二十多岁的年轻女子。

“我私自邀请了其他人一起参加我们的饭局”，斯托曼诚恳地说，他狡猾地微笑着，跟当年和他在帕罗奥多的饭店里见面时看到一样。

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，理查德去接受武田奖”，李普曼当时是这么跟我说的⁷。

在去饭店的路上，我听莎拉讲了她和理查德头一次见面的情景。有意思的是，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年年初就开始约会了，莎拉回忆道。

“我非常景仰理查德建立起这整个自由软件运动的方式，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：“那么，他真正的想法是什么？”

莎拉马上回答说：“粉碎孤独。”

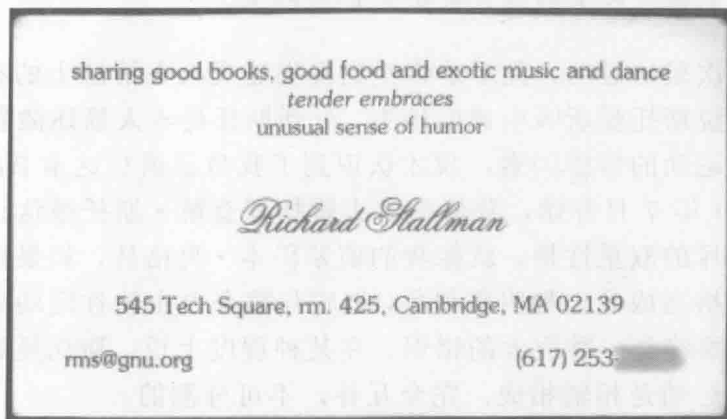
晚餐期间，我尽量让两位女士多说点话，并且尝试着看看一年来斯托曼是不是多少有些改观，不再那么倔强。但是一路谈下来，他还是依旧固执己见。斯托曼的眼光数次停留在我妻子的胸部，表现得比我印象中的样子更为过分，但他还是跟平时一样保持着敏感的状态。我的妻子无意中说了声“愿上帝不会让这样的事再次发生”，结果得到了一个典型的斯托曼式的批评。

“我不想打断你，但是上帝并不存在！”斯托曼说。

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

幸运的是我错了。”

告别前，斯托曼给了我一张他的“快乐卡”，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。



图四 斯托曼的“快乐卡”，是在我们一起吃饭那个晚上给我的

次日，我们几个人一起去吃点心，斯托曼看上去比前一天晚上更沉浸于热恋中。回忆起他在克利尔舍堂的宿舍管理员有关“长生不老药”利弊的争执，斯托曼表示希望有一天科学家们能研究出让人类永生的关键因子。“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，斯托曼没有意识到身体和精神层面的孤独和黑客层面的孤独之间的联系，他说：“分享代码的推动力来自友情，而友情在更低的一个层面。”不过，后来又提到这个话题时，斯托曼承认了他对孤独，或者说是持续的孤独的恐惧。在 GNU 工程最初的岁月里，这样的孤独帮助他做了的各种重要的决定。

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，并把它们抢回来。”

这次采访过后，我不禁感受到我们这类人在情感上的相似性。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，我才认识到了我自己撰写这本书的源动力。从 2000 年 7 月开始，我就学着去赞扬理查德·斯托曼这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，如果把斯托曼的这种性格当成是一种附带现象，把它与整个自由软件运动隔离开来去分析，那将是一种巨大的错误。在某种程度上说，斯托曼的性格与自由软件运动是相辅相成，完全互补，不可分割的。

我肯定不会每一位读者对于斯托曼都有相似程度的认同感——即使在读完这本书后，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。很少人能做到像理查德·M·斯托曼这样特立独行。但我真诚地希望这本书能起到一个抛砖引玉的作用，在 GFDL 的激励下，能有更多人产生与我写作本书时类似的动力，向本书中补充更多的观点，让斯托曼的形象更加地丰满起来。

尾注

1. 参考“Safari Tech Books Online; Subscriber Agreement: Terms of Service.” <http://safari.oreilly.com/mainhlp.asp?help=service>。
2. 参考“Safari 在线科技书籍；订阅者协议：服务条款”，<http://safari.oreilly.com/mainhlp.asp?help=service>。

3. 参考“The Open Publication License: Draft v1.0”(1999年7月8日), <http://opencontent.org/openpub/>。
4. 参考“The GNU Free Documentation License: Version 1.1”(2000年3月), <http://www.gnu.org/copyleft/fdl.html>。
5. 参考: <http://www.gnu.org/philosophy/license-list.html>。
6. 我会很乐意向任何愿意把本书“移植”到 Uadanax 平台的人提供帮助, Uadanax 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。
7. 另外, 我没有找到有关武田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, 可以参考武田基金会的网站 <http://www.takea-foundation.jp>。

附录 A

术 语

在大多数情况下，我使用 GNU/Linux 这个词来表示基于 Linux 内核的自由软件操作系统，使用 Linux 这个词来特指驱动这个操作系统的内核。但在第 9 章中，这个原则不太适用。在这章的最后部分，我介绍了 Linux 作为 Minix 的衍生作品的早期演化历史。在这个项目开始开发的前两年中，托瓦兹和他的同事们所开发的操作系统与斯托曼所设想的 GNU 系统并没有很相似之处，但他们开始慢慢地共享一些关键的组件，比如 GNU C 编译器和 GNU 调试器。

这样的决定使得斯托曼在 1993 年以前并没有很重视要求在 Linux 前加上 GNU 的标识。

有些人可能会认为在后期把这个操作系统称为 GNU/Linux 有点专制。我想指出的是，这是让斯托曼与我合作完成本书的一个基本前提。对于这个问题，我有着自己的观点，一方面在于这个操作系统的模块化设计和围绕它的社区，另一方面在于 Linux 这个名字本身并不具备政治意义。由于这是一本理查德·斯托曼的传记，如果这个操作系统里没点政治含义，多少会让人觉得怪怪的。

在本书即将完稿之际，确定了 O'Reilly & Associates 将成为本书的出版商，斯托曼要求我在书中使用“GNU/Linux”而不是“Linux”，只有这样，他才会答应给在本书出版后，帮助 O'Reilly 推销本书。当我得知这个消息以后，我坚定了我先前的决定，由斯托曼自己去判断这本书最终是否满足了他的期望。直到现在，我还不知道斯托曼的判断结果会是什么样的。

在“自由软件”和“开源”这两个词上也有类似的问题。我仍然选用政治意味稍重的“自由软件”一词用于描述那些包含可以随意复制和修改的源代码的软件。虽然“开源”一词更为大众化，但我只在描述那些声援这个词的团体和公司时才使用。在少数的场合下，这两个词也是完全可以互换的，我是依据克里斯汀·彼德森的建议做出的决定，他是如此定义这两个名词的：“应该尽可能使用‘自由软件’一词，”彼德森说，“‘开源’这个词只是为了需要而被创造出来，它并不是一个理想的词汇。”

附录 B

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年来演化过程。

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菲尔·阿格雷(Phil Agre)所撰写的短文，这篇文章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。”

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（折腾）”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乐趣和创新点的精神。

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，如果一个学生花上大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 20 世纪 50 年代的演化，“Hack”这个词的含义变得更加尖锐和反叛。20 世纪 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作“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。俱乐部的成员只需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承。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地宣布，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得更快，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。

“太空大战 (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，因此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。

20 世纪 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地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气氛。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。

这是早期计算机黑客们惊人技能的一种证明，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 20 世纪 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，它的含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含义。要成为一个黑客，不仅仅需要去编写一些有意义的软件，同时还需要能够融入黑客“文化”并崇尚它的传统，就像一个中世纪的酿酒师需要加入一个酿酒协会一样。黑客社区的结构不像协会那么严格，但是在麻省理工学院、斯坦福和卡耐基梅隆大学这样的精英大学中，黑客们开始公开宣传“黑客道德”：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，作者史蒂芬·李维在经过了很多研究和咨询以后，把黑客道德定义为五种核心的黑客信条。

从很多层面上来说，李维所列出的这些核心信条都进一步定义了计算机黑客文化的核心思想。但是黑客社区给人们留下的协会般的印象，并没有被普通人占据优势地位的软件工业所压制。20世纪80年代初，计算机开始在各行各业中广为应用，那些原本只在顶级大学或公司才能接触到电脑的程序员，现在可以通过 ARPAnet 与一流的黑客进行交流。通过交流，这些程序员开始渐渐接触到像麻省理工学院那些黑客文化中的无政府主义哲学。然而，那些麻省理工学院原生黑客文化中的禁忌，比如禁止恶意行为，在文化的转移中却渐渐地消失了。一些年轻的程序员开始在一些错误的方向上应用他们的计算机才能，发明和传播计算机病毒、攻击军事计算机网络、故意让类似于 ARPAnet 网关的 MIT Oz 这样的计算机死机。“黑客”这个词的含义开始渐渐滑向一个无政府主义的边缘。当警察和工商界开始发现计算机犯罪的来源是少数不守规矩的程序员，而这些程序员又片面地引用一些黑客的信条来为他们的行为辩解，黑客这个词也就更多的在报纸和杂志上以负面形象出现。尽管像《黑客》这样的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“电子盗贼”的同义词。

尽管黑客这个词被这样错误地使用了近二十年，这个词叛逆的内涵让人很难分辨一个 15 岁有软件开发经验且绕过了现代加密系统和一个在 20 世纪 60 年代破坏门锁或直接损坏房门访问办公室电脑终端的大学生有什么分别。毕竟，一个人想要颠覆权威的创新精神就是让另一个人头疼的安全问题。即使是这样，作为信条的核心禁忌，反对恶意行为这一条仍然有很深远的影响力。因此，大部分黑客都喜欢用“骇客 (Cracker)”这个词来描述这类故意破解计算机安全系统去盗取或破坏数据的人，这些所谓的“黑客”使用他们的计算机技能去做一些恶意的事情。

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，计算机黑客的概念演化了很多。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 fucking 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，它精妙地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崇尚的，就像在纯科学的领域，”《波士顿环球报》的记者伦道夫·赖安 (Randolph Ryan) 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。”赖安写道：“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。所以“黑帽”与“白帽”之间的差异也越来越明显：“黑帽”黑客把新的想法转变为破坏性的、恶意的行为，而“白帽”黑客把新想法转变为正面的或者至少是有益的方向。

“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。至于这个台球下一步会弹去哪个方向，人们还是只能众说纷纭。

附录 C

GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other

written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above,

and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until

at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications

given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you

may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify
this document

under the terms of the GNU Free Documentation License,
Version 1.1

or any later version published by the Free Software
Foundation;

with the Invariant Sections being LIST THEIR TITLES,
with the

Front-Cover Texts being LIST, and with the Back-Cover
Texts being LIST.

A copy of the license is included in the section entitled
"GNU

Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.