

自由軟件，自由社會

理查德·斯托曼選集（第三版）

[美] Richard M. Stallman 著
北京 GNU/Linux 用戶組 譯²

2021 年 07 月

²本書基於 GNU 自由文檔協議 1.3 版（GNU Free Documentation License 1.3）授權發佈，可自由複製和分發，和/或基於 GNU 自由文檔協議 1.3 版或自由軟件基金會發布的更高版本，做出修改。本書使用 Markdown 書寫，源碼地址：<https://github.com/beijinglug/fsfs-zh/>。

Copyright © 2019 Beijing GNU/Linux User Group <<https://beijinglug.club>>.

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

A copy of the license is included in the section entitled “GNU Free Documentation License”.

Original Copyright Notice:

Verbatim copying and distribution of this entire book are permitted worldwide, without royalty, in any medium, provided this notice is preserved. Permission is granted to copy and distribute translations of this book from the original English into another language provided the translation has been approved by the Free Software Foundation and the copyright notice and this permission notice are preserved on all copies.

Free Software Foundation 51 Franklin Street, Fifth Floor Boston, MA 02110-1335 Copyright © 2002, 2010, 2015 Free Software Foundation, Inc.

ISBN 978-0-9831592-5-4

譯者序

Copyright © 2019 北京 GNU/Linux 用戶組

<https://beijinglug.club>

《自由軟件，自由社會：理查德·斯托曼選集》（Free Software, Free Society: Selected Essays of Richard M. Stallman）是一本指導了無數人的書籍，它將理查德·斯托曼的自由軟件哲學和思想，闡述得淋漓盡致，此書第一版首發於 2002 年。也是在 2002 年，北京 GNU/Linux 用戶組（BLUG）成立了，經過十多年的發展壯大，已經成為中國推廣自由軟件的先鋒力量。2015 年，《自由軟件，自由社會》第三版發佈。到同年冬天，BLUG 成員中已經有很大一部分成為了自由軟件基金會（FSF）和電子前哨基金會（EFF）成員。2016 年春，北京 GNU/Linux 用戶組正式成為中國第一個被 GNU 工程認可的 GNU 用戶組（GNU User Group）¹。

成為 GNU 用戶組，意味著 GNU 和自由軟件基金會認可了北京 GNU/Linux 用戶組，十多年來在推廣自由軟件哲學和技術上的貢獻。同時更是一種督促，督促我們需要為推廣自由軟件做更多貢獻。所以您手上這本書，就是我們為 GNU 工程和自由軟件做出的新的貢獻。

翻譯這本書的想法，來源於人民郵電出版社 2015 年出版的鄧楠、李凡希翻譯的理查德·斯托曼傳記《若為自由故：自由軟件之父理查德·斯托曼傳》。看到此書出版，讓我們意識到僅有他個人的傳記是遠遠不夠的，傳

¹相關信息可見 <https://savannah.gnu.org/projects/blug>

記只能讓大家理解和懂得理查德·斯托曼的人生經歷，卻完全不足以理解他對自由軟件哲學的思考和闡述，因此我們決定將他對自由軟件的闡述和哲學思想的選集——《自由軟件，自由社會》一書翻譯出來，以自由文檔許可證公開對外發布。

自由軟件來到中國，已經有二十多年的歷史，從 1998 年引入 GNU/Linux 到現在很多公司基於 GNU/Linux 開發的所謂“國產操作系統”，雖然走過了二十多年的道路，但我們開發了多少真正尊重用戶的軟件？最近幾年，國家大力推進“自主可控”的信息產業建設，雖然湧現出了很多優秀的國產“開源”軟件，然而這些軟件是真正尊重用戶“自主可控”權力的嗎？這些讓我們深深意識到，只有社會中每一個人都能對運行在自己計算機（當然也包括智能手機、智能穿戴設備等）上的軟件“自主可控”，整個國家的信息產業才能真正實現“自主可控”，畢竟這個國家是由每個如你我這樣的普通公民組成的。

翻譯《自由軟件，自由社會》一書，也希望可以團結國內自由開源軟件社區和高校社團，啟發所有參與者找回初心，合力推動自由軟件在中國落地生根，推動中國的自由軟件開發更合規，融入世界潮流。

選擇翻譯此書第三版，有很多考量。除了因為第三版是 2015 年發行的最新版，更是因為此版本有至少三分之一都是新文章，特別是理查德·斯托曼最近幾年在《紐約時報》和《連線》雜誌上發表的重要文章，反映了最新的技術革新和社會變化：比如《如今自由軟件更加重要》(p. 32) 一文則提到了隨著移動互聯網、所謂“雲計算”技術的發展，新時代自由軟件的重要性比之前更加重要；又如《自由硬件和自由硬件設計》(p. 53) 一文就覆蓋了近年來熱門的“創客”運動和自由硬件開發；還比如《民主可以承受多少監控？》(p. 323) 一文就反映了自 2013 年 NSA 前僱員愛德華·斯諾登披露美國大規模監控醜聞以來，自由軟件社區在“大數據”時代需要面對的機遇和挑戰。第三版中有大量篇幅闡述軟件專利和版權的內容，而這些內容是前兩版較少涉及的部分，更是中國的自由軟件（也包括開源軟件）用戶和開發者很少關注和理解的，我們認為將這一部分翻譯出來，會極大推動國產自由軟件的合規化。

此書的翻譯，本著尊重原文，力求精品的原則。很多人認為理查德·斯托曼的觀點偏激，甚至是極端的，我們翻譯時在理解他論點的前提下，只直譯了原文的文本，除非必要很少意譯，以防曲解甚至誤解作者原意，力求傳達出理查德·斯托曼闡述自由軟件哲學原始的本真的意思。這樣也給讀者留下自行判斷的空間：斯托曼的觀點是不是偏激極端。為了幫助讀者理解，翻譯時我們還加入了必要的“譯者注”，以降低閱讀門檻。

本書的翻譯參考了既有的一些中文譯版，特別感謝 GNU 網站中文化團隊以及“哲思社區”曾翻譯過此書的第一版和第二版的部分文章，他們高質量的翻譯為本書的翻譯提供了很多參考。術語和專業性內容我們也參考了國內包括港澳臺專家撰寫的相關論文、書籍，還有中文維基百科。同時為了減少重複勞動，以合規性為前提，我們也將幾篇既有的優秀中文翻譯（主要是許可證文本）加入其中，感謝他們曾做出的巨大貢獻。

參與本書翻譯和校對的成員有（按貢獻量排序）：tonghuix、nadebula、Hagb、mytbk、MandyMY、persmule、wxy、biergaizi、liushuyu、IceCatX 和 Artoria2e5。其中 nadebula 提供了 25 篇高質量翻譯，特別是翻譯了大量版權和專利相關的文章，而這一部分則是此前很少發表的內容；其他成員則參與了翻譯和校對；感謝 mytbk 幫助解決了此書 PDF 及 EPUB 格式生成的問題；liushuyu 重新制作了“自由軟件之歌”中文版樂譜，並可以通過自由的樂譜軟件 MuseScore 播放聆聽¹；Hagb 志願幫助將此書新翻譯的文章向 GNU 官網反饋，並解決了多年未解決的交叉引用問題。

本書的翻譯依然有不完善之處，我們非常歡迎各界朋友雅正。本書的翻譯項目託管在 <https://github.com/beijinglug/fsfs-zh>，您可以直接在那裡提交 Issue 或 Pull Request 來幫助我們改正。此處亦附上此書英文原版下載地址 <https://www.gnu.org/doc/fsfs3-hardcover.pdf>，以便對照。

北京 GNU/Linux 用戶組 (<https://beijinglug.club>)

2019 年 4 月於北京

¹參見本書附錄 C 的《自由軟件之歌》(p. 347)

第三版序言

Copyright © 2015 自由軟件基金會。

給理查德·斯托曼的情書，作者雅各·阿貝爾鮑姆 (*Jacob Appelbaum*)
我們生活在由機器構成的信息社會里。

軟件和硬件對我們的信息時代和互聯網一樣重要。自由軟件被稱為是革命性的政治理論，就如同我們可以控制自己的生活，我們也應該能夠控制那些延伸我們身體的機器。這個被自由軟件基金會支持的理論，已經成為過去三十年來數萬人的實踐和傳統。

自由軟件作為一種政治理論承認了軟件和硬件系統在我們社會中所扮演的角色。批判之前和現在的制度是必要的。我們也許會發現自己無法修改既存制度。我們成為了別人製造的那些不公道制度的犧牲品，而他們本身就是不公道的。而這些制度的產出並非總是顯而易見，特別是當我們被迫接受這些制度安排，尤其是當它們已變成標準化和品牌標準的時候。自由軟件並不僅僅只是批評：它可以成為提供自由的替代品，依賴於自由的標準，自由地為大眾使用。

自由軟件提供了一種轉變的範例，我們可以自由的從前人那裡理解並學習，自由地成長和分享，從錯誤中學習，從學習中受益，並分享這些益處給每一個人。當我們用 Copyleft 的時候，我們能確保以後的用戶都能從我們的作品中獲得同樣的自由。自由軟件確保了我們的子孫後代也能解碼數據的全部歷史。這不僅確保了我們的自由權益，同樣也包括他們的。

在當今這個大規模監控的時代，自由軟件通過它的可驗證性機制帶來了必須的透明性。自由軟件讓我們可以通過加密，來確保數據完整性、身份認證以及匿名我們自己。在一個不斷深入私有化的世界裡，我們發現自由軟件給出了通向自由社會的一個公共支柱。我們無法列舉自由軟件的好處，因為我們無法道盡自由本身的好處。推廣自由軟件的事業是不會終了的，就如同為正義的聲索，並且需要時刻保持警惕。推廣自由軟件的事業是困難的，我們需要秉持排除萬難的堅定意志來倡導和推動自由軟件。

投身於自由軟件不僅僅是對於知識，同時也包括授權：授權學習，授權修改，授權分享，以及授權使之能夠被分享。爭取信息時代的自由需要拒絕在自由軟件核心原則上妥協，並且要有犧牲精神。很多人可能拒絕這些負擔，他們只想獲得更多的財富；而其他人卻在努力增加人類知識的深度和廣度。通過實現自由軟件，我們能發現一種面向長期願景的可持續性發展模式，不僅增加了知識，同時切實可行地將這些知識自由地分享給所有人。這份崇高的事業以其無差別的關懷澤及我們每一個人；從現實到超現實，自由軟件都被涉及。

理查德·斯托曼是一位革命家和理論家，他為這個世界推出了自由軟件。而他撰寫文章中涉及的論題，被信息時代和之後創立制度及系統的人們廣泛閱讀和詮釋，已成為數十年來必讀的經典。他一生致力於人性解放事業，同時這本書也介紹瞭如何為這項事業提供幫助。

雅各·阿貝爾鮑姆¹

JACOB APPELBAUM

¹雅各·阿貝爾鮑姆（Jacob Appelbaum），美國獨立記者，計算機安全研究員，藝術家和黑客（Hacker）。受僱於華盛頓大學，曾是 Tor 項目的核心開發者。——譯者注

第一版序言

Copyright © 2002 自由軟件基金會。

每一個時代都有引領時代發展的哲學家——也許是一個作家亦或一個藝術家。有時這些人家喻戶曉，有些則是需要幾代人的努力將其理念變為現實。無論是否為人熟知，一個時代會為這些抒發理想的人留下印記，他們或是盈盈細語於詩歌，或是揭竿而起於政治運動。

我們這個時代有一位哲學家。他不是藝術家，也不是專職作家。他是一位程序員。理查德·斯托曼（Richard Stallman）是 MIT 前程序員以及操作系統架構師。作為程序員和架構師，他創立了一項在這個日益由“代碼”決定的世界裡爭取自由的運動，從此開始了他在大眾生活舞臺上的職業生涯。

“代碼”是一項使計算機運行的技術。無論是寫在軟件裡或者燒在硬件上，都是由一系列指揮機器運行並寫成文字的指令組成。這些機器——計算機日益定義並控制了我們的生活。它們決定了電話如何接通，電視上播什麼節目；它們決定了視頻是否能夠通過視頻流傳到計算機上；它們也控制了計算機彙報給其製造商哪些信息。這些機器為我們服務，而代碼控制這些機器。

我們應該如何控制這些代碼？怎麼理解它們？啟用控制時應該有什麼自由與之匹配？有什麼樣的權力？

這些問題也挑戰著斯托曼的生活。通過他及他的作品，讓我們明白了代碼“自由”的重要性。這裡的 free 並非表示寫代碼的人不能得到報償，而

是意味著程序員將代碼透明給所有人，任何人都可以控制代碼，並按照自己的需求修改。這就是“自由軟件”；“自由軟件”為這個由代碼構成的世界做了回答。

“Free”，斯托曼在解釋這個詞的時候頗有怨言。然而這並沒有什麼可悲嘆的，模糊的概念迫使人們去思考，“Free”這個詞著實足夠啟發思考。對當代美國人而言，“自由軟件”聽起來像是烏托邦，完全無法實現的，空洞的，甚至連午餐，也是免費的。那些控制著世界上最重要機器的代碼怎麼能是“免費的”。一個理性的社會怎麼能主張這樣的理想？

但“free”一詞的奇怪意思只是對我們而言，而不是這個詞彙本身。Free 一詞有很多意思，只有一個是表示免費。更多的則表示是自由。斯托曼說，把它稱為“言論自由”，或者更準確地說是“自由勞動”較為恰當，並不是表示沒有花費，而是表示沒有被他人控制和限制。

這種機制通過一種“copyleft”模式的 GPL 許可證得以體現。基於 copyleft 條款賦予的權力，“自由軟件”不僅僅是開放的，更保證了修改自由，且其他使用自由軟件（技術上稱為“派生”）的也必須是自由的。如果你使用了一個自由軟件程序，並公開發布了修改版，那麼修改版也和原版一樣是自由的。這是必須的，否則會違反版權法。

“自由軟件”與自由社會一樣，也有其敵人。微軟發起了針對 GPL 的戰爭，警告所有人 GPL 是一種“危險的”許可證。事實上，它說的危險性很大程度上是欺騙性的。還有反對者認為 GPL “強迫”修改版也必須是自由的。光看這一條不能判斷是否強迫。微軟拒絕允許用戶發佈修改過的 Office 軟件而不需要支付（假設）數百萬的費用，如果這還不是強迫的話，那麼 GPL 堅持的修改自自由軟件的軟件也必須是自由的，也不是強迫。

有些人認為斯托曼的言論太過極端，然而並不極端。事實上，顯而易見的，斯托曼的作品只是將代碼出現之前世界的自由簡單翻譯過來了。“自由軟件”能確保代碼世界的秩序與之前的傳統是一樣“自由的”。

比如：一個“自由的社會”是由法律來規範的。然而任何自由社會的法律對這些都有限制：沒有自由社會可以用祕密法律。在傳統上沒有政府可以對其治理的對象隱藏規範。法律能夠起效，只在明顯公平正義的情況下

才可以。只有當法律的條款是可以被其監管的人或其管理的管理者（律師，立法機構）知悉和控制時，法律才是可見的。

法律的這種條件超出了立法者的工作。來想想美國法院的司法實踐吧。當事人僱用律師來提高利益，有時候這種利益通過訴訟得以提升。在訴訟過程中，律師寫出訴訟狀，這些訴訟狀反過來影響法官寫下的意見。這些意見決定誰能夠在案件中獲勝，或者某法律條文是否符合憲法的精神。

這個過程中的所有材料都符合斯托曼所說的自由。訴訟簡報是公開的，別人可以自由使用。論據是透明的（不等於說就是好的），論證過程也無需原律師的允許就可以拿走。法官意見可以在之後的訴訟摘要裡引用，並可以複製或綜合到別的訴訟簡報或意見中。美國法律的“源代碼”在設計上和原則上，對任何使用它的人都是開放和自由的。拿律師來說，藉由重複使用之前案件的材料，律師發揮了創造力，可以作出最佳的訴訟簡報。原始資料是自由的，創新和經濟都是在其上建立的。

自由代碼（這裡指法律代碼）的經濟並沒有扼殺律師。儘管任何人都可以使用和複製他們做成的東西，律師事務所仍然有足夠動機來作出好的訴訟簡報。律師也是手藝人，他或她的產品是公開的。但是這種工藝不是慈善行為。律師能得到收入，公眾沒有不給他們工作報酬。相反，這種經濟能夠蓬勃發展恰是因為之後的工作可以彌補前面的。

我們來設想一種不同的法律實踐——案情和證據都是保密的，判決只有結果而沒有論證，法律只由警方把持而其他人看不到，法規條令也不經解釋如何執行。

我們儘可以設想這樣的社會，但卻不能稱其為“自由”。無論是否有更好的動機，或更有效率的分配，這樣的社會都無法認為是自由的。自由社會中的自由生活理想並不僅僅只是高效的分配。而是在開放和透明的前提下建立法律的限制，不能因為對領導人有利就加入相應的選項。由軟件代碼治理的生活也不能比這個更差。

編寫代碼不是訴訟。它更好，更豐富也更有建設性。創造性和動機並不依賴於對創作產品的完全控制，法律就是一個明顯的例子。類似爵士樂、小說或者建築，法律條文是基於既有的基礎而建立的。這種增補和改變一

直是創新的體現。一個自由的社會必須保證其最重要的資源一直是自由的，這才像話。

這本書以一種使之更微妙且權力更明確的方式，收錄了理查德·斯托曼的著作和演講。這些文章跨度很大，從版權到自由軟件運動的歷史，包含了許多不為人知的論點，其中尤其是對數字世界中版權受到懷疑的變化的情況的深刻見解。這些文章是理解他思想的源泉，理解他的激情和他的正直，即使其他方面都不及此。這些文章將會啟發那些接受他的理念並將之發揚光大的人們。

我並不十分了解斯托曼，但我很清楚他是一個嚴厲的人。他求勝心切，常常不耐煩。他對朋友可以像對敵人一樣火冒三丈。他很強硬，不屈不撓，但卻也很有耐心。

當我們的世界最終理解了代碼的威力和危險，終於認識到代碼和法律、政府一樣，必須透明才能獲得自由時，不妨回首看著這位強硬且不屈不撓的程序員，會發現他終身奮鬥的景象終於實現：自由和知識可以在編譯器下倖存。我們會認識到沒有人能像斯托曼這樣，可以為社會獲得自由而奮鬥和發聲。

然而我們尚沒有獲得自由。我們在捍衛自由的時候也許會失敗。然而無論勝敗，這些作品都展現了自由的景象。創作這些作品的生活中，會啟發那些就像斯托曼一樣，為自由而戰的人們。

勞倫斯·萊斯格¹

LAWRENCE LESSIG

¹勞倫斯·萊斯格（Lawrence Lessig），是一位美國學者暨學術與政治的行動主義者，哈佛法學院法學教授。他還是知識共享（Creative Commons）發起委員、軟件自由法律中心（SFLC）委員、陽光基金會諮詢委員與電子前哨基金會（EFF）前任委員。——譯者注

前言

此第三版《自由軟件，自由社會》(Free Software, Free Society) 將第二版的很多文章更新了，並加入了很多新文章，其中三分之一都是新文章。

與前幾版一樣，此書首先介紹了自由軟件的原則和哲學。講述了軟件為何必須自由，並解釋了我們的原則是如何指導實踐活動的，以及解決了有關硬件自由設計的問題。

如何稱呼和框定一個問題會影響我們對這個問題的思考。企業通過術語來推廣他們的理念，接受了相應術語也就支持了他們。因此，在這一版中，我們加入了新的文章來講解 FSF（自由軟件基金會，Free Software Foundation）是如何以及為什麼這麼稱呼這些術語。

“版權”這一部分節錄了討論版權相關法律問題的演講，並說明如何修改這些法律。

“專利”這一部分設想了一個解決計算機領域專利問題的方案。我將專利和版權分開討論，是因為這兩點是不能混淆在一起的。

“許可證”這一部分大體並沒有改變，依舊講述了 GNU 的各個許可證，還有 Brett Smith 講述的各個許可證的歷史變遷，並用一篇文章解釋了為什麼軟件項目必須升級到 GNU 通用公共許可證第三版。

這一版繼續闡述自由軟件社區需要面對和解決的各種危險和陷阱，包括了當下的非自由遊戲、電子書和持續威脅的電子監控。

我希望此書可以告訴你，我們很容易就會失去自由，教會你如何保護，並啟發你珍視自由。

要感謝 Jeanne Rasata 管理此項目，編輯書目，排版文字，並建立目錄。
還要感謝 Karl Berry 協助解決 Texinfo 技術問題，以及 Kyle Winfree 設計和
排版此書的封面。

理查德·斯托曼
RICHARD STALLMAN

目錄

譯者序	i
第三版序言	v
第一版序言	vii
前言	xi
目錄	xiii
1 GNU 計劃和自由軟件	1
1.1 什麼是自由軟件？	1
1.2 GNU 工程	8
1.3 GNU 操作系統的初始公告	29
1.4 如今自由軟件更加重要	32
1.5 為什麼學校應該只使用自由軟件	38
1.6 政府推動自由軟件的措施	41
1.7 為什麼自由軟件需要自由的文檔	46
1.8 售賣自由軟件	49
1.9 自由硬件和自由硬件設計	53
1.10 應用自由軟件判斷準則	63

2	名字的含義	71
2.1	名字的含義？	71
2.2	Linux 和 GNU 操作系統	75
2.3	自由與非自由軟件的分類	80
2.4	為什麼說開源漏掉了自由軟件的要點	90
2.5	您說過“知識產權”嗎？這是一種迷惑的幻景	99
2.6	為何稱之為詐騙（Swindle）？	104
2.7	應避免使用（或慎用）的詞語，由於它們是不公正的或者引起混淆的	107
3	版權和不公	129
3.1	閱讀的權利	129
3.2	對版權的錯誤解讀——一系列錯誤	137
3.3	科學必須擺脫版權束縛	151
3.4	計算機網絡時代的版權與社區之爭	154
4	軟件專利：對程序員的威脅	175
4.1	軟件專利和文學專利	175
4.2	軟件專利的威脅	180
4.3	保護軟件領域免受專利困擾	202
5	自由軟件許可證	205
5.1	許可證簡介	205
5.2	如何為你的作品選擇一份許可證	211
5.3	X Window 系統的陷阱	216
5.4	程序不得限制它們的自由運行	219
5.5	什麼是 Copyleft？	222
5.6	為什麼使用 Copyleft	225
5.7	Copyleft：務實的理想主義	226
5.8	GNU 通用公共許可證	229

5.9	為何升級到 GPLv3	244
5.10	GNU 寬通用許可證	248
5.11	GNU 自由文檔許可證	252
5.12	關於出售例外對 GNU GPL 的影響	276
6	陷阱和挑戰	279
6.1	您能夠信任您的計算機嗎?	279
6.2	JavaScript 陷阱	286
6.3	如果您在大學工作，請發佈自由軟件	292
6.4	GNU/Linux 上帶有數字限制管理 (DRM) 的非自由遊戲：是 好是壞?	295
6.5	電子書的威脅	297
6.6	電子書必須增進我們的自由而非限制我們的自由	299
6.7	服務器真正是在為誰服務?	302
7	珍視社區和你的自由	311
7.1	避免破壞性的妥協	311
7.2	克服社會慣性	316
7.3	自由還是權力?	318
7.4	缺陷並不同於壓迫	321
7.5	民主可以承受多少監控?	323
附錄 A	關於軟件的基礎知識	337
附錄 B	不同語言對“自由軟件”和“免費軟件”的翻譯	341
附錄 C	自由軟件之歌	347

第 1 部分

GNU 計劃和自由軟件

1.1 什麼是自由軟件？

Copyright © 1996–2002, 2004–2007, 2009–2015 自由軟件基金會。
此文最初於 1996 年發表在 <http://gnu.org>。

1.1.1 自由軟件的定義

自由軟件的定義給出了一個標準，即一個特定的軟件是否有資格被稱為自由軟件。我們時不時修改這一定義，以澄清和解決與之相關的各種細微問題。關於我們對自由軟件定義所作的修改記錄，請參閱 <http://gnu.org/philosophy/free-sw.html> “歷史” 部分。

“自由軟件”（Free Software）表示的是那些尊重用戶和社區自由的軟件。粗略的說，它賦予用戶運行、複製、分發、學習、修改並改進軟件的自由。因此，“自由軟件”是有關用戶的自由權益（liberty），而不是指免費（free）的價格。為了解理解這個概念，你需要將“free”一詞理解成“言論

自由”中的“自由”，而不是“免費啤酒”裡的“免費”。有時我們會稱之為“Libre Software”，以避免帶有免費的意思。

我們爭取這些自由，是因為每個人都應該擁有它。有了這些自由，用戶（包括個體和團體）就可以控制程序為己所用。當用戶無法控制程序時，我們稱這樣的軟件為“非自由”（Nonfree）或“專有”（Proprietary）程序。非自由的程序控制了用戶，而開發者控制著程序：這就讓程序成為了非正義權力的幫凶¹。

如果一個軟件的用戶擁有以下四項基本自由，那麼該軟件可以被稱為自由軟件：

- 基於任何目的，按你的意願運行軟件的自由（自由之零）。
- 學習軟件如何工作的自由，按你的意願修改軟件以符合你的計算的自由（自由之一）。可訪問源代碼是此項自由的先決條件。
- 分發軟件副本的自由，因此你可以幫助他人（自由之二）。
- 將你修改過的軟件版本再分發給其他人的自由（自由之三）。這樣可以讓整個社區有機會共享你對軟件的改動。可訪問源代碼是此項自由的先決條件。

如果一個軟件可以充分授予用戶所有這些自由，它就是自由軟件，否則就是非自由軟件。即使我們可以依據它們與這四項自由有多大的差異來區分不同的非自由發行方式，我們依然認為這些非自由軟件是不符合倫理的。

在任何情況下，這些自由都必須應用於所有要使用的代碼，或引導其他人使用這些軟件。例如，程序 A 會自動調用程序 B 來處理一些事情，如果我們獨立發佈 A 也意味著用戶還需要 B，因此我們需要判斷 A 和 B 是否都是自由的。如果我們修改了 A 使得它不再需要 B，那麼只需要 A 是自由的，就可以忽略 B。

¹可參見《如今自由軟件更加重要》(p. 32) 一文。

本文餘下的內容，將會澄清關於特定自由是否適用的一些關鍵點。

再分發軟件的自由（自由之二和自由之三）表示你可以在任何地方發佈軟件的副本給任何人，無論是否有過修改，無論是免費或收取費用。自由地再分發意味著（排除其他因素）你不需要必須事前徵得任何發行許可或為此支付額外費用。

你應該有修改軟件的自由並將軟件用於私人的工作或娛樂，甚至不需要提到你的修改。如果你發佈了自己的修改版，你不應被要求通知特定的人或以特定的方式發佈。

運行軟件的自由，即賦予任何個人或組織在任何計算機系統上，基於任何工作方式或任何目的運行軟件，而無需與任何開發者或特定實體溝通的自由。在這個自由中，重要的只是用戶的目的，而非開發者的目的；你作為用戶有基於任何目的運行軟件的自由；如果你將軟件發佈給了其他人，則他也有按自己目的運行軟件的自由，但你不能將自己的目的強加給他。

以你的意願運行軟件的自由也表示你不能被要求禁止或停止運行該程序，這無關軟件的功能，或對你來說它是否有用。

再分發軟件副本的自由，表示分發該程序的二進制或可執行格式時必須附帶源代碼，包括修改過的和未修改版（以可運行格式發佈程序可以方便在自由的操作系統上安裝）。對特定程序也可以不發佈和產生二進制可執行格式（因為一些編程語言不支持這個特性），但你仍有發佈這些格式，或開發一種方法來產生這些格式文件的自由。

為了實現自由之一和自由之三（修改軟件併發布修改版的自由），你必須能夠訪問程序的源代碼，因此有權力訪問程序的源代碼對自由軟件來說是一個必要條件。而混淆過的“源代碼”不能算真正的源代碼¹。

自由之一允許你用你修改的版本來替代原始的版本。如果一個程序從產品設計的角度發佈只能用其他人修過的版本而不能用你的修改版——例如

¹這裡指的是“混淆代碼”（Obfuscated code）的做法，是將計算機程序的代碼，轉換成一種功能上等價，但是難於閱讀和理解的形式的行為。——譯者注，摘自維基百科

所謂的“Tivo 化”¹或“鎖定”，或者（用行業內的話來說）“安全啟動”（Secure Boot）——這樣自由之一就被架空而無法實現，即便這些軟件的二進制是從自由的源代碼編譯而來，也不能算是自由軟件。

修改軟件的一個重要方法是合併可用的自由子程序或模塊。如果一個程序的許可證讓你不能將一個合理授權發佈的現有模塊融合進來——例如要求你是你所添加進的代碼的版權所有者，那麼該許可證過於嚴苛以致不能被視為自由的。

自由之三允許你以自由軟件的方式發佈修改版。一個自由許可證可允許以其他形式發佈；也就是說，這不一定是 copyleft 許可證。不過如果一個許可證要求以非自由的方式發佈修改版，則是非自由的。

為了讓這些自由得以實現，在你遵守許可的條件下，這些條款必須是永久且不可撤銷的。在你遵守許可的情況下，如果軟件的開發者有權力撤銷，或者添加限制性條款，那麼該軟件就不是自由的。

然而，一些對自由軟件發佈方式的限制規則是可以接受的，前提是與核心自由不衝突。例如，copyleft（最簡單地說）規定你不能在分發程序的時候添加限制性條款以拒絕其他人的核心自由。這一規則與核心自由並不衝突，反而是在保護核心自由。

在 GNU 工程中，我們用 copyleft 合法保護每個人的四項基本自由，我們相信使用 copyleft 是非常重要的。然而非 copyleft 的許可證也是合乎倫理的。有關“自由軟件”、“copyleft 軟件”和其他類型軟件的話題可以參見本書《自由與非自由軟件的分類》（p. 80）一文。

“自由軟件”並不意味著“非商業”。一個自由軟件允許商業使用，商業開發以及商業發佈。自由軟件的商業開發已經不是一件特殊的事情。這對自由軟件是非常重要的，你可能需要為獲取自由軟件的副本而付費，也可能不需要付費。但是無論如何獲得副本，你都有複製並修改軟件的自由，甚至有重新銷售的自由。

¹“Tivo”是美國一大有線電視機頂盒產品，“Tivo 化”（Tivoization）是指該產品包含了以 GPL 許可證發佈的軟件，但實際上用戶不能修改，因為一旦設備發現軟件經過修改就會自動關機。相關內容可參考本書《為何升級到 GPLv3》（p. 244）一文。

對程序的修改是不是改進程序，這只是一個主觀判斷。如果你對軟件的修改權實際上僅限於作出被他人認作改進的修改，該程序並不自由。

不過，規範修改版打包行為的條款是可以接受的，因為實質上這沒有限制你發佈修改版的自由，或者私下使用的自由。因此，許可證的一些條款是可以接受的，比如要求你改變修改版的名字，移除徽標（LOGO），或者修改版為你自已所有。只要這些要求不那麼繁冗以至於影響了正常發佈，都是可以接受的。既然你都已經做過一些修改，也就無所謂再多做一些了。

“如果你的版本在這種方式下可用，也必須能夠在另一種方式下可用”，類似這樣的條款也是可以接受的。同樣的，比如有規則要求如果你發佈了修改版，也必須給前一個開發者發送一份，也是可以接受的（注意這樣的條款仍然讓你有自由選擇是否公開發布的權力）。要求你為公開發布版的程序附帶源代碼，也是可以接受的。

這裡有個特殊問題，如果許可證要求你改變程序被其他程序調用時的名字。這樣做實際上阻礙了你發佈修改版以便在其被其它程序調用時取代原始版。這種情況僅當通過一種別名機制來標示原始程序名是修改版的別名時才可被接受。

有時政府的出口管制或者貿易制裁會影響你國際範圍內發佈軟件的自由。軟件的開發者沒有權力消除或覆蓋這些限制，但他們可以做的是拒絕強制要求使用程序時接受這些條件。這樣，這些限制就不會影響到那些政府管轄以外的國家和人民的活動。因此自由軟件許可證為了行使這些必要的自由，不能要求服從任何這些出口限制條款。

我們僅僅只是提及這些出口限制條款，而沒有將這些條款作為軟件許可證的條件，因為這樣並沒有限制用戶，所以也是可以接受的。如果該出口限制條款對自由軟件來說是非常普遍的，將其作為需要的條件也不是一個問題。然而這會產生一個潛在問題，出口限制法規若之後有一定修改，可能會讓軟件變成非自由的。

自由的許可證不能要求遵守非自由程序的許可證。所以，如果一個許可證要求你必須遵守“你使用的所有程序”的許可證，那麼運行非自由程序的用戶在這種情況下意味著需要遵守非自由程序的許可證，那麼這將使

得該許可證變成非自由的。

自由軟件許可證可以指定與軟件相關的訴訟需要遵守哪個地區的法律，或需要在哪裡提起訴訟。

大多數自由軟件許可證是基於版權的，而何種要求能通過版權施加是有限制的。如果一個基於版權的許可證尊重如上文所述的自由，那麼也就不會發生我們意料之外的問題（儘管這也許會發生）。然而有些自由軟件許可證是基於合同的，而合同可以增加更大範圍的限制。這樣就有更大可能性使得該許可證出現無法接受的限制或非自由。

我們無法將這些可能的情形一一列出。如果一個基於合同的許可證以不同於基於版權許可證的方式限制了用戶，此處也沒有表明其為合法，那麼我們就需要考察這個許可證，並且很可能會將之視為非自由的。

在談到自由軟件的時候，儘量不要用“贈予”（Give Away）或“免費”（For Free），因為這些詞都是暗示價格上的免費而非自由的。一些常用語比如“盜版”也體現了一些我們希望你不會認同的意見。相關詳情參見本書《應避免使用（或慎用）的詞語，由於它們是不公正的或者引起混淆的》（p. 107）一文。我們同時還根據不同語言列出了“自由軟件”（Free Software）一詞的翻譯，請參見本書附錄 B（p. 341）。

最後，這些有關自由軟件的標準需要謹慎的解讀。判斷一個軟件許可證是否是自由軟件許可證，標準就是看它是否符合自由軟件精神以及用語是否確切。如果一個許可證包含了不合理的限制，即便是此文中我們沒有預料到的問題，我們也會拒絕它。有些時候一個許可證會帶來新的問題和思考，這需要我們與律師諮詢以後，共同思考和研判。我們最終得出結論以後，將會更新這份標準以使其能更容易體現特定的許可證是否符合標準。

如果你對特定的許可證感興趣，可以參考我們的許可證列表，位於<http://gnu.org/licenses/license-list.html>。如果裡面沒有列出你所關心的，可以發郵件詢問我們：licensing@gnu.org。

如果你打算起草一份新的許可證，請首先通過上面這個郵箱聯繫自由軟件基金會。不同的自由軟件許可證意味著用戶需要花費更多精力來理解；我們也許可以幫你從現有的自由軟件許可證裡找到符合你需要的。

如果這樣依然不行，你真的需要一份新的許可證，有了我們的幫助也可以確保許可證符合自由軟件標準並避免一些現實問題。

1.1.2 軟件之外

軟件手冊必須是自由的¹，就如同軟件必須是自由的一樣，因為手冊會對軟件產生一部分影響。

同理可證其他各種具有實用功能的作品，也就是說，任何體現知識可用性的作品也應該如此，比如教育資源和參考資料。維基百科（Wikipedia）就是最著名的例子。

任何領域的作品都可以是自由的，對自由軟件的定義已經擴展為了對自由文化的定義，可應用於任何領域的作品²。

1.1.3 開放源代碼？

另一些用戶使用“開放源代碼”（“開源”，Open Source）一詞來表示與“自由軟件”相近（但不盡相同）的意思。我們傾向於使用“自由軟件”一詞，因為一旦你理解它表示自由而不是價格，這麼稱呼可以表達自由。“開放”（Open）一詞並不能表達自由之要義³。

¹可參見《為什麼自由軟件需要自由的文檔》（p. 46）一文

²可參見 <http://freedomdefined.org>

³可參見《為什麼說開源漏掉了自由軟件的要點》（p. 90）一文

1.2 GNU 工程

Copyright © 1998, 2001, 2002, 2005–2008, 2010 Richard Stallman。
本文最初以標題“GNU 操作系統和自由軟件運動”發表於由 Chris DiBona 等人編寫的《開源軟件文集：開源革命之聲》(Sebastopol: O'Reilly Media, 1999)¹。儘管我不是“開放源碼”的支持者，我還是貢獻了這篇文章，這樣自由軟件運動的思想不會在那本書中完全失聲。

1.2.1 第一個軟件分享社區

當我在 1971 年開始在 MIT（麻省理工學院）的人工智能實驗室工作的時候，我成為了那裡一個已存在數年之久的軟件分享社區的一員。在我們這個特別的社區裡，分享軟件不受任何限制；這和計算機的歷史一樣悠久，正如分享菜譜的行為乃是和做飯的歷史一樣久遠的。但我們分享得比大多數人更多。

人工智能實驗室使用一個叫 ITS（不兼容分時系統）的分時操作系統，由實驗室的黑客²員工們設計，並以 Digital PDP-10——當年的大型機之一——的彙編語言寫成。作為這個社區和人工智能實驗室系統黑客員工的一員，我的工作便是改進這個系統。

那時我們並不稱我們的軟件為“自由軟件”，因為那個詞尚不存在；但它們實際上就是。只要其他大學和公司的人想要移植和使用我們的程序，我們都十分歡迎。要是你看誰在用一個沒見過而有趣的程序，你總可以提出要看看源代碼，以便閱讀，改動，或者吸收其部件以創造新的程序。

¹《開源軟件文集：開源革命之聲》一書中譯本已由中國電力出版社於 1999 年 12 月出版，洪峰譯。——譯者注

²一部分大眾傳媒混淆地將“黑客”一詞用來表示“安全破壞者”。我們作為黑客拒絕認可這個含義，並繼續用這個詞表示“那些喜愛編程、享受有趣的才智，或兩者兼備的人。”見拙作“On Hacking”，於 <http://stallman.org/articles/on-hacking.html>。

1.2.2 社區的解體

80 年代初，Digital 關停了 PDP-10 系列，與此同時，形勢發生了劇變。PDP-10 在 60 年代優雅而強大的架構無法自然地擴展到 80 年代開始可用的更大的地址空間上。這意味著幾乎所有組成 ITS 的程序都要作廢。

人工智能實驗室的黑客社區不久前就解體了。在 1981 年，附屬的 Symbolics 公司幾乎僱走了人工智能實驗室的所有黑客，而嚴重減員使得社區已無法自持（由 Steve Levy 撰寫的《黑客》一書記述了這些事件，並給出了一幅社區全盛時期的清晰圖景¹）。人工智能實驗室在 1982 年買了一臺新的 PDP-10，而其管理員打算使用 Digital 的非自由分時系統取代 ITS。

那時的新計算機，如 VAX 或 68020，都自帶著操作系統，但沒有一個是自由軟件：僅僅為了得到一份可以運行的副本，你就得籤保密協議。

這意味著使用計算機的第一步就是要你下保證不會去幫助他人。協作的社區被禁止了。私有軟件所有者訂立的規則是：如果和他人分享，你就是海盜（在英語中也用該詞表示“盜版者”——譯者注）。想要改動，就來求我們吧。

說私有軟件的社會制度——不許分享和改造軟件的制度——是反社會的、是不道德的、是完全錯誤的，可能令不少讀者吃驚。但是對於一個建立在分裂群眾並保持用戶無助的基礎之上的制度，我們還能說什麼呢？對上述觀點吃驚的讀者可能已經將私有軟件的社會制度視為理所當然，或用帶有私有軟件行業暗示的詞語來判斷。軟件出版商花了大量的力氣和時間去使人們相信對這個問題只有一種看法。

當軟件出版商談論“行使”他們的“權利”或“停止盜版”²時，他們實際“說”的是次要的。這些聲明真正傳達的是他們將未闡明的假設視為理所當然；公眾被要求不加審視地接受這些。還是讓我們來仔細審視一番吧。

其中一個假設就是，軟件公司對擁有軟件有著毋庸置疑的自然權利並

¹ 《黑客：計算機革命的英雄》一書已由機械工業出版社於 2011 年出版，趙俐、刁海鵬、田俊靜譯。——譯者注

² 參見《應避免使用（或慎用）的詞語》一文的“盜版（Piracy）”一節（p. 121）查看“盜版”一詞的更多錯誤用法。

因此有權置身所有用戶之上。（如果這真是一個自然權利，那麼無論它對公眾有多大害處，我們也不能反對。）有趣的是，美國憲法和法律慣例排斥這種觀點：版權不是自然權利，而是一個政府強加的、限制用戶自然地複製權的人為壟斷。

另一個未闡明的假設是，對軟件唯一重要的是它能讓你做什麼工作——我們計算機用戶不應關心我們應當擁有什麼樣的社會。

第三個假設是，如果我們不授予軟件公司凌駕於程序的用戶之上的權利，我們將沒有軟件可用（或者決不會有一個程序來做這個或那個特定的工作）。這個假設，在自由軟件運動展示了我們能夠製造豐富的有用的軟件而不在上面拴任何鎖鏈之前，也許看似是有理的。

如果我們拒絕接受這些假設，並基於普通常識下的道德把用戶放在首位來判斷這些問題，我們會得到非常不同的結論。計算機用戶應當有自由去修改程序以適應他們的需求，並自由地共享軟件，因為幫助別人是社會的根基。

篇幅所限，結論背後的展開敘述不在此詳述，讀者請參閱文章《為何軟件不應有主》於 <http://gnu.org/philosophy/why-free.html>，以及《如今自由軟件更加重要》(p. 32)。

1.2.3 嚴酷的道德抉擇

沒了社區，繼續像從前那樣是不可能了。與此相反，我面臨著一個嚴酷的道德抉擇。

最簡單的選擇是加入私有軟件世界，簽署保密協議並下保證不去幫助黑客同仁。很可能我也會開發在保密協議下分發的軟件，因而壓迫他人並迫使他們也去背叛他們的夥伴。這樣我可以掙錢並在寫代碼時樂在其中。但我知道當我在事業的盡頭，回望築牆分化人們的歲月時，我將發現我竟耗費了生命以使這個世界變得更糟。

當有人拒絕把我們的打印機控制程序（該程序缺少部分特性使得打印機極其難用）的源代碼交給我和 MIT 人工智能實驗室時，我已經嘗過了當保密協議接受端的滋味。所以我無法對自己說保密協議是無辜的。當他拒

絕和我們分享代碼時我很生氣，己所不欲，毋施於人。

另一個選擇是離開計算機領域，直截了當但令人不快。這樣我的技藝不會被濫用，但它們會荒廢。我將不會因分化限制計算機用戶而被指責，但這種事仍然會發生。

所以我求索一條路線讓程序員可以做一些好事。我們心自問，有沒有可能有什麼項目或程序可以由我來寫，以再重新成立一個社區？

答案很清晰：我們首先需要的是一個操作系統。它是開始使用一臺計算機的至關重要的軟件。有了操作系統你可以做很多事；沒有操作系統你就完全無法使用計算機。有了自由的操作系統，我們就可以再次擁有互助的黑客社區——並邀請任何人加入。而且任何人都能使用計算機而毋須圖謀去剝奪他們的朋友。

作為一個操作系統開發者，我擁有適當的技藝。故我覺上天降此大任於我，縱不覺註定成功。我決定將系統做成和 Unix 兼容以使其可移植，而且這樣 Unix 用戶可以很容易切換到它。追隨黑客傳統，我選了 GNU 這個名字，即“GNU’s Not Unix”（意為“GNU 不是 Unix”）的遞歸縮寫。

一個操作系統並不意味著僅僅是一個內核，這樣僅僅足夠來運行其他程序。在 20 世紀 70 年代，能稱得上操作系統的軟件都包含了命令處理器、彙編器、編譯器、解釋器、調試器、文本編輯器、郵件程序，以及其它許多程序。ITS、Multics、VMS 和 Unix 都有這些。GNU 操作系統也得有。

後來我聽到希勒爾的這些話¹：

我不為我誰為我？我只為我我為何？此時不作更待何時？

啟動 GNU 工程的決定正是基於類似的情懷。

1.2.4 Free 是“freedom”的“free”

“free software”一語常常引發誤解——它無關價格、它關乎自由。以下是自由軟件的定義：

一個軟件對你，一個特定用戶而言是自由軟件，當：

¹作為無神論者，我不追隨任何宗教領袖，但我有時發現我欽佩他們說過的一些話。

- 不論目的為何，有使用該軟件的自由。
- 有修改軟件以符合自身需求的自由。（為使得該自由有效行使，你必須能夠訪問源代碼，因為在沒有源代碼的情況下修改程序非常困難。）
- 有重新分發該軟件副本的自由，既可免費亦可收費。
- 有發行該軟件的修改版的自由，這樣社區將從你的改進中受益。

因為“free”指的是自由，而不是價格，所以在銷售副本和自由軟件之間沒有矛盾。事實上，銷售副本的自由是很重要的：以 CD-ROM 形式賣出的自由軟件集對社區很重要，並且出售它們是為自由軟件開發籌集資金的重要方法。因此，無法給予人們這些自由的程序就不是自由軟件。

因為“free”一詞的二義性，人們用了很長時間找尋替代詞，但更好的詞彙尚未找到。英語比其它語言有更多的單詞和詞彙間的微妙差別，但它缺少一個簡單而明確的單詞用來表示“自由”，就像在“freedom”一詞中——最接近此含義的單詞是“unfettered”。諸如“liberated”，“freedom”和“open”等替代都有錯誤的含義或一些其它缺點。

1.2.5 GNU 軟件和 GNU 系統

開發一個系統是個非常大的工程。為了達成目標，我決定儘可能適配並使用現有的自由軟件。例如，一開始我就決定用 $\text{T}_{\text{E}}\text{X}$ 來做主要的文本排版器；若干年後，我決定使用 X 窗口系統而不是為 GNU 再寫一個窗口系統。

由於上述決定，和其他類似的決定，GNU 系統不同於所有 GNU 軟件的集合。GNU 系統包含非 GNU 的軟件，這些程序是由其他人或項目為了他們自己的目的而開發的。我們之所以能用它們是因為它們是自由軟件。

1.2.6 工程啟動

我在 1984 年 1 月辭去了 MIT 的工作而開始編寫 GNU 軟件。離開 MIT 是必要的，這樣 MIT 就無法干涉我將 GNU 作為自由軟件發行。如果我還在職，MIT 可能會要求擁有這些作品，並強加他們自己的發行條款，甚至

將它們變成一個私有軟件包。我不希望做大量工作卻只是看到它背離其初衷：創建一個新的軟件分享社區。

儘管如此，Winston 教授，後來的 MIT 人工智能實驗室的領導，友善地邀請我繼續使用實驗室的設施。

1.2.7 第一步

GNU 工程開始前不久，我聽說了自由大學編譯器工具包，又稱 VUCK（荷蘭語的“自由”一詞以 V 開頭）。該編譯器設計成支持多種編程語言，包括 C 和 Pascal，並支持多種目標機器，我曾寫信給其作者詢問 GNU 是否可以使用它。

他帶著嘲弄回答了，說大學是自由的而編譯器不是。因此我決定首先為 GNU 工程而寫的程序就是一個支持多種語言，多平臺的編譯器。

我希望能避免只靠自己編寫整個編譯器，因此我要來了 Lawrence Livermore 實驗室開發的多平臺編譯器 Pastel 的源碼。它支持一種適合系統編程的 Pascal 語言的擴充版本，並由該語言寫成。我給它加上了一個 C 語言前端，並開始將其移植到 Motorola 68000 計算機。但當我發現該編譯器需要數兆字節的棧空間，而可用的 68000 Unix 系統僅允許 64k 時，我只得放棄。

隨後我瞭解到 Pastel 編譯器的工作方式是分析整個輸入文件得到一個語法樹，將整個語法樹轉化為一條“指令”鏈，再產生整個輸出文件，整個過程不釋放任何內存空間。到此為止，我總結我只得從頭開始寫一個新的編譯器。那個新編譯器現在叫 GCC：其中沒有一點 Pastel 編譯器的內容，不過我仍努力把之前寫的 C 前端適配上去並用上。但那是幾年後的事了：首先，我做出了 GNU Emacs。

1.2.8 GNU Emacs

我從 1984 年 9 月開始寫 GNU Emacs，從 1985 年初它就開始可用了。這使得我開始可以使用 Unix 系統編輯文件；因為沒有興趣學用 vi 或 ed，

在那之前我在其他類型的機器上編輯文件。

這時候，人們開始想用 GNU Emacs，因此出現了該如何發行它的問題。當然，我把它放在了我在 MIT 時用的計算機的匿名 ftp 服務器上（那臺計算機 `prep.ai.mit.edu`，因而成了主要的 GNU ftp 發行站點，當它一年後退役時，我將其域名轉移到我們的新 ftp 服務器上）。但在那時，不少感興趣的人們並不在互聯網上因而無法通過 ftp 得到副本。所以問題是，我該跟他們說什麼？

我可以說，“找個上網的朋友幫你下載一個”。或者我可以像對原來 PDP-10 Emacs 那樣做：跟他們說，“寄一盤磁帶和 SASE（貼足郵資寫明發信人）來，我會把 Emacs 寫到磁帶上寄回去。”但我沒有工作，而我正在尋找通過自由軟件掙錢的方法。所以我宣佈我會寄一盤磁帶給任何想要的人，要價 150 美元。我以這種方式啟動了發行自由軟件的事業，那是今天發行整個 GNU/Linux 系統的公司們的先驅。

1.2.9 是一個對任何用戶都自由的程序嗎？

如果一個程序離開作者的手時是自由軟件，這並不一定意味著它對擁有其副本的每一個人都是自由軟件。例如，公有領域的軟件¹（沒有版權的軟件）是自由軟件；但任何人都可以製作由它修改而來的私有版本。類似的，不少自由程序是被版權保護但按照一個簡單的、允許私有修改版的寬容性協議發行。

這個問題的典型例子是 X 窗口系統。這是一個由 MIT 開發並以一個寬容性協議發行的自由軟件，它很快被各個計算機公司接受。它們把 X 以僅有二進制的形式加到它們的私有 Unix 系統中，並被同樣的保密協議控制著。這些 X 的副本和 Unix 一樣，已經不再是自由軟件。

X 窗口系統的開發者們並不認為這是一個問題——他們期望並有意使其發生。他們的目標不是自由，而僅僅是“成功”，那種定義為“有許多用戶”的成功。他們不在意這些用戶是否擁有自由，只是希望他們人數眾多。

這導致一個矛盾的情形，兩種不同的自由度計算方法對同一個問題

¹參見《自由與非自由軟件的分類》中的更多公有領域軟件的相關內容 (p. 82)。

“這個程序自由嗎？”給出不同的回答。如果你基於 MIT 許可證的發行條款給出的自由作判斷，你就會說 X 是自由軟件。但是如果你以一般 X 用戶的自由來衡量，你就只能說它是私有軟件。大多數 X 用戶當時正在使用的是隨 Unix 系統而來的私有版本，而不是自由版本。

1.2.10 左版 (Copyleft) 和 GNU GPL

GNU 的目標是給與用戶自由，而不僅僅是流行。所以我們需要使用可以阻止 GNU 軟件被轉變成私有軟件的發行條款。我們使用的方法叫“左版”¹。

左版使用版權法，但使它的作用與之通常的作用相反：它成為一種保持程序自由的手段，而不是限制程序的手段。

左版的核心思想是給與任何人運程序、複製程序、改寫程序，和發行改寫後的程序的許可——但那些人無權添加自己的限制。這樣一來，定義“自由軟件”的關鍵自由藉由每個擁有副本的人得以保證，這些自由成了不可剝奪的權利。

對於一個有效的左版，修改版也得是自由的。這確保了建立在我們的工作基礎上的作品發佈後將有用於社區。當以程序員為業者志願改進 GNU 軟件時，左版能防止他們的僱主說：“你不能分享那些改進，因為我們要用它們來做一個我們的私有版本。”

如果我們要確保程序的每個用戶的自由，就需要做出修改保證必須的自由。那些私有化 X 窗口系統的公司通常作了一些修改以將其移植到它們的系統和硬件。這些改動與 X 的大規模擴展相比而言是較小的，但是它們並非微不足道。如果進行修改是拒絕用戶自由的一個藉口，任何人來利用這一藉口都是非常容易的。

一個有關的問題涉及將自由的程序和非自由的代碼結合到一起。這樣

¹在 1984 或 1985 年，Don Hopkins（一位非常有想象力的朋友）寄給我一封信。在信封上他寫下了不少有趣的話，包括這句：“Copyleft——逆轉一切權利。”我用“copyleft”一詞來命名我正在開發的發行理念。（本文採用了賈星克、李極光發表在雲南師範大學學報的《論自由軟件運動》一文中對 copyleft 一詞的譯法，即“左版”。——譯者注。）

的結合體將不可避免變得不自由；任何一個在非自由部分上缺失的自由也將在整體上缺失。允許這樣的結合將會打開足以沉掉一艘船的缺口：任何添加或結合到左版程序上的東西必須使得更大的結合版也是自由和左版的。

我們明確為大多數 GNU 軟件使用了左版的 GNU 通用公共許可證，或簡稱 GNU GPL。我們在特定場合下有其它種類的左版可以使用。GNU 手冊也是左版的，但使用一個非常簡化的左版類型，因為 GNU GPL 的複雜性對手冊是不需要的¹。

1.2.11 自由軟件基金會

隨著使用 Emacs 的興趣的增長，開始有其他人加入 GNU 工程，我們感到再次籌集資金的時候到了。所以我們在 1985 年建立了自由軟件基金會（FSF），一個發展自由軟件的免稅慈善機構。FSF 也接手了發行 Emacs 磁帶的工作；後來通過將其他自由軟件（既有 GNU 的也有非 GNU 的）加到磁帶上，和賣軟件的自由手冊擴展了該業務。

FSF 的大部分收入曾經來自銷售自由軟件的副本和其他相關服務（源代碼的 CD-ROM、二進制文件的 CD-ROM、精心印刷的手冊，都有著再發行和修改的自由），以及豪華的發行版（我們為客戶選擇的平臺定製的完整的軟件集）。今天 FSF 仍然銷售手冊和其他部件²，但大部分的資金來自成員的會費。你可以通過 <http://fsf.org/join> 來加入 FSF。

自由軟件基金會的僱員已經編寫並維護了大量的 GNU 軟件包。兩個值得注意的是 C 庫和 shell。GNU C 庫是運行於 GNU/Linux 系統上的任一程序用於和 Linux 通信的組件，由自由軟件基金會的成員之一，Roland McGrath 所開發。用於大部分 GNU/Linux 系統的 shell 是 BASH，“Bourne Again Shell”³，由 FSF 僱員 Brian Fox 編寫。

¹我們現在對文檔使用 GNU 自由文檔許可證。

²於我們的在線商店中可見，位於 <http://shop.fsf.org>。

³“Bourne Again Shell”是對“Bourne Shell”——Unix 上普遍使用的 shell 玩的文字遊戲。（Bourne 與表示出生的 born 諧音——譯者注）

我們資助了這些程序的開發是因為 GNU 工程並不僅僅與工具和開發環境有關。我們的目標是一個完整的操作系統，而該目標需要這些程序。

1.2.12 支持自由軟件

自由軟件的哲學抵制一種特定的分佈極廣的商業實踐，但它不反對商業。當商業尊重用戶的自由時，我們祝願它們成功。

銷售 Emacs 的副本展現了一種自由軟件的生意。當 FSF 接手了這項生意之後，我需要另一種方法謀生。我在銷售與我所開發的自由軟件相關的服務中找到了它。它包括教人們諸如對 GNU Emacs 編程，定製 GCC，和主要是移植 GCC 到新平臺的軟件開發的課題。

今天，每個此類的自由軟件生意被許多公司實踐著。有的通過 CD-ROM 發行自由軟件集；其他的則銷售服務，在從回答用戶問題，到改正程序錯誤，以至增加大的新功能等不同層次上。我們甚至開始看到基於發起新的自由軟件產品的自由軟件公司。

值得特別注意的是，不少公司儘管將它們自己與“open source”一詞聯繫在一起，實際上它們的生意是基於與自由軟件一起工作的非自由軟件。它們不是自由軟件公司，它們是私有軟件公司，其產品誘惑用戶遠離自由。它們稱此為“增值包”，反映了它們希望我們接受的價值觀念：便利在自由之上。如果我們更珍惜自由，我們應該稱它們為“去自由的”包。

1.2.13 技術目標

GNU 的首要目標是作為自由軟件。即使 GNU 對 Unix 沒有技術優勢，它也有允許用戶合作的社會優勢；它還有道德優勢：尊重用戶的自由。

但是將已知好的實踐檢驗過的標準應用到工作上是很自然的——例如，動態地分配數據結構以避免武斷地固定大小限制，並在任何有意義之處處理所有可能的 8 位代碼。

另外，我們放棄了面向小內存的 Unix 設計，決定不支持 16 位機器（顯然 32 位機器在 GNU 系統完成時將成為主流），並且不為了減少內存使

用而作任何努力，除非超過了一兆字節。在處理非常大但不是至關緊要的文件程序中，我們鼓勵程序員們將整個文件讀入內存，然後掃描其內容而不必顧慮輸入輸出的問題。

這些決定使得不少 GNU 程序在可靠性和速度上超越了在 Unix 上的對應的程序。

1.2.14 捐來的計算機

當 GNU 工程聲名鵲起，人們開始給該工程捐贈運行 Unix 的計算機。這非常有用，因為開發 GNU 組件最輕鬆的方法就是在一個 Unix 系統上做，然後一個一個地替換掉 Unix 系統上的組件。但這引發了一個道德問題：我們擁有 Unix 的副本從根本上說是不是正當的。

Unix 以前是（並且現在還是）私有軟件，而 GNU 工程的哲學說我們不該用私有軟件。然而，應用那些與推論出“自衛的暴力是正當的”相同的理由，我的結論是：在開發用於幫助他人停用私有軟件包的自由軟件代替品的關鍵時刻，使用私有軟件包是合理的。

但是，儘管這是一個可以合理化的罪惡，它仍然是罪惡。今天我們已經不再持有任何 Unix 的副本，因為我們已經用自由的操作系統取代了它們。如果我們不能將一臺計算機的操作系統換成自由的，我們就把整臺計算機都換掉。

1.2.15 GNU 任務清單

隨著 GNU 工程的進行，以及越來越多的系統組件被找到或開發，最終使得整理一份未完成工作的清單變得很有用。我們用它來招募開發者來編寫缺失的部分。它被稱為 GNU 任務清單。除了尚未完工的 Unix 組件外，我們列出了額外的各種各樣的其它有用的軟件和文檔項目，我們認為，這些是一個真正完整的系統所應當擁有的。

如今¹，留在 GNU 任務清單中的 Unix 組件除了一些無關緊要的之外已經幾乎沒有了——它們都已經被完成了。但清單中充滿了可以被稱為“應用程序”的項目。將任何不止能吸引一小部分用戶的程序加到操作系統中都是有益的。

就連遊戲都在任務清單裡——而且從一開始就有。Unix 包含遊戲，所以 GNU 自然也該包含。但兼容性對遊戲來說不是問題，所以我們沒有跟著 Unix 已有的遊戲列表走。作為代替，我們列出了用戶可能會喜歡的一系列不同種類的遊戲。

1.2.16 GNU 庫 GPL

GNU C 庫使用一種特別的左版，稱作 GNU 庫 GPL (LGPL)²，允許將私有軟件鏈接到該庫上。為什麼需要這樣的特例？

這不是個原則問題：沒有哪個原則說私有軟件產品有資格包含我們的代碼。（為什麼要為一個嚴辭拒絕與我們分享的工程作貢獻呢？）為 C 庫，或任何庫使用 LGPL，是個策略問題。

C 庫做的是通用的工作：每個私有系統或編譯器都附帶 C 庫。因此，讓我們的 C 庫只能為自由軟件所用不會為自由軟件帶來任何優勢——這隻會嚇阻人們使用我們的庫。

有一個系統是這個的例外：在 GNU 系統（包括 GNU/Linux）中，GNU C 庫是唯一的 C 庫。所以 GNU C 庫的發行許可決定了是否可以為 GNU 系統編譯私有程序。沒有道德理由允許私有應用在 GNU 系統中運行，但從戰略上來看禁止它們會更多地嚇阻人們使用 GNU 系統，而不是鼓勵開發自由應用。所以使用庫 GPL 對 C 庫是個好策略。

對於其他的庫，策略性的決定需要具體問題具體分析。當一個庫做的

¹寫於 1998 年。2009 年我們不再維護長任務列表。社區的自由軟件開發得很快，我們沒法全部追蹤到。取而代之，我們有一個高優先級項目的列表——一個更短的列表，列舉了我們十分想要鼓勵人們去寫的項目。

²這個許可證現在叫 GNU Lesser General Public License (GNU 寬通用公共許可證)，以免給出所有庫都該使用它的印象。更多信息見《為什麼你不應該在下一個函數庫中使用寬 GPL》於 <http://www.gnu.org/philosophy/why-not-lgpl.html>。

是一種能幫助編寫特定種類程序的特殊工作時，那麼將其用 GPL 發行，限制其只能被用於自由軟件，是一種幫助其它自由軟件開發者的方法。這給了他們面對私有軟件的一個優勢。

想一下 GNU Readline，一個被開發用來為 BASH 提供命令行編輯功能的庫。Readline 是用普通的 GNU GPL 而不是庫 GPL 發行的。這可能確實減少了 Readline 的使用量，但這對我們沒有損失。與此同時，至少有一個可用的應用特地為了能夠使用 Readline 而變成了自由軟件，那是社區真正的收穫。

私有軟件有金錢提供的優勢；自由軟件開發者則要相互取得優勢。我希望有朝一日我們能擁有大量受 GPL 保護的、沒有可用的私有替代的庫，提供作為新自由軟件的磚石的有用模塊，併為進一步的自由軟件開發添加巨大的優勢。

1.2.17 搔到癢處？

Eric Raymond¹ 說“每一個優秀的軟件作品都從搔到開發者個人的癢處（意為‘解決開發者個人的問題’——譯者注）開始²。”也許有時是這樣。但不少 GNU 軟件的關鍵部分是為了一個完整的自由操作系統而開發的。它們來自願景和計劃，而不是衝動。

例如，我們開發了 GNU C 庫因為類 Unix 系統需要一個 C 庫，開發了 BASH 因為類 Unix 系統需要一個 shell，還有 GNU tar 因為類 Unix 系統需要一個 tar 程序。我們自己的程序也同樣如此——GNU C 編譯器，GNU Emacs，GDB 和 GNU Make。

有些 GNU 程序是為應對我們的自由的特定威脅而開發的。為此，我們開發了 gzip 來取代因 LZW 專利而從社區流失的 Compress 程序。我們發現有人開發 LessTif，後來我們還開始了 GNOME 和 Harmony，來解決因某些私有軟件庫（見下文）所帶來的問題。我們正在開發 GNU 隱私衛士（即

¹Eric Raymond 是開源的主要擁護者；參見《為什麼說開源漏掉了自由軟件的要點》(p. 90)。

²埃裡克·雷蒙德 (Eric. S. Raymond)，《大教堂與集市》；此書已由機械工業出版社於 2014 年出版，衛劍飢譯。

GnuPG 簡稱 GPG。基於 OpenPGP 協議實現的自由的用於加密、數字簽名及產生非對稱密鑰對的軟件。——譯者注）來取代流行的非自由加密軟件，因為用戶不應該在隱私和自由之間作出選擇。

當然，寫這些程序的人們變得對這項工作感興趣，許多人為了自己的興趣和需要給它們添加了很多功能。但那並不是這些程序存在的原因。

1.2.18 出乎意料的發展

在 GNU 工程剛剛開始的時候，我覺得我們將開發整個 GNU 系統，然後整個發行。而這並沒有發生。

因為 GNU 系統的每一個組件都是在 Unix 系統中實現的，每個組件早在一個完整的 GNU 出現以前就都可以在 Unix 系統中運行。這些程序有的變得流行，而用戶們開始擴充並移植它們——到各種互不兼容的 Unix 版本上，有時也會移植到其他系統。

這個過程使得這些程序更加強大，且為 GNU 工程引來了資金和貢獻者。但或許也使得一個最小可用系統延遲了數年，因為 GNU 的開發者們把時間投入到維護這些移植版和為已有組件增加特性，而不是去編寫一個個缺失的組件上。

1.2.19 GNU Hurd

到了 1990 年，GNU 系統幾乎已完成了：唯一主要的缺失部分是內核。我們已經決定將我們的內核實現為一組運行在 Mach 上的服務進程。Mach 是一個由卡內基梅隆大學，而後在猶他大學開發的微內核；GNU Hurd 是運行於 Mach 之上一組服務（正如一群牛羚——GNU）（Hurd 與 herd 諧音，而 herd 有“群”的意思——譯者注），負責 Unix 內核的各種任務。開發的啟動有所延誤，因為我們要等 Mach 像它承諾的那樣作為自由軟件發行。

選擇這種設計的一個原因是為了避免此工作中看起來似乎是最困難的一部分：在沒有一個源碼層調試器的條件下調試內核程序。這部分工作在 Mach 中已被完成，因此我們期待能將 Hurd 服務作為用戶程序來調試，用

GDB。但這花了很長時間才做到，而互相發消息的多線程服務群實際上非常難以調試。這使得讓 Hurd 能夠穩定工作的進程延長了很多年。

1.2.20 Alix

GNU 的內核原本並未打算叫 Hurd。它原本的名字是 Alix——以我當時的戀人命名。她，作為一個 Unix 系統管理員，指出她的名字是多麼符合 Unix 系統版本的一般命名模式；作為玩笑，她跟朋友說：“有人一定得用我的名字給一個內核起名。”我什麼都沒說，但打算用一個叫 Alix 的內核讓她吃一驚。

事情並沒有保持不變。Michael（現在叫 Thomas）Bushnell，內核的主要開發者，鐘意 Hurd 這個名字，並重新定義了 Alix 來表示內核的某個特定部分——用於捕獲系統調用並向 Hurd 服務器發消息以處理的那個部分。

後來，Alix 和我分手了，她還改了名字；與此獨立地，Hurd 的設計改變了，以至於 C 庫直接向服務器發消息，而這使得那個 Alix 組件從設計中消失了。

但在這些事情發生以前，她的一個朋友在 Hurd 的源碼中偶然見到了 Alix 這個名字，並告訴了她。所以她確實有機會發現有一個內核以她命名。

1.2.21 Linux 和 GNU/Linux

GNU Hurd 尚不適合用於生產，我們也不知道它還能不能適合。這個基於能力的設計有著直接來源於設計靈活性的問題，而且不知道有沒有解決方案。

幸運的是，有另一個內核可用。在 1991 年，Linus Torvalds 開發了一個 Unix 兼容的內核並稱之為 Linux。它一開始是私有的，但在 1992 年，他使其成為自由軟件；整合 Linux 和尚不完整的 GNU 得到了一個完全自由的操作系統（當然，整合工作自身也很重要）。正是因為 Linux，我們今天終於能運行 GNU 系統的一個版本。

我們稱這個版本的系統為 GNU/Linux，以表達他是由 GNU 系統和作

為內核的 Linux 整合而成的。不要實用主義地稱整個系統為“Linux”，因為那將我們的工作歸於他人。請同等程度地提及我們¹。

1.2.22 未來的挑戰

我們已經證實了我們開發多種類型的自由軟件的能力。但這並不意味著我們是不可戰勝、不可阻擋的。一些挑戰使得自由軟件的未來變得不確定；與它們會戰將需要堅實的努力和耐力，有時要奮戰數年。這將會需要那種當人們珍惜他們的自由，並且不讓任何人將其奪走時所顯示的決心。

下邊四個段落將討論這些挑戰。

機密硬件

硬件廠商愈發傾向於對硬件規格保密。這使得編寫讓 Linux 和 XFree86 能支持新硬件的自由驅動程序變得很難。現在我們有了完整的自由操作系統，但是如果不能支持明天的計算機，我們將會在明天失去它們。

有兩種方法來應付這個問題。程序員可以採取逆向工程的手段來了解如何支持這些硬件。其他的人則可以選用被自由軟件支持的硬件；隨著我們的人數增加，規格保密將成為一個自取滅亡的策略。

逆向工程是件大工作；我們會有程序員具備足夠的決心去擔負這件工作嗎？是的——如果我們已經建立了認為自由軟件是個原則問題，而私有驅動程序不可容忍的堅定信念。我們中的大多數人會額外花錢，或甚至額外花時間，從而我們可以使用自由驅動程序嗎？是的，如果擁抱自由的決心被廣泛傳播。

[2008 腳註：這個問題同樣延伸到 BIOS。有一個自由的 BIOS，Libre-Boot²（coreboot 的一個發行版）；這個問題對計算機變得重要起來，因為 LibreBoot 可以不用非自由的“Blob”就能支持它們。]

¹更多信息見“GNU/Linux FAQ”，於 <http://gnu.org/gnu/gnu-linux-faq.html>，和《Linux 和 GNU 操作系統》（p. 75）。

²參見 <http://libreboot.org>。

不自由的庫

運行於自由操作系統上的非自由庫表現得就像針對自由軟件開發者的陷阱一般。庫的誘人特性是誘餌；如果你用了這個庫，你就掉進了陷阱，因為你的程序無法有用地成為自由操作系統的一部分（嚴格來說，我們可以包含你的程序，但沒有了該庫它就無法運行）。更糟糕的是，如果一個使用私有庫的程序流行起來，它可引誘其他沒有懷疑的程序員們落入陷阱。

這樣的程序的第一個實例是 80 年代的 Motif 工具箱，儘管那時還沒有自由操作系統，但是很顯然 Motif 以後會對自由操作系統引發什麼問題。GNU 工程通過兩種方式迴應這個問題：通過請求個別的自由軟件工程在支持 Motif 的同時也支持自由的 X 部件工具箱，並請求一些人編寫替代 Motif 的自由軟件。該工作花費許多年時間；由匈牙利程序員們開發的 LessTif，在 1997 年才變得足夠強而得以支持大多數 Motif 應用程序。

在 1996 到 1998 年間，另一個不自由的 GUI 工具箱庫，叫 Qt，被用在 KDE 桌面這一包含大量自由軟件的集合中。

自由的 GNU/Linux 系統不能使用 KDE，因為我們不能使用那個庫。但是一些不嚴格堅持自由軟件的 GNU/Linux 系統商業發行者將 KDE 加入到它們的系統中——而產生了有更強能力，和更少自由的系統。KDE 小組積極地鼓勵更多的程序員們使用 Qt，成百萬的新“Linux 用戶”從來都不知道有這樣一個問題的存在。情形相當糟糕。

自由軟件社區以兩種方法應對這個問題：GNOME 和 Harmony。

GNOME，GNU Network Object Model Environment（譯為“GNU 網絡對象模型環境”，現在已經不再用這個說法——譯者注），是 GNU 的桌面項目。從 1997 年開始，由 Miguel de Icaza 在紅帽軟件的支持下開發，GNOME 開始提供類似的桌面工具，但排它地只使用自由軟件。它也有技術上的優勢，如支持多種語言，而不僅僅是 C++。但是它的主要目的是自由：不需要使用任何非自由軟件。

Harmony 是一個兼容的替代庫，設計為使得無需 Qt 運行 KDE 軟件成為可能。

在 1998 年 11 月，Qt 的開發者聲明改動許可證，當其實施後，應該會

使得 Qt 成為自由軟件。雖然沒辦法確信，但是我想這應該部分歸功於社區對 Qt 是非自由軟件時所造成的問題的堅定迴應（新的許可證既不方便也不公正，所以仍舊值得去避免使用 Qt）。

[後記：2000 年 9 月，Qt 按 GNU GPL 發行，實際解決了這個問題。]

我們將如何應對下一個誘人的非自由庫呢？整個社區會明白要遠離陷阱嗎？或者我們中的許多人將為了方便而放棄自由，從而產生一個大問題？我們的未來將取決於我們自身的哲學。

軟件專利

我們面對的最惡劣的威脅來自軟件專利，它可以對自由軟件加上算法和功能的限制多達二十年。LZW 壓縮算法的專利申請於 1983 年，而我們仍然無法發行能夠生成適當壓縮過的 GIF 的自由軟件。[直到 2009 年這些專利才過期。]1998 年，由於專利訴訟威脅，一個用於生成 MP3 壓縮音頻的自由程序被迫從發行版中移除。

仍有辦法對付專利：我們可以尋找證據以證明一個專利是無效的，也可以尋找替代方法來完成工作。但是這每一種方法只是偶爾才起作用；當它們都失敗時，一個專利可能會迫使所有的自由軟件都缺少某些用戶想要的功能。當這種事發生時我們可以做些什麼呢？

我們中因自由而重視自由軟件的人們無論如何都將與自由軟件共進退。我們將設法不用專利保護的功能而完成工作。但是那些認為自由軟件技術出眾而重視它的人們，有可能在專利抑制自由軟件時認為這是自由軟件的失敗。因而，雖然討論軟件開發的“市集”模式的實用效力和一些自由軟件的可靠性和能力是有用的，但我們決不能止步於此。我們必須探討自由和原則。

自由文檔

我們自由操作系統的最大不足不是在軟件中——而是缺乏可以包含在我們系統中的優秀自由手冊。文檔資料是任何軟件包的要害部分；當一個重要的自由軟件包沒有與優秀的自由手冊一起出現，那就是一個重大缺

陷。今天我們有許多這樣的缺陷。

自由文檔，像自由軟件一樣，是自由問題，不是價格問題。自由手冊的標準幾乎與自由軟件完全相同：它是為了給予所有用戶某種自由。必須允許重新發布（包括商業銷售），不論是在線還是書面形式，因而手冊能夠伴隨每個程序的每個副本。

改動的許可同樣至關重要。作為一個普遍規律，我不相信人們有必要擁有修改所有種類文章和書籍的許可。例如，我不認為你或我應該感激被給予修改像本文這樣描述我們行為和我們觀點的文章的許可權。

但是有一個特殊的原因說明為什麼修改自由軟件文檔資料的自由是要緊的。當人們行使他們修改軟件的權利，並且增加或改變其功能時，如果他們是盡職的，則他們也會同時修改文檔和資料——因而他們能隨著修改過的程序一起提供正確和可用的文檔資料。一個不允許程序員們盡職並完成該工作的手冊不符合我們社區的要求。

關於修改應該如何完成的一些限制並不會造成問題。例如，保持原作者的版權聲明，發行條款，或作者列表的要求是正當的。要求修改後的版本包括它們是修改版本的聲明也是沒有問題的，即使有整個章節不能刪除或修改，只要這些章節處理的是非技術主題。這些類型的限制不是問題，因為它們不阻止盡職程序員修改手冊以適應修改過的程序。換種說法，它們不妨礙自由軟件社區完全利用該手冊。

然而，必須能夠修改手冊中的所有技術性內容，並將結果以所有常規介質，通過所有常規渠道分發；否則，這些限制將對社區造成阻礙，手冊將變得不自由，而我們需要另一手冊。

自由軟件開發者們會有覺悟和決心去生產全系列的自由手冊嗎？再說一次，我們的未來取決於哲學。

1.2.23 我們必須談論自由

估計當今有數千萬的用戶使用諸如 Debian GNU/Linux 和紅帽“Linux”這樣的 GNU/Linux 系統。自由軟件已經發展到了這樣實用的優勢，使得用戶純粹為了實用原因而聚集到它身邊。

這種現象的好結果是明顯的：更多人有興趣開發自由軟件，更多用戶參與到自由軟件產業，以及更能鼓勵公司開發商業自由軟件而不是私有軟件產品。

但是對軟件的興趣增長快於對其指導哲學的瞭解，這帶來了一定麻煩。我們面對上述挑戰和威脅的能力依賴於堅決主張自由的意志。為了確定我們的社區擁有這個意志，我們需要在新用戶來到社區時向他們傳播這樣的思想。

但是我們正在這點上失敗：吸引新用戶加入社區的努力大大超越了教育他們成為我們社區的好公民的努力。我們需要做這兩件事，而且我們也需要保持這兩個努力的平衡。

1.2.24 “開放源代碼”

當 1998 年一部分社區決定停止使用術語“自由軟件”並改為說“開源軟件”時，教導新用戶有關自由的觀念變得更加困難。

一些喜歡該術語的人想要避免“自由”與“免費”的混淆——這個目標是正當的。其他人卻打算將激勵了自由軟件運動和 GNU 工程的原則精神拋到一邊，反而迎合行政和商業用戶，而這些用戶中的許多人持有一種將利潤置於自由，社區和原則之上的意識形態。因而，“開源”的花言巧語集中在製作高質量，強有力軟件的潛能上，但是避開自由，社區和原則的思想。

《Linux》雜誌是一個清晰的例子——其中充滿了利用 GNU/Linux 運行私有軟件的廣告。當下一個 Motif 或 Qt 出現時，這些雜誌將警告程序員們遠離它還是為它登載廣告呢？

商業支持能以許多方式為社區作貢獻；其它種類的支持也都一樣，它是有益的。但是為了贏得他們的支持而減少對自由和原則宣講可能損失慘重；它使得前述“軟件推廣和公民意識教育”之間的失衡變得愈加糟糕。

“自由軟件”和“開放源碼”或多或少地描述了同一個軟件的類型，但對軟件本身和價值觀的描述是不同的。GNU 工程繼續使用“自由軟件”這個術語，來表達自由是重要的思想，而不僅僅是技術。

1.2.25 嘗試！

猶達大師的名言“沒有‘嘗試’”聽似優雅，但對我不起作用。我已經完成了我的大部分工作，與此同時擔心我能不能做，且不確定如果我做了，我的努力夠不夠完成目標。不過無論如何我嘗試了，因為在敵人和我的城池之間除了我沒有任何人。令我吃驚的是，我有時會成功。

有時我失敗了；我的一些城池陷落了。隨後我發現另一個受到威脅的城池，併為另一場戰鬥做好準備。天長日久，我學會了尋找威脅並將我置於城池和威脅之間，並召集其他黑客和我聯合。

今天，我不是在單打獨鬥。當我看見一個團的黑客挖戰壕堅守戰線時的感覺是一種安慰和樂趣，我意識到，目前，這個城池也許能倖存。但是危險與年俱增，並且現在 Microsoft 已經明確地將目標對準我們的社區。我們不能把自由的將來視為天命。別把它當作天命！如果你想要保持你的自由，你必須備戰以保衛它。

1.3 GNU 操作系統的初始公告

這是 GNU 工程的原始通告，由理查德·斯托曼（Richard Stallman）於 1983 年 9 月 27 日發表。

縱觀歷史，可以發現 GNU 工程在很多地方都與這份初始通告有很多差異。比如實際是拖延到了 1984 年 1 月才開始。而自由軟件的很多哲學理念也是數年之後才得以釐清。

```
From mit-vax!mit-eddie!RMS@MIT-OZ
From: `RMS%MIT-OZ@mit-eddie`
Newsgroups: net.unix-wizards,net.usoft
Subject: new Unix implementation
Date: Tue, 27-Sep-83 12:35:59 EST
Organization: MIT AI Lab, Cambridge, MA
```

自由的 Unix!

今年的感恩節我要去寫一個完整的類 UNIX 軟件系統，命名為 GNU (Gnu's Not Unix)，並以自由¹的方式開放給所有人使用。非常歡迎大家貢獻時間、金錢、程序和設備來參與其中。

首先，GNU 將是一個內核加上編寫和運行 C 程序所需的所有工具：編輯器，Shell，C 語言編譯器，鏈接器，彙編器，以及一些其它的東西。在此之後，我們將添加一個文本排版工具，一個 YACC，一個帝國的遊戲，電子表格，以及數百種其它的東西。我們希望最終可以提供一切通常和 UNIX 系統一起出現的有用的東西，並包括一份在線的和印刷版的文檔。

GNU 可以運行 UNIX 程序，但不會與 UNIX 完全相同。我們會基於在其它系統上的經驗完善之以使其更舒適。特別的，我們

¹這裡用詞沒有注意，初衷是想說沒有人需要為使用 GNU 系統而索要授權。然而這個詞沒有說的很清晰，結果人們經常以為獲取 GNU 的副本需要很少或者免費。而這從來都不是目的。

計劃使用更長的文件名，文件版本號，抗崩潰的文件系統，可能還有文件名自動補全，獨立於終端的顯示支持，以及一個基於 Lisp 的窗口系統，最終所有 Lisp 程序和 Unix 程序都可以共享同一個屏幕。C 語言和 Lisp 都可以作為系統編程語言。還會有基於 MIT 的 chaosnet 協議的網絡軟件，會遠遠優於 UUCP。我們還會有某些東西來兼容 UUCP。

我是誰？

我是理查德·斯托曼，發明了備受模仿的原版 EMACS 編輯器，現在供職於 MIT（麻省理工大學）的人工智能實驗室。我廣泛從事過編輯器、編譯器、調試器、命令解釋器、不兼容分時系統、Lisp 機器操作系統。我率先在終端獨立顯示支持 ITS。此外我還實現了一個可靠的文件系統和兩個 Lisp 機器的窗口系統。

我為什麼必須寫 GNU

我信奉的一條金科玉律就是如果我喜歡一個程序，那麼我必須與其他人一起分享。我不能憑良心簽署保密協議或軟件許可協議。

因此我不能繼續使用那些破壞我原則的電腦，因此我決定將所有自由軟件組合在一起，這樣我就可以不使用任何不自由的軟件了。

如何貢獻其中

我正在向計算機廠商索要捐獻一些機器和金錢。還向一些個體索要程序和作品。

其中一家廠商已經提供了我一臺機器。但我們可以用更多。如果你能捐助更多機器，那麼 GNU 將會更早期在上面運行。機器最好能夠在一個住宅區內進行操作，並且不需要複雜的冷卻或供電。

獨立程序員可以寫一些 Unix 工具的兼容複製品並將其貢獻給

我。對大多數項目而言，如此大規模的分佈式項目通過兼職很難協作，獨立完成的部分可能很難組合在一起。然而對替換 Unix 的任務而言，這個問題並不存在。大多數接口規格已經通過 Unix 兼容固定下來了。如果每個貢獻者的作品都可以和 Unix 的剩餘部分工作在一起，那麼多半一樣可以和 GNU 的剩餘部分一起工作。

如果我獲得了捐助的金錢，我可能需要聘用一些人全職或者兼職工作。薪水可能不高，但是我希望找到這樣的人，對他們而言認識到自己工作是幫助人類和賺錢一樣重要。我將其看作一種讓有獻身精神的人們無需按它法謀生，而將他們的全部精力投入到 GNU 的工作上的途徑。

聯繫我以獲取更多資訊。

Arpanet 郵件：

RMS@MIT-MC.ARPA

Usenet：

...!mit-eddie!RMS@OZ ...!mit-vax!RMS@OZ

美國郵政：

Richard Stallman
166 Prospect St
Cambridge, MA 02139

1.4 如今自由軟件更加重要

Copyright © 2015 理查德·斯托曼 (Richard Stallman)。此文大幅修改後發表於《連線》雜誌 (Wired) 網站，標題是“[Why Free Software Is More Important Now Than Ever Before](#)”，2013 年 9 月 28 日。

自 1983 年以來，自由軟件運動一直在為計算機用戶的自由而戰——用戶控制他們使用的軟件，而不是相反。當一個程序尊重用戶和社區的自由時，我們把它稱為“自由軟件”。

我們有時也稱之為“libre software”以便澄清我們關注的是自由，而不是價格。一些專有（非自由）軟件比如 Photoshop，非常昂貴；而其他比如 Flash 播放器則免費提供——所以價格並不是大問題。無論高價還是免費，它們都將程序開發者的權力凌駕於用戶之上，而這種權力是不應該存在的。

同時這些軟件還有個共同點：他們都是惡意軟件。也就是說從功能上都是設計用來虐待用戶的。今天的專有軟件往往是惡意軟件，因為開發者的權力腐化了它們¹。而自由軟件，用戶控制著程序，既是單獨用戶控制同時也是群體控制。這樣他們就控制著計算機做的事情（假設這些程序都是忠實完成用戶的指令）。

專有軟件程序控制著用戶，另一些（開發者或“所有者”）則完全控制著程序。因此專有程序給了開發者凌駕用戶的權力。而這本身就是不公平的，並且誘使開發者用其他方式虐待用戶。

自由意味著你控制著自己的生活。如果你使用程序來打理你的生活，你的自由就取決於你如何控制著這些程序。你應該擁有對程序的控制權，更何況這些程序控制著你生活中重要的事情。

用戶對程序的控制需要四項基本的自由²。

1. 基於任何目的，按你的意願運行軟件的自由。

¹參見 <http://gnu.org/proprietary/proprietary.html> 可知這些不斷增加的威脅。

²自由軟件的完整定義可參見《什麼是自由軟件？》(p. 1) 一文。

2. 學習軟件“源代碼”並修改的自由，這樣可以讓程序執行你想做的事情。程序是由程序員使用編程語言編寫的（比如結合英語和代數），這種形式稱為“源代碼”。任何熟悉編程，並能以源代碼形式編程的人，都可以讀源代碼，懂得其邏輯，並可以修改之。當你只能得到可執行格式，也就是對計算機來說能理解，但對人類極難讀懂的一系列數字時，讀懂並修改該形式的程序難如登天。
3. 將當前副本重新分發的自由（這不是一種義務，這樣做是你的選擇。如果程序是自由的，並不意味著別人有義務提供給你一份副本，或者你有義務為其他人提供副本。分發程序而沒有自由，會虐待用戶。然而如果不分發軟件——只是私下使用——則不會虐待任何人）。
4. 隨時分發你修改過的版本的自由。

前兩個自由表示用戶可以實際控制程序。而剩下的自由，表示任何用戶團體都可以集體控制程序。具備這四個自由，用戶就可以完全控制程序。這四個自由缺一不可，否則程序就是專有的（非自由的），並且不道德。

任何實用性的作品都可以適用，包括烹飪菜譜，教育作品比如教科書，參考書比如字典和百科全書，顯示文章的字體，硬件設計的電路程序，3D打印用的（不只是裝飾品）模型文件。因為這些並不是軟件，所以自由軟件運動嚴格上來說並不包括它們，但同理可證：這些作品也需要符合上面這四項自由。

一個自由的程序允許你按照自己的想法去改造。對於將專有軟件看成密封盒子的人來說，改造軟件看起來很荒謬，然而在自由的世界裡則是非常普遍，並且對學習編程非常有利。傳統美國消費者對汽車的改造並不順暢，恰是因為汽車包含非自由的軟件。

1.4.1 專有化的不公

如果用戶沒有控制程序，那麼程序就在控制用戶。對專有軟件，有一些實體比如開發者或者程序“所有者”，控制著程序，將權力凌駕於用戶之上。一個非自由的程序就像個操縱桿，是一把操縱不公權力的工具。

令人髮指的是（這種情況很常見）專有程序設計用來窺視、限制、審查甚至虐待用戶¹。例如蘋果的 iThings 操作系統做了所有這些，同樣的基於 ARM 芯片移動設備上的 Windows 系統也是如此。Windows 手機固件以及 Google Chrome 的 Windows 版包含了通用後門，可以讓一些公司不經過用戶同意就遠程修改程序。亞馬遜的 Kindle 則通過後門刪除用戶的電子書。

“物聯網”（internet of things）產品上使用的非自由程序會將物聯網變成“騙聯網”（internet of telemarketers）²或“窺聯網”（internet of snoopers）。

為了結束非自由程序的不公，自由軟件運動開發了自由的程序，這樣用戶可以解放自己。我們首先在 1984 年開始開發自由的操作系統 GNU。現在數以萬計的計算機運行著 GNU，主要是 GNU/Linux 結合體³。

分發非自由的程序給用戶是殘害用戶的行為；然而如果不分發程序則不會傷害任何人。如果你寫了一個程序並私下使用，不會傷害到任何人。你也許會失去做好事的機會，但這與做錯事是不一樣的。因此，我們說所有程序都應該自由，意思是所有副本都應該遵循這四個自由，但並不意味著別人有義務提供給你一份副本。

1.4.2 非自由軟件與 SaaS

非自由軟件是公司控制人們電腦的首選方案。今天，還有另外一種方式，稱為“替代軟件的服務”（SaaS，Service as a Software Substitute）。這意思是讓別人的服務器去做你自己的計算任務。

SaaS 並不意味著運行在服務器上的程序是非自由的（雖然大多數確實是非自由的）。然而，使用 SaaS 會導致與使用非自由程序一樣的不公：殊途同歸。就拿 SaaS 翻譯服務為例：用戶發送文本到服務器，服務器將其翻譯好發回用戶（比如從英文翻譯為西班牙文）。那麼翻譯的工作是由服務器運營商控制的，而不是用戶。

¹參見本文的第一個腳註。

²參見 Marcelo Rinesi 於 2015 年 8 月 6 日發表的文章“The Telemarketer Singularity”<http://ieet.org/index.php/IEET/more/rinesi20150806>

³關於 GNU 操作系統的歷史可參見《GNU 工程》(p. 8) 一文，以及“GNU/Linux FAQ”<http://gnu.org/gnu/gnu-linux-faq.html>。

如果你使用 SaaS，服務器運營商控制著你的計算過程。它需要委託所有數據到服務器運營商那裡，而這些數據也可能會被迫出讓給國家——畢竟誰是服務器真正服務的人¹？

1.4.3 主要和次要的不公

當你使用專有軟件或者 SaaS，首先你對自己不好，因為這給了別的實體可以用不公的權力凌駕你之上。所以為你著想，你必須遠離非自由的程序。如果你承諾不分享程序，你就在對他人不好。這個承諾本身就是罪惡，不那麼罪惡的是打破承諾；最好的做法就是你應該完全不作出這樣的承諾。

有一些非自由的程序會直接給用戶施壓。Skype 就是一個例子：當用戶使用了一個非自由的 Skype 客戶端軟件，它會要求其他人使用同樣的軟件——這樣兩個人都將自由放棄了（Google Hangouts 也有同樣的問題）。建議使用這樣的軟件也是同樣的錯誤。我們必須堅決拒絕它們，即使是短時間，即使是在別人的電腦上。

另一個使用非自由程序和 SaaS 的害處是會嘉獎肇事者，鼓勵他們開發更多這樣的程序或“服務”，導致更多人掉入公司的陷阱。

所有這些間接傷害發生在公共實體或學校的時候會更加顯著。

1.4.4 自由軟件與國家

公共機構是為人民服務的，而不是為他們自己。在計算機領域也是如此。他們有義務完全控制計算過程以確保恰當地為人民服務（也就是所謂的國家計算主權）。他們決不能讓計算機的控制權落入私人之手。

為了維持對為人民服務的計算過程的控制，公共機構必須不能使用專有軟件（一個由國家之外的實體掌控的軟件），同時不能委託國家機構以外的實體編寫或運行服務，因為這會是 SaaS。

專有軟件在非常時期是沒有安全可言的——因為無力抵抗其開發者。

¹參見《服務器真正是在為誰服務？》(p. 302) 一文。

甚至開發者會幫助其他人攻擊。微軟會在修復 Windows 的 bug 之前將其展示給 NSA¹（美國數字間諜機構）。我們不知道蘋果是不是也這樣做，但他們同樣受到與微軟一樣的政府壓力。如果其他國家政府使用這樣的軟件，會危害國家安全²。你會希望 NSA 攻入你自己國家政府的計算機嗎？

1.4.5 自由軟件與教育

學校（包括所有這類教育活動）通過他們的教學會影響社會的未來。為了做善事他們必須只教自由軟件。教授專有程序會產生依賴性，這與教育的使命是相悖的。通過培訓使用自由軟件，學校會將社會的未來轉向自由，並幫助天才的程序員掌握這門手藝。

他們同時也教育了學生協作的習慣，幫助其他人。每個班級都要有這樣的規則：“對學生而言，這個班級是分享知識的地方，如果你將一個軟件帶來，不僅僅是你自己用，同時你必須將副本分享給班裡其他人——包括源代碼——以備其他人想要學習。因此，不允許將專有軟件帶到課堂，除非學習逆向工程。”

專有軟件的開發者會懲罰那些好心分享軟件的學生，並阻撓學生企圖修改的好奇心。這是很爛的教育³。

1.4.6 自由軟件：不止“優勢”

經常有人讓我描述自由軟件的“優勢”。然而“優勢”這個詞對自由而言太弱了。沒有自由的生活是一種壓迫，無論是我們平時的生活還是計算機領域。我們必須拒絕讓軟件或計算機服務的開發者控制我們的計算機。這才是我們要做的事情，雖然有自私的因素；但不僅僅是自私的考量。

自由包括與其他人協作的自由。拒絕人們的這項自由意味著讓人孤立，

¹參見 Sean Gallagher 於 2013 年 6 月 14 日發表的文章“NSA Gets Early Access to Zero-Day Data from Microsoft, Others”<http://arstechnica.com/security/2013/06/nsa-gets-early-access-to-zero-day-data-from-microsoft-others/>

²關於我們建議的政策可參見《政府推動自由軟件的措施》(p. 41) 一文

³有關自由軟件在學校的討論可參見<http://gnu.org/education>

會成為對人壓迫的開始。在自由軟件社區，我們深刻的意識到與他人協作的重要性因為我們的工作正是有組織的協作。如果一個朋友看到你在用一個程序，她也許會要一份副本。而禁止人們再分發，或者說你“不應該”這麼做的程序，是反社會的。

在計算機領域，協作包括向用戶再分發原始副本，也包括分發你修改過的版本。自由軟件鼓勵所有這些協作，而專有軟件禁止這些。專有軟件禁止再分發副本，並拒絕提供給用戶源代碼，封鎖人們對軟件的修改。SaaS 也有同樣的效果：如果你的計算過程是由互聯網上其他人的服務器，其他人的程序副本做出的，你不能看到或碰觸到這些計算用的軟件，因此你就不能再分發或修改了。

1.4.7 結論

我們應該控制我們自己的計算機，如何贏回控制權？可以拒絕使用我們自己或平時所用計算機上的非自由軟件，拒絕 SaaS。對於我們這些程序員而言，可以開發自由軟件¹，還可以拒絕開發或者推廣非自由軟件或 SaaS，並廣泛散播這些理念給其他人²。

我們以及上千用戶從 1984 年開始就這麼做，這樣才有了現在我們使用的自由的 GNU/Linux 操作系統，所有人——無論是否是程序員——都可以使用。以一個程序員或活動家的身份，加入我們的事業。讓我們一起解放所有計算機用戶吧。

¹關於許可證的建議可參見《如何為你的作品選擇一份許可證》(p. 211)一文中的推薦許可證。

²各種幫助的方式可參見 <http://gnu.org/help>

1.5 為什麼學校應該只使用自由軟件

Copyright © 2003, 2009, 2014 Richard Stallman 此文最早於 2003 年發佈在 <http://gnu.org>。

教育性的活動（包括學校裡的）有道義責任只教授自由軟件。

所有計算機用戶都應該堅持使用自由軟件：這會讓你獲得控制你自己計算機的權利——專有軟件更願意做其擁有者或開發者想要做的事情，而不是用戶想讓它們做的事情。自由軟件還會賦予用戶自由協作的權利，這將會引領一種積極向上的生活方式。這些理由適用於學校，而且也適用於每個人。但是，本文的目的不僅限於此，還會講到專門適用於教育領域的特殊原因。

自由軟件可以幫助學校節省經費只是一個連帶好處。節省經費是因為自由軟件賦予學校，如同賦予其他個人用戶那樣，自由複製和分發軟件的權利。教育系統可以給每一所學校一份副本，而且每一所學校都可以將其安裝在學校的每一臺電腦上，這樣做無需支付任何費用。

儘管這項好處很有用，但我們堅定地拒絕將其置於首位，因為與道義層面性命攸關的重要問題相比，它顯得微不足道。推動學校使用自由軟件，不只是讓教育變得稍微“好一點”的一個方法，也是一件讓好教育替代爛教育的大事。所以，我們來探討一下這個更深層次的問題。

學校有一項社會使命：要教導學生成為一個堅強、有才能，獨立、相互協作並且自由的社會中的一名公民。學校應該像推廣保育和選舉一樣推廣自由軟件的使用。通過教授自由軟件，學生能成為自由的數字化時代的合格公民。這項工作能幫助社會從整體上脫離大集團公司的統治。

相比之下，如果我們教學生使用非自由軟件，就等同於培養依賴性，這將違背學校的社會使命，學校應該極力避免這種事情的發生。

為什麼呢？畢竟一些專有軟件開發者可以免費給學校的提供非自由程序的副本。那是因為，就像菸草公司免費向學生髮放香菸一樣，他們希望

利用學校給學生灌輸這種對他們產品的依賴性¹。

當學生畢業之後，他們就會終止這種免費行為，而且，他們也不會為畢業生就職的公司或機構提供免費服務。你一旦對這些軟件形成了依賴，你很可能就會付費，並且之後的升級可能會很昂貴。

自由軟件可以讓學生了解和掌握這些軟件的工作原理。有些學生，或者那些天才程序員，青少年時代就對計算機和軟件充滿好奇心，他們急切地想要獲知他們想要知道的一切。他們非常渴望閱讀那些他們每天都在使用的軟件的源代碼。

專有軟件阻抑了學生對知識的渴望：他們被告知，“你想要獲得的知識是一個祕密，禁止學習！”專有軟件是教育精神的敵人，所以，除非作為逆向工程的目的，學校應該拒絕使用它們。

自由軟件鼓勵每個人學習。自由軟件社區反對“高高在上的技術”——這會讓大眾對技術的基本原理敬而遠之。我們鼓勵任何年齡層次學生或個人閱讀源代碼，而且，我們希望他們學得越多越好。

使用自由軟件的學校將會成為那些喜愛編程學生的樂土。你知道喜歡編程的學生是怎樣成為優秀的程序員的嗎？他們必須要閱讀和理解人們真實使用的軟件的源代碼。通過閱讀和編寫大量的代碼，你才有可能寫出高質量清晰的計算機程序。目前來看，只有自由軟件允許你這樣做。

如何學習為大型軟件項目編寫代碼？最好的辦法就是為現有的大型項目編寫大量代碼。自由軟件鼓勵你這樣做，但專有軟件會禁止如此。任何學校都可以給學生提供這種掌握編程技術的機會，但只有使用自由軟件的學校才真正有這個可能。

學校使用自由軟件最深層次原因是可以進行德育教育。我們希望學校教給學生基本的知識和有用的技巧，但是這只是學校工作的一部分。學校最根本的一項職能，就是培養好公民——其中包括助人為樂的習慣。在計算機領域，這就意味著我們需要教授學生分享軟件的精神。從幼兒園開始，

¹參見《雷諾菸草公司因向兒童發放捲菸樣品，於 2002 年被罰 1500 萬美元》（RJ Reynolds Tobacco Company was fined \$15m in 2002 for handing out free samples of cigarettes at events attended by children）http://bbc.co.uk/worldservice/sci_tech/features/health/tobaccotrial/usa.htm

學校就應該告訴學生，“如果你把軟件帶到學校來，你應該與其他同學一起分享。一旦有人想要學習一下它的工作原理，你就應該把源代碼展示給大家。所以說，非自由軟件最好不要帶到學校來，除非出於反向工程的目的。”

當然，學校必須履行其諾言：在課堂上只允許自由軟件出現（除非為了逆向工程的目的），在分享軟件的同時，源代碼也應該一併分享，這樣的話，學生們就能複製它們，把它們帶回家，甚至還可以將它們再次分享給其他人。

教授學生使用自由軟件，讓他們參與自由軟件社區，就是一門生動鮮活的公民教育課程。這門課程教授將會把學生培養成為具有公共服務精神的行為模範，而不是壟斷企業的超級大亨。所有學校都應該使用自由軟件。

如果你和學校有一定聯繫——如果你是一名學生、一名教師、一位員工、一名管理者、一位捐贈人，或者這些人的父母——你有義務和責任向學校推廣自由軟件。如果私人請求無法解決這個問題，可將這個問題公開出來提交到社區。這樣就能讓更多的人意識到這件事情的重要性。你可以尋找同盟，讓大家一起發起這項推廣自由軟件的運動。

1.6 政府推動自由軟件的措施

Copyright © 2011–2014 自由軟件基金會，此文於 2011 年首次發表在 <http://gnu.org>。

本文提出了一些推動自由軟件在國內發展的強力政策，並引領國內其他方面轉向軟件自由。

國家的使命是通過整合社會資源為人民自由謀福祉。這個使命在計算機領域的一個方面就是鼓勵用戶使用自由軟件：代表用戶自由的軟件¹。而專有（非自由）軟件則踐踏了人民的自由；這是一個國家應該努力去根除的社會問題。

從國家的計算主權角度來說，國家需要堅持自由軟件的主張（國家控制其計算過程）。所有用戶理所應當控制自己的計算過程，但國家有代表人民對計算過程維持控制的責任。現在大多數政府活動都依賴計算過程，而對這些活動的控制則依賴於對計算過程的控制。而在任務很關鍵的機構裡失去這種控制將會極大危害國家安全。

將國家機構轉向自由軟件同時也會提供額外的好處，比如節省經費和鼓勵本地的軟件支持行業等等。

本文中，“國家實體”指代各級政府，以及公共機構包括學校，公共—私立夥伴關係，主要由國家出資的活動比如特許公立學校，以及由國家控制或由國家特許經營或注資的“私營”企業。

1.6.1 教育領域

教育領域的政策非常重要，因為會影響國家的未來

- 只能教授自由軟件

教育活動或者至少這些國立教育實體，必須只教授自由軟件（因此，他們決不能引導學生使用非自由程序），並教育學生堅持使用自由軟

¹參見《什麼是自由軟件？》(p. 1) 一文了解自由軟件的定義

件背後的公民原因。教授非自由程序就是在教授依賴性，這與學校的使命是相悖的。

1.6.2 國家與公眾

這些政策對個人與組織使用軟件的影響也很巨大：

- 永遠不要求非自由程序

法律和公共部門實踐活動必須改革永遠不能要求或施壓給個人或組織使用非自由程序。他們還必須不能在通信或出版行業引入類似的規定（包括數字限制管理 Digital Restrictions Management¹）。

- 只散播自由軟件

國家實體無論在什麼時候將軟件公開散播，包括網頁中包含或附帶的程序，必須作為自由軟件散播，且必須與只運行自由軟件的平臺兼容。

- 政府網站

政府實體的網站和網絡服務必須設計成用戶能夠只用自由軟件而沒有障礙（相對專有軟件——譯者注）地使用。

- 自由的格式和通信協議

國家實體必須只使用自由軟件廣泛支持的文件格式和通信協議，特別是已經公開的規格（我們並不說“標準”，因為它應該適用於標準和非標準的接口）。例如，他們必須不能散播使用 Flash 或者非自由解碼器播放的音頻或視頻文件，公立圖書館必須不能包含數字限制管理。

- 解除對計算機的限制性許可證

¹參見反 DRM 活動網站 http://defectivebydesign.org/what_is_drm 以及《應避免使用（或慎用）的詞語》一文中有關 DRM 的章節（p. 115）瞭解相關問題。

銷售計算機必須不能要求購買一份專有軟件許可證。必須通過法律來約束銷售方提供給消費者購買計算機時不需要專有軟件以及不需要為此付費的選項。強制付費是另一個錯誤，不能讓我們從專有軟件的不公中分散注意力，應更關注失去自由而導致的損失。雖然如此，強迫用戶為其付費等於是給了特定專有軟件開發者一個額外的不平等優勢——讓他們來決定用戶的自由。國家應該防止這種情況發生。

1.6.3 國家計算主權

很多政策會影響到國家計算主權。國家實體必須維持對它們的計算資源的控制，而不是放棄控制而委之於私人之手。這一點適用於所有計算機，包括智能手機。

- 遷移到自由軟件

國家實體必須遷移到自由軟件，並且不能安裝和繼續使用非自由軟件，除非臨時例外。只有一個機構可以有權授予這些臨時例外，且需要令人信服的理由。這個機構的目標是將這些例外必須減少到零。

- 開發自由的 IT 解決方案

當政府實體需要為開發計算機解決方案付費時，合同必須要求以自由軟件形式交付，並且方案必須設計為可以在百分之百自由軟件環境中運行和開發。所有合同都需要這條，這樣如果開發商一旦不符合需求，作品就不給報酬。

- 為自由軟件選配計算機

當政府實體購買或租借計算機時，必須在其等級中選擇幾乎無需專有軟件即可運行的型號。政府需要為每個等級的計算機維護一個按此標準授過權的型號的列表。對於公開型號和政府採購型號，應該選擇只對政府開放的型號。

- 與廠商協商

國家必須經常與廠商協商市場（對外銷售或/和麵向政府）上適合的硬件產品，在所有相關產品領域，都不能需要專有軟件。

- 與其他國家聯合

本國需邀請其他國家來和廠商一同協商適合的硬件產品。聯合它們獲得更多影響力。

1.6.4 國家計算主權之二

國家計算主權（和安全）包括控制計算機做國家的事情。這需要 (1) 避免“替代軟件的服務”（Service as a Software Substitute）¹——除非這類服務是運行在國家機關的同分支機構，以及 (2) 其他降低國家對計算資源控制的實際行動。因此：

- 國家必須控制其計算機

每一臺國家使用的計算機必須從屬於政府的同一個要使用它們的分支機構或由此機構租借，而這個政府分支機構必須不能把決定誰有權物理訪問計算機，誰能維護（軟件和硬件），或者能夠安裝什麼軟件的權力放棄給外人。如果計算機不可移動，那麼計算機所在的物理空間必須是國家所有（無論是所有者或者承租人）。

1.6.5 對開發的影響

國家政策會影響自由和非自由軟件的開發：

- 鼓勵自由

國家需要鼓勵開發者創立或完善自由軟件並將其公開，比如通過減稅或其他財政補貼的方式。相反，對非自由軟件的開發者則沒有此類財政補貼。

¹參見《服務器真正是在為誰服務?》(p. 302) 一文的相關章節，瞭解 SaaS。

- 不要鼓勵非自由

特別的，專有軟件開發者不應該“捐獻”軟件副本給學校，並根據軟件的售價申請減稅優惠。專有軟件在學校是不合法的。

1.6.6 電子垃圾

自由不應意味著產生更多電子垃圾：

- 可替代性軟件

很多現在計算機設計為無法用自由軟件替代其預裝的軟件。因此唯一解放它們的方式就是丟掉，而這對社會不利。

因此，對於新（即，非二手）計算機或基於計算機的產品，阻礙用戶開發、安裝或阻礙用戶替代任何內部能由製造方升級的預裝軟件的，硬件接口保密或有意設限的，大量銷售、進口或分發的行為都應該是非法的，或者至少課以重稅來勸退。這特別適用於任何需要“越獄”方可在其上安裝不同操作系統的設備，或包含保密接口外設的設備。

1.6.7 技術中立

通過本文中給出的措施，政府可以恢復對其計算過程的控制，並引導國家公民、商業和組織控制自己的計算過程。然而，很多反對意見認為這會違反技術中立的“原則”。

技術中立是指國家不應該在技術選型時強加干涉。無論這是不是有效的原則或是爭議，這僅僅只涉及到技術問題。而這裡提出來的措施解決的是道義、社會和政治問題，因此超出了技術中立的範疇¹。除非他們覺得其政府需要在主權和公民自由上保持“中立”。

¹參見我的文章“Technological Neutrality and Free Software” <http://www.gnu.org/philosophy/technological-neutrality.html>

1.7 為什麼自由軟件需要自由的文檔

Copyright © 1996–2007, 2009 自由軟件基金會。

自由的操作系統最大的不足並不在於軟件——而是缺乏可以收錄其中的優秀自由手冊。我們很多重要的程序還沒有完善的手冊。文檔是任何軟件包的重要組成部分；當一個重要的自由軟件包發佈，卻沒有附帶自由的手冊，這會是一個嚴重的缺陷。而這樣的缺陷現在有很多。

曾幾何時，多年以前，我想學習一下 Perl。並得到了一份自由手冊，但我發現很難讀懂。當我尋問 Perl 用戶有沒有替代品，他們告訴我，有更好的入門手冊，但那些不是自由的。

為什麼會如此？作者為 O'Reilly 協會編寫了優秀的手冊，而 O'Reilly 附加了限制條款出版這些手冊——不能複製、不能修改、不能獲得源文件——這使得它們被排除在自由軟件社區之外。

這類事情絕不是第一次發生，而且（對我們社區是巨大的損失）遠非最後一次。專有手冊出版商誘使許多作者限制他們的手冊。很多次我聽到 GNU 用戶急切地告訴我他正在寫一本手冊，他希望以此來幫助 GNU 計劃，但是隨後我的希望破滅了，因為他接著解釋說，他已經與出版商簽署了一項限制它的合同，所以我們不能使用它。

鑑於編寫好的英文手冊是程序員中難得的技巧，因此我們就更承擔不起因此而失去手冊。

和自由軟件一樣，自由文檔是一個關乎自由的問題，並非價格。這些手冊相關的問題不是因為 O'Reilly 對印本收取了費用——這本身不是問題（自由軟件基金會也銷售自由的 GNU 手冊印刷版¹）。但 GNU 手冊能以源代碼形式獲得，而這些（專有）手冊只能以紙張的形式獲得。GNU 手冊允許複製和修改；Perl 的手冊則不可以，這些限制就是問題所在。

自由手冊的規範與自由軟件的規範大致一樣：它給所有的用戶一定的自由。必須允許重新發布（包括重新商業發佈），因此手冊可以隨著每一份程序，發佈到網絡上或發行印刷版。允許修改手冊也是十分關鍵的。

¹參見 <http://shop.fsf.org/category/books/> 和 <http://gnu.org/doc/doc.html>

作為一般準則，我不認為允許人們修改各種文章和書籍是必要的。寫作問題不一定和軟件問題相同。例如，我不認為你或我有權利去修改像這樣一篇表明我們行為和看法的文章。

不過，自由軟件文檔的修改對自由很重要，這裡有個特殊的原因。當人們行使他們修改軟件的權力的時候，添加或是修改軟件功能時，若有責任心的人，他們也會同時修改軟件的手冊——以便為修改後的軟件提供準確可用的文檔。一本手冊若讓編程人員不能盡責完成他的工作，或更確切地說要求程序員修改程序之後重新寫一份手冊，是不能滿足我們社區的需求的。

儘管全面禁止修改是不可接受的，但一些受限的修改方法還是可以接受的。例如，要求保留原作者的版權聲明，發佈條款或是作者名單，上述這些都是可以的。要告知版本已經被修改，甚至不許刪除或修改整個章節，只要這些章節處理的是一些非技術性話題，像這些要求都沒問題（一些 GNU 手冊就是如此）。

這類限制不是問題所在，因為作為一個實際問題，他們沒有阻擋有責任心的程序員去修改手冊以適應修改過的程序。換句話說，他們沒有阻止自由軟件社區充分利用該手冊。

然而，它必須能修改手冊所有的技術內容，且隨後通過所有常規渠道，在所有常規媒介中發佈結果。否則，這些限制一定會阻礙社區，將手冊變得不自由，而因此我們將會需要另一本手冊。

很不幸的是，在專有手冊存在的情況下往往很難發現有人寫了另一本手冊。問題在於很多用戶認為專有手冊已經足夠好，因此他們沒有意識到撰寫自由手冊的必要性。他們沒有意識到自由操作系統還有缺陷仍需完善。

為什麼用戶認為專有手冊足夠好了呢？因為有些人不考慮這個問題。我希望本文將為改變這種狀況做點貢獻。

其他用戶認為，專有手冊可以被接受和許多人認為專有軟件可以被接受有著同樣的原因：他們純粹是以實際需求來評判，而不將自由作為一項判斷依據。這些人有權表達觀點，但由於這些觀點滋生於不自由的價值觀，所以無法指導我們這些切實重視自由的人。

請將這個問題公之於眾。我們仍然認為需要減少專有手冊的出版量。如果我們傳播專有手冊不夠好的言論，也許下一個願意幫助 GNU 寫文檔的人將會在不是太晚的時候認識到，他必須首先使它自由。

我們也可以鼓勵商業出版商銷售自由、Copyleft 的手冊以代替專有手冊¹。而幫助這項事業的方法是在購買之前檢查手冊的分發條款，儘量購買 copyleft 的手冊，而不是非 copyleft 的手冊。

¹其他出版商的自由圖書可見此列表：<http://gnu.org/doc/other-free-books.html>

1.8 售賣自由軟件

Copyright © 1996–1998, 2001, 2007, 2015 自由軟件基金會。此
文於 1996 年首發於 <http://gnu.org>

很多人認為 GNU 工程的精神就是不該對軟件發行的副本收費，或者應只收取很少的錢——只要是成本價即可。這是一種誤解。

事實上，我們鼓勵重新發布自由軟件¹的人儘可能多的收取他們想要的費用。如果一個許可不允許用戶銷售生成的副本，它就是個非自由許可。如果這讓你感到驚訝，請繼續往下讀。

“free” 這個詞有兩種合理通用的解釋，它可以被解釋為自由亦或是免費價格。當我們談到 “free software” 時，我們所指的是自由，而不是價格。（可以理解為 “言論自由”，而不是 “免費啤酒”）。明確地講，自由軟件的意思就是任何用戶可以自由的運行、修改和免費或收費地重新發布修改過或未修改的程序。

自由的程序有時是免費發佈的，而有時則需要收費。有時同一個程序可以在不同的地方分別以這兩種方式發佈。無論其價格如何，這種程序都是自由的，因為用戶在使用時是自由的。

非自由程序²通常以高價出售，但有時銷售商會贈送你一份免費副本。儘管如此，那也不能使其成為自由軟件。無論收費與否，這個程序都是不自由的，因為用戶沒有自由。

因為自由軟件無關價格，所以低價並不意味著更多自由，或者更加接近自由。因此如果你重新發布自由軟件的副本，你也可以收取基本的費用來獲得收益。重新發布自由軟件是一種很好而且合法的行為；如果你那樣做，你同樣也可能從中獲利。

自由軟件是一項社區工程，每個依賴社區的人都應該盡其可能為社區建設做出貢獻。對一個發佈者來說，這樣做可以使一部分收益流向自由軟件開發項目或者自由軟件基金會。只有這樣才能推動自由軟件在世界的發

¹參見《什麼是自由軟件？》(p. 1) 一文

²也稱為 “專有軟件”，參見《自由與非自由軟件的分類》的相關章節 (p. 87) 一文

展。

發佈自由軟件是籌集資金的一個機會。不要浪費這個機會！

為了捐獻金錢，你需要收取一些額外的費用。如果你收費過低，那就不能為支持自由軟件發展而儲備資金。

1.8.1 過高的發行費用會傷害一些用戶嗎？

有時人們擔心過高的發行費用會使自由軟件超出低收入用戶的預算。對於專有軟件，高昂的費用確實如此——但自由軟件與之不同。

不同之處在於自由軟件天生就易於傳播，可以通過多種方式來獲取。

在你沒有支付標準價格之前，軟件囤積者盡他們該死的最大努力阻止你使用專有軟件卻不按標價付錢。如果這個定價很高，的確會讓一些用戶很難使用這個軟件。

對於自由軟件，用戶不必須為了使用軟件而支付發行費用。可以從擁有一份副本的朋友那裡複製程序，或者請能上網的朋友下載程序。或者很多用戶可以一起分攤費用買一張 CD-ROM，然後輪流安裝軟件。當軟件自由時，高價的 CD-ROM 就不再是主要障礙。

1.8.2 高昂的發行費用不利於自由軟件的使用嗎？

另一個經常擔心的問題是自由軟件的傳播。人們認為高昂的發行費用會減少用戶數量，或者低廉的價格可能會鼓勵用戶去使用。

對於專有軟件確實是這樣——但是自由軟件與之不同。可以通過如此多的方式來獲得軟件副本，因此發行服務的費用對自由軟件的普及影響很小。

從長遠來看，使用自由軟件的用戶數量主要取決於自由軟件可以做什麼，以及自由軟件有多好用。很多用戶並不把自由放在首位；如果自由軟件不能做他們要做的所有工作，許多用戶將繼續使用專有軟件。因此，如果我們想從長遠上來增加自由軟件的用戶數量，我們的首要任務是開發更多的自由軟件。

最直接的方式就是自己寫一些需要的自由軟件¹或手冊²。不過如果你是發行方而不是編寫軟件，你提供幫助的最好方式就是提供資金給其他人來寫自由軟件和手冊。

1.8.3 “售賣軟件” 這個說法也很迷惑

嚴格來說，“售賣”是指通過貨物交易來獲取金錢。銷售一份自由軟件的副本是合法的，而且我們鼓勵這種行為。

然而，當人們想到“售賣軟件”時³，他們通常想象成大多數公司的做法：使軟件專有化，而不是自由。

因此，除非你打算仔細分析甄別，正如這篇文章所言，我們建議最好避免使用“售賣軟件”這個說法，而選用其他說法來代替。例如，你可以說“為發行自由軟件收費”——這樣就不會有歧義了。

1.8.4 收費的高低與 GNU GPL

除非一種特殊情況，GNU 通用公共許可證（GNU GPL）對於你發佈自由軟件的副本時收取多少費用並沒有要求。你可以免費，一美分，一美元，或者十億美元。這完全取決於你和市場，所以如果沒有人願意花十億美元買你的軟件請不要向我們抱怨。

唯一例外的情況是隻發佈二進制（的格式）而沒有（發佈）相對應的完整源代碼。這樣做的人，GNU GPL 會要求其為後期需求提供源代碼。如果對於源代碼沒有價格上的限制⁴，他們可以確定一個沒有人能支付得起的龐大數目——比如十億美元——並且假裝發佈源代碼，但本質是在隱藏源代碼。因此，在這種情況下我們必須限制源碼價格，以確保用戶的自由。一般情況下，我們沒有理由限制發行費用，所以我們不限制他們。

¹參見 Savannah 任務列表 <http://savannah.gnu.org/projects/tasklist>

²參見 <http://gnu.org/doc/doc.html>

³參考《應避免使用（或慎用）的詞語》一文的“販賣軟件（Sell Software）”一節（p. 124）一文了解為何“售賣軟件”這個說法有歧義

⁴參見《GNU 通用公共許可證》一文的第六段（p. 234）

有時公司的活動超出了 GNU GPL 所許可的範圍，他們辯解說“沒有對 GNU 軟件收費”或者其他類似的理由。無論怎麼樣，他們都沒有因此得逞。自由軟件是關於自由的，並且遵循 GPL 就是捍衛自由。當我們維護用戶自由時，我們沒有因為思考需要收取多少發佈費用而心煩意亂。自由才是問題所在，是全部的問題，也是唯一的問題。

1.9 自由硬件和自由硬件設計

Copyright © 2015 Richard Stallman。本文分成兩部分，分別以標題“為什麼我們需要自由的電子硬件設計”<http://wired.com/2015/03/need-free-digital-hardware-designs>和“硬件設計必須自由，如何行動見此”<http://wired.com/2015/03/richard-stallman-how-to-make-hardware-designs-free>，發表於《連線》雜誌網站。2015 年本文發佈於 <http://gnu.org>。

自由軟件的思想向硬件延伸到什麼範圍？讓我們的硬件設計自由是不是個道德義務，正如使我們的軟件自由那樣？維持我們的自由是否需要抵制出自非自由設計的硬件？

1.9.1 定義

自由軟件事關自由，而非價格；寬泛地講，這意味著用戶有使用軟件的自由和複製並再分發軟件的自由，免費或收費。更精確地，其定義可用這四項基本自由¹公式化的來考察。為強調“free”一詞指的是自由而非價格，我們經常把法語和西班牙語中的“libre”一詞和“free”一起使用。

將同樣的概念直接應用於硬件，自由硬件意為用戶有使用、複製和免費或收費地再分發硬件的自由。然而，除了鑰匙、DNA、和塑料製品的外觀之外，沒有用於硬件的複製機制。大多數硬件是按照一定的設計裝配出來。設計先於硬件存在。

因此，我們實際需要的概念是類似自由硬件設計的東西。很簡單：其意指一種允許用戶使用（即，在其指導下裝配硬件）和複製並免費或收費地再分發的設計方案。這種设计方案必須提供和自由軟件一致的四項基本自由。

這樣我們可以將按照自由的設計裝配出的硬件說成“自由硬件”，或“設計自由的硬件”以避免可能的誤解。

¹參見《什麼是自由軟件？》(p. 1) 以查看四項基本自由的列表。

首次接觸自由軟件思想的人們經常認為其含義是你可以免費獲得一份副本。很多自由程序可以不要錢地獲取，因為下載你自己的副本無需代價，但那不是“free”在此處的含義（事實上，有些間諜軟件程序——例如 Flash 播放器和《憤怒的小鳥》是免費的，儘管它們不自由）。將“libre”一詞和“free”一起使用有助於澄清觀點¹。

對硬件，混淆趨於走向另一方向；硬件要花錢生產，故商業化生產硬件無法免費（除非是賠本賺吆喝或添頭），但這並不妨礙其設計方案變得自由。由你的 3D 打印機生產出的東西可以非常便宜，但並非精確地免費因為你需要支付原料成本。在道德方面，自由的問題完全壓倒了價格問題，因為對其用戶拒絕自由的設備還不如沒有。

由某些人使用的“開放硬件”和“開放源碼硬件”的說法具有和“自由硬件”相同的具體含義，但這些說法降低了對自由議題的重視。它們由“開放源碼軟件”的說法派生而來，其多少指的是自由軟件，但缺少了對自由的討論或將自由的議題展示為關乎對錯的事²。為著重強調自由的重要性，一旦關乎自由我們就要注意談到自由；因為“開放”做不到這一點，讓我們不要用它來取代“自由”一詞。

1.9.2 硬件和軟件

硬件和軟件在根本上不同。一個程序，即使取編譯後的可執行形式，也是一組可被計算機解釋為指令的數據集合。和任何其他數字作品相同，它可以用計算機來複製和改寫。程序的副本沒有固有的物理形式或化身。

與之相對，硬件是一個物理構造且其物質性是關鍵。雖然硬件的設計可以表示為數據，甚至在某些場合下表示為程序，設計仍然不是硬件本身。為 CPU 做的設計不能執行程序。你基本上不可能試著用為鍵盤做的設計打字或在為顯示屏做的設計上顯示像素。

更進一步，雖然你可以用計算機來修改或複製一份硬件設計，計算機

¹參見 <http://gnu.org/philosophy/proprietary/proprietary-surveillance.html> 以查看一個在工業中擴散的監視手段的快速增長的列表。

²參見《為什麼說開源漏掉了自由軟件的要點》(p. 90) 一文

並不能將這份設計轉化為其描述的物理結構。那需要裝配設備。

1.9.3 硬件和軟件的邊界

在數字設備中，硬件和軟件的邊界是什麼？它遵從其定義。軟件是計算機可以操作修改和複製的部分，而硬件則不可以。這是理清其區分的正確方法，因為關乎實際的後果。

在硬件和軟件之間有一個包含可被更新或替換，卻並未打算在產品售出後更新或替換的灰色地帶——固件（Firmware）。從概念上說，這個灰色地帶十分狹窄。在實踐上，它很重要，因為很多產品落在其中。我們可以把這樣的固件看作硬件的小小延伸。

有人會說原裝的固件程序和場效應可編程邏輯門陣列（FPGA）“模糊了硬件和軟件的邊界”，但我認為這是對事實的誤讀。在使用過程中安裝的固件是軟件；設備自帶並不可修改的固件雖天生是軟件，但我們可以把它當成一種電路。如對 FPGA，FPGA 本身是硬件，但加載到 FPGA 中的邏輯門模式是一種固件。

在 FPGA 上運行自由的邏輯門模式可以是一種使得數字設備從電路層變自由的潛在有效方法。然而為了使 FPGA 在自由世界中可用，我們需要為其設計自由的開發工具。其障礙是加載進 FPGA 的邏輯門模式文件的格式是私密的。多年以來，沒有一種 FPGA 的型號能夠讓我們不使用不自由（私有）的工具就能夠為其生成邏輯門模式文件。

到了 2015 年，用來給一種通用的 FPGA 型號——Lattice iCE40¹用硬件描述語言（HDL）寫成的輸入文件編程的自由軟件工具出現了。現在也可以用自由工具編譯 C 程序並在 Xilinx Spartan 6 LX9 FPGA 上運行它們²，但這些工具不支持 HDL 輸入。我們建議您抵制其他型號的 FPGA 直到它們也能被自由的工具支持。

對 HDL 代碼本身而言，它可以表現為軟件（當運行於模擬器上或載入 FPGA 中），也可以表現為硬件設計（當實現為不可變的硅晶或電路板）。

¹參見 <http://clifford.at/icestorm/>.

²參見 <https://github.com/Wolfgang-Spraul/fpgatools>

1.9.4 3D 打印機的道德問題

道德上，軟件必須自由¹；不自由的程序是不義的。我們是否應當對硬件設計取同樣的觀點？

我們肯定應該，在 3D 打印（或者，更普遍地，任何個人裝配行為）可處理的領域。用於製造有用的、實用的物品（即，功能性而非裝飾性）的打印模式必須是自由的，因為它們是為了實用而產生的作品。用戶應得到對這些作品的控制，正如他們應得對他們使用的軟件的控制。發佈一個不自由的功能性物品的設計和發佈非自由程序一樣有錯。

請仔細甄別，並選擇那些只用自由軟件就能驅動的 3D 打印機；自由軟件基金會為這樣的打印機背書²。有的 3D 打印機是按照自由的硬件設計製成的，但 MakerBot 的硬件設計是不自由的³。

1.9.5 我們是否必須抵制不自由的數字硬件？

不自由的數字硬件⁴設計是否不義？我們是否必須為了我們的自由抵制所有按照不自由的設計製造出的數字硬件，如同我們必須抵制非自由軟件？

因為硬件設計和軟件源碼在概念上是平行的，許多硬件黑客像對非自由軟件那樣迅速聲討非自由硬件設計。我對此並不同意，因為硬件和軟件的情況不同。

今天裝配芯片和電路板的技術類似印刷業：它適於工廠中的大規模生產。比起今天的軟件複製，它更類似 20 世紀 50 年代的書籍複印。

複製和修改軟件的自由在道德上是必要的，因為那些活動對使用軟件的人們來說是可行的：你能使用軟件的設備（計算機）同樣適合用來複制

¹參見《如今自由軟件更加重要》(p. 32) 一文

²參見 <http://fsf.org/resources/hw/endorsement>

³Rich Brown, “Pulling Back from Open Source Hardware, MakerBot Angers Some Adherents,” 27 September 2012, <http://cnet.com/news/pulling-back-from-open-source-hardware-makerbot-angers-some-adherents/>.

⁴如本文中的用法，“數字硬件”包括在數字部分之外還有模擬部分的硬件。

和改寫它。今天的移動計算機太弱小而難以適用於此，但任何人都能找到足夠有力的計算機。

而且，計算機足以用來下載並運行那些懂得如何修改軟件的人修改的版本，即使你不是程序員。當然，非程序員每天都下載並運行軟件。這就是為什麼自由軟件對非程序員有實在的貢獻。

這些在多大程度上適用於硬件？並非使用數字硬件的每個人都知道如何修改電路設計，或芯片設計，但任何有 PC 的人都有這麼做的必要設備。到目前為止，硬件平行於軟件，但接下來區別就大了。

你不能在你的計算機上運行電路設計或芯片設計。構造大型電路是一系列艱苦的工作，之後你才能得到電路板。裝配芯片在今天並不適合個人完成；只有規模化生產才能讓它們足夠便宜。以今天的硬件技術，用戶們無法像運行軟件黑客某甲（原文是 John S Hacker——譯者注）改造過的軟件那樣，下載並運行硬件黑客某乙（原文是 John H Hacker——譯者注）改造過的硬件設計。因此，四項基本自由無法像給與用戶們對程序的集體控制那樣給與對硬件設計的集體控制。這就是一切軟件都必須自由的原因無法適用於硬件的地方。

1983 年沒有自由的操作系統，但顯然如果有了，我們可以立刻用上它並擁有軟件自由。缺少的只是這樣一個操作系統的代碼。

而到了 2014 年，就算我們有了適用於 PC 的 CPU 芯片的自由設計，按此設計大規模生產出來的芯片也不能為我們在硬件界給與我們同樣的自由。如果我們要購買從工廠大規模生產出來的產品，對工廠的依賴會引發大多數和非自由設計相同的問題。我們需要未來更好的裝配技術，這樣才能讓我們自由的設計得到硬件的自由。

可以展望這樣的未來，我們的個人生產者能製造芯片，機器人可以把芯片和變壓器、開關、按鍵、屏幕、風扇等組合焊接在一起。在這樣的未來裡，我們都可以製造自己的計算機（還有裝配機和機器人），而且我們每個人都能從懂硬件的人做出的改造設計中得到好處。那時抵制非自由軟件的議題才也能適用於硬件設計。

這個未來距今至少有數年之久。在此期間，原則上沒有必要抵制非自

由設計的硬件。

1.9.6 我們需要自由的硬件設計

儘管在今天的狀況下我們無需抵制按照非自由設計製造出的數字硬件，不過我們仍需要開發自由的設計並在條件允許時使用它們。這些已經在今天提供了優勢，而且在未來它們可能是使用自由軟件的唯一方法。

自由的硬件設計提供了實用上的優勢。很多公司可以裝配，這減少了对單一賣家的依賴。集團可以安排大量裝配。擁有電路圖或 HDL 代碼使得學習設計以尋找錯誤或惡意功能成為可能（NSA 在一些計算機中促成惡意弱點之事已被人所知）。並且，自由設計可作為設計計算機和其他複雜設備的構造磚塊，其規格會被公開，對我們不利的部件也會更少。

在我們可以這樣製造整臺計算機之前，自由硬件設計可以用於計算機和網絡的某些部件，還有嵌入式系統。

如果它成為阻止非自由軟件的唯一方法，自由硬件設計甚至可能會在我們能以個人之力裝配硬件以前就變得至關重要。因為一般的商業硬件正越發設計得使用戶屈服，由於私密的規格，甚至需要你以外的某人簽名代碼，它們越發不兼容於自由軟件。甚至蜂窩電話的調制解調器芯片和一些圖形加速器已經需要製造商為固件簽名。任何在你計算機上允許別人改但不許你改的程序，都是強加於你不義強權的體現；強加了那種要求的硬件是惡意硬件。所以蜂窩電話調制解調器芯片，現在可用的所有型號都是惡意的。

某一天，自由設計的數字硬件可能是唯一可以讓自由系統運行的平臺。讓我們專注於在那一天之前擁有必要的自由硬件設計，並希望我們有方法為所有用戶將它們足夠便宜地裝配出來。

如果你設計硬件，請讓你的設計變得自由。如果你使用硬件，請催促公司們並向其施壓，讓他們使硬件設計變得自由。

1.9.7 設計的層次

軟件有實現的層次；例如，一個軟件包可能包括庫、命令和腳本。但這些層次對軟件的自由沒有明顯的貢獻，因為可以把所有這些層次都變得自由。設計程序組件和設計把組件組合在一起的代碼是同類的工作；與其類似，從源代碼編譯程序組件和編譯把這些組件組合在一起的程序是同類的操作。使整個系統自由只需繼續工作直到完成整件任務。

因此，我們堅持主張程序要在全部層次上自由。對一個分類為自由軟件的程序，組成它的每一行源代碼都得是自由的，這樣你才能只用自由的代碼重建這個程序。

與此相反，物理對象經常是由不同類型的工廠設計製造的組件所構造而成的。例如，計算機由芯片構成，但設計（或裝配）芯片和從芯片設計（或裝配）計算機是非常不同的。

因此，我們需要區分數字產品（可能還有其他類型的產品）的設計中的層次。連接芯片的電路是一個層次；每個芯片的設計是另一層次。在 FPGA 中，基本單元間的互聯是一個層次，而基本單元本身是另一層次。理想的未來中我們會希望設計在所有層次上都是自由的。在當前狀況下，僅僅使一個層次變得自由已經是一個明顯的進步。

然而，如果一個設計在同一層次上組合了自由和非自由的部分——例如，一個結合了私有“軟核”的“自由”HDL 電路——我們必須下結論說此設計作為一個整體在那個層次上是不自由的。類似地，對非自由的“嚮導”或“宏”，如果它們指的是芯片間的互聯或芯片內部結構間的可編程互聯的一部分。其自由部分可以是通向自由設計未來目標的一個步驟，但到達那一目標需要把不自由的部分替換掉。它們在自由的世界中決不能接受。

1.9.8 自由硬件設計的許可和版權

你通過將一個硬件設計按照自由的許可發行來使其成為自由設計。我們建議您使用第三版或更新的 GNU 通用公共許可證。我們帶著這種用途的視點設計了 GPL 第三版。

在電路和非裝飾性的物體外觀上的左版 (Copyleft)，不會像想象一般獲得成功。這些設計的版權僅適用於這些設計是如何被畫出或寫下的。左版是一種利用版權法的手段，所以其效果僅限於版權法起效的範圍。

例如，一個電路，作為一種拓撲關係，無法擁有版權（也因此無法使用左版）。由 HDL 寫成的電路定義可以擁有版權（因此可以使用左版），但左版只能保護這 HDL 代碼的具體表達方式，而非依其生成的電路拓撲。與此類似，電路的圖紙或佈局可以擁有版權，因此可以使用左版，但僅僅保護圖紙或佈局本身，而非電路拓撲。任何人都可以以不同的外觀畫出同樣的電路拓撲，或寫出生成相同電路的不同 HDL 定義。

版權不能保護物理電路，所以當人們構建電路的實例時，設計的許可證對他們能構建出的設備做什麼沒有法律效力。

對物件的圖紙而言和 3D 打印模型而言，版權不能阻止以不同的方式繪製同樣功能性物體的外觀，它也不能保護按照圖紙製造功能性物質的實體。只要版權考慮到，每個人都可以自由地製造和使用它們（而這是我們十分需要的自由）。在美國，版權不涉及某個設計描述的功能特性¹，但卻涉及裝飾特性。當一個物體既有裝飾特性又有功能特性時，你將進入一個棘手的境地²。這一切也許在你的國家也是如此，也許不是。在商業性或大規模生產之前，你需要諮詢當地的律師。你需要考慮的不只是版權。你可能會踩到專利地雷，其大部分被跟你正在使用的設計的誕生毫無關係的實體持有，而還可能有其他的法律問題。

切記版權法和專利法完全不同。假設他們有任何相同之處都是個錯誤。這就是“知識產權”的說法是完全的誤導而應被徹底抵制的原因³。

¹參見美國版權局定義的“useful article”，於 <http://copyright.gov/register/useful.html>。

²一篇由 Public Knowledge 所寫的文章“為你的 3D 打印產物賦予許可的三個步驟” https://publicknowledge.org/assets/uploads/documents/3_Steps_for_Licensing_Your_3D_Printed_Stuff.pdf（2015 年 3 月 6 日）。對其在美國的複雜性給出了很有用的信息，儘管該文落入了使用偽概念“知識產權”，和不應該和版權聯用的宣傳用語“保護”的常見謬誤。若問為何，參見《應避免使用（或慎用）的詞語，由於它們是不公正的或者引起混淆的》一文的“保護（Protection）”一節（p. 122）。

³參見《您說過“知識產權”嗎？這是一種迷惑的幻景》（p. 99）一文。

1.9.9 通過文件倉庫提倡自由硬件

通過人們發表硬件設計的文件倉庫的規則來勸說他們把已發表的設計變得自由是最有效的方式。倉庫的操作員應當把將要使用設計的人們的自由置於設計者的偏好之上。這意味著要求有用物品的自由設計是在該倉庫中發表它們的一個條件。

對裝飾性物品，此論點不適用，所以我們不必堅持它們必須是自由的。但是，我們必須堅持它們是可分享的。因此，一個既能處理裝飾性物品模型也能處理功能性物品模型的倉庫必對每一類存檔都有合適的許可政策。

對數字設計，我建議倉庫堅持使用“GNU GPL 第三版或更新”、Apache 2.0 或 CC-0。對功能性三維設計，倉庫要勸用戶選擇以下四種許可之一：“GNU GPL 第三版或更新”、Apache 2.0、CC-SA、CC-BY 或 CC-0。對裝飾性設計，要選“GNU GPL 第三版或更新”、Apache 2.0、CC-0 或任何 CC 族許可。

該倉庫應當要求所有設計都以源代碼形式發表，且只能被私有軟件使用的私密格式源代碼並不真正足夠。對三維模型而言，STL 格式並不是用來修改設計的最佳格式因此不算源代碼，所以倉庫不該接受此格式，除非它們可能和真正的源代碼一同發表。

沒有理由為硬件設計的源代碼只選擇單一的格式，但尚未能被自由軟件處理的源碼格式充其量只應該勉強接受。

1.9.10 自由硬件和擔保

一般說來，自由硬件設計的作者沒有對按其設計裝配硬件者提供擔保的道德義務。這和出於銷售硬件實體的問題不同，銷售硬件實體必須由其賣家和/或製造者提供擔保。

1.9.11 結論

我們已經有了可以讓我們的硬件設計自由的合適許可證。我們需要的是作為一個社區將其認作我們所必做之事而在我們自行裝配物品時堅持自由的設計。

1.10 應用自由軟件判斷準則

Copyright © 2015 理查德·斯托曼 (Richard Stallman)

四項基本自由為判斷某一特定代碼片斷是否為自由的（即尊重用戶自由）提供了準則¹。我們應當如何將它們應用於判斷一個軟件包、一套操作系統、一臺計算機、或是一個網頁是否適合被推薦使用呢？

一個程序是否是自由的首先影響到的是我們對於自己的私人行為的決定：為了捍衛我們自己的自由，我們需要拒絕那些將會剝奪我們自由的程序。然而，這也會影響到我們應當對別人怎樣說和怎樣做。

非自由程序是不公正的。發佈一個非自由程序、向他人推薦非自由程序、或是更為普遍地將它們引入課程而引導人們使用非自由軟件，以上這些行為意味著引導人們放棄自己的自由。可以肯定的是，引導人們使用非自由軟件並不同於在他們的計算機上安裝非自由軟件，但我們不應該將人們引入歧途。

在更深層次上，我們不能提出一個非自由程序作為一個解決方案，因為這將會承認其合法性。非自由軟件是一種問題；將其以一種解決方案的方式呈現否認了這一問題的存在²。

本文闡述了我們應當如何應用自由軟件的基本準則來判斷不同種類的事物，並且決定是否應該推薦它們。

1.10.1 軟件包

一個軟件包若要成為自由的，其中所有代碼必須都是自由的。但不僅限於代碼。由於文檔文件包含手冊、自述（README）、更新日誌等，這些都是軟件包的必要的技術組成部分，它們必須也是自由的³。一個軟件包通常與很多其他軟件包一起使用，並且與其中的一些進行交互。與非自由軟件進行的何種交互才是倫理上可接受的呢？

¹參見《什麼是自由軟件？》(p. 1) 一文以獲知自由軟件的完整定義。

²我的文章《避免破壞性的妥協》(p. 311) 詳細論述了這一問題。

³參見《為什麼自由軟件需要自由的文檔》(p. 46) 以獲知關於這一問題的更多細節。

我們著手開發 GNU 的目的是帶來一款自由的操作系統，由於在 1983 年還沒有這樣的自由操作系統。當我們於 20 世紀 80 年代開發出最早的 GNU 組件時，其中每個組件都依賴於非自由軟件是不可避免的。例如，沒有任何一個 C 程序可以離開非自由的 C 編譯器而運行，直到 GCC 可以正常工作，並且它們也不能離開 Unix libc 而運行，直到 glibc 可以正常工作。每個組件都只能運行在非自由操作系統上，因為當時所有的操作系統都是非自由的。

當我們發佈一款可以運行在某些非自由操作系統上的組件後，用戶就把它移植到其他的非自由操作系統上。從倫理上講，這些移植並不比我們曾經用於開發這些組件的限定平臺代碼更壞，因此我們整合了用戶的這些補丁程序。

當 Linux 內核於 1992 年變為自由之時，它填補了 GNU 操作系統的最後一塊空缺（Linux 最初於 1991 年以一種非自由許可證發佈）。GNU 和 Linux 的組合成為了一種完全自由的操作系統——GNU/Linux¹。

此時，我們可以選擇移除對非自由平臺的支持，但是我們決定不這樣做。非自由操作系統是不公的，但用戶運行非自由操作系統並不是我們的過錯。支持該非自由操作系統上的自由軟件並不會進一步惡化這種不公。並且這將是實用的，不僅對於那些非自由操作系統用戶，也對於吸引更多人為開發該自由軟件做貢獻。

然而，在自由程序上運行非自由程序是一個完全不同的問題，因為這是在誘導用戶在自由之路上倒退。在某些情況下我們完全禁止這樣做：例如 GCC 禁止任何非自由插件²。當一個程序允許非自由擴展的時候，它至少不應該引導用戶使用它們。例如，我們更傾向於選擇 LibreOffice 而非 OpenOffice，由於後者提示用戶使用非自由擴展，而 LibreOffice 則避開了它們。我們開發冰貓（IceCat）³起初也是為了避免向用戶推廣由火狐（Firefox）建議使用的非自由擴展。

¹參見《Linux 和 GNU 操作系統》(p. 75) 一文以獲知更多信息。

²如需獲知為何 GCC 拒絕任何非自由插件，參見我在 GCC 郵件列表中的回覆，它位於 <https://gcc.gnu.org/ml/gcc/2014-01/msg00247.html>。

³參見 <http://directory.fsf.org/wiki/IceCat>。

事實上，如果冰貓解釋如何在 macOS 上運行冰貓，這將不會引導用戶去運行 macOS。但如果它介紹了一些非自由擴展，它將會鼓勵冰貓用戶安裝這些非自由擴展。因此，冰貓軟件包及其手冊和網站不應該介紹這些東西。

有時一個自由軟件和一個非自由軟件協同工作，但其中任何一方都不是另一方的基礎。我們針對這種情況的規則是，如果該非自由軟件非常有名，我們應當告知人們如何使用我們的自由軟件與之工作；但如果該專有軟件鮮為人知，我們不應該暗示其存在。有時我們會在該非自由軟件存在的情況下提供互操作支持，但避免告知用戶這麼做的可能性。

我們拒絕任何僅可用於某一非自由操作系統的“增強組件”。它們會鼓勵人們使用該非自由操作系統而非 GNU，如同自擺烏龍。

1.10.2 GNU/Linux 發行版

隨著 Linux 內核於 1992 年自由化，人們開始開發 GNU/Linux 發行版 (distros)。但只有少數發行版是完全由自由軟件構成的。

適用於軟件包的規則也適用於發行版：一個符合倫理的發行版必須僅包含自由軟件並且只將用戶向自由軟件的方向引導。但是，對於一個發行版而言，“包含”一個特定的軟件包是什麼意思呢？

某些發行版從二進制包安裝作為其發行版一部分的軟件；而其他發行版從上游源代碼構建每個軟件，並且從字面意義上講，它們所包含的只是需要下載並構建的列表。對於自由的問題，發行版怎樣安裝一個給定的軟件包並不重要；如果它將某個軟件包作為可選項提供，或者它的網站這樣做，我們稱該發行版“包含”該軟件包。

自由操作系統的用戶擁有對它的控制權，於是他們可以安裝他們想要安裝的任何東西。自由發行版提供了用戶可用於安裝他們自己的程序以及他們對於自由軟件的修改版本的通用工具；他們也可以安裝非自由軟件。在這些發行版中提供這些通用工具並不是倫理瑕疵，由於該發行版的開發者對於用戶基於其自身的主動權獲取並安裝什麼軟件並不負有責任。

然而，當開發者引導用戶走向非自由軟件的時候，他們對於用戶安裝

非自由軟件就應當負有責任了——例如，將非自由軟件置於流行發行版的軟件包列表中，或者從它們的服務器進行分發，或者將其呈現為一種解決方案而非一種問題。這正是為何大多數 GNU/Linux 發行版具有倫理瑕疵的問題之所在。

自行安裝軟件包的用戶通常具有一定的判斷能力：如果我們告訴他們“Baby 包含非自由代碼，而 Gbaby 是自由的”，我們可以預見他們能夠留心記住哪個是哪個。但發行版是推薦給那些不瞭解這些細節的普通用戶的。他們將會想“我應該使用他們所說的哪個呢？我想它應該是 Baby”。

因此，要想向公眾推薦一款發行版，我們堅持要求它們的名字不能與我們所拒絕的某個發行版相似，唯有如此，我們關於僅僅推薦自由發行版的信息才能被可靠地傳達。

發行版和軟件包之間的另一個區別在於向其中添加非自由代碼的可能性。程序開發者會仔細檢查他們向其中添加的代碼。如果他們決定使該程序成為自由的，他們不太會向其中添加非自由代碼。不過也有例外，包括 Linux 內核中添加“二進制 blobs”這樣惡劣的案例，但它們與現存的自由軟件相比只佔一小部分。

與之相反，一個 GNU/Linux 發行版通常包含數千個軟件包，並且其開發者可能每年都會向其中添加數百個新的軟件包。如果未能盡力避免那些包含某種非自由軟件的軟件包，幾乎肯定會有一些非自由軟件混入其中。由於自由發行版的數量很少，作為列出那些發行版的條件，我們要求每位自由發行版的開發者通過移除任何非自由代碼或惡意代碼來承諾保持該發行版成為自由軟件。參見《GNU 自由操作系統發行版指南》<http://gnu.org/distros/free-system-distribution-guidelines.html>。

我們不要求自由軟件包的開發者也做出這樣的承諾，這是不現實的，幸運地是，這也不是必需的。得到超過 30000 個自由軟件的開發者的承諾也許能夠避免少數問題，但其代價是極大增加自由軟件基金會（FSF）員工的工作量；此外，這些開發者中的大部分與 GNU 計劃並無關係，他們也不願意向我們做出任何承諾。因此我們只需在發現問題的時候應對這些使自由軟件變為非自由軟件的少數案例。

1.10.3 外設

計算機外設要求計算機中的軟件被操作系統加載到其中以便使其工作——這些軟件可以是驅動程序或固件。因此，如果一件外設可以在一臺未安裝任何非自由軟件的計算機上使用——如果該外設的驅動程序以及任何需要由操作系統加載到其中的固件都是自由的，那麼它是可接受並使用以及推薦的。

驗證這一點是簡單的：將該外設連接到一臺運行完全自由的 GNU/Linux 發行版的計算機上並且觀察它是否正常工作。但是大多數用戶需要在購買外設之前獲知這一點，因此我們在 h-node.org 列出了很多外設的信息，這是一個完全自由的操作系統的硬件數據庫。

1.10.4 計算機

一臺計算機在不同層次上包含不同的軟件。我們應當基於什麼準則來判斷一臺計算機是否“尊重您的自由”呢？

顯而易見的是：操作系統和其中的任何軟件都必須是自由的。在 20 世紀 90 年代，啟動加載軟件（當時是“基本輸入/輸出系統”，即 BIOS）成為可替換的，並且由於它運行在中央處理器（CPU）上，它與操作系統所存在的是同一類的問題。因此，諸如固件或驅動程序，不論安裝在操作系統中，或是隨操作系統一起安裝，或是啟動加載程序都必須是自由的。

如果一臺計算機擁有某些要求在操作系統中安裝的非自由驅動程序或固件的硬件功能，我們可能仍然能夠推薦它。如果它在沒有那些功能的情況下仍然可用，並且我們認為大部分人不會為了使該功能可用而被引導安裝非自由軟件，那麼我們仍然能夠推薦它們。否則我們就不能。這將是一種主觀判斷。

一臺計算機可能在較低的層次上帶有預裝的可修改固件和微碼。它也可能在真正只讀的存儲器中擁有代碼。我們決定在現今我們所使用的認證準則中忽略這些程序，這是由於如若不然就沒有任何計算機可以滿足，並且因為通常不會被更改的固件在倫理上與電路相同。因此我們的認證準則

僅僅覆蓋那些運行在計算機的主處理器上且儲存在非真正只讀存儲器中的代碼。當在其他層次上運行自由軟件成為可能，我們也會要求這些層次上的軟件是自由的。

由於認證一款產品是對它的積極推廣，我們要求它們的販賣者以支持我們作為回報，這可以通過談論自由軟件而非開源軟件¹以及將 GNU 和 Linux 的結合體稱為 GNU/Linux²來做到。我們沒有義務積極支持那些不認可我們的工作或是不支持我們運動的項目。

參見 <http://www.fsf.org/resources/hw/endorsement/criteria> 以獲知我們的認證準則。

1.10.5 網頁

現在的很多網頁都包含複雜的 JavaScript 程序並且需要它們才能工作。這是一種有害的實踐，因為它阻礙用戶對他們自己計算的控制。更壞的是，這些程序中的大部分是非自由的，這是一種不公。JavaScript 代碼常常窺探用戶³。JavaScript 已經變成了一種對用戶自由的威脅。

為了解決這一問題，我們開發了 LibreJS，這是一種用於阻止非普通非自由的 JavaScript 代碼的火狐瀏覽器擴展（沒有必要阻止簡單的腳本，如果它們只是實現一些次要的用戶界面特性）。我們請求網站將它們的 JavaScript 程序自由化並且標記其許可證以便 LibreJS 識別。

與此同時，鏈接至一個包含非自由 JavaScript 程序的網頁是否符合倫理呢？如果我們堅決不做任何妥協，我們將只能鏈接至自由的 JavaScript 代碼。然而，很多網頁即使不運行它們的 JavaScript 代碼也能工作。此外，除了我們的鏈接，您會經常在其它網站遇到非自由的 JavaScript 代碼；為了避免這些情況，您必須使用 LibreJS 或禁用 JavaScript。因此，我們決定做出讓步並且鏈接那些不運行非自由 JavaScript 程序也能工作的網頁，同時鼓勵用戶在普遍意義上保護自己不受來自非自由 JavaScript 程序的威脅。

¹參見《如今自由軟件更加重要》(p. 32) 和《為什麼說開源漏掉了自由軟件的要點》(p. 90)。

²參見《名字的含義？》(p. 71) 一文

³參見《JavaScript 陷阱》(p. 286) 一文

然而，如果某個網頁不運行非自由 JavaScript 程序就不能實現其功能，鏈接到它將會不可避免地要求人們運行該非自由 JavaScript 代碼。出於原則，我們不會鏈接這些網頁。

1.10.6 結論

將“軟件應當是自由的”這一基本理念應用到不同場合要求不同的實踐策略。隨著新情況的出現，GNU 計劃和 FSF 將會適配我們的自由準則，不論在實踐中還是理論上，都將計算機用戶引向自由。通過僅僅推薦尊重用戶自由的程序、發行版和硬件產品，並且宣示您的立場，您可以為自由軟件運動提供它所急需的支持。

第 2 部分

名字的含義

2.1 名字的含義？

Copyright © 2000, 2006, 2007 Richard Stallman.

關於此主題了解更多，你也可以閱讀我們的 GNU/Linux FAQ <http://gnu.org/gnu/gnu-linux-faq.html> 和本書的《Linux 和 GNU 操作系統》(p. 75) 一文，它講了 GNU/Linux 系統的歷史，和這命名的問題有關。以及另一篇文章《從未聽過 GNU 的 GNU 用戶》<http://gnu.org/gnu/gnu-users-never-heard-of-gnu.html>。

顧名則思義；我們所選的名字代表了我們所要傳達的思想。一個不恰當的名字會向人們傳遞錯誤的思想。玫瑰不管叫什麼聞起來總是甜的——但是如果你把它叫作筆，當人們用它寫字時將會很失望。同樣地，如果你把筆叫作“玫瑰”，人們可能就不知道它能做什麼了。如果你把我們的操作系統稱為“Linux”，則會傳達一些關於此系統起源、歷史和目的的錯誤概念。如果你稱之為 GNU/Linux，這將傳達（雖然不詳細）一個準確的概念。

這對我們的社區重要嗎？人們是否瞭解系統的起源、歷史和目的的重要嗎？是的——因為忘記歷史的人往往註定要重蹈覆轍。目前圍繞 GNU/Linux 建立起來的自由世界仍然無法保證生存；那些迫使我們開發 GNU 的問題並沒有被徹底根除，而且他們總是想復辟。

當我解釋為什麼把這個操作系統稱為“GNU/Linux”比“Linux”要好時，人們有時這樣迴應：

就算 GNU 工程因其工作而應該得到讚譽，但如果人們不給予它這種讚譽，就真的值得如此大驚小怪的嗎？不管是誰做的，反正工作被完成了，這不是重要的事嗎？你應該心態平和，以幹好工作為榮，而不是對榮譽的問題耿耿於懷。

這似乎是個明智的建議，如果所有的情況都像那樣，大家幹完工作後都不計得失，那這建議還真是對的！但現實充滿挑戰，現在還不是那樣對未來想當然的時候。我們社區的力量靠的是保證自由與合作。使用 GNU/Linux 這個名字是一種讓人們提醒自己和他人瞭解這些目標的方式。

雖然不考慮 GNU 也能寫出優秀的自由軟件；許多優秀的工作是在 Linux 的名義下完成的。但是“Linux”（這個名字）從它被創造的那一刻起，就與一種不對自由協作做承諾的哲學聯繫在一起。隨著這個名字的廣泛商業使用，當我們把它和社區精神相聯繫時，將會遇到更多的麻煩。

對自由軟件來說，未來的一大挑戰是來自於那些“Linux”發行版公司，為提高易用性和增強功能，將非自由的軟件添加到 GNU/Linux 中。所有主要的商業發行版都這樣做；沒有一個讓自己限於自由軟件的。他們中的大多數都在其發行版中沒有區分非自由軟件包。很多甚至開發非自由軟件並加入到其系統發行中。還有些甚至蠻橫的地宣佈那些“Linux”系統是“席位許可”的，也就是可以給用戶和 Microsoft Windows 一樣的“自由”。

人們用“普及 Linux”的名義替這種加入非自由軟件的行為辯護——事實是，更重視普及度而不是自由。有時這是被公開承認的。例如，《連線》雜誌（Wired）對 Linux Magazine 期刊的編輯 Robert McMillan 說，“感覺推動開源軟件應該用技術，而不是政治決策”。並且 Caldera 的 CEO 公開勸

說用戶放棄自由的目標，並轉而為“普及 Linux”而工作¹。

如果普及率指的是使用混合非自由軟件的 GNU/Linux 的用戶數量，添加非自由軟件到 GNU/Linux 系統或許會增加普及率。但同時，它含蓄地鼓勵社區將非自由軟件當好東西接受，而忘記自由的目標。如果你的方向不對，那麼車開得再快也沒用。

當非自由的“附加組件”是一種庫或者編程工具時，就會成為自由軟件開發者的陷阱。當他們編寫基於非自由軟件包的自由軟件時，他們的軟件不能成為完全自由系統的一部分。Motif 和 Qt 曾以這種方式誘騙了大量的自由軟件，導致的問題花費數年才能解決。Motif 的一些問題直到它被廢棄並不再被使用都沒有完全解決。Sun 公司實現的非自由 Java 直到現在也有類似的效果：Java 陷阱²，慶幸的是大多已經修復。如果我們的社區持續向這個方向發展，將會把 GNU/Linux 的未來引到自由和非自由拼接在一起的地步。從現在起的五年裡，我們仍然確信將會有大量的自由軟件產生；但如果我們不謹慎一些，用戶所期望的自由軟件就很難在脫離非自由軟件的情況下使用。如果這事發生了，我們為自由做出的努力就前功盡棄了。

如果發佈自由的替代品只是簡單的編程問題，隨著我們社區開發資源的增加，未來解決問題可能就變得愈加簡單。但是我們面臨的障礙使得這更加困難：法律禁止自由軟件。當軟件專利出現，並且像數字千年版權法案（DMCA）這樣的法律被用來阻止開發一些用於重要工作中的自由軟件時（例如看 DVD 或者聽 RealAudio 音頻流），我們會發現，除了拒絕使用調用它們的非自由程序，我們自己並沒有明確的方法來抵禦專利和私密的數據形式。

面對這些挑戰需要許多不同的努力。但是應對任何挑戰，我們首先需要牢記自由協作的目標。我們不能僅僅依靠渴望強大、可靠的軟件來激勵人們作更多的努力。我們需要大家有一種決心，一種為他們的自由和他們的社區而戰的決心——一種鑠而不捨的決心。

在我們的社區裡，這個目標和決心主要源自 GNU 工程。我們就是這樣

¹參見 Dietmar Muller 於 2001 年 7 月 10 日發表的文章《“Stallman: Love Is Not Free”》<http://zdnet.com/article/stallman-love-is-not-free/>

²參見《“自由的桎梏——Java 陷阱”》<http://gnu.org/philosophy/java-trap.html> 一文

的人——把自由和社區當作不可讓步的事情；而那些稱之為“Linux”的組織通常不說這些。關於“Linux”的雜誌更是打滿了非自由軟件的廣告；打包“Linux”的公司把非自由軟件添加進系統；其它公司通過開發在 GNU/Linux 上運行的非自由的應用來“支持 Linux”；連“Linux”的用戶群也邀請推銷員展示這些應用。人們在社區中最有可能接觸到自由的理念和決心的地方，就是在 GNU 工程中。

但是當人們接觸到它時，能體會到這與他們息息相關嗎？

那些知道自己正在用的系統是源自 GNU 工程的人，能看到他們自身與 GNU 之間有一個直接的關係。他們不會自動地同意我們的哲學，但至少他們會看到一個嚴肅思考它的理由。相反，那些認為自己是“Linux 用戶”的人，相信 GNU 工程只是“開發了被證實在 Linux 中非常有用的工具”，他們通常只瞭解 GNU 與其間接的關係。當他們接觸到 GNU 哲學時，就會忽視它。

GNU 工程是理想主義的，但是那些鼓勵理想主義的人面臨很大的障礙：普遍的意識形態鼓勵人們把理想主義當作“不現實的”而摒棄。我們的理想主義其實已經極其實用：這是我們擁有一個自由的 GNU/Linux 操作系統的原因。喜歡這個系統的人應該知道，正是我們的理想主義實現了它。

如果“工作”真的做了，如果除了信譽沒有其他利害攸關，也許放棄可能會更明智。但是我們沒有處在那個境地。為了鼓勵人們做那些需要做的工作，我們需要讓我們已經做了的工作被認可。要幫助我們就請將這個系統稱為 GNU/Linux 吧。

2.2 Linux 和 GNU 操作系統

Copyright © 1997–2002, 2007, 2014 Richard Stallman

每天，大量的用戶在毫不知情的情況下在他們的計算機上運行著修改版的 GNU 操作系統¹。而這個現在廣為使用的修改版的 GNU 操作系統通常被稱為“Linux”，但是很多它的用戶並不知道，它其實就是 GNU 計劃開發的 GNU 操作系統²。

Linux 也確實存在，並且那些用戶也確實是在使用它，不過它僅僅是這些用戶所使用的操作系統的一部分。Linux 僅僅是一個內核：它就是分配你硬件設備上的資源給其他你所使用的程序的一個特殊程序。內核是操作系統中不可或缺的一部分，但是隻有內核是遠遠不夠的。內核僅能在完整操作系統的環境下才能正常工作。一般來講，Linux 會與 GNU 操作系統合起來使用：整個系統就是添加了 Linux 內核的 GNU 操作系統，或者簡而言之，GNU/Linux。所有所謂的“Linux”發行版其實就是 GNU/Linux 發行版。

很多用戶並不能區分內核（即 Linux）和整個操作系統（他們也稱為“Linux”）。這種混淆視聽的稱呼並不能幫助人們正確理解這一區別。所以人們通常會認為 1991 年 Linus Torvalds 在沒有藉助太多其他幫助的情況下就開發出了整個操作系統。

程序員們基本上都知道 Linux 只是個內核。但是他們通常會聽到別人把整個操作系統都稱為 Linux，然後他們就會跟從接受以內核名稱命名操作系統的傳統。比如，很多人都相信當 Linus Torvalds 編寫完 Linux 內核之後，它的用戶們去尋找其他與之搭配使用的自由軟件時，（毫無道理地）發現用於構建類 Unix 系統的一切其實都已然存在。

真實的情況是，當時的他們發現了並不完全完整的 GNU 系統，而

¹參見《自由與非自由軟件的分類》查看更多有關 GNU 操作系統的信息 (p. 85)。

²詳情請見“GNU Users Who Have Never Heard of GNU”<http://gnu.org/gnu/gnu-users-never-heard-of-gnu.html> 和“Overview of the GNU System”，<http://gnu.org/gnu/gnu-history.html>。

這一發現決非偶然。當時已有的自由軟件¹加在一起一同組成了一個完整的操作系統。這是因為 GNU 項目從 1984 年就開始致力於實現這一目標。在 GNU 宣言 (GNU Manifesto)²當中，我們就確立了開發一個稱為 GNU 的類 Unix 的自由操作系統的目標。GNU 計劃的初始聲明 (The Initial Announcement)³還包括了 GNU 操作系統開發計劃的部分大綱。在 Linux 開始開發前，GNU 幾乎就要完成了。

絕大多數的自由軟件項目都是以“為某項具體的工作開發一款具體的軟件”為目的。比如，Linus Torvalds 開發了類似於 Unix 的內核 (Linux)；Donald Knuth 編寫了一個格式化文本工具 (T_EX)；而 Bob Scheifler 則開發了一種窗口管理系統 (X 窗口管理系統)。對於這類軟件而言，衡量特定的軟件對該項目的貢獻是簡單而自然的。

但是如果我們以這種方法去衡量 GNU 項目的貢獻的話，我們會得出怎樣的結果呢？某個 CD-ROM 的供應商發現在他們的“Linux 發行版”中，GNU 軟件⁴是其中最大的必備依賴軟件，約佔全部源代碼數量的 28%，並且還包括了構成完整操作系統不可或缺的一部分組件。而 Linux 自己只佔 3%（截至 2008 年，此比例仍然準確：在 gNewSense 的“主源”當中，Linux 佔 1.5%，而 GNU 軟件包占了 15%）。所以如果你想以寫操作系統的人命名這個操作系統的話，最為確切的單詞名字恐怕是“GNU”。

但這並不是考慮這個問題最深層次的方法。GNU 項目過去不是，現在也不是僅僅只創造某些特定軟件的一個項目。它不是隻開發一個 C 語言編譯器的項目⁵，雖然我們確實開發了一個 C 語言編譯器。它也不是隻開發一個文本編輯器的項目，雖然我們也做過。GNU 項目是為了開發一個完全自由的類 Unix 操作系統——GNU。

很多人都為自由軟件做過貢獻，他們理應被提名。但是問題在於這是一個完整的操作系統——並不是一個有用軟件的簡單集合——因為 GNU

¹參見《什麼是自由軟件？》(p. 1) 來獲得自由軟件的完整定義。

²查看“GNU 宣言 (GNU Manifesto)”，位於 <http://gnu.org/gnu/manifesto.html>。

³參見《GNU 操作系統的初始公告》(p. 29)。

⁴參見《自由與非自由軟件的分類》瞭解 GNU 軟件的更多信息 (p. 86)。

⁵GCC 主頁：<http://gnu.org/software/gcc/>。

項目就是要創造這樣一個操作系統。我們已經列出了一個完整操作系統所需軟件的清單，並且我們系統性地發掘、編寫，或者找人編寫清單上的所有條目。我們編寫了一些不太吸引人卻又不可或缺的組件¹，因為操作系統的正常運行不能沒有這些軟件。在我們系統組件中，有一部分是編程工具，並且在程序員中流行了起來。但是我們也編寫了很多不是工具的軟件²。我們甚至還編寫了一個棋類遊戲，GNU Chess，因為完整的操作系統還需要一些遊戲。

在 90 年代初期，我們已經搞定了一個除了內核之外的系統。我們其實也開始開發一個內核，GNU Hurd (<http://gnu.org/software/hurd/hurd.html>)，一個運行於 Mach 的內核。開發這個內核的難度遠超我們的想象；GNU Hurd 在 2001 年終於能穩定運行了，但是距離被人們日常使用的目標仍然相差甚遠³。

幸運的是，因為有 Linux 的存在，我們並不需要等待 Hurd。在 1992 年，Torvalds 自由化了 Linux 之後，它填補了 GNU 操作系統中重要的最後一道坎。人們從此可以將 GNU 操作系統⁴和 Linux 合併使用來創造一個完全自由的操作系統——一個包含 Linux 的 GNU 操作系統，即 GNU/Linux 操作系統。去讓它們能很好地協同工作並不是一件容易的事。一些 GNU 組件⁵需要作出一些必要的改動才能和 Linux 一起使用。將一個完整系統整合成一個能“開箱即用”的發行版也是一個不小的挑戰。這需要解決一個我們之前沒有遇到過的問題——如何安裝和啟動系統，因為我們還沒有研究到那一步。因此，製作了各種各樣發行版的人們進行了大量的必要工

¹這些不太引人注意而不可或缺的組件包括 GNU 彙編器 (GAS) 和鏈接器 (GNU ld)，它們現在都是 GNU Binutils 軟件包 (<http://gnu.org/software/binutils/>) 的一部分，還有 GNU tar (<http://gnu.org/software/tar/>) 等軟件。

²比如，Bourne Again Shell (BASH)，還有 PostScript 解析器 Ghostscript (<http://gnu.org/software/ghostscript/ghostscript.html>)，以及 GNU C 運行庫都不是編程工具。並且 GNUcash，GNOME，和 GNU Chess 都不是。

³參見 <http://gnu.org/software/hurd/hurd-and-linux.html> 就會明白為什麼 FSF 開發了 GNU Hurd 內核。

⁴參見“Linux 0.01 發行註記”(“Notes for Linux Release 0.01.”)：<http://ftp.funet.fi/pub/linux/historical/kernel/old-versions/RELNOTES-0.01>。

⁵比如 GNU C 運行庫 (GNU C Library) <http://gnu.org/software/libc/libc.html>。

作。但是這就像萬物的規律一樣，最終會有人去解決這個問題的。

GNU 項目支持 GNU/Linux 系統和 GNU 操作系統。FSF 資助了重寫 GNU C 運行庫中與 Linux 有關的擴展的工作，這樣，最新的 GNU/Linux 操作系統就能毫不修改地使用最新版本的庫了。FSF 還資助了 Debian GNU/Linux 的早期開發。

現在，GNU/Linux 操作系統有非常多的發行版（英文中常稱為“distros”）。它們當中的絕大部分遵從 Linux 的哲學而不是 GNU 的哲學。不過徹底自由的 GNU/Linux 發行版也是存在的¹。FSF 贊助了 gNewSense 計算機等設備（<http://gnewsense.org>）。

製作一個完全自由的 GNU/Linux 發行版並不只是刪去非自由軟件程序。現今，原版 Linux 也包含非自由程序。這些程序是在系統啟動時加載進 I/O 設備的，大量的這些程序被包含在 Linux 的“源代碼”中。因此，維護自由版本的 GNU/Linux 發行版現在還指維護一個自由版本的 Linux（<http://directory.fsf.org/project/linux>）。

不管你是否使用 GNU/Linux，請不要使用歧義詞“Linux”迷惑大眾。Linux 是內核，系統中主要的不可或缺的組件之一。而整個系統其實是加入了 Linux 的 GNU 操作系統。當你提及這個組合時，請說“GNU/Linux”。

這篇文章和“GNU 計劃”都是宣揚“GNU/Linux”的好選擇。如果你提到 Linux 內核，並且你還想引用更有深度的參考資料，FOLDOC（計算機自由在線詞典）<http://foldoc.org/linux> 是一個你可以信賴的網站。

2.2.1 附言

除了 GNU，另外一個項目也獨立完成了一個自由的類 Unix 操作系統。這個操作系統叫 BSD，研發於加州大學伯克利分校（UC Berkeley）。在八十年代期間，它並不是自由的，但是在九十年代早期，它變成了自由的。

¹參見 <http://gnu.org/distros/> 以獲得我們知道的所有完全自由的發行版的列表。

現存的自由操作系統幾乎是非 GNU 即 BSD¹。

人們有時候會問 BSD 是不是也是某種版本的 GNU，就像 GNU/Linux 那樣。其實 BSD 開發者們也是得到了 GNU 項目的啟發才使他們的代碼自由了。而且很顯然這種行為是由 GNU 的積極分子努力遊說的結果，但是他們的代碼與 GNU 並無太多交集。BSD 操作系統現在也使用一些 GNU 軟件，就像是 GNU 操作系統及其變種也使用一些 BSD 軟件一樣。然而，總體而言，它們是兩個完全不同、分開發展的操作系統。BSD 開發者沒有採用編寫內核然後放進 GNU 操作系統的做法，所以將 BSD 稱為 GNU/BSD 很顯然是不合適的²。

¹撰寫此文時，一個類 Windows 的接近全部自由的操作系統已經開發了出來，但是從技術角度來講，它不像 GNU 或者 Unix，所以並沒有受此問題的影響。Solaris 的大部分內核都已自由開放，但是如果你想基於 Solaris 實現一個完全自由的操作系統，除了需要替換內核中缺失的實現外，還需要放進 GNU 或者 BSD 當中。

²從另一個方面來講，在此文撰寫的這幾年間，GNU C 運行庫已經被移植到了很多版本的 BSD 內核上，這為集成 GNU 操作系統到該內核上帶來了方便。就像 GNU/Linux 那樣，GNU 還有好多個變種，比如 GNU/kFreeBSD 和 GNU/kNetBSD。普通桌面用戶可能不好區分 GNU/Linux 和 GNU/*BSD。

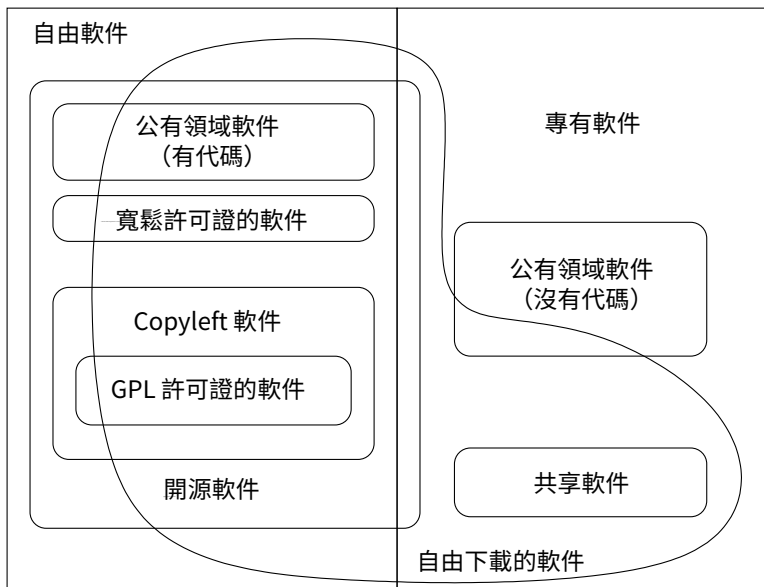


圖 2.1: 自由與非自由軟件的分類

2.3 自由與非自由軟件的分類

Copyright © 1996–1998, 2001, 2006, 2007, 2009, 2011, 2012, 2014, 2015 自由軟件基金會。本文最初於 1996 年發表於 <http://gnu.org>

參見《應避免使用（或慎用）的詞語，由於它們是不公正的或者引起混淆的》(p. 107) 一文。

本圖最初由 Chao-Kuei 創作並且經過其他人的更新，解釋了不同類別的軟件之間的區別。它有矢量圖版，位於 <http://gnu.org/philosophy/category.svg>（以及中文矢量圖版，位於 <https://www.gnu.org/philosophy/category.zh-cn.svg>），

以及 XFig 格式版本,位於<http://gnu.org/philosophy/category.fig>。本圖採用 GNU 通用公共許可證 (GNU GPL) 2.0 或更高版本、GNU 自由文檔許可證 (GNU FDL) 1.2 或更高版本、或者創作共用-署名-相同方式共享 (CC BY-SA) 許可證 2.0 或更高版本。

2.3.1 自由軟件 (Free Software)

自由軟件是指附帶這樣許可的軟件：它允許任何人使用、複製和/或再分發，不論是逐字再分發還是再分發帶有更改的版本，也不論免費還是有償。特別地，這意味著源代碼必須可獲得。“如果它不是源代碼，它就不是程序。”這是一種簡化的描述；您可以在《什麼是自由軟件？》(p. 1) 找到自由軟件的完整定義。

如果一個程序是自由的，它就有可能被包含在一種自由的操作系統中，例如 GNU，或者 GNU/Linux 操作系統的自由版本¹。

還有很多種方式可以使程序成為自由的——有很多細節的問題，對這些問題可以有多種方式的決定，而仍然能夠使程序成為自由的。一些可能的變體將會在下文進行描述。如果需要獲得關於具體的自由軟件許可證的信息，參見《許可證列表》頁面<http://gnu.org/licenses/license-list.html>。

自由軟件關乎自由而非價格。但是專有軟件公司一般使用“free software”這一短語來指代價格。有時它們的意思是您可以免費獲得一份二進制副本；有時它們的意思是在您所購買的計算機上捆綁了一份副本，而整臺計算機的價格包含了二者各自的價格。不管是哪種方式，這都與我們在 GNU 計劃中所指的自由軟件無關。

由於這種潛在的混淆，當一家軟件公司宣稱它的產品是“free software”的時候，務必查看實際的發佈條款以確認用戶是否真正擁有自由軟件所蘊含的所有基本自由。有時它真的屬於自由軟件，有時它不是。

很多語言都擁有兩個獨立的單詞來表示自由和免費。例如，法語中有

¹參見《Linux 和 GNU 操作系統》(p. 75) 一文以獲得更多信息。

“libre” 和 “gratuit”。而英語則不是：在英語中確實有一個單詞 “gratis” 用於無歧義地表示免費，但沒有普通的形容詞用於無歧義地表示自由。因此，如果您說另一種語言，我們建議您在將 “free” 翻譯為您的母語的時候澄清其涵義。參見我們整理的將 “free software” 這一短語翻譯為多種其他語言的詳細列表（見附錄 B (p. 341)）。

自由軟件常常比專有軟件更可靠¹。

2.3.2 開源軟件（Open Source Software）

“開源軟件” 這一短語被某些人用於指代與自由軟件或多或少地相同的一類軟件。它們並不是與自由軟件嚴格一致的一類軟件：它們接受某些在我們看來約束性過強的許可證（條款），也有一些自由軟件許可證（條款）是它們所拒絕接受的。然而，兩者外延的差別並不大：就我們所知，只有很少的開源軟件不是自由軟件。理論上，可能也會有自由軟件不被接受為開源軟件，但是我們還沒有看到實際的例子。

我們傾向於使用自由軟件這一概念，由於它代表自由——這是 “開源” 這一短語所不能體現的²。

2.3.3 公有領域軟件（Public Domain Software）

公有領域軟件是指不受版權保護的軟件。如果源代碼也在公有領域中，這將成為一種非 copyleft 自由軟件的特例。這意味著某些副本或者修改版本可能完全不是自由的。

在某些情況下，可執行程序可能位於公有領域中，但源代碼不可獲得。這不是自由軟件，由於自由軟件要求源代碼的可獲得性。與此同時，大部分自由軟件不在公有領域中，它們是受版權保護的，並且版權持有人通過使用自由軟件許可證合法地賦予任何人使用它們的自由許可。

有人以一種不嚴格的方式來使用 “公有領域” 這一概念來指代 “自由”

¹ 參見《自由軟件更值得信賴！》一文，位於 <http://gnu.org/software/reliability.html>。

² 參見《為什麼說開源漏掉了自由軟件的要點》(p. 90) 一文。

或“免費得到”。然而，“公有領域”是一個法律概念，準確表示“不受版權保護”之意。為了澄清，我們建議將“公有領域”僅僅用於它所嚴格指代的涵義，而使用其他短語來指代其他涵義。

根據伯爾尼保護文學和藝術作品公約（大多數國家已經簽署）¹，任何寫下的東西自動獲得版權，這也包括計算機程序。因此，如果您想要使您的程序進入公有領域，您必須採取某些法律步驟以放棄其版權；否則該程序是受版權保護的。

2.3.4 Copyleft 軟件（Copylefted Software）

Copyleft 軟件是一類自由軟件，其發佈條款可以保證它所有版本的所有副本都或多或少帶有相同的發佈條款。例如，這意味著 copyleft 許可證普遍禁止他人為該軟件添加額外的限制條款（儘管有限的一些安全的附加條款可能被允許），並且要求源代碼可獲得。這可以保護該程序及其修改版本，使得常見的某些方式無法將其變為專有軟件。

一些 copyleft 許可證例如 GNU 通用公共許可證（GNU GPL）第三版阻止了其他一些專有化的方式，例如 Tivo 化（tivoization）²。

在 GNU 計劃中，我們為我們所寫的幾乎所有軟件使用了 copyleft 許可證，由於我們的終極目標是賦予每位用戶自由軟件所蘊含的所有基本自由。參見我們關於 copyleft 的文章《什麼是 Copyleft?》(p. 222) 以獲知關於 copyleft 許可證如何發揮作用以及我們為何使用它們的更多解釋。

Copyleft 是一個通用的概念，為了以 copyleft 許可證發佈一個實際存在的程序，您需要使用特定的一系列發佈條款。有多種可能的方式編寫 copyleft 發佈條款，因此原則上可以有很多種 copyleft 自由軟件許可證。然而實際上幾乎所有 copyleft 軟件都使用 GNU GPL。兩種不同的 copyleft 許可證通常不兼容，這意味著直接將使用一種許可證的代碼與使用另一種許

¹ 《伯爾尼保護文學和藝術作品公約》（法語：Convention de Berne pour la protection des œuvres littéraires et artistiques），簡稱《伯爾尼公約》，是關於著作權保護的國際條約。《公約》生效至今（2013 年）進行過 8 次補充和修訂，共有 167 個簽約國。——譯者注，摘自維基百科

² 參見《為何升級到 GPLv3》(p. 244) 以獲得更多信息。

可證的代碼合併將會是非法的；因此，人們使用同一種 copyleft 許可證將會有利於社區。

2.3.5 非 copyleft 的自由軟件 (Noncopylefted Free Software)

非 copyleft 的自由軟件帶有來自作者的再分發和修改的許可，也帶有添加額外限制條款的許可。

如果一個程序是自由的但並未採用 copyleft 許可證，那麼某些副本或者修改版本可能完全不是自由的。軟件公司可以編譯該程序，不論是否對其進行修改，都可以將可執行文件作為專有軟件產品發佈。

X Window 系統闡釋了這一點。X 聯盟將 X11 以某種非 copyleft 發佈條款發佈，使其成為非 copyleft 的自由軟件，並且後續開發者幾乎沿用了相同的方式。一份帶有此類發佈條款的副本是自由軟件。然而，也有專有版本的存在，並且確實有（或者至少確實曾經有）流行的用於工作站或個人計算機（PC）的圖形顯示卡僅支持專有版本。如果您使用的是這類硬件，那麼 X11 對您來說不是自由軟件。X11 的開發者甚至曾經在一段時期將 X11 作為專有軟件發佈¹；他們之所以能夠如此，是由於其他人使用相同的非 copyleft 許可證貢獻了他們的代碼。

2.3.6 使用包容型許可證的軟件 (Lax Permissive Licensed Software)

包容型許可證包括 X11 許可證和兩種 BSD 許可證²。這些許可證幾乎允許對其代碼進行任何方式的使用，包括髮布專有的二進制版本，不論是否更改了源代碼。

¹ 參見《X Window 系統的陷阱》(p. 216)。

² 參見 “The BSD License Problem,” 一文，位於 <http://gnu.org/philosophy/bsd.html>。

2.3.7 使用 GPL 許可的軟件 (GPL-Covered Software)

GNU GPL 是一類特定的發佈條款集合，用於使該程序成為 copyleft 軟件。GNU 計劃將其用作大部分 GNU 軟件的發佈條款。

因此，將自由軟件等同於使用 GPL 許可的軟件是一種錯誤。

2.3.8 GNU 操作系統 (The GNU Operating System)

GNU 操作系統是一種“類 Unix 操作系統”，它完全由自由軟件構成，我們在 GNU 計劃中於 1984 年開始 GNU 操作系統開發¹。

一款類 Unix 操作系統由眾多程序構成。GNU 操作系統包括所有 GNU 程序包。它還包括很多其他程序包，諸如 X Window 系統以及 T_EX，它們不是 GNU 軟件。

完整的 GNU 操作系統的首個測試版於 1996 年發佈，它包括 GNU Hurd，我們的內核，後者於 1990 年開始開發。直到 2001 年，GNU 操作系統（包括 GNU Hurd）開始能夠相對可靠地運行，但是 Hurd 仍然缺少一些重要功能，因此它不能被廣泛使用。與此同時，GNU/Linux 操作系統，作為使用 Linux 內核而非 Hurd 內核的 GNU 操作系統的衍生版本，自 20 世紀 90 年代起已經獲得了巨大成功²。這表明了 GNU 操作系統不是一個靜態的程序集合；用戶和再分發者可以根據他們的需求和偏好來選擇不同的軟件包。其結果仍是一種 GNU 操作系統的變體。

由於 GNU 操作系統的目標是自由，GNU 操作系統中的任何一個組件都是自由軟件。但是，它們不必都是 copyleft 的，任何類別的自由軟件都可以被合法並且合適地包含進來，如果它有助於實現技術上的目標。

¹參見“GNU 系統概覽”一文，位於 <http://gnu.org/gnu/gnu-history.html> 以獲知更多歷史背景。

²參見《Linux 和 GNU 操作系統》(p. 75) 一文以獲得更多信息。

2.3.9 GNU 程序 (GNU Programs)

“GNU 程序”是 GNU 軟件的同義詞。若程序 foo 是一個 GNU 程序，那麼它也是一個 GNU 軟件。我們有時也稱之為 GNU 軟件包。

2.3.10 GNU 軟件 (GNU Software)

“GNU 軟件”是指在 GNU 計劃支持下發布的軟件¹。如果一個程序是 GNU 軟件，我們也可稱之為 GNU 程序或 GNU 軟件包。GNU 軟件包的自述文件 (README) 或手冊應當聲明它是一個 GNU 軟件包；同時，自由軟件目錄²標識了所有 GNU 軟件包。

大部分 GNU 軟件是 copyleft 的，但不是全部；然而，所有 GNU 軟件都必須是自由軟件。

有些 GNU 軟件是由自由軟件基金會 (FSF) 員工所編寫的，但是大部分 GNU 軟件來自眾多志願者³ (某些志願者由商業公司或者大學支付薪酬，但他們確實是我們的志願者)。某些貢獻的軟件由 FSF 擁有版權，有些由編寫它們的貢獻者擁有版權。

2.3.11 FSF 擁有版權的 GNU 軟件 (FSF-Copyrighted GNU Software)

GNU 軟件包的開發者可以選擇將版權轉讓給 FSF 或者自己保留。這是他們的選擇權利。

如果他們將版權轉讓給 FSF，該程序成為 FSF 擁有版權的 GNU 軟件，此時 FSF 可以強制實行它的許可證。如果他們選擇自己保留版權，強制實行其許可證是他們自己的責任。

FSF 不會接受非官方 GNU 軟件包的軟件的版權轉讓，這是一條規定。

¹參見“GNU 系統概覽”一文，位於 <http://gnu.org/gnu/gnu-history.html> 以獲知更多歷史背景。

²參見 <http://directory.fsf.org>。

³參見 <http://gnu.org/people/people.html>。

2.3.12 非自由軟件 (Nonfree Software)

非自由軟件是指任何不是自由軟件的軟件。它的使用、再分發或修改被禁止，或者要求您請求授權許可，或者被嚴格限制以致於您事實上不可能自由地進行以上行為。

2.3.13 專有軟件（或譯為“私有軟件”，Proprietary Software）

專有軟件是非自由軟件的另一種叫法。在過去，我們曾將非自由軟件進一步細分為“半自由軟件”（semifree software），它們允許非商業性的修改和再分發；以及專有軟件，它們禁止任何修改或再分發。但我們現在已經放棄了這種區分，並且現在將“專有軟件”用作非自由軟件的同義詞。

FSF 遵循這樣的規則：我們不能在自己的計算機上安裝任何專有軟件，除非暫時性地用於一種特定用途，即編寫一個自由軟件來取代它。除此之外，我們感覺沒有任何可能的藉口來安裝一款專有軟件。

例如，在 20 世紀 80 年代，我們認為在我們的計算機上安裝 Unix 是合理的，由於我們需要用它編寫一個可以取代 Unix 的自由操作系統。而現在，由於自由的操作系統已經有了，因此這一藉口不再適用；我們不會使用任何專有操作系統，並且我們所組裝的任何一臺新計算機都必須運行一款完全自由的操作系統。

我們並不堅持要求 GNU 的用戶或者貢獻者也必須嚴格遵守這條規則。它只是我們對自己制定的規則。但我們希望您也願意遵循它，為了您自己的自由。

2.3.14 免費軟件 (Freeware)

“Freeware”這一短語沒有明確並且公認的定義，但它通常被用於指代那些允許再分發但禁止修改（並且其源代碼不可獲得）的軟件包。這些軟件包不是自由軟件，因此請您不要使用“freeware”來指代自由軟件。

2.3.15 共享軟件 (Shareware)

共享軟件是指那些附帶有允許人們再分發副本的許可協議的軟件，但它宣稱任何人如果想要繼續使用其一份副本就必須支付一筆授權許可費用。

共享軟件不是自由軟件，甚至不是半自由軟件。有兩大原因：

- 對於大多數共享軟件，源代碼不可獲得；因此，您完全不能修改它們；
- 共享軟件並不帶有允許人們在不支付授權許可費用的條件下複製副本並且安裝它們的許可條款，甚至對於那些從事非盈利性活動的個人用戶也不提供這樣的許可（實際上，人們經常不遵守這種發佈條款並且仍然這樣做，但許可條款並不允許）。

2.3.16 私人軟件 (Private Software)

私人軟件或者定製軟件是指專為某一特定用戶（通常是一家組織機構或者商業公司）開發的軟件。該用戶保存和使用它，並且不會以源代碼或者二進制格式將其對公眾發佈。

一款私人軟件可以是自由軟件（儘管這對他人幫助不大），如果它的終極用戶擁有四項基本自由。特別地，如果其用戶對私人軟件擁有完整的權利，該軟件就是自由的。然而，如果用戶向其他人分發副本但不隨之提供四項基本自由，那些副本就不是自由軟件。

自由軟件關乎的是自由，而非可獲得性。總的來說，我們不認為開發一款軟件但不發佈它是一件壞事。確實有這樣的情況，若一款軟件舉足輕重，以至於人們要爭論，獨佔這款軟件而拒絕對公眾發佈是在對全人類犯錯。然而，這樣的情況畢竟罕見。大部分軟件並非如此生死攸關，拒絕將它對公眾發佈並不是特別地壞。因此，開發私人軟件或者定製軟件的實踐與自由軟件運動的原則之間並無衝突。

幾乎所有受僱傭的程序員都在開發某種定製軟件；因此大部分編程工

作就是或者可以是一種與自由軟件運動相容的方式完成的。

2.3.17 商業軟件 (Commercial Software)

“商業”和“專有”並不同！商業軟件是由企業作為其業務的一部分所開發的軟件。大部分商業軟件是專有軟件，但確實也有商業化的自由軟件，並且也有非商業化的專有軟件。

例如，GNU Ada 編譯器是由一家商業公司開發的。它一直以 GNU GPL 的條款發佈，每一份副本都是自由軟件；但它的開發者販賣其支持合同。當它們的銷售員同潛在的客戶交談時，有時客戶會說“我們更加信賴商業化的編譯器”。銷售員回答說“GNU Ada 確實是一款商業化的編譯器；它恰好也是一款自由軟件”。對於 GNU 計劃，上述優先級應該是另一種順序：重要的是 GNU Ada 是一款自由軟件；它同時也是一款商業軟件只是細節。然而，由於其作為一款商業軟件而帶來的額外發展是大有裨益的。請您幫助我們宣傳這種認識：自由的商業軟件是可能的。您可以努力通過在您想要表達“專有”的時候避免使用“商業”來達到這樣的效果。

2.4 為什麼說開源漏掉了自由軟件的要點

Copyright © 2007, 2008, 2010, 2012–2015 Richard Stallman 此文
最早於 2007 年發佈於 <http://gnu.org>

當我們說軟件“自由”的時候，我們意指它尊重用戶最根本的自由：即運行、學習和修改軟件，或者重新發布軟件副本（無論是否修改過）的自由¹。這是一個關於自由的問題，而非價格。因此我們應該理解為“言論自由”，而不是“免費啤酒”。

這些自由極其重要。它們是必要的，不僅是因為它們滿足用戶的個體利益，更是因為它們促進社會的團結——也就是分享與協作。由於我們的文化和生活變得越來越數字化，因此這些自由就變得越來越重要了。在一個由數字化的聲音、圖像和文字組成的世界裡，自由軟件正在逐漸地趨近於通常意義上的自由。

現在全世界有數以千萬計的人使用自由軟件；印度和西班牙的一些學校正在教授所有的學生使用自由的 GNU/Linux 操作系統²。但是，大多數用戶都沒有聽說過我們開發這個系統以及建立自由軟件社區的倫理原因，因為現在這個系統和社區更多地被描述為“開放源代碼”（簡稱“開源”），並將其歸屬為另一種不同的、幾乎不提及自由的哲學。

自 1983 年以來，自由軟件運動一直為計算機用戶的自由而戰。1984 年，我們發起開發自由的操作系統 GNU，避免那些否定用戶自由的非自由操作系統。在（二十世紀）八十年代，我們開發出了這個系統的重要組件，還有 GNU 通用公共許可證（GNU GPL）——一個專門用於保護所有用戶自由的許可證。

並不是所有的自由軟件用戶和開發人員都認同自由軟件運動的目標。1998 年，一部分人從自由軟件社區中分裂出去，並且開始了以“開源”為旗號的運動。提出“開源”這個說法原本是為了避免“free software”可能產生的一些誤解，但卻很快就與自由軟件運動的哲學觀點分道揚鑣了。

¹有關自由軟件的完整定義，參見《什麼是自由軟件？》(p. 1) 一文。

²關於操作系統可參見《Linux 和 GNU 操作系統》(p. 75) 一文

一些“開源”的支持者認為它是“自由軟件的商業市場運動”，因為突出現實利益能吸引商業執行部門，並且避免糾纏於他們不想聽的是非觀點。其他支持者甚至斷然否定自由軟件運動的倫理和社會價值觀。不管他們的觀點是哪一種，“開源”運動中都沒有談起或提倡這種價值觀。“開源”這個說法很快便與一些現實想法和論點勾連在一起了，比如構建強大而穩定的軟件。從那時起許多“開源”的支持者也因此而加入其中，並持同樣觀點。

自由軟件和“開源軟件”這兩個說法其實說的幾乎都是同一類軟件，但是它們所秉持的價值觀是根本不同的。開源是一種開發方法論，自由軟件是一場社會運動。對自由軟件運動而言，自由軟件是一種道義責任，尊重用戶最根本的自由。而另一方面，開源的哲學更關注如何讓軟件“更好”——只關心實用價值。開源認為，非自由軟件只是手邊問題的欠妥解答，關於“開源”的大多數討論並不關注對與錯，只關注知名度和成功¹。

然而自由軟件運動中，非自由軟件是一個社會問題，解決方案就是停止使用並轉向自由軟件。

“自由軟件”，“開放源代碼”，如果指代的是同樣（或幾乎相同²）的軟件，你用什麼名字很重要嗎？是的，因為不同的文字傳達了不同的理念。雖然現在一個以其他方式命名的自由程序能給你同樣的自由，但是可持續地構築自由依賴於教育人們珍惜自由。如果你願意提供幫助，最基本的就是提到“自由軟件”。

自由軟件運動中我們並不把開源陣營看作敵人；敵人是專有（非自由）軟件。但是我們希望人們知道我們代表自由，所以我們不能接受把我們和開源支持者混為一談。

¹一個典型的例子，比如 Jay Lyman 的文章 “Open Source Is Woven Into the Latest, Hottest Trends” (2013 年 9 月 12 日) <http://www.linuxinsider.com/story/Open-Source-Is-Woven-Into-the-Latest-Hottest-Trends-78937.html>

²參見 “How Free Software and Open Source Relate as Categories of Programs” 一文，位於 <http://gnu.org/philosophy/free-open-overlap.html>

2.4.1 自由軟件和開放源代碼的實際差別

實際上，開源的標準比自由軟件要弱一些。據我們所知，目前所有的自由軟件都算是開源軟件。幾乎所有開源軟件也都是自由軟件，但也有例外。首先是一些開源許可證過於嚴苛，因此並不能認定為是自由許可證。比如“Open Watcom”的許可證就不是自由的，因為它不允許將修改後的軟件私人使用。幸運的是，只有很少一些程序使用這樣的許可證。

其次，實踐中更重要的一點是，很多產品包含了計算機可以檢查可執行程序簽名的方法，以便阻止用戶安裝其它版本的可執行程序。只有這個有特權的公司可以構建能在設備上運行，或允許訪問其全部功能的可執行程序。這種設備我們稱之為“暴君”（Tyrants），而這種行為稱之為“tivo 化”（tivoization），以我們第一次看見這樣做的產品（Tivo）而得名。即使可執行文件是從自由的源代碼構建出來，用戶卻無法運行修改版，所以這個可執行文件也是非自由的。

開源標準並沒有認識到這個問題。它只關注源代碼的許可證問題。因此，從比如像 Linux 這種開源且自由的源代碼構建，卻不可修改的可執行文件只是開源軟件但不是自由的。很多 Android 產品包含了這種基於 Linux 的非自由 tivo 化的可執行文件。

2.4.2 對“自由軟件”和“開放源代碼”的常見誤解

術語“free software”容易被誤解：一個下意識的理解，“你可以無償得到的軟件”，而其真正含義“給予用戶真正自由的軟件”同樣與其名稱相符。我們通過廣為散播自由軟件的定義和強調“理解 free 為‘自由的言論’，而並非‘免費啤酒’”，來解決這個問題。這不是一個完美的解決方法；它不能完全排除這個問題。如果它沒有其它問題的話，也許一個沒有歧義且正確的說法會更好一些。

然而很不幸，英語裡所有可選的說法都存在各種問題。我們曾看到過很多這種情況，人們收到很多建議，但沒有一個是明顯“正確”的選擇。（例如，在某些環境中法語和西班牙語單詞“自由”是有效的，但印度人卻完

全不認可。) 每個提交的用來替代“自由軟件”的詞語都存在一些語義問題——這也包括“開源軟件”。

“開放源代碼軟件”的官方定義（由開放源代碼促進會公佈，這裡引用太長了¹⁾）是間接引述自我們的“自由軟件”的標準。與自由軟件不同；在一些方面稍微有點寬鬆。儘管如此，他們的定義大多數情況還是非常接近我們的定義的。

然而，“開源軟件”這個概念表面的意思是“你可以看源代碼”，因此多數人似乎都把這當作它的真正意義。這個標準比自由軟件更脆弱，甚至比開源的官方定義都脆弱。這樣就包括了很多既不自由也不開源的軟件。

由於“開放源代碼”的表面含義並非擁護者們所設想的，結果大多數人曲解了這個說法。以下是撰稿人 Neal Stephenson 所定義的“開源”：“Linux 是‘開源’軟件的意思，簡單地說，就是任何人都可以得到源代碼的副本²⁾。”我並不認為他有意牴觸或者爭論官方的定義。我想他只是用了英語的習俗來提出開源的含義。堪薩斯州曾經發表過類似的定義：“使用開源軟件(OSS)。開源軟件是源代碼可自由、公開使用的軟件，但特定的許可證規定了人們可以用代碼做些什麼³⁾”。

《紐約時報》的一篇文章引申了開源的意思，認為開源就是讓用戶做測試⁴⁾——讓一小部分用戶測試產品的早期版本並給出反饋——而這已經是專有軟件數十年來一直在做的事情。

¹⁾全部定義可參見 <http://opensource.org/docs/osd>

²⁾Neal Stephenson, *In the Beginning...Was the Command Line* (紐約, HarperCollins 出版, 1999 年), p 49。

³⁾堪薩斯州全州技術架構, “信息架構”, version 8.0, 20.3.8, 2011 年 10 月 11 日。 <https://web.archive.org/web/20001011193422/http://da.state.ks.us/ITEC/TechArchPt6ver80.pdf>

⁴⁾Mary Jane Irwin, “The Brave New World of Open-Source Game Design” (開源遊戲設計的勇敢新世界), 紐約時報, 在線版, 2009 年 2 月 7 日。 <http://www.nytimes.com/external/gigaom/2009/02/07/07gigaom-the-brave-new-world-of-open-source-game-design-37415.html>

這個說法還被引申為設計和發佈沒有專利的產品¹。無專利的設備對社會而言確實值得稱讚，但是“源代碼”並不屬於這些。

開源的支持者們試圖通過指出他們的官方定義來應對這個問題，但收效甚微，甚至還不如直接用“自由軟件”。“free software”這個說法天生就只有兩種含義，其中一個是我們設想的含義，所以領會了“言論自由，而並非免費的啤酒”的人就不會曲解它。但是“開源”只有一個天然的含義，但是這個含義卻和它的支持者們預想的不同。所以不存在一個一勞永逸的方法來解釋和證明其官方定義。這就產生了更大的歧義。

另外一個關於“開源”的誤解認為它的意思是“不使用 GNU GPL”。這還會產對一個對“自由軟件”的誤解，認為自由軟件就是“GPL 許可證保護的軟件”。這樣都不對了，因為 GNU GPL 被當作是一個開源許可證，而大多數開源許可證被當作是自由軟件許可證。而且還有很多不是 GNU GPL 的自由軟件許可證²。

“開放源代碼”甚至已經被引申和應用到了其他活動中，比如政府、教育和科學，那些沒有源代碼的，與軟件的標準並不相關的領域。這些活動唯一的共同點是他們只是以某種方式邀請人們參加。而目前為止這個概念只是被引申成了“參與”或“透明”而已，甚至更淺。在最壞的情況下，它已經成為一個空洞的流行語³。

2.4.3 不同的價值觀會得出類似的結論……但並非總是如此

在二十世紀六十年代，激進組織以派別紛爭而聞名：一些組織因為對策略上細節的分歧而導致分裂。儘管依然有著相似的基本目標和價值觀，

¹Karl Mathiesen 和 Tess Riley, “Texas Teenager Creates \$20 Water Purifier to Tackle Toxic E-Waste Pollution” (德州青少年創建 \$20 水淨化器，以解決有毒電子廢物汙染環境)，2015 年 8 月 27 日。<http://theguardian.com/sustainable-business/2015/aug/27/texas-teenager-water-purifier-toxic-e-waste-pollution>

²參見“Various Licenses and Comments about Them”<http://gnu.org/licenses/license-list.html>

³Evgeny Morozov, “Open and Closed” (開放和封閉)，2013 年 3 月 16 日，<http://www.nytimes.com/2013/03/17/opinion/sunday/morozov-open-and-closed.html>

但是分裂出來的兩個組織往往把對方當作敵人。右翼利用了這一點，來抨擊整個左派。

有些人試圖把我們和開源之間的分歧比作這些激進組織之間的分歧，以此來詆譏自由軟件運動。但是他們完全錯了。我們雖然和開源陣營在根本目標和價值觀上無法達成共識，但是他們和我們的觀點在許多情況下是一樣的，都引領了實際行動——比如開發自由軟件。

結果是，來自自由軟件運動和開源陣營的人們經常在一些諸如軟件開發的實際項目中協同工作。不同的哲學觀點能夠如此頻繁地激勵不同人參與到相同的項目中，這是非同尋常的。不過因為這些觀點相差很大，所以也存在導致行為相差很大的情況。

開放源代碼的理念是允許用戶修改和重新發布軟件以使它更強大、更可靠。但是這些並不能得到保證。專有軟件的開發者們並不一定就不稱職。有時他們開發的程序也是強大而且可靠的，儘管它並沒有尊重用戶的自由。自由軟件積極分子和開源愛好者對此的反應是非常不同的。

一個從未受到自由軟件思想影響的純開源愛好者，會說：“我非常驚訝你們不使用我們的開發模式，居然能讓程序工作得這麼好，但是你們確實做到了。我怎樣才能得到一份副本呢？”。這種態度會鼓勵那些奪走我們自由，或有損自由的項目。

自由軟件積極擁護者會說：“你的程序很吸引人，但我更珍視我的自由。所以我抵制你的程序。我會以其他方式努力，支持一個開發自由替代品的項目。”如果我們珍視我們的自由，我們就能行動起來保衛它。

2.4.4 強大而可靠的軟件可能是件壞事

我們想讓軟件變得強大而可靠的想法，來自“軟件是被設計用於服務其用戶”的假設。如果軟件強大而可靠，這意味著它將更好地服務於用戶。

然而只有在軟件尊重了用戶的自由時，才能說它服務於用戶。如果軟件被設計的目的是使用戶枷鎖纏身會怎樣呢？那樣的話，強大僅僅意味著枷鎖套得更緊，而可靠則使枷鎖更難去除。惡意的功能，比如針對用戶的間諜行為、限制用戶、後門和在專有軟件裡很常見的強制性升級，甚至有

些開源的支持者也想實現這些。

在電影和唱片公司的壓力下，個人使用的軟件逐漸被有意地設計成能夠限制用戶的軟件。這種惡意的特徵被稱為 DRM 或者數字限制管理 (Digital Restrictions Management, 參見 DefectiveByDesign.org)，而這正是與自由軟件所要提倡的自由精神相對立的。並且不僅僅在精神上：因為 DRM 的目標是踐踏你們的自由，DRM 的開發者們試圖使你修改實現了 DRM 的軟件變得困難、或者不可能，甚至非法。

然而，一些開源支持者提出了所謂“開源 DRM”軟件。他們的想法是，通過公開並允許人們修改那些被設計成用來限制你訪問加密媒體的程序的源代碼，生產更加強大、可靠的軟件來限制像你這樣的用戶。然後，再將這些程序放到那些不允許你修改的設備中發佈。

這種軟件或許是開放源代碼的，而且使用了開源的開發模式，但它並不是自由軟件，因為並沒有尊重實際運行它的用戶的自由。如果開源開發模式通過開發更強大和更可靠的軟件來限制用戶而獲得了成功，這將使它變得更糟糕。

2.4.5 對自由的敬畏

起初，開源陣營從自由軟件運動中分裂出來，是因為“自由軟件”的倫理觀念使一些人不安。這的確是事實：談論關於自由、道義問題、可靠性和方便性，提醒人們去考慮那些可能被他們忽略的問題，比如他們的行為是否是道德的。這會引起不舒服，而且一些人甚至會迴避它。但這並不意味著我們應該停止討論這些事情。

然而，這正是開源的引領者決定要做的。他們希望停止對倫理規範和自由問題的討論，而僅僅關注某個自由軟件直接的實際利潤，這樣，他們或許就能更有效地將軟件“賣”給某些用戶，特別是商業用戶。

用他們自己的話說，這種方法已經被證實是有效的。開源的修辭方式已經吸引了許多商業和個人用戶來使用、甚至開發這些壯大我們社區的自由軟件——但僅僅是在表面的、實用上的。開源哲學，和它純粹的實用主義價值觀，阻礙了人們對自由軟件更深層觀點的理解；它將許多人帶進了

我們的社區，但卻沒有教他們去捍衛社區。就目前的現狀來看還是不錯的，但是它不足以捍衛自由。要吸引自由軟件用戶，需要讓他們成為個體自由的捍衛者。

遲早這些用戶會因為一些實際利益而轉向專有軟件。無數的公司試圖提供這種誘惑，有些甚至提供免費副本。為什麼用戶會減少？只有他們領悟了自由軟件所賦予用戶的自由，去珍惜自由的價值，而不只關心特定自由軟件在技術上和實用性上的方便性價值，他們就會明白了。為了傳播這種思想，我們必須談論自由。對商業採取一定的沉默方式對於社區來說是有利的，但是，如果大家普遍地把對自由的熱愛看作是一種怪癖，那將是很危險的。

這種危險確實已經發生。大多數投入自由軟件的人，特別是發行者，很少談論“自由”——通常是因為他們追求“更容易被商業接受”。幾乎所有的 GNU/Linux 操作系統發行版都在基本的自由系統上加入了專有軟件包，而且他們讓用戶認為這是優點，而不是與“自由”思想背道而馳的。

加入專有版權的軟件和部分非自由的 GNU/Linux 發行版之所以找到了孕育的溫床，是因為我們的許多社區沒有在他們的軟件上貫徹自由。這並非巧合。大多數 GNU/Linux 的用戶是通過討論“開源”而引入到這個系統的，而這些討論卻並沒有將“自由”作為目標。那些不支持自由的行為和不探討自由的言論並肩而行，相互促進。為了避免這種趨勢，我們需要更多地，而不是更少地，來談論“自由”。

2.4.6 “FLOSS”和“FOSS”

為了在自由軟件和開源之間保持中立，常用“FLOSS”和“FOSS”¹這種說法。如果你的目標是中立性，那麼“FLOSS”更好，因為這真的很中立。如果你更想表達自由，那麼用中立的說法並不好。站在自由的一邊，讓人們看到你對自由的支持。

¹可參見《應避免使用（或慎用）的詞語》一文的“自由和開源軟件（FLOSS）”一節（p. 116）和《FLOSS 和 FOSS》<http://www.gnu.org/philosophy/floss-and-foss.html>

2.4.7 對立的傳播

“自由”和“開放”在傳播上是一組對立的概念。“自由軟件”和“開源”雖然是不同的思想，但大多數人認為這兩個在概念上是相同的。當人們變得習慣於說和思考“開源”，這對他們獲取自由軟件運動的理念和思考是一個障礙。如果他們已經將“開放”和我們聯繫起來，在他們認識到我們代表其他事情之前，也許需要衝擊他們的理性。任何推廣“開放”的活動實際上都會更深的讓自由軟件運動隱藏在幕後。

因此，自由軟件的活躍者經過深思熟慮以後決定拒絕參加那些標榜自己是“開放”的活動。即使活動本身並不錯，但每次為“開放”多做的貢獻，都會對另一方作出傷害。也有很多很好的活動是“自由”或“解放”的。而參與這些活動，會對這些項目有小小的額外好處。既然有這麼多有益的項目可選，為什麼不選能更有好處的呢？

2.4.8 結論

開源的擁護者把新用戶引入我們的社區，我們這些自由軟件支持者必須做更多地工作來讓這些新用戶關注自由的問題。我們必須大聲且更大聲的高呼：“是自由軟件給予了你們自由！”每當你說“自由軟件”，而不是說“開源”的時候，你就是在為我們助陣。

備註

Karim R. Lakhani 和 Robert G. Wolf 的論文 “Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects” (劍橋: MIT 出版社, 2005) <http://ocw.mit.edu/courses/sloan-school-of-management/15-352-managing-innovation-emerging-trends-spring-2005/readings/lakhaniwolf.pdf>, 在探討自由軟件開發者的動機時，指出相當一部分人被“軟件必須是自由的”的觀點所鼓舞。這裡忽略了他們調查的對象是 SourceFourge 的開發人員這一事實，而該站點並不支持這是個倫理問題的觀點。

2.5 您說過“知識產權”嗎？這是一種迷惑的幻景

Copyright © 2004, 2006, 2007, 2009, 2010, 2013, 2015 Richard Stallman. 本文創作於 2004 年並且發表在 Policy Futures in Education, vol. 4, n. 4, pp. 334–336, 2006。

有這樣一種時興的趨勢：將版權、專利、商標——三種相互獨立而又不同，並且涉及三種獨立而又不同的法律概念——再加上其他幾十種法律一股腦地裝進一個罐中，並稱之為“知識產權”。這種涵義扭曲而又使人混淆的概念變得普遍流行並非偶然。從這種混淆中得利的商業公司推動了這種混淆。要想避免這種混淆，最明瞭的方式就是完全拒絕這一概念。

根據現在供職於斯坦福法學院的 Mark Lemley 教授的觀點，對“知識產權”這一概念的廣泛使用變得流行是始於 1967 年世界“知識產權”組織（WIPO）的成立，而這一概念只是在最近數年才變得真正普遍（WIPO 是聯合國的正式機構之一，但它實際上代表的是版權、專利及商標持有人的利益）。這一概念的廣泛使用始於 1990 年前後。

這一概念本身具有一種不難覺察的偏見：它暗示人們在思考版權、專利和商標的時候同實物的產權進行類比（這種類比與版權法、專利法、商標法存在矛盾，但只有專業人士瞭解這種矛盾）。這些法律實際上並不像物權法，但這一概念的濫用使得立法者將它們修改得越來越像後者。由於這種轉變正是那些行使版權、專利和商標權力的商業公司所求之不得的，於是這種由“知識產權”這一概念所帶來的偏見迎合了它們。

這種偏見已經足以成為拒絕這一概念的理由，並且人們經常建議我為這一整體類別起個別的名字——也有人提出了他們自己起的名字（通常是幽默的）。這些名字包括“小惡魔”（IMPs，意為 Imposed Monopoly Privileges，即“強制壟斷特權”）以及“魔像”（GOLEMS，意為 Government-Originated Legally Enforced Monopolies，即“政府主導的合法強制壟斷”），還有人說“專屬權利王國”（exclusive rights regime，ERR），但將“限制”（restriction）指代為“權利”也是一種雙關。

有些別名可能確實是更好的稱謂，但是，使用它們中的任何一個來指

代“知識產權”則是一種錯誤。由於其他名字都未能揭露這一概念的深層次問題：過度廣義化。並不存在諸如“知識產權”這樣高度統一的事物——它是一種幻景。人們之所以會誤認為“知識產權”是一種合乎邏輯的分類，原因僅僅在於這一概念的廣泛使用對於人們理解相關法律產生了誤導。

“知識產權”這一概念充其量只是把不同類的法律混裝在一起的雜物容器。非法律專業的人們在聽到這一適用於不同法律的概念時，傾向於想象它們相似地基於某種共同的原則和功能。

然而，事實遠非如此。這些法律是獨立起草的，以不同方式演進的，覆蓋不同的行為，擁有不同的規則，並且由此帶來了不同的公共政策問題。

例如，版權法被設計為提升作者身份和藝術，並且覆蓋了作品表達的細節。專利法的本意是促進發表有用的想法，其代價是賦予想法的發表者關於此想法的暫時壟斷權——這種代價對於某些領域是值得付出的，而對於其他領域則不然。

與之不同的是，商標法的本意並非提升任何行為方式，而只是讓消費者知道他們所購買的東西是什麼。受“知識產權”概念影響的立法者卻將它變成了一種刺激廣告行為的模式。而這些只是他們所談論的眾多法律裡的三種。

由於這些法律是獨立發展的，它們在任何細節上都是不同的。它們的基本目的和方法也是不同的。因此，如果您知道某些關於版權法的細節，並且您明智地想象專利法在這些方面與之不同，您將幾乎不會犯任何錯誤！

事實上，您所遇到的那些基於“知識產權”所闡述的廣義化的陳述幾乎全部是錯誤的。例如，您可能會看到這樣的宣傳：“知識產權”的目的是“促進創新”，但這僅僅適用於專利法，也許還有植物品種壟斷權。版權法與創新沒有關係，一支流行歌曲或是一部小說即使沒有任何創新也可以獲得版權。商標法和創新也沒有關係；如果我自營一家茶店並將其命名為“RMS 茶”，它將成為合法的商標，即使我以和任何其他人相同的方式販賣茶飲。商業機密法和創新也沒有關係，即使有，也只是稍微沾邊；我的茶店客戶清單可以成為商業機密，但這與創新沒有關係。

您也會見到這樣的論斷，“知識產權”與創造力有關，但事實上這隻適

用於版權法。除了創造力之外，還需要其他東西才能打造出一件可獲專利的發明。商標法和商業機密法與創造力沒有關係；“RMS 茶”的名字並不體現任何創造力，我的茶店客戶機密清單也和創造力無關。

當人們談論“知識產權”時，他們通常真正想要表達的是或多或少一系列法律。例如，富國經常向窮國強行施加不平等法律以榨取其錢財。這些不平等法律中的一些也位於它們所稱的“知識產權法”之中，而其他一些則不是；然而，對這一行為的批評通常也會使用“知識產權”這一標籤，由於他們對此很熟悉。通過使用這一概念，他們對問題的本質進行了錯誤解讀。最好能夠使用一個準確的術語，諸如“立法殖民”，這樣的術語可以切中要害。

非專業人士並不是被“知識產權”這一概念所迷惑的唯一人群。即使是講授這些法律的法學教授也會被這一概念的迷惑性所引誘和誤導，並且做出與其所知的事實相違背的廣義化陳述。例如，一位教授在 2006 年寫道：

與現在效力於 WIPO 的繼承者不同，美國憲法的制定者對知識產權持有原則堅定、促進競爭的態度。他們知道權利可能是必需的，但是……他們束縛了國會的雙手，並且以多種不同的方式限制它的權力。

這段陳述所指的是美國憲法第 I 章，第 8 條，第 8 款，它許可了版權法和專利法。但這一條款與商標法、商業機密法或是任何其他不同的法律無關。而“知識產權”這一概念誘使那位教授做出了錯誤的廣義化。

“知識產權”這一概念也會導致過度簡單化的思考。它促使人們僅僅關注這些不同類法律所擁有的，微小的，形式上的共同點——於是他們為某些群體創造了人為的特權——而無視那些構成它們基礎的細節：每種法律為公眾施加的特別的限制及其帶來的後果。這種對形式的過度簡單化關注助長了一種應對所有這些問題的“經濟主義”方式。

如同它經常發揮作用的方式，經濟在這裡成為了未經檢驗的設想的載體。這包括了關於價值的設想，諸如產量與之有關，而自由和生活方式與之無關；以及那些經事實證明大部分是錯誤的設想，諸如音樂版權是在支

持音樂家，或是藥品專利是在支持救生藥物的研究。

還有一個問題是，由於“知識產權”這一概念在很大程度上的含混不清，由不同法律引起的特定問題變得幾乎不可見。這些特別的問題源自每一條特定的法律——而這恰恰是“知識產權”這一概念試圖誘導人們所忽略的。例如，版權法所帶來的問題之一是音樂分享是否應該被允許；而專利法與此無關。專利法引起的問題諸如窮國是否應該被允許生產救生藥物並且以較低的價格出售以挽救生命，而版權法卻與此無關。

這些問題究其本質都不是純粹的經濟問題，它們在非經濟層面有著顯著差別；基於經濟方面的膚淺的過度廣義化概念來試圖思考這些問題，意味著忽略它們之間的區別。將兩種法律一同裝進“知識產權”的罐中將會阻礙對其任意一方清晰的思考。

因此，任何關於“知識產權問題”的觀點以及任何關於這種假想類別的廣義化幾乎一定是愚蠢的。如果您認為所有這些法律都是一個問題，您就會傾向於從過於籠統的過度廣義化概念中選擇一種作為您的觀點，而這些都是有百害無一利的。

如果您想要對專利法、版權法、商標法或是其他不同法律所引起的問題進行清晰的思考，第一步就是忘記將它們不分青紅皂白地放在一起的想法，並且將它們作為獨立的問題區別對待。第二步是拒絕“知識產權”這一概念所提示的狹隘視角和過度簡單化的情景。只有對每個問題進行獨立且完整的思考，您才能有機會清晰地思考它們。

2.5.1 註記

- 參見“The Curious History of Komongistan (Busting the Term ‘Intellectual Property’)”一文，位於<http://gnu.org/philosophy/komongistan.html>。
- 可以用非洲各國之間的關係類比這些法律之間的關係，並且“非洲”是一個合乎邏輯的地理概念；然而，談論“非洲”而非某個特定的國

家將會引發大量混亂¹。

- Rickard Falkvinge 也支持拒絕使用“知識產權”這一概念²。

¹Nicolas Kayser-Bril, “Africa Is Not a Country”, 於 2014 年一月 24 日發表於 <http://theguardian.com/world/2014/jan/24/africa-clinton>。

²“Language Matters: Framing the Copyright Monopoly So We Can Keep Our Liberties”, 於 2013 年七月 14 日發表於 <http://torrentfreak.com/language-matters-framing-the-copyright-monopoly-so-we-can-keep-our-liberties-130714>。

2.6 為何稱之為詐騙 (Swindle)?

Copyright © 2013 Richard Stallman.

我有意地用具有批判性的名字來稱呼醜惡的東西。我將蘋果迫使用戶屈從於它們的計算機稱為“iThings”，而將亞馬遜虐待讀者的電子書閱讀器稱為“詐騙” (Swindle)。有時我將微軟的操作系統稱為“Losedows”，而將微軟的第一款操作系統稱為“MS-Dog”¹。當然，我如此做是為了表達自己的不滿以及作為消遣。然而這種消遣並不是個人層面的，它服務於一項重要目的。通過嘲諷我們的敵人，我們可以將幽默的力量注入我們的事業。

扭曲名字是一種不尊重對方的行為。如果我們尊重這些產品的製造商，我們將會使用它們所選定的名字……並且那將是恰如其分的。這些惡意的產品只配得到我們的鄙視而非我們的尊重。每一款私有軟件有迫使其用戶屈從於某些實體的權力，但如今最常用的私有軟件已經不限於窺探用戶和限制用戶，甚至是任意擺佈用戶：趨勢是這些產品正在變得越來越醜惡。它們理應被消滅乾淨，其中帶有數字限制管理 (DRM) 的產品應該被判定為非法。

當我們談論它們的時候，我們應當表達出我們對它們的譴責，那麼，還有什麼方式比扭曲它們的名字更容易呢？如果我們不如此做，就很可能在稱呼它們的同時未能表達出我們的譴責。例如，當它們出現在某些其他話題之中時，下筆千言來解釋它們為什麼是壞的，就會顯得離題萬里。

點名談論這些產品但未能表達出我們的譴責，將會產生使其合法化的反效果，這是與它們所應得的譴責正好相反的。

商業公司將為其產品選擇名字作為其市場計劃的一部分。它們通常選擇那些它們認為人們更傾向於重複的名字，然後在市場營銷活動中投入數以百萬計的資金以迫使人們重複並且思考這些名字。通常，這些營銷活動的意圖是促使人們基於其表面的誘人之處來崇拜這些產品，而忽視它們所造成的危害。

¹採取行動反對 iThings，位於 u.fsf.org/ithings；反對亞馬遜詐騙 (Swindle) 位於 u.fsf.org/swindle 和 u.fsf.org/ebookslit；以及反對 Windows，位於 upgradefromwindows.org。

每當我們用製造商所使用的名字來稱呼這些產品的時候，我們是在為它們的市場營銷活動做貢獻。重複這些名字是對其產品的積極支持；而扭曲它們的名字則是在拒絕給予這些產品支持。

除了產品名稱以外，其他術語也會帶來類似的問題。例如，數字限制管理 (DRM) 指的是為了某些其他人的利益而製造技術產品以限制用戶。這種不可原諒的行徑只配得到我們的怒火，直到我們將其徹底消滅。很自然地，那些應該為此受到譴責的人們基於他們看待此問題的立場為它起了這樣一個名字：數字版權管理。這個名字是一場致力於贏得從各級政府到萬維網聯盟 (W3C) 的各種實體的支持的公共關係運動的基礎。¹

使用它們的稱謂就是支持他們的立場。但如果那不是您的立場，為什麼要給予它們默許的支持呢？

我們站在用戶的立場上，並且從用戶的觀點出發，那些惡意功能所管理的不是權利而是限制，因此我們稱之為“數字限制管理”。

這兩種稱謂都不是中立的：在從中選擇一種稱謂的同時，您選擇了一種立場。請您選擇用戶的立場並且將其發揚光大。

有一次，我的演講聽眾中的某人宣稱“數字版權管理”才是 DRM 的官方名稱並且是唯一可能的正確名稱，由於這是它的第一個名字。此人爭論以此出發，我們稱之為“數字限制管理”是錯誤的。

製造產品或者從事商業行為的實體通常會在我們尚未獲知其存在的時候就為其選定了一個名字。如果它們的暫時優勢地位迫使我們使用它們所起的名字，它們就將在無形之中獲得額外的優勢，這基於它們的資金、媒體影響力以及技術地位。我們將會不得不言不由衷地對抗它們。

有人不喜歡扭曲名字的方式並且認為這是“幼稚”或者“不專業”的。他們的意思是，這聽起來並不顯得古板而缺乏幽默——後者是一件好事，由於在我們的立場上，如果我們試圖表現得“專業”，就不能擁有歡樂。反抗壓迫遠比從事專業工作更加嚴肅，因此我們必須要有一些詼諧的放鬆。這需要真正的成熟，這裡包括一些幼稚，而非僅僅是“表現得像個成年人”。

如果您不贊同我們所選定的帶有諷刺意味的諧音或形近單詞，您可以

¹參見 <https://u.fsf.org/drm> 以獲得關於數字限制管理 (DRM) 的更多信息。

創造您自己所喜好的名字，越多越歡樂。當然，還有其他方式以表達譴責。如果您想要表現得“專業”，您可以採用其他方式將其呈現。它們能夠表達您的用意，但需要更多時間和努力，特別是當您不想使用嘲諷的手法時。注意到這一點，您的譴責才不會有所欠缺；不要讓關於反對這種“題外話”的壓力迫使您未能對您所提到的那些齷齪的東西進行足夠的批判，因為這將會起到使其合法化的反效果。

2.7 應避免使用（或慎用）的詞語，由於它們是不公正的或者引起混淆的

Copyright © 1996–1999, 2001–2004, 2007–2015, 自由軟件基金會。本列單最初於 1996 年發表於 <http://gnu.org>。

有些單詞或短語是我們建議避免使用或者避免在特定上下文或應用場景使用的。其中的一些詞語具有歧義或誤導性；而其他一些詞語則預設了某種我們所不贊同的觀點，並且我們希望您也不會同意那些觀點。（參見《自由與非自由軟件的分類》(p. 80) 和《為何稱之為詐騙 (Swindle)?》(p. 104) 兩篇文章。)

2.7.1 可獲得性 (Access)

有一種普遍的誤解認為，自由軟件意味著公眾必須可以獲得某一程序。這並非自由軟件的本意。

自由軟件的準則¹並不是關於誰能夠獲得某一程序的；四項基本自由所關注的是擁有它的一份副本的用戶可以用它做什麼。例如，自由之二稱該用戶擁有為其複製一份副本並且送給或賣給您的自由。但是任何用戶都沒有為您複製副本的義務；並且您也沒有權利強行要求任何用戶為您提供副本。

特別地，如果您為您自己編寫了一個程序並且完全不向任何人提供副本，該程序也是自由軟件（儘管是在一種平凡的意義上），由於您（作為擁有它的唯一用戶）擁有四項基本自由。

事實上，當很多用戶擁有某一程序的副本時，某些人確實會將其發佈到互聯網上，並且供其他人訪問。我們認為人們應當如此，如果該程序確實有用。但這並非自由軟件的強制要求。

確實有一種特殊情況，此時是否擁有可獲得性與自由軟件直接相關：GNU 通用公共許可證 (GNU GPL) 允許賦予一位特定用戶下載程序源代

¹參見《什麼是自由軟件?》(p. 1) 以獲得自由軟件的完整定義。

碼的訪問權，作為以實物載體的形式為該用戶提供源代碼副本的替代方案。這條規則也適用於用戶已經擁有一份非源代碼形式的副本這一特例。

2.7.2 選擇 (Alternative)

我們從不將自由軟件描述為私有軟件之外的選擇，由於這一單詞假設所有“選擇”都是合法的，並且每增加一種“選擇”都對用戶有利。事實上，這一單詞假設自由軟件應該和那些不尊重用戶自由的軟件共存。

我們堅信以自由軟件形式發佈是發佈軟件以供他人使用的唯一符合倫理的方式。而其他方式，無論是私有軟件還是“作為軟件替代品的服務”(SaaS) 都會迫使用戶屈從¹。因此我們認為不向用戶提供這些自由軟件之外的“選擇”才是好的。

2.7.3 BSD 風格 (BSD-Style)

“BSD 風格許可證”這一表述將會引起混淆，由於它將具有顯著不同的許可證混為一談²。例如，最初的 BSD 許可證由於帶有廣告條款從而與 GNU GPL 不兼容，但是改進過的 BSD 許可證與 GPL 兼容。

為了避免混淆，最好在談論中明確給出許可證的名字³並且避免使用含混不清的短語“BSD 風格”。

2.7.4 閉源 (Closed)

將私有軟件描述為“閉源”很明顯是與“開源”概念相對。在自由軟件運動中，我們不希望被別人同開源陣營混為一談，因此我們謹慎地避免使用那種促使人們將我們與他們混在一起的表達方式⁴。例如，我們避免將非

¹參見《如今自由軟件更加重要》(p. 32) 和《服務器真正是在為誰服務?》(p. 302) 以獲得更多信息。

²參見“The BSD License Problem”一文，位於 <http://gnu.org/philosophy/bsd.html>。

³參見“Various Licenses and Comments about Them”，位於 <http://gnu.org/licenses/license-list.html>。

⁴參見《為什麼說開源漏掉了自由軟件的要點》(p. 90)。

自由軟件描述為“閉源”，我們稱之為“非自由”或“私有”¹。

2.7.5 雲計算（Cloud Computing）

“雲計算”這一短語（或者在表示計算的上下文中進一步簡稱為“雲”）是一種含混不清的市場化流行語，它沒有任何合乎邏輯的涵義。它被用於指代一系列不同的活動，其唯一的共同點是它們都使用互聯網進行文件傳輸以外的事情。因此，這一短語是在散佈混淆。如果您基於它進行思考，您的思考將會是混亂的。

當您思考或是迴應一個由他人提出的並且使用這一概念的論述時，第一步是要澄清話題。該論述是關於什麼場景的？適用於此場景的恰當並且清晰的概念是什麼？只有當話題被清晰闡述之時，合乎邏輯的討論才是可能的。

“雲計算”的眾多可能的涵義之一是將您的數據存儲於在線服務上。在大多數場景中，這是愚蠢的做法，由於這使您暴露在監控之下²。

另一種可能的涵義（與上一條有所重疊但又不完全相同）是“作為軟件替代品的服務”（SaaS），它拒絕了您對於您自己的計算的控制權。您應該從不使用 SaaS³。

另一種可能的涵義包括租賃遠程的實體或虛擬服務器。這些實踐在某些特定環境下是可以接受的。

另一種可能的涵義是通過您自己的移動設備訪問您自己的服務器。這並不會產生特別的倫理問題。

美國國家標準技術研究所（NIST）為“雲計算”的定義⁴提出了三種場

¹參見《自由與非自由軟件的分類》以獲得關於私有軟件的更多信息（p. 87）。

²John Harris, “Why Hackers and Spooks Want Our Heads in the Cloud”, 2011 年四月 25 日, <http://guardian.co.uk/commentisfree/2011/apr/25/hackers-spooks-cloud-antiauthoritarian-dream>。

³參見《服務器真正是在為誰服務?》（p. 302）以獲得關於此問題的更多信息。

⁴Peter Mell 和 Anthony Grance, “The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology”, NIST Special Publication 800-145 (2011 年九月), <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>。

景，它們會帶來不同的倫理問題：軟件即服務（SaaS）、平臺即服務（PaaS）、基礎設施即服務（IaaS）。然而，這種定義與“雲計算”通常的用法並不匹配，由於它並未包括向在線服務中存儲數據。由 NIST 定義的 SaaS 在很大程度上與 SaaS 重疊，後者虐待它的用戶，但這兩個概念並不同。

這些不同的計算實踐甚至不屬於同一個討論範圍。避免使用“雲計算”這一帶來混淆的概念，最佳方式是不使用術語“雲”與計算相關聯。只談論您想要表達的話題，並且以一種具體的概念稱呼它。

有趣的是，Larry Ellison，一位私有軟件開發者也注意到了“雲計算”這一概念的空泛性¹。但他決定仍然使用這一概念，因為他是一位私有軟件開發者，畢竟此人與我們不是由相同理念所驅使的。

2.7.6 商業（Commercial）

請不要將“商業”用作“非自由”的同義詞。這將兩種完全不同的問題混為一談。

一個程序稱為商業軟件，如果它是作為一種商業行為而被開發的。一個商業軟件可以是自由或非自由的，取決於它的發佈方式。類似地，由學校或個人開發的程序也可以是自由或非自由的，取決於它的發佈方式。這兩個問題——程序由何種實體開發，以及它的用戶應當擁有什麼自由——是不相關的。

在自由軟件運動的最初十年，自由軟件包幾乎都是非商業的；GNU/Linux 操作系統的組件由個人或者諸如自由軟件基金會（FSF）和大學的非盈利性組織所開發。其後，在 20 世紀 90 年代，自由的商業軟件開始出現。

自由的商業軟件是對我們的社區的貢獻，因此我們應當鼓勵。但是那些認為“商業”等同於“非自由”的人們傾向於認為“自由商業”的組合是自相矛盾的並且否定其可能性。讓我們小心對待，不要以這種方式使用“商業”一詞。

¹Dan Farber, “Oracle’s Ellison Nails Cloud Computing”, 2008 年九月 26 日, http://news.cnet.com/8301-13953_3-10052188-80.html。

2.7.7 補償（Compensation）

當與版權聯繫在一起的時候談論“補償作者”將會帶來以下兩種假設：（1）版權是以作者之名而存在的；以及（2）每當我們閱讀任何東西的時候，我們已經欠下了作者一筆債務並且必須補償。第一個假設是錯誤的¹，而第二個假設是不可容忍的。

而“補償版權持有人”的鬼話在此基礎上又附加了一層詐騙：您很可能認為這是在補償作者，儘管在少數情況下確實是這樣，但是絕大多數情況下這是在補償出版商，正是那些出版商對我們強行施加了不平等的法律。

2.7.8 消費（Consume）

消費是指我們對食物所做的事情：當我們消化了它以後，它就不再作為食物繼續存在。作為類比，我們可以將相同的詞語用於那些我們可以將其消耗殆盡的其他產品。將其用於耐用品諸如衣物或電器是一種涵義的延伸。然而，如果將其應用於發表的作品（計算機程序、保存於盤片或文件中的唱片、保存於紙上或文件中的書籍），這些作品的本質意義是永續保存並且可以被無限次地運行、播放或閱讀，這顯然就是一種錯誤。播放唱片或者運程序這些行為並沒有消費它們。

“消費”一詞是與不可複製的實物產品的經濟層面相關聯的，而引導人們在毫無察覺的情況下將這一結論套用到可複製的數字作品上——這種錯誤正是私有軟件開發者（以及其他出版商）所熱切希望並且積極推動的。他們的這種扭曲的觀點在一篇 Business Insider 文章中暴露無遺²。這篇文章還將出版的作品稱為“內容”（content）。

與我們“消費內容”相關的狹隘想法為諸如數字千年版權法案（DMCA）這樣的法律奠定了基礎，它們禁止用戶破解數字設備中的數字限制管理

¹參見《對版權的錯誤解讀——一系列錯誤》（p. 137）一文以獲得關於此問題的更多信息。

²Lara O'Reilly, “A Former Googler Has Declared War on Ad Blockers with a New Startup That Tackles Them in an Unorthodox Way”, 2015年六月18日, <http://uk.businessinsider.com/former-google-exec-launches-sourcepoint-with-10-million-series-a-funding-2015-6?r=US&IR=T>。

(DRM)。如果用戶認為他們對這些設備所做的事情是“消費”，他們就可以將這種限制視為自然的要求。

它還鼓勵人們利用 DRM 對數字唱片的“流媒體”服務進行限制，這種形式也可以認為適用於“消費”這一概念。

為何這種不合理的用法廣泛流傳？有人也許認為這一概念過於高深；如果它吸引了您，用強有力的理由來拒絕它就會顯得更為高深。其他人也許是出於（他們自己的或是其僱主的）商業利益。他們在各種有聲望的地方對這一概念的使用將會給人以它是“正確”的錯覺。

說“消費”音樂、小說或是任何其他藝術作品就是將它們視為產品而非藝術。如果您不想散播這種態度，您應當努力拒絕對它們使用“消費”這一概念。我們建議您說某人“體驗”一件藝術作品或者一件表達觀點的作品，以及某人“使用”一件實用作品。

2.7.9 消費者 (Consumer)

當“消費者”一詞被用於指代計算機用戶的時候，它被附加了一種我們應當拒絕的假設。其中一些假設來自於使用程序就是“消費”程序這一理念（參見上一條目），這將會引導人們將那些從不可複製的實物產品中得出的結論也強行套用到可複製的數字作品上來。

此外，將軟件用戶描述為“消費者”指的是這樣一種境地，人們被限制為只能在“市場”中可以買到的“產品”中做出選擇。這裡沒有這種理念的容身之地，即用戶可以對程序所做的事情直接行使控制權¹。

為了描述那些並不被限制為被動使用作品的人們，我們建議使用“個人”或者“公民”而非“消費者”。

“消費者”一詞所帶來的問題已經在此前有所註釋²。

¹ 參見《如今自由軟件更加重要》(p. 32) 一文以獲得關於此問題的更多信息。

² Owen Hatherley, “Be a User, Not a Consumer: How Capitalism Has Changed Our Language”, 2013 年八月 11 日, <http://theguardian.com/commentisfree/2013/aug/11/capitalism-language-raymond-williams>。

2.7.10 內容（Content）

如果您想要描述一種舒適和滿足的感覺，您一定會說“content”，但如果將這個詞語用作名詞以指代具有作者權的出版物和作品，這將服務於一種您可能很想避免的態度，這種態度將它們視為商品，其目的是裝滿包裝盒並且用於賺錢。事實上，這貶低了作品本身。如果您不同意這種態度，您可以稱之為“作品”或者“出版物”。

那些使用“內容”一詞的人們通常是那些以作者（它們稱之為“創造者”，creator）之名攫取更多版權權力的出版商。“內容”一詞暴露了它們對於這些作品及其作者的真實態度（參見 Courtney Love 致 Steve Case 的公開信¹並且在頁面中搜索 content provider（內容提供商）。哎呀，Love 女士沒有注意到“知識產權”一詞也是帶有偏見和欺騙性的²）。

然而，只要其他人仍然使用“內容提供商”這一概念，持不同政見者也可以稱他們自己為“惡意內容提供商”。

“內容管理”（content management）這一短語的涵義空泛性無出其右。“內容”指代“某種信息”，而“管理”在此上下文中指代“對它們做一些事情”。因此某種“內容管理系統”是一種用於對某種信息做出某種事情的系統。幾乎所有計算機程序都符合這個定義。

在大多數情況下，這一概念指代用於更新網站頁面的系統。對於這種定義，我們推薦“網站修改系統”（WRS，web site revision system）這一概念。

2.7.11 創作共用許可的（Creative Commons Licensed）

對於一篇作品的許可證，最重要的方面是它是否自由。創作共用（CC）發佈了七種許可證；其中三種（CC BY、CC BY-SA 和 CC0）是自由的，而其他幾種則是非自由的。因此，將一篇作品描述為“創作共用許可的”未能

¹美國搖滾音樂家 Courtney Love 於 2000 年五月 16 日在數字好萊塢在線娛樂會議上的演講的未編輯抄本可以在此找到：http://www.salon.com/2000/06/14/love_7/。

²參見《您說過“知識產權”嗎？這是一種迷惑的幻景》（p. 99）以及本文的“知識產權（Intellectual Property）”一節（p. 118）獲知這一問題的原因。

說明它是否是自由的，並且暗示這個問題並不重要。這一論述也許是準確的，但它對於自由的省略是有害的。

為了鼓勵人們關注最重要的區別，務必具體指定哪種創作共用許可證被使用，例如“使用 CC BY-SA 許可證”。如果您不知道一篇特定的作品應該使用哪種許可證，您需要設法獲知這一點才能下定結論。

2.7.12 創造者 (Creator)

當“創造者”一詞應用於作者時，這是在暗含著將他們比作神（造物主，deity）。這一概念被出版商所使用，以便將作者的道德高度提升到普通人之上，進而以此為理由賦予他們更多的版權權力，然後出版商就能夠以作者之名來行使這種權力。我們建議您仍然說“作者”。然而，在很多情況下，“版權持有人”才是您所真正想要表達的。這兩個概念不是等效的，“版權持有人”通常不是作者。

2.7.13 數字商品 (Digital Goods)

當“數字商品”這一短語被應用於具有作者權的作品副本時，這將它們與實物商品等同起來了——這些實物商品是不可複製，並且因此必須被量產並且銷售。這種比喻鼓勵人們將他們用於評估實物商品的觀點和直覺也套用到評估軟件或者其他數字作品上來。這種比喻同樣將問題限定在經濟方面，其膚淺和有限的價值並不包括自由和社區。

2.7.14 數字鎖 (Digital Locks)

“數字鎖”這一短語被某些批評者用於指代數字限制管理 (DRM)。這一短語的問題在於它未能批判 DRM 的危害。而那些接受這一概念的人們未能對其進行透徹的思考。

鎖並不一定是壓迫性的或者壞的。您可能擁有很多把鎖，以及它們的鑰匙或代碼；您可能覺得它們有用或是會帶來麻煩，但它們並未壓迫您，

由於您可以自己打開或鎖上它們。類似地，我們發現加密¹對於保護我們的數字文件是無價之寶。這也是一種數字鎖，並且您擁有其控制權。

而 DRM 就像是由其他人為您安置的鎖，他們拒絕給您鑰匙——換言之，就像手銬。因此，對於 DRM 的恰當的比喻是“數字手銬”而非“數字鎖”。

一系列反對 DRM 的運動為其選擇了不明智的術語“數字鎖”；為了讓事情重回正軌，我們必須堅持改正這一錯誤。FSF 可以為一場反對“數字鎖”的運動提供支持，如果我們同意其基本立場；然而，當我們表示我們的支持的時候，我們將會引人注目地將該短語更改為“數字手銬”，並且解釋為何如此稱呼。

2.7.15 數字版權管理（Digital Rights Management）

“數字版權管理”（簡寫為 DRM）是指那些被設計用於對計算機用戶強行施加限制條件的技術機制。其中“權利”（rights）一詞的使用是一種鼓吹，其用意是誘使您在不經意間用那些施加這些限制的少數人的觀點來看待這一問題，並且忽略被強行施加了這些限制的公眾。

較好的替代短語包括“數字限制管理”（Digital Restrictions Management）和“數字手銬”（digital handcuffs）。

請您簽名支持我們致力於廢除 DRM 的運動，它位於 DefectiveByDesign.org。

2.7.16 生態系統（Ecosystem）

我們不建議將自由軟件社區或是任何人類社區描述為“生態系統”，由於這一短語暗示倫理評價的缺失。

“生態系統”這一短語暗含著建議這樣一種不加批判地服從態度：不要

¹Cory Doctorow, “Encryption Won’t Work If It Has a Back Door Only the ‘Good Guys’ Have Keys To”, 2015 年五月 1 日, <http://theguardian.com/technology/2015/may/01/encryption-wont-work-if-it-has-a-back-door-only-the-good-guys-have-keys-to->。

問什麼應該發生，只要學習並理解發生了什麼。在生態系統中，一些機體吃掉其他一些機體。在生態學中，我們從不追問一隻貓頭鷹吃掉一隻老鼠或是一隻老鼠吃掉一顆種子是好是壞，我們只需觀察到它們這樣做。物種種群的增長或衰退基於它的生存條件；這無所謂對錯，這只是一種生態現象，即使到了某一物種滅絕的程度。

與之相反，對其生存環境持有倫理立場的人們可以決定保護那些如果沒有人為干預就會消亡的東西——諸如公民社會、民主、人權、和平、公共衛生、穩定的氣候、潔淨的空氣和水、瀕危物種、傳統藝術……以及計算機用戶的自由。

2.7.17 自由和開源軟件（FLOSS）

FLOSS 意指“自由和開源軟件”（Free/Libre and Open Source Software），這一概念試圖在自由和開源之間尋求中立¹。如果保持中立就是您的最終目標，使用 FLOSS 是保持中立的最佳方式。但如果您想要表達您對自由的支持，就不要使用這種中立性的短語。

2.7.18 免費（For Free）

如果您想要表述一個程序是自由軟件，不要說它是可以“免費獲得”的。這一短語明確表示“零價格”。自由軟件關乎的是自由而非價格。

自由軟件副本通常可以免費獲得——例如，從文件傳輸協議（FTP）服務器下載。但自由軟件副本也可以存儲在只讀光盤（CD-ROM）上販賣；與此同時，私有軟件副本有時也可以作為促銷手段而免費獲得，並且一些私有軟件也通常免費提供給某些特定用戶。

為了避免混淆，您可以稱該程序可以“作為自由軟件（而獲得）”。

¹參見 <http://www.gnu.org/philosophy/floss-and-foss.html> 以獲得關於此問題的更多信息。

2.7.19 自由和開源軟件（FOSS）

FOSS 意指“自由和開源軟件”（Free and Open Source Software），這一概念試圖在自由和開源之間尋求中立¹。如果保持中立就是您的最終目標，使用 FLOSS 更好。但如果您想要表達您對自由的支持，就不要使用這種中立性的短語。

2.7.20 可自由獲得（Freely Available）

不要將“可自由獲得的軟件”用作自由軟件的同義詞。這兩個短語並不同等。如果任何人可以輕鬆獲得一份副本，就可稱其為“可自由獲得”。而自由軟件的定義是基於那些擁有其一份副本的用戶所擁有的自由。這是對於不同問題的回答。

2.7.21 免費軟件（Freeware）

不要將“免費軟件”用作自由軟件的同義詞。“免費軟件”這一短語在 20 世紀 80 年代常用於指代那些僅以可執行文件形式發佈的程序，其源代碼不可獲得。現在這一短語沒有任何特別的公認定義。

當使用英語以外的語言時，請避免直接借用英語短語諸如“free software”或“freeware”。最好能將 free software 翻譯為您所說的語言。（參見本書附錄 B (p. 341) 以獲得將 free software 無歧義地翻譯為多種語言的方式。）

通過使用您自己語言中的詞彙，您表明了真正想要表達的自由之意，而非只是對某些神祕的外國市場概念鸚鵡學舌。首次引用自由的概念可能會使您的同胞感到奇怪和不安，但是一旦他們發現它所表達的正是它所宣稱的涵義，他們將會真正地理解問題的實質。

¹參見上一條腳註。

2.7.22 贈送軟件（Give Away Software）

用“贈送”指代“將程序作為自由軟件發佈”是誤導性的。這種表達方式與“免費”一詞相似：它暗示問題的關鍵是價格而非自由。避免這種混淆的方式要說成“作為自由軟件發佈”。

2.7.23 谷歌（Google）

請不要將“谷歌”用作動詞以指代在互聯網上進行搜索。它只是眾多搜索引擎中的一種。我們建議使用“網絡搜索”這一短語。儘可能使用一種尊重您隱私的搜索引擎；DuckDuckGo 聲稱它不會跟蹤其用戶¹，儘管我們不能確認這一點。

2.7.24 黑客（Hacker）

黑客是指享受智慧樂趣的人們²——並不一定與計算機有關。在 20 世紀 60 至 70 年代，麻省理工學院（MIT）的自由軟件社區的程序員稱他們自己為黑客。大約在 1980 年，那些發現了黑客社區的記者們錯誤地將這一詞語用於指代“安全破壞者”。

請不要散播這種錯誤。那些破壞安全的人稱為駭客（cracker）。

2.7.25 知識產權（Intellectual Property）

出版商和律師喜好將版權描述為“知識產權”——這一短語也被應用於專利、商標以及其他更為晦澀的法律領域。這些法律之間共同特性如此之少，而它們之間的區別如此之大，以至於對它們進行廣義化是無謀的。最好還是具體地討論版權，或是專利，或是商標。

“知識產權”這一短語帶有一種隱藏的假設——即對於那些不同類的問題的思考方式應當基於同實物對象的類比，以及我們關於它們的概念應當

¹“DuckDuckGo Privacy Policy”，最後更新於 2012 年四月 12 日，<https://duckduckgo.com/privacy>。

²參見我的文章“On Hacking”，位於 <http://stallman.org/articles/on-hacking.html>。

基於同實物產權的類比。

當涉及複製的時候，這種類比抹殺了實物對象和信息之間的本質區別：信息可以被毫不費力地複製，而實物對象則不能。

為了避免不必要的偏見和混淆，最好能夠採取這樣一種嚴格的策略，不允許將“知識產權”作為討論甚至是思考的基礎。

虛偽地將這些權力稱為“權利”已經開始使得世界“知識產權”組織（WIPO）感到尷尬¹。

2.7.26 LAMP 系統（LAMP System）

LAMP 是指“Linux, Apache, MySQL 和 PHP”——一種常用於網絡服務器的軟件組合，除了此上下文環境中的“Linux”實際上是指 GNU/Linux 操作系統。因此，LAMP 應該改為 GLAMP，即“GNU, Linux, Apache, MySQL 和 PHP”。

2.7.27 Linux 系統（Linux System）

Linux 是 Linus Torvalds 於 1991 年開始開發的內核的名字，使用 Linux 內核的操作系統基本上是由 GNU 再加上 Linux 內核組成的。將整個操作系統稱為“Linux”是不公平並且引起混淆的。請將此完整的操作系統稱為 GNU/Linux，既尊重 GNU 計劃，也將整個操作系統和它的內核區分開來²。

2.7.28 市場（Market）

將自由軟件用戶，或者更普遍地將軟件用戶稱為“市場”是誤導性的。

這並不意味著自由軟件社區之中沒有市場的立錐之地。如果您擁有一項自由軟件的支持業務，您就可以擁有客戶，並且您可以在市場中同他們

¹Richard Stallman, “Public Awareness of Copyright, WIPO, June 2002”, 最後更新於 2014 年, <http://gnu.org/philosophy/wipo-PublicAwarenessOfCopyright-2002.html>。

²參見《Linux 和 GNU 操作系統》(p. 75) 一文以獲知 GNU/Linux 操作系統的更多歷史背景，由於它與本文所提到的命名問題相關。

進行交易。只要您仍然尊重他們的自由，我們就會祝願您在自己的市場中取得成功。

但是，自由軟件運動是一場社會運動而非一項業務，其致力於的成功並非某種市場上的成功。我們試圖通過賦予公眾自由來服務於公眾——而非通過競爭從對手手中奪取市場份額。將這場為自由而戰的運動同某種僅僅是為了商業上的成功而進行的行為等量齊觀，就是否認了自由的重要性，並且使得私有軟件合法化了。

2.7.29 貨幣化 (Monetize)

“monetize”一詞的恰當定義是“將某物用作貨幣”。例如，人類社會曾經將金、銀、銅、印刷的紙、某些種類的貝殼、大塊的岩石等貨幣化。然而，我們現在見到一種趨勢正在將這個詞語用作另一種用途，即“將某物用作盈利的基礎”。

這種用法將利潤置於首要位置，而將用於盈利的東西置於次要位置。將這種態度用於軟件工程是令人反感的，因為這將引導開發者將軟件變為私有的，如果他們得出的結論是使其成為自由軟件不能為其帶來足夠的利潤。

一家富有生產力並且合乎倫理的企業也可以盈利，但只要它將所有其他東西置於利潤之下，它就不再是合乎倫理的。

2.7.30 MP3 播放器 (MP3 Player)

在 20 世紀 90 年代後期，製造便攜式、固態存儲的數字音頻播放器成為可能。其中的大部分支持受專利限制的 MP3 編碼解碼器 (codec)，但這並不是全部。某些播放器支持不受專利限制的音頻編碼解碼器諸如 Ogg Vorbis 或者自由無損音頻編碼解碼器 (FLAC)，還有些播放器甚至完全不支持 MP3 編碼的文件，這顯然是為了迴避這些專利。將這些設備稱為“MP3 播放器”不僅僅是引起混淆的，它也賦予了 MP3 額外的好處，而這正是我們應當拒絕的。我們建議使用“數字音頻播放器”或者更簡單的“音

頻播放器”，如果上下文允許如此簡略。

2.7.31 開放（Open）

請不要將“開放”或者“開源”用作自由軟件的替代詞語。這些短語基於不同的價值觀代表不同的立場¹。自由軟件是一場政治運動；開源只是一種開發模式。

當指代開源的立場時，使用它的名字是恰當的；但不要為我們或我們的作品貼上“開源”的標籤——那將會使人們誤認為我們也持有那樣的立場。

2.7.32 個人計算機（PC）

使用 PC 這一縮略詞指代某一特定類型的計算機硬件是可以的，但是請您不要用它來暗示該計算機必須運行微軟 Windows 操作系統。如果您在同一臺計算機上安裝 GNU/Linux，它仍然是一臺 PC。

建議使用 WC 來稱呼一臺運行 Windows 的計算機。

2.7.33 Photoshop（PS）

請不要將 photoshop（PS）用於動詞以在普遍意義上指代任何照片處理或圖像編輯。它只是一款特定的圖像編輯軟件的名字，我們應當儘量避免使用它，由於它是私有軟件。有很多自由軟件可用於圖像編輯，例如 GIMP²。

2.7.34 盜版（Piracy）

出版商通常將不被它們批准的複製行為稱為“盜版”。通過這種形式，它們暗示這在倫理上與在外海攻擊船隻以及綁架謀殺船上的人一樣壞。基

¹參見《為什麼說開源漏掉了自由軟件的要點》（p. 90）以獲得完整解釋。

²參見 <http://directory.fsf.org/wiki/GIMP>。

於這樣一種鼓吹，它們在世界上的大部分地區騙取到了法律支持以便在絕大多數（有時是全部）環境下禁止複製（它們仍在繼續施壓以使得這些禁令更加完整）。

如果您不認可未經批准的複製行為就像綁架謀殺的理念，您可能也不願意使用“盜版”一詞來描述此行為。中立性的短語諸如“非授權複製”（或者“被禁止的複製”，用於此種行為不合法的情況）可作為替代。我們中的一些人甚至傾向於使用某種褒義詞，諸如“與他人分享信息”。

一位主持了版權侵犯審判的美國法官認可“盜版”和“盜竊”屬於誹謗性的詞語¹。

2.7.35 PowerPoint (PPT)

請不要使用 PowerPoint 指代任何類型的幻燈片演示文稿。它只是一款特定的用於製作演示文稿的私有軟件的名字。為了您的自由，您應該只用自由軟件來製作演示文稿。推薦的可選工具包括 T_EX 的 beamer 類以及 OpenOffice.org（或 LibreOffice——譯者注）的 Impress。

2.7.36 保護 (Protection)

出版商的律師喜愛使用“保護”一詞來描述版權。這個單詞暗示防止破壞或者避免痛苦；因此，它鼓勵人們去同情那些通過版權得利的版權持有人和出版商，而非那些受到版權限制的用戶。

避免說“保護”而改用中立性的詞語是容易做到的。例如，與其說“版權保護持續很長時間”，不如說“版權持續很長時間”。

類似地，與其說“受版權保護”，不如說“被版權覆蓋”或者更簡單的“具有版權”。

如果您想要批判版權制度而非只是想保持中立，您還可以使用“版權

¹Ernesto Van der Sar, “MPAA Banned from Using Piracy and Theft Terms in Hotfile Trial”, 2013 年十一月 29 日, <http://torrentfreak.com/mpaa-banned-from-using-piracy-and-theft-terms-in-hotfile-trial-131129>。

限制”這一短語。因此，您可以說“版權限制持續很長時間”。

“保護”一詞還被用於描述惡意功能。例如，“複製保護”是一種干擾複製行為的功能。從用戶的觀點來看，這是一種阻礙。因此我們將此惡意功能稱為“複製阻礙”。它更多地被稱為數字限制管理（DRM）——參見 Defective by Design 運動，它位於 DefectiveByDesign.org。

2.7.37 合理與無差別(RAND, Reasonable and Non-Discriminatory)

一些標準化組織推廣那些禁止自由軟件實現的，受專利限制的標準，這些組織通常擁有這樣的專利許可獲得政策，它要求符合此標準的軟件的每份副本支付一筆固定的專利費。它們通常將這樣的專利許可條款稱為 RAND，即“合理與無差別”。

這一短語洗白了一系列通常既不合理也並非無歧視的專利許可條款。確實，這些許可條款並不歧視任何特定的個人，但它們確實是在歧視自由軟件社區，這使得它們成為不合理的。因此，RAND 的一半是欺騙性的，而另一半是帶有偏見的。

標準化組織應該認識到這些許可條款是具有歧視性的，並且放棄使用“合理與無差別”或者 RAND 來描述它們。在它們這麼做之後，那些不希望加入這種“洗白”說法的政策制定者才能拒絕這一短語。僅僅因為那些持有大量專利的商業公司使其變得普遍流行就接受並使用這一短語，將會允許那些商業公司支配您所表達的觀點。

我們建議使用“僅限統一付費”（uniform fee only）或者簡寫為 UFO 來作為替代，這種描述是精準的，由於這些許可條款的唯一條件就是統一的權利金。

2.7.38 軟件即服務（SaaS, Software as a Service）

我們曾經說“軟件即服務”是一種不公，但我們隨後發現人們對於哪些活動屬於 SaaS 的理解還存在諸多分歧。於是我們轉為使用一個新的短

語，“作為軟件替代品的服務”，即 SaaS。這個短語有兩個優點：由於它從未被使用過，因此我們的定義是其唯一定義；另外它解釋了這種不公包括哪些方面。

參見《服務器真正是在為誰服務?》(p. 302) 一文以獲得關於此問題的更多討論。

在西班牙語中，我們繼續使用 “software como servicio” 這一短語，這是由於 “software como ser vicio”¹ 這個笑話太精妙絕倫了，以至於我們不捨得放棄它。

2.7.39 販賣軟件 (Sell Software)

“販賣軟件” 這一短語是含混不清的。嚴格地說，用一份自由軟件副本換取一筆資金是在販賣該軟件，這樣做也無可厚非。然而，人們通常將 “販賣軟件” 與對該軟件的後續使用的私有限制相關聯。您可以保持清醒並且拒絕混淆，通過說 “有償發佈軟件副本” 或者 “為軟件的使用施加私有限制”。

參見《售賣自由軟件》(p. 49) 一文以獲得關於此問題的更多討論。

2.7.40 分享經濟 (Sharing Economy)

“分享經濟” 這一短語不是一種用於指代諸如 Uber 或者 Airbnb 這樣的在人與人之間安排商業交易服務的好方式。我們使用 “分享” 這一詞語來指代非商業合作，包括對已發佈的作品的原始版本進行非商業性的再分發。將 “分享” 一詞的涵義延伸至包含這些商業交易的程度破壞了它的本意，因此我們不會在這種上下文中使用它。

用於描述諸如 Uber 的業務的更好的短語是 “計件工作服務經濟” (piecework service economy)。

¹“軟件，作為惡意的” 之意。

2.7.41 Skype

請不要將 Skype 作為動詞使用，用於從一般意義上指代任何類型的視頻通訊或者網絡電話。Skype 只是一種特定的私有軟件的名字，它會窺探用戶。¹如果您想要以一種同時尊重您的自由和隱私的方式進行網絡音視頻通話，您可以試用眾多自由的 Skype 替代品，詳情 https://libreplanet.org/wiki/Group:Skype_Replacement。

2.7.42 軟件工業（Software Industry）

“軟件工業”一詞鼓勵人們想象軟件總是由某種工廠開發並且送達“消費者”手中的。自由軟件社區已經證實了事情並非如此。軟件企業確實存在，並且不同的企業開發自由和/或私有軟件，但那些開發自由軟件的企業並不是像工廠那樣運營的。

“工業”一詞正在被軟件專利的倡導者用作一種鼓吹。他們將軟件開發稱為“工業”，然後試圖爭論這意味著它也應該受到專利壟斷權的支配。歐洲議會於 2003 年拒絕軟件專利，並且通過表決，將“工業”一詞定義為“實物商品的自動化生產”²。

2.7.43 源代碼模型（Source Model）

維基百科（Wikipedia）以一種使人混淆的方式使用“源代碼模型”這一概念。從表面看，它指的是程序的源代碼如何分發，但其文本將這一概念與開發方法論混淆起來。它將區分“開放源代碼”和“共享源代碼”作為答案，但它們是重疊的——微軟將後者作為一種市場概念以覆蓋一系列實踐，其中的一些也是“開放源代碼”的。因此，這一概念實際上未能傳達任

¹參見這份說明以獲得更多信息 <http://gnu.org/proprietary/proprietary-surveillance.html/#SpywareInSkype>。

²歐洲議會，“Directive on the Patentability of Computer-Implemented Inventions”，2003 年九月 24 日，<http://web.archive.org/web/20071222001014/http://www.swpat.ffii.org/papers/euoparl0309>。

何合乎邏輯的信息，但它卻為連篇累牘地把自由軟件描述為“開源”提供了機會。

2.7.44 終端 (Terminal)

移動電話和平板設備是計算機，並且人們應當被允許在它們上面使用自由軟件以進行他們的計算。將它們稱為“終端”暗示它們僅僅適合於連接到服務器，這是您操控自己計算的一種壞方式。

2.7.45 盜竊 (Theft)

那些支持過於嚴格和專制版權制度的支持者，通常使用諸如“盜版”或者“盜竊”這樣的詞語來指代侵犯版權的行為。這是一種偏見，但是它們希望您將其作為客觀事實而接受。

在美國法律體系中，版權侵犯並不屬於盜竊。適用於盜竊的法律並不適用於版權侵犯。專制性的版權制度的支持者正在請求權力機構——並且錯誤地解讀權力機構所說的話¹，闡明瞭什麼才能被恰當地描述為“版權盜竊”。

非授權複製在很多情況下（不是全部！）被版權法所禁止，但是被禁止並不意味著它是錯誤的。從普遍意義上說，法律並不定義正確與錯誤。法律充其量只能試圖實施公正。如果法律（實現）不能適應我們的是非觀（具體），那麼這樣的法律就應該被改變。

一位主持了一場版權侵犯審判的美國法官認可“盜版”和“盜竊”屬於誹謗性的詞語²。

¹為了駁斥他們，您可以援引這一真實案例：Harper Lee 起訴她的代理人涉嫌誘騙她將 *To Kill a Mockingbird* 一書的版權指認給他。

²Ernesto Van der Sar, “MPAA Banned from Using Piracy and Theft Terms in Hotfile Trial”, 2013 年十一月 29 日, <http://torrentfreak.com/mpaa-banned-from-using-piracy-and-theft-terms-in-hotfile-trial-131129>.

2.7.46 信任計算（Trusted Computing）

“信任計算”是其倡導者所起的名字，用於指代這樣一種重新設計計算機的陰謀，這使得應用程序開發者可以信任您的計算機將會服從他們而非您¹。從他們的角度看，這是“信任”；從您的角度看，這是“背叛”。

2.7.47 供應商（Vendor）

請不要使用“供應商”一詞一般性地指代任何開發或者打包軟件的人。很多程序被開發的目的是販賣其副本並且它們的開發者自然也是其供應商；這甚至也包括部分自由軟件包。然而，很多程序是由志願者或者並非致力於販賣副本的組織開發的。這些開發者不是供應商。類似地，只有部分 GNU/Linux 發行版的打包者是供應商，我們建議使用“提供者”（supplier）這一通用性的概念。

¹參見《您能夠信任您的計算機嗎？》（p. 279）以獲得關於此問題的更多信息。

第 3 部分

版權和不公

3.1 閱讀的權利

Copyright © 1996, 2002, 2007, 2009, 2010, 2014 Richard Stallman。
本文撰寫於 1996 年並以標題 “The Right to Read: A Dystopian Short Story”（閱讀的權利：一段反烏托邦的短故事）於 1997 年二月發表於 Communications of the ACM, vol. 40, n. 2。

本文摘自《第谷之路》（*The Road to Tycho*），這是一系列關於“月球人革命”的前奏的文章，於 2096 年發表於月球城。

對於 Dan Halbert 而言，他的“第谷之路”始於大學時代——當 Lissa Lenz 向他借用計算機的時候。她的計算機發生了故障，並且除非她能夠借到另一臺計算機，她將肯定不能完成中期項目。她不敢向其他任何人求助，除了 Dan。

這將 Dan 置於兩難境地，他有責任幫助她——但如果將自己的計算機借給她，她將有機會閱讀他的書籍。拋開如果您允許其他人閱讀您的書籍，您將被處以很多年監禁這一事實，這種想法首先使他感到震驚。與其他任何人一樣，他從小學開始就被這樣教育：分享書本是齷齪並且非法

的——只有盜版者才會去做這些壞事。

並且，他不被 SPA——軟件保護權力機構——抓住的機會並不大，在他的計算機軟件課上，Dan 知道了每本書都帶有一個版權監視器，它將會將此書於何時何地何人被何人閱讀的信息上報給中央授權許可機構（該機構對這些信息的利用不僅僅是為了抓住閱讀盜版者，也會將個人閱讀興趣偏好出賣給圖書零售商）。當他的計算機下一次聯網時，中央授權許可機構將會發現這一切。他作為計算機擁有者將會受到最嚴厲的處罰——由於他未能盡力阻止這種犯罪。

當然，也許 Lissa 本意上並不一定要閱讀他的書籍，她有可能只是想借用計算機來編寫她的中期彙報。但是，Dan 知道她來自於中產階級家庭，並且很難負擔得起學費，遑論她的各種閱讀費用。閱讀他的書籍可能是使她能夠順利畢業的唯一出路。他理解這種情形；他自己也必須借錢以支持其所需的科研論文閱讀費用（這些費用的 10% 將會被付給撰寫了這些論文的科研人員；由於 Dan 致力於他的學術事業，他可以指望自己的科研論文被頻繁引用，這樣他就可以賺到足夠的錢來支付這筆閱讀費用）。

其後，Dan 將會了解到，以前曾經有一段時間，任何人都可以去圖書館閱讀期刊文章甚至是圖書而無需支付任何費用。那時也有獨立的學者，他們可以閱讀數千頁的文獻而無需得到政府圖書館的批准。但是，在 20 世紀 90 年代，商業性和非盈利性的期刊出版者都開始徵收訪問費用。直到 2047 年，提供對學術文獻的免費公開訪問的圖書館已經成為一種灰暗的記憶。

當然，也有一些方式可用於繞過軟件保護權力機構（SPA）和中央授權許可機構的審查。它們本身是非法的。Dan 曾經有一位計算機軟件專業同學，Frank Martucci，此人得到了一種非法的調試工具，並且將其應用於在閱讀時繞過版權監視器的代碼。但是，他將此事告知了太多的朋友，其中之一將他告發至 SPA 以獲取一份獎賞（債臺高築的學生很容易被利誘叛變）。2047 年，Frank 被捕入獄，並非由於盜版閱讀，而是由於非法擁有一份調試工具。

其後，Dan 將會知道曾經有一段時間，任何人都被允許擁有調試工具，

那時甚至有可以從光盤（CD）獲得或者從網絡上下載得到的自由調試工具。但是，隨著普通用戶將其用於繞過版權監視器，最終一位法官裁決非法目的成為了調試工具在實際中的主要用途。這意味著它們是非法的，調試工具的開發者都被捕入獄。

當然，程序員們仍然需要調試工具，但在 2047 年，調試工具廠商限量發佈它們，並且僅對得到政府官方授權許可與擔保的程序員分發。Dan 用於計算機軟件課上的調試工具位於一道特殊的防火牆後面，使得它只能被用於課堂練習。

從前，通過安裝某種修改過的操作系統內核來繞過版權監視器也是可能的。Dan 最終將會了解到這些自由內核，甚至整個自由操作系統。這些東西直到世紀之交前後都還存在。但是，它們不僅像調試工具那樣是非法的——甚至您即使擁有它們也不能安裝，因為您不知道您計算機的 root 口令。而美國聯邦調查局（FBI）或者微軟客服都不會告知您那個口令。Dan 得出結論，他不能簡單地將計算機借給 Lissa，但他又不能拒絕幫助她，因為他愛著她。每次和她說話的機會都會使他充滿愉悅。而她選擇向他求助，這意味著她也愛著他。

Dan 做出了一些更加不可想象的舉動來解決這種兩難境地——他將計算機借給了她，並且將他的口令告知了她。通過這種方式，如果 Lissa 閱讀了他的書籍，中央授權許可機構將會認為實際上是他在閱讀這些書。這仍然是一種犯罪，但是 SPA 不會自動發現這一點，除非 Lissa 將他舉報。

當然，如果學校最終發現他將自己的口令給了 Lissa，這將使得他們二人的學生身份終結，不論她使用這個口令做了什麼。學校的政策是這樣的，任何對其監控學生如何使用計算機的行為的干涉是懲戒行為的底線，不論您是否做了任何有害的事——這種所謂的攻擊行為使得學校管理人員難於對您進行監控。他們認為這意味著您正在做一些其他的被禁止的事情，並且他們不需要知道這些事情到底是什麼。

學生們通常不會因此被開除——不是直接被開除。而是他們將會被禁止使用學校的計算機系統，這將使他們不可避免地不能通過他們的任何課程。

其後，Dan 將會知道，這種大學政策只是始於 20 世紀 80 年代，那時大學生們開始大規模使用計算機。在此之前，大學一直保持著另一種學生懲戒制度；它們將會懲罰那些有害的行為，而非那些僅僅是引起懷疑的行為。

Lissa 沒有向 SPA 舉報 Dan，而他關於幫助她的決定使得他們終成眷屬。這也促使他們質問他們在兒時曾被教給的關於盜版的東西。這對夫婦開始閱讀關於版權的歷史的東西，關於蘇聯以及它對複製分享的限制，甚至是關於最初的美國憲法。他們移居月球，在那裡他們遇到了其他同樣選擇逃離 SPA 魔掌的人們。當第谷於 2062 年爆發起義的時候，普遍的閱讀權利立即成為這場起義的核心訴求。

3.1.1 作者註記

- 本文據說應該是由其他人在未來撰寫的一篇歷史文章，描述了 Dan Halbert 在一種專制社會下的青年生活，這種專制社會是由那些將“盜版”作為宣傳口號的敵人創立的。因此本文使用那種社會下的用語。我嘗試從今天的視角呈現它，使其看起來更具壓迫性。有關“盜版”可參考《應避免使用（或慎用）的詞語》一文的“盜版（Piracy）”一節（p. 121）。
- 以下註記自本文初次發表之後經過數次更新。

閱讀的權利是一場於今天打響的鬥爭，儘管可能需要經過 50 年的時間才能使我們現在的生活方式成為灰暗的記憶，上文所述的大部分特定法律和實踐已經被提議，它們中的許多已經在美國和其他地區被通過併成為正式的法律。在美國，1998 年的數字千年版權法案（DMCA）為限制閱讀與出借計算機化的圖書（及其他形式的作品）建立了法律基礎。歐盟在 2001 年通過的一項版權指導意見中實施了類似的限制。在法國，隨著“信息社會的作者權利與相關權利法案”（DADVSI）於 2006 年通過，僅僅是擁有 DeCSS，一款用於解密數碼多功能影音光盤（DVD）的自由軟件，的一份副本即被視為犯罪。

在 2001 年，由迪士尼贊助的參議員 Hollings 提出了一項稱為“安全系統與標準認證法案”（SSSCA）的法律，其要求每臺新計算機帶有強制性的限制複製設備並且禁止用戶繞過它。繼 Clipper 芯片及類似的美國政府關於密鑰保管的提案之後，這顯示了一種長期趨勢：計算機系統越來越傾向於賦予不在場者對該計算機系統的實際使用者進行有效控制的權力。SSSCA 隨後更名為“消費者寬帶和數字電視促進法案”（CBDTPA）這一不可拼讀的名稱，它被評註為“消費但不要嘗試編程法案”（Consume But Don't Try Programming Act）。

隨後不久，共和黨人控制了美國參議院。他們與好萊塢的聯繫不如民主黨人那樣密切，因此他們沒有強行推動這些提案。現在民主黨人重新獲得控制權，我們所面臨的威脅進一步升高。

在 2001 年，美國開始試圖利用它所倡議的美洲“自由貿易”區（FTAA）條約來向整個西半球強制施加相同的規則。FTAA 是一種所謂的“自由貿易”條約，它們實際上是被設計用於賦予商業企業更多凌駕於民主政府之上的權力；強制推行諸如 DMCA 的法律正是這種精神的典型體現。而 FTAA 事實上由時任巴西總統盧拉否決，此人拒絕接受 DMCA 及其他類似法規的條件。

此後，美國通過一系列“自由貿易”協定將類似的條件強行施加給諸如澳大利亞和墨西哥等國，並且通過另一紙條約，中美洲自由貿易協定（CAFTA），將其強行施加給哥斯達黎加等國。時任厄瓜多爾總統科雷亞拒絕與美國簽署一項“自由貿易”協定，但我聽說厄瓜多爾已經於 2003 年採納了類似於 DMCA 的法案。

這個故事中的某一種理念直到 2002 年還未在現實中被提議，這種理念是由 FBI 和微軟掌控您的個人計算機的 root 口令並且不會讓您得到它們。

這一陰謀的倡導者將其命名為諸如“信任計算”或者“Palladium”。我們稱之為“背叛計算”¹，由於其效果是使您的計算機聽命於商業公

¹參見《您能夠信任您的計算機嗎？》（p. 279）一文以獲得更多關於“信任計算”的信息。

司，並且到了不再聽命於您甚至是違抗您的地步。這在 2007 年作為 Windows Vista 的一部分被實施¹；我們可以預見蘋果也會做出類似的事情。在這種陰謀中，由計算機制造商掌控其祕密代碼，但 FBI 對它同樣是唾手可得。

確切地說，微軟所掌控的並不是傳統意義上的口令；由於無人能夠在終端上輸入它。與之相反，它是一種數字簽名和加密密鑰，它對應著存儲在您的計算機上的另一個密鑰。這將允許微軟以及任何潛在與微軟合作的網站對於用戶能夠在其計算機上做的事情進行完全控制。

Vista 同樣賦予了微軟額外的權力；例如，微軟可以強制安裝更新，它可以命令所有運行 Vista 的計算機拒絕運行某一特定的設備驅動程序。Vista 所施加的諸多限制的主要目的是強制實施用戶所不能克服的數字限制管理（DRM）。DRM 的威脅正是我們發起 Defective by Design 運動的原因 DefectiveByDesign.org。

當這段故事初次編寫之時，SPA 威脅小型互聯網服務供應商（ISP），要求它們允許 SPA 監控所有用戶。大多數 ISP 在受到威脅的時候都妥協了，由於它們打不起官司。一家位於加利福尼亞州奧克蘭的 ISP，Community ConneXion 拒絕了這一要求，並且確實被起訴了。SPA 隨後撤訴，但它得到了 DMCA 的支持，這賦予了它們一直渴求的權力。SPA，實際上是軟件發行商協會之意，由商業軟件聯盟（BSA）取代了其類似於警察的職責。在今天，BSA 不是一支官方警察力量，但它卻在以非官方的方式行使這種權力。通過使用讓我們想起前蘇聯的手段，它邀請人們舉報其同事和朋友。於 2001 年在阿根廷發生的一起 BSA 恐怖行動幾乎毫無遮掩地威脅人們：分享軟件將被處以酷刑。上文描述的大學安全政策並非毫無實據。例如，芝加哥地區的某大學的一臺計算機在登錄時顯示了以下信息：

本系統僅供授權用戶使用。對於任何未經授權或者超出其授權範圍使用本系統的個人，其在本系統上進行的所有活

¹參見 <http://badvista.fsf.org/> 以瞭解我們反對 Windows Vista 的運動。

動將被監控並且記錄。在對任何人不當使用本系統進行監控或者系統維護期間，授權用戶的行為也可能被監控。任何使用本系統的個人必須明確同意接受這種監控，並且此監控一旦發現可能指證非法行為或者侵犯大學規定的行為的證據，本系統可能將通過此監控發現的證據移交至大學權力機構和/或官方執法機構。

這是一種類似於美國憲法第四修正案的有趣方式：首先施壓以迫使幾乎所有人放棄他們的權利。

3.1.2 壞消息

關於閱讀的權利的鬥爭已經在進行中了，敵人已經有備而來嚴陣以待，而我們還沒有。因此這場鬥爭正在向著對我們不利的方向進行。自本文初次發表之日起發生的一些代表性的壞消息包括：

- 今天的商業電子書剝奪了讀者的傳統自由。參見《電子書的威脅》(p. 297) 一文以獲得更多信息；
- 某一“生物課本”網站¹允許您訪問其出版物，僅當您簽訂合同以保證不會將其借給任何人閱讀²。而出版商可以隨意廢除此合同；
- 電子出版物限制其用戶的自由³；
- 計算機中的圖書⁴：用於控制誰可以在一臺計算機上閱讀圖書和文檔

¹Nature America Inc., “Announcing Principles of Biology, an Interactive Textbook by Nature Education,” http://nature.com/nature_education/biology.html.

²Nature America Inc., “Principles of Science Privacy Notice,” accessed August 2015, <http://nature.com/principles/viewTermsOfUse>.

³參見 Don Clark 的文章 “Seybold Opens Chapter on Digital Books” (31 August 1999, <http://www.zdnet.com/article/seibold-opens-chapter-on-digital-books/>) 以獲知有關以電子形式發佈圖書以及影響到閱讀其副本的權利的版權問題的信息。

⁴“Microsoft Announces New Software for Reading on Screen,” 30 August 1999, <http://microsoft.com/en-us/news/press/1999/Aug99/SeyboldPR.aspx>.

的軟件。

如果我們想要阻止這樣的壞消息並且帶來一些好消息，我們需要組織起來並且進行鬥爭。自由軟件基金會（FSF）的 Defective by Design 運動已經為這場鬥爭拉開序幕；訂閱該運動的郵件列表以提供支持。並且加入 FSF 以資助我們的工作。

3.1.3 參考文獻

- 美國專利及商標局（USPTO），Intellectual Property and the National Information Infrastructure: The Report of the Working Group on Intellectual Property Rights, Washington, DC: GPO, 1995.（參見《您說過“知識產權”嗎？這是一種迷惑的幻景》（p. 99）一文以獲知為何“知識產權”這一短語是含混不清的並且永遠不應被使用）。
- Samuelson, Pamela, “The Copyright Grab,” *Wired*, January 1996, 4.01, http://wired.com/wired/archive/4.01/white.paper_pr.html.
- Boyle, James, “Sold Out,” *New York Times*, 31 March 1996, sec. 4, p. 15; 也可以在這裡找到：https://law.duke.edu/boylesite/sold_out.htm.
- 社論，*Washington Post*, “Public Data or Private Data,” 3 November 1996, sec. C, p. 6, <http://web.archive.org/web/20130508120533/http://www.interesting-people.org/archives/interesting-people/199611/msg00012.html>.
- 公有領域聯盟（Union for the Public Domain）——一個致力於抵抗並且逆轉版權和專利的權力過度擴張的組織。

3.2 對版權的錯誤解讀——一系列錯誤

Copyright © 2002, 2003, 2007, 2009–2011 自由軟件基金會。本文最初於 2002 年發表於 <http://gnu.org>

版權法中發生了一些奇怪而又危險的事情。在美國憲法環境下，版權存在的意義是使用戶獲益——用戶是指那些讀書、聽音樂、看電影或者運行軟件的人們——而非為了出版商或作者。然而，即使人們越來越傾向於拒絕並且反抗那些“為了他們自己的好處”而強行施加給他們的各種版權限制，美國政府仍然正在施加更多的限制，並且試圖以新的嚴厲懲罰措施來恐嚇公眾以迫使其服從。

那麼，版權政策是如何一步一步地走到與其宣稱的初衷背道而馳的地步的呢？而我們又如何才能使其重新回到與其初衷相符的正軌呢？為了解這些，我們應當首先審視美國版權法的根基：美國憲法。

3.2.1 美國憲法中的版權

當美國憲法被起草之時，作者們應當被賦予某種版權壟斷權這一想法被提出——然後又被拒絕了。我們國家的創始人採納了另一種假設，即版權並非作者們的一種自然的權利，而是以進步之名，人為地為他們做出的一種認可。憲法通過這一條款（第 I 章，第 8 條，第 8 款）許可了這樣一種版權體系：

（國會應當擁有權力）通過在一段限定的時間內保證作者和發明者對其作品或發明擁有專屬權利，以促進自然科學和實用技藝的進步。

美國最高法院已經一再強調，促進進步意味著保護那些版權作品的使用者的利益。例如，在 *Fox Film 起訴 Doyal* 的案例¹中，法庭說道：

¹Fox Film Corp. v. Doyal, 286 US 123, 1932.

美國的根本利益以及賦予（版權）壟斷權的主要目的是基於那些由公眾得來、通過作者的勞動獲得的普遍利益。

這一根本性的決定解釋了為何版權不是被憲法強制要求的，而只是作為一種選項而被許可——以及為何只應該在一段“限定的時間”內有效。如果版權是一種自然的權利，即某種作者們由於應當擁有它而擁有的東西，那麼沒有任何理由可以解釋為何可以在一定時期之後終止這種權利，就如同不能解釋為何每個人的房子在建成一定時期之後應當成為公共財產。

3.2.2 版權交易

版權體系通過為出版商和作者提供特權以保證其利益進而維持其自身運轉；然而，這麼做並非為了他們的利益。與之相反，它如此運作是為了改變他們的行為：即刺激作者們創作並且發表更多作品。事實上，政府是在以公眾的名義消費公眾的自然權利，作為交易的一部分為公眾帶來更多的發表作品。法學家們將這一概念稱為“版權交易”。就如同政府花費納稅人的錢財以購買一條高速公路或者一架飛機，區別只是在於此處政府花費的是我們的自由而非我們的錢財。

但是，這種版權交易對於公眾而言真的是一種公平的交易嗎？眾多其他交易方式都是可能的；哪一種方式最好呢？版權政策的每一個問題都是這個問題的一部分。如果我們對這個問題的本質產生誤解，我們將會傾向於對這些問題做出不恰當的抉擇。

憲法批准了將版權權力賦予作者。而實際上，作者們通常將其轉讓給出版商；因此實際上通常是出版商而非作者在行使這些權力從而得到大部分利益，儘管作者有時也能得到一小部分利益。因此通常只是那些出版商在遊說政府以進一步增加版權權力。為了更好地反映關於版權的事實而非那些鬼話，本文將會把出版商而非作者稱為版權權力的持有者，並且將有版權作品的使用者稱為“讀者”，即使對這些作品的使用並非只能是閱讀，這是由於“用戶”一詞過於遙遠和抽象。

3.2.3 第一個錯誤：尋求平衡

版權交易將公眾置於首要地位：公眾讀者的利益是其一端；出版商的利益（如果有）只是達成這一目的的一種方式。因此，讀者的利益和出版商的利益首先在質上是不對等的。對版權的目的的第一步錯誤解讀是將出版商的利益的重要性提升到與讀者的利益相同的級別上來。

通常有這樣的說法，美國版權法旨在在出版商和讀者的利益之間“尋求平衡”。那些援引這種解讀方式的人們將其呈現為一種對憲法中所陳述的基本立場的複述，換言之，它應當與所謂的版權交易等價。

但是，這兩種解讀其實有著天壤之別；它們不僅在概念上截然不同，在涵義上也是完全不同的。這種“平衡”的概念假設讀者和出版商的利益在重要性上的差別只是量上的，即我們應當分別給予它們多少權重，以及它們適用於哪些場合。“利益相關方”的概念通常被用於以這種方式描述這一問題。這種觀點抹殺了讀者和出版商的利益在質上的區別，而這種區別正是政府參與版權交易的基本點。

這種偷換概念的影響是深遠的，由於在版權交易中對公眾利益的強有力的保護——版權特權只能以讀者之名被合理化，而絕非以出版商之名被合理化這一理念——被這種“平衡”的解讀方式所拋棄了。由於出版商的利益也被視為其中一端，它也可以使版權特權合理化；換言之，“平衡”概念宣稱版權特權可以以公眾以外的某人之名被合理化。

從實踐角度講，這種“平衡”概念的結果是逆轉了在版權法中做出更改所需正當理由的負擔。版權交易將這種負擔置於出版商一方，他們需要說服讀者轉讓部分自由。而平衡概念逆轉了這種負擔，實事求是地說，由於對於出版商將會通過額外的特權得到好處這一點沒有異議。除非這對讀者造成的傷害能夠被證實，並且這種傷害大到了已經超出讀者所能得到的好處的程度，我們將會得出結論：出版商被賦予了它們要求的幾乎所有特權。

由於在出版商和讀者之間“尋求平衡”的理念否定了讀者本應擁有的首要位置，我們必須堅決反對。

3.2.4 針對什麼的平衡？

當政府為公眾購買任何東西之時，它是在以公眾的名義進行交易；它的責任是試圖得到最佳的可能交易方式——對公眾而言的最佳，而非對於交易中的另一方。

例如，當政府與建設公司簽署合同以修建高速公路時，政府應當致力於花費盡可能少的公眾資金。相關政府機構將會通過競爭性投標的方式來降低總價。

從實踐角度講，這一價格不可能為零，因為承包商不可能叫出那麼低的價格。儘管沒有獲得特別報酬的資格，它們在自由社會中仍然享有通常的公民權利，包括拒絕對其不利的合同的權利；即使最低競價也足以使某些承包商有利可圖。因此，這裡確實存在著某種平衡。但這並非由於利益雙方各自要求得到特殊報酬而達成的一種蓄意的平衡。這是一種在公眾目標和市場力量之間達成的平衡。政府試圖為駕車的納稅人獲得他們在自由社會和自由市場的大背景下可能得到的最佳交易。

在版權交易中，政府花費的是我們的自由而非我們的錢財。自由遠比金錢更加珍貴，因此政府對於理性而節約地花費我們的自由的責任更重於花錢的責任。政府絕不應該將出版商的利益與公眾的自由相提並論。

3.2.5 不是平衡，而是折中

將讀者利益與出版商利益進行平衡是對版權政策的錯誤判斷，但是，此處確實有兩種利益需要被權衡：關於讀者的兩種利益。讀者自身有其關於使用出版作品的自由的利益；取決於所處環境，他們還可能擁有通過某種激勵系統鼓勵發表作品的權利。

由於在討論版權問題時，“平衡”一詞被用於指代在讀者和出版商之間“尋求平衡”這一理念。因此，對於讀者自身的兩種利益使用“平衡”一詞

將會產生歧義¹。我們需要另一個術語。

通常意義上，當某一團體擁有兩種部分衝突的目標而不能兩全之時，我們稱之為“折中”。因此，與其說在兩個團體之間“尋求正確的平衡”，不如說“在花費與保留我們的自由之間尋求最佳的折中”。

3.2.6 第二個錯誤：最大化一項輸出

版權政策中的第二個錯誤包括將最大化——而非僅僅是增加——出版作品數量作為最終目標。第一個錯誤概念“尋求平衡”將出版商的地位提升到與讀者對等的位置，而這第二個錯誤則將它們的地位進一步提升到遠在讀者之上的位置。

當我們購買物品時，我們通常不會買空所有庫存或者只買最貴的型號。與之相反，我們將會為其他購買需求預留資金，對於任意特定的物品，只買我們所需要的，並且選擇一種能夠滿足需求的型號而非最高端的型號。根據報酬遞減原理，將我們的所有可用資金花在某一特定物品上很可能是一種低效的資源分配方式；我們一般會選擇將部分資金留作他用。

報酬遞減原理也適用於版權交易，如同它適用於任何其他購買行為。我們應當最先賣出的自由是那些使我們失去得最少的那部分自由，並且這部分自由的犧牲能夠換取對作品出版的最大激勵。隨著我們繼續賣出的自由越來越關乎自身核心利益，我們將會發現每一筆新的交易相對於上一筆都是更大的犧牲，同時它們所能帶來的出版活力增量越來越小。而當這種增量遠未減到零之前，我們就會說不值得為之付出不斷增加的代價；而後我們將會達成某種交易，其最終結果是增加了出版量，但並未達到可能的極限程度。

將最大化出版量作為最終目標將會預先拒絕所有這些更加明智、更加有利的交易方式——它強制規定公眾必須出賣幾乎全部自由來使用出版作

¹參見 Julian Sanchez 的文章 “The Trouble with ‘Balance’ Metaphors” (4 February 2011, <http://juliansanchez.com/2011/02/04/the-trouble-with-balance-metaphors/>) 以獲得關於“合理判斷與平衡權重之間的類比能夠怎樣以有害的方式限制我們的思考”這一問題的深入調查。

品，而僅僅是為了換取一點點出版量的增加。

3.2.7 具有欺騙性的最大化

在實踐中，將最大化出版量作為最終目標而全然不顧自由的代價這一理念是由一系列具有欺騙性的說法所支撐的。這種說法宣稱公眾對出版物進行復制的行為是非法、不公平、並且本質上錯誤的。例如，出版商將複製出版物的人們稱為“盜版者”，這種誹謗性的稱謂將與他人分享信息的行為與攻擊船隻的行為等同起來（這一誹謗性的稱謂最初被作者用於描述那些找到了以某種合法方式出版未經授權許可的版本的出版商；而它被出版商所使用的現代用法幾乎與其原意完全相反）。這種欺騙性的說法直接抹殺了版權的憲法基礎，而是如同表現美國法律體系中那些公認的傳統那樣呈現其自身。

“盜版”這一具有欺騙性的說法之所以被普遍接受，是由於它充斥於各種媒體，以至於極少有人意識到它到底有多麼重要。它是如此地強有力，因為如果由公眾進行復制是根本非法的，我們將完全不能對抗出版商強制要求我們犧牲自己的自由的行為。換言之，當公眾被威脅給出理由以證明為何出版商不應獲得某些額外權力之時，所有理由當中最重要的一條——“我們想要複製”——已經被預先否決了。

這封死了通過爭論反對與日俱增的版權權力的所有出路，除了一些細枝末節的問題。因此，今天對於日益強大的版權權力的反對幾乎只能援引一些旁枝問題，並且從來不敢援引本應作為一種合法公眾價值的再分發副本的自由。

從實踐角度講，出版量最大化這一最終目標使得出版商可以如此論述：“某種行為正在削減我們的銷量——或者我們認為它將會這樣——因此，我們假定它使得出版量下降了一定的未知數量，因此它應當被禁止。”我們將得到這樣一種不可容忍的結論：公眾的利益應當以出版商的銷量來衡量，但凡對大眾媒體公司有利的東西，對美國也都是好的。

3.2.8 第三個錯誤：最大化出版商的權力

一旦出版商得到了這樣的許可，即為了達到出版輸出最大化的政策目標可以不計任何代價，那麼下一步就是推斷出這需要賦予它們最大可能的權力——使得版權覆蓋於一篇作品任何想象到的使用方式，或者應用某些其他法律工具諸如“拆封包裝”許可證來實現同樣的效果。這一目標涉及廢除“合理使用”和“首次銷售權”，它已經被各級政府強制執行，從美國各州到各個國際機構。

這一步驟是錯誤的，由於嚴酷的版權規則阻礙了新的有益作品的創作。例如，莎士比亞從其他人於數十年前發佈的作品中借用情節用於自己的劇本，因此，如果今天的版權法在當時有效，他的劇本將成為非法。

即使我們真的想要最大可能的出版量而無視公眾為之付出的代價，最大化出版商的權力也是達成這一目標的錯誤方式。作為一種本意在於促進進步的方式，這種做法是自相矛盾的。

3.2.9 三大錯誤的後果

當前的版權立法趨勢是賦予出版商更寬泛的、持續時間更長的權力。而版權的概念基礎自從被這一系列錯誤所扭曲之後，幾乎沒有給讀者留下任何說“不”的權利。立法者口頭上大肆鼓吹版權服務於公眾這一理念，而實際上卻一直在賦予出版商們想要的任何東西。

例如，參議員 Hatch 在介紹 S.483 法案¹時如是說，這條於 1995 年通過的法案將版權的期限延長了 20 年。

我相信我們現在需要搞清楚這個問題：當前的版權期限是否足以保護作者的利益，以及與之相關的另一個問題：當前的版權保護期限是否足以繼續為作者創作新作品提供足夠的激勵。²

¹Congressional Record, S. 483, “The Copyright Term Extension Act of 1995,” 2 March 1995, pp. S3390–4.

²Congressional Record, “Statement on Introduced Bills and Joint Resolutions,” 2 March 1995, p. S3390, <http://gpo.gov/fdsys/pkg/CREC-1995-03-02/pdf/CREC-1995-03-02-pt1-PgS3390-2.pdf>.

這一法案延長了那些自 20 世紀 20 年代以後創作的已發表的作品的版權期限。這一改變是對出版商的慷慨贈予，而不可能為公眾帶來任何好處，由於現在不可能以任何方式逆動地增加當時的出版量。然而，這剝奪了公眾的一項在今天看來非常有意義的自由——再分發當時出版的作品的自由。注意，他們所鼓吹的“保護”¹一詞的用法中包含了三大錯誤中的第二項。

這條法案同時延長了將要被創作的作品的版權期限，對於出資聘用作者創作的作品，版權期限將會持續 95 年而非當前的 75 年。理論上這將會增加對創作新作品的激勵；但是，任何宣稱需要這項額外刺激的出版商應當被要求以未來 75 年預期的財務狀況表來證明這一要求的合理性。

不言而喻的是，國會並不會質疑出版商的論證：一項版權延長法律於 1998 年實施。它的官方名稱是 Sonny Bono 版權期限延長法案，它是以一位當年早些時候去世的支持者命名的。我們通常稱之為米老鼠版權法案，由於我們認為它的真實動機是防止米老鼠的形象的版權過期。Bono 的遺孀以他的名義完成了剩下的工作，並作了如下聲明：

事實上，Sonny 希望版權保護期限持續到永遠。我被一位工作人員告知這樣一種改變將會違反憲法。我現在邀請各位與我共同努力，以所有可能的方式強化我們的版權法。你們應當知道，還有 Jack Valenti²的提案建議版權的保護期限為“比永遠少一天”。也許委員會可以期待下一屆國會。³

美國最高法院隨後聽說一個案例，這一案例試圖在根本上推翻該版權法律，由於反動地延長版權期限未能符合憲法關於促進進步的終極目標。法庭以放棄其判決該問題的責任作為迴應；在版權問題上，對憲法只需要空口應酬。

¹參見《應避免使用（或慎用）的詞語》一文的“保護（Protection）”一節（p. 122）以獲知為何在與版權相關聯時應當避免使用“保護”一詞。

²Jack Valenti 是美國電影協會（MPAA）常任主席。

³Congressional Record, remarks of Rep. Bono, 7 October 1998, p. H9952, <http://gpo.gov/fdsys/pkg/CREC-1998-10-07/pdf/CREC-1998-10-07-pt1-PgH9946.pdf>.

另一條於 1997 年通過的法律使得為任何已發表作品複製達到一定數量的副本成為一項重罪，即使您將它們送給遵紀守法的朋友。而在此之前，這在美國甚至不是一項罪行。

一條更壞的法律，數字千年版權法案 (Digital Millennium Copyright Act, DMCA)，被設計為重新帶回一種當時被稱為“複製保護”的東西——現在稱之為數字限制管理 (DRM)¹——這是一種已經被用戶所唾棄的東西，它使得破解這些限制規則，甚至只是發表關於如何破解它們的信息成為犯罪行為。這條法律應當被稱作“媒體公司支配法案” (Domination by Media Corporations Act)，由於它事實上賦予了出版商編寫自家版權法的權力。它宣稱它們可以對作品的使用強行施加任何限制，並且這些限制將具有法律效力，如果該作品包含某種加密或者許可證管理器之類的強制實行措施。

用於支持該法案的一種論證是它可以實現一種現代條約以增加版權權力。該條約由世界“知識產權”組織 (WIPO)² 所大肆宣揚，這一組織被版權與專利持有人的利益所支配，並且得到了克林頓政府以施壓方式提供的支持；由於該條約只是強化了版權權力，至於它在任何國家是否有利於公眾利益尚存爭議。無論如何，這條法案的效力大大超出了這項條約所要求的。

圖書館曾經是反對此法案的重要來源，尤其針對該法案中禁止某種可被視為合理使用的複製行為這一方面。那麼出版商一方是如何迴應的呢？前代理人 Pat Schroeder，現在作為美國出版商協會 (AAP) 的說客如是說：“出版商與圖書館的訴求不共戴天”。由於圖書館的訴求只是維持部分現狀，人們可以以此作為迴應：出版商到底是怎樣活到今天的。

國會議員 Barney Frank 在一次與我和其他反對該法案的人士舉行會談時，展示了美國憲法中關於版權的觀點在多大程度上被無視了。他宣稱由犯罪刑罰所支撐的新的版權權力是急迫需要的，由於“電影行業感到擔憂”，同樣還有“音樂行業”和其他“行業”。我問他：“但是，這是為了公眾

¹ 參見 <http://gnu.org/proprietary/proprietary-drm.html> 以獲得關於這一問題的更多信息。

² 參見《您說過“知識產權”嗎？這是一種迷惑的幻景》(p. 99) 一文以獲知為何這一短語是有問題的。

的利益嗎？”他如此迴應：“你有什麼資格談公眾利益？那些富有創造性的人們不需要為了公眾利益而犧牲他們的權利！”在這裡“行業”被視同它所僱傭的“富有創造性的人們”，版權被視為它應有的權利，而憲法的本意則被徹底顛倒了。

DMCA 於 1998 年被實施。隨之，它宣稱合理使用在名義上尚屬合法，但它允許出版商禁止所有那些您可能對其進行合理使用的軟硬件。事實上，合理使用已被禁止。

基於這一法律，電影業對用於讀取和播放數碼多功能影音光盤（DVD）的自由軟件，乃至關於如何讀取它們的信息強行實施了審查。2001 年四月，普林斯頓大學 Edward Felten 教授受到來自美國唱片業協會（RIAA）的法律訴訟的威脅以撤銷一篇關於他從一種被提議用於限制對錄製音樂訪問的加密系統中所學到的東西的科研論文。

我們同樣開始見到那些剝奪了眾多讀者的傳統閱讀自由的電子書——例如，將書借給您的朋友，或是賣給二手書店，或是從圖書館借閱，或是無需將您的名字告知公司數據庫便可購書，甚至是再次閱讀某本書等等這些自由。加密的電子書通常限制了以上這些活動——您只能利用某種被設計用於限制您自由的特殊私密軟件來讀取它們。

我堅決不會購買任何一部這種加密、受限的電子書，並且我也希望您也能拒絕它們。如果一部電子書不能賦予您與紙版書相同的自由，就不要接受它！

任何人擅自發布可用於閱讀受限的電子書的軟件將面臨被起訴的風險。一位俄國程序員 Dmitry Sklyarov 於 2001 年訪問美國期間參加一次會議演講時被逮捕，由於此人在俄國編寫了一款這樣的程序，而這當時在俄國尚屬合法。現在俄國正準備通過一條法律以禁止這種行為，而歐盟最近已經採納了這樣的一條法律。

面向大量最終用戶銷售的電子書目前看來在商業上並不很成功，但這並非由於讀者選擇捍衛他們的自由；而是由於其他原因不受歡迎，例如計算機屏幕並不是一種適合於輕鬆閱讀的平面。我們不可能長期依賴這種好的缺陷來保護我們自己；用於促銷電子書的下一步嘗試是使用電子

紙——一種像書本一樣的東西，可以向其中下載加密、受限的電子書。如果這種像紙一樣的平面能夠提供比現在的顯示屏更有吸引力，我們將不得不捍衛我們的自由以便繼續擁有它。與此同時，電子書正在利基市場（niche market）中覓得商機：紐約大學（NYU）和其他牙科學校要求其學生以受限的電子書格式購買它們的教材。

媒體公司對此仍不滿足。2001 年，由迪士尼贊助的參議員 Hollings 提議了一條“安全系統標準與認證法案”（Security Systems Standards and Certification Act, SSSCA）¹，它要求所有計算機（及其他數字錄製與回放設備）必須帶有政府強制要求的限制複製系統。這是它們的最終目的，但它們日程上的第一件事是禁止任何可以調諧數字高清電視（HDTV）的設備，除非它被設計為使得公眾不可能對其進行“惡意篡改”（即以公眾自己的目的進行修改）。由於自由軟件是用戶可以修改的，我們在此首次遭遇了這樣一種被提議的法律，它將明確禁止自由軟件被用於某種特定任務。可以預見對於其他任務的自由軟件禁令也一定會隨之而來。如果美國聯邦通信委員會（FCC）採納這一規則，現存的自由軟件諸如 GNU Radio 將會被審查。

阻止這些法案和規則的通過和實施需要政治行動²。

3.2.10 尋找正確的版權交易

那麼，決定版權政策的恰當方式是什麼呢？如果版權是一筆以公眾為名進行的交易，它應當把服務於公眾利益放在至高無上的地位。政府在消費公眾自由的時候應盡的職責是隻去花費那部分必須花費的，並且要換取儘可能多的回報。最起碼的要求是，我們應當儘可能減少版權權力的程度，同時又能維持與之相適應的出版量級別。

¹由於它被更名為不可拼讀的 CBDTPA，一種有助於理解和記憶的解讀是“消費但不要嘗試編程法案”（Consume, But Don't Try Programming Anything），而其真正名稱是“消費者寬帶和數字電視促進法案”（Consumer Broadband and Digital Television Promotion Act）

²如果您真的想要提供幫助，我推薦以下網站：<http://defectivebydesign.org>, <http://publicknowledge.org>, and <http://eff.org>.

既然我們不能通過競標來找到自由的最小代價，如同我們在對待建設工程時那樣。我們怎樣才能找到正確的版權交易方式呢？

一種可能的方式是分階段減少版權特權並且考察其效果，通過觀察出版量是否發生了顯著減少以及何時發生，我們可以獲知為了達到公眾預期的目標所真正必需的版權權力是多少。我們必須基於真實的觀察來評估這一點，而非根據出版商所說將會發生什麼來進行判斷。由於它們會盡其所能做出誇張的悲觀預期，如果它們的權力以任何形式受到限制。

版權政策包含多種獨立的方面，它們可以被單獨評估。當我們找到了它某一政策方面所必需的最低限度以後，仍然可能繼續減少其他方面的版權限制，以維持理想的出版量水平。

版權的一個重要方面是它的持續時期，而現在的版權期限通常是以一個世紀作為數量級的。將這種對複製的壟斷權縮短到10年，從該作品發表日期開始計算可能是一種合理的第一步。而版權的另一方面，即覆蓋了創作其衍生作品的方面，可以持續稍長一段時間。

為何從作品發表之日開始計算？由於未發表的作品的版權並未直接限制讀者的自由；我們是否擁有複製一篇作品的自由並不重要，如果我們還沒有得到它的副本。因此，給予作者一段較長時間來發表作品並不會造成傷害。作者們（通常在作品發表之前確實擁有其版權）很少會僅僅為了推遲其版權終止的日期而選擇延遲發佈作品。

為何是10年？由於這是一種安全的提案；我們能夠以基於實踐的理由確信這一期限的減少並不會為當今的總體出版活力帶來什麼衝擊。在大部分媒體和流派當中，成功的作品只在短短的幾年之內能夠帶來豐厚利潤，並且即使是成功的作品通常也會在遠未到達10年期限之時絕版。即使是對於參考文獻類的作品，它們的有用生命週期可能長達幾十年，10年的版權期限也是足夠的：由於更新的版本還將正常出版，並且很多讀者將會願意購買仍然受到版權保護的當前版本而非複製一份已經處於公有領域中的，十多年前的舊版副本。

10年也許仍然比真正必需的時間要長；一旦這個問題得到解決，我們可以嘗試進一步縮短這一期限來對版權體系進行調整。在一次有關文學作

品版權慣例的研討會上，我提出了 10 年期限的提議，坐在我身旁的一位著名科幻作家表示強烈反對，他認為超過 5 年期限的任何限制條件都是不可容忍的。

但是，我們不必對所有類別的作品都採用相同的時間期限。維持版權政策的絕對平均對於公眾利益並非至關重要，並且版權法對於某些特殊應用場景和媒體類型已經有了很多例外條款。為每項高速公路工程都按照國內成本最高地區的最困難工程所必需的價格標準付費是愚蠢的，而為每種類型的藝術作品都按照我們所發現的對於任何一類作品所必需的自由代價中的最高標準來“花費”自由同樣愚蠢。

因此也許小說、字典、計算機程序、歌曲、交響樂、電影等等應當具有不同的版權期限，於是我們可以將任何一類作品的版權期限縮短到眾多待發表的同類作品所必需的程度。也許一小時以上的電影可以擁有 20 年的版權，考慮到創作它的開銷。在我本人的領域，計算機編程，3 年應當足夠，由於產品的生命週期遠比這一時限更短。

版權政策的另一方面是合理使用的限度：對已發表的作品的全部或部分進行復制的某些方式為法律所許可，即使該作品受版權保護。減少這方面的版權權力的第一步自然是允許私人進行偶爾的小批量非商業性複製並且對個人用戶分發。這將阻止版權警察對人們私人生活的入侵，但對於已發佈的作品的銷售很可能並不會造成什麼影響（可能有必要採取其他法律步驟以保證拆封包裝許可不能被用於取代版權並且藉此限制這種複製的權利）。Napster 的經驗顯示我們還應允許面向公眾的非商業性逐字再分發——由於公眾之中有那麼多的人想要複製和分享，並且他們發現這是那麼有用，只有德拉古式的暴政才會禁止這些，而公眾理應得到他們想要的。

對於小說及其他用於消遣的一般作品形式，非商業性的逐字再分發可能足以保證讀者的自由。計算機程序被用於功能性的目的（用於完成某項任務）需要在此基礎上的更多自由，包括髮布改進版本的自由。參見本書《什麼是自由軟件？》(p. 1) 一文中“自由軟件定義”章節以獲取關於計算機軟件用戶應當享有的自由的解釋。而在程序發佈之後兩三年才能使得這些自由全部可獲得也不失為一種可接受的妥協。

以上這些改變將使得版權迴歸公眾希望的，使用數字技術進行復制的意願軌道上來。出版商將會毫無疑問地發現這些提案是“不公平”的；它們也許會威脅退出出版業，但卻不會真正如此，由於這一行業仍然有利可圖，並且仍將是最好的行業。

在我們思考如何減少版權權力的同時，我們必須保證媒體公司不會簡單地將其改為最終用戶許可協議（EULA）。有必要禁止使用合同來對版權範圍以外的複製施加限制。對於面向大量最終客戶的不可談判合同所能要求的東西進行限制是美國法律體系中的一個標準組成部分。

3.2.11 個人註記

我是軟件設計者而非法學學者。我之所以對版權問題感到擔憂是由於這在計算機網絡世界，例如互聯網中，是無法迴避的。作為一個用了計算機和網絡 30 多年的用戶，我衡量了我們已經失去的以及將要失去的自由的价值。作為一位作者，我有權拒絕將作者比作半神的創造者這一富有幻想性的神話，而出版商經常援引這一點作為依據以支持增加作者的版權權力——然而作者隨後就會把這些權力出讓給出版商。

本文的大部分內容包含了您可以查證的事實和推理，還有您可以用於構建您自己想法的提議。但是我再次請求您接受我個人的看法：諸如我這樣的作者並不應該擁有凌駕於您之上的權力。如果您想要對我所編寫的軟件或書籍表示感謝，我將會心存感激地接受一張支票——但請您不要以我的名義放棄您的自由。

3.3 科學必須擺脫版權束縛

Copyright © 2001, 2012 Richard Stallman. 本文最初於 2001 年 6 月 8 日發表於 Nature 雜誌的 Web Debates 論壇。

很多觀點都指向這一結論：軟件自由必須是普遍性的，這一結論通常同樣適用於其他形式的表達性的工作，儘管是以多種不同的方式。本文專注於討論將這些軟件自由領域相關的原理應用於文獻領域的情況。通常地，這些議題與軟件自由相正交，但我們仍然決定將與本文類似的幾篇文章添加進來，由於諸多對自由軟件感興趣的人們也想獲知關於如何才能將這些原理應用於除了軟件之外的其他領域的細節。

科學文獻存在的理由應當是傳播科學知識，並且科學期刊存在的理由應當是促進這一進程，這應該是一種不言自明的真理。由此可以推論，使用這些科學文獻的規則應當被設計為有助於實現這一目標。

我們現在所擁有的規則，稱之為版權，最初建立於印刷機時代，這是一種從本質上來說相對中心化的印刷量產方式。在這樣一種印刷環境下，期刊文章的版權僅僅約束期刊發行者——要求它們獲得出版文章的許可——以及潛在的抄襲者。它有助於期刊運營以及傳播知識，而並未影響科學家們或者學生們的有用成果，不論對於文章的作者還是讀者。這種規則可以很好地適應這種體系。

但是，用於科學文獻發表的現代技術是萬維網（World Wide Web）。那麼，什麼樣的規則才能確保在網絡上傳播科學文獻和知識的效果最大化呢？文獻應當以非私有格式發佈，並且對所有人開放訪問權限。而且每個人應當有權利為文獻提供“鏡像”，即在滿足適當署名的條件下逐字發表原文。

這些規則應當同樣適用於過去與未來的文獻，只要它們是以電子形式發佈的。但是，現在並不急迫需要改變當前的版權體系，由於它們仍然適合於紙版期刊，此處的問題不屬於那個領域。

不幸的是，看起來並非所有人都認可本文開頭提出的那種不言自明的理念。很多期刊發行者看起來所堅信的是，科學文獻存在的目的是使它們能夠出版期刊以便收取科學家們和學生們的訂閱費用。這種想法可稱為“方法與目的的混淆”。

它們採取的方式是僅僅允許那些能夠並且願意為之付費的人們訪問，甚至只是允許他們閱讀科學文獻。它們以版權法為理由禁止科學家們選擇新的規則。儘管版權法對於計算機網絡時代有著諸多不妥，它們仍然具有法律效力。

以科研協作與人類未來之名，我們必須從根本上拒絕這種方式——不僅僅是在其上建立起來的阻礙進步的體系，還有那些促成了這些體系形成的錯誤前提。

期刊出版商有時宣稱在線訪問需要昂貴並且耗能的服務器，於是它們必須收取訂閱訪問費用以維持其服務器開銷。這個“問題”是其自身的“解決方案”所造成的。如果賦予每個人鏡像的自由，世界各地的圖書館將會建立其自己的鏡像站點以滿足此要求。這種去中心化的解決方案將會減少網絡帶寬開銷並且提供更快速的訪問，還能防止學術記錄意外丟失。

出版商還會爭論它們需要收取訂閱訪問費用以支付編輯的薪酬。我們姑且接受這一假設，即編輯必須得到薪酬；然而這是本末倒置的。編輯一篇普通論文的成本大約是產出它們的科研項目經費支出的 1% 到 3%。這樣小的支出比例很難使得以此為理由阻礙對其結果的使用的行為合理化。

與之相反，編輯的成本可以通過諸如按版面向作者收費的方式得到補償，作者則可以從科研項目贊助者那裡得到補償。贊助者應當不會介意，即使它們當前不得不以一種繁雜的方式支付論文發表費用，例如通過大學圖書館訂閱期刊產生的間接費用。通過改為向科研贊助者徵收編輯費用，我們可以消除以此支持限制訪問的表面理由。對於作者不屬於某一科研機構或公司，以及作者沒有科研贊助的少數情形，可免除其版面費用，這部分成本改為向附屬於科研機構的作者徵收。

另一種支持徵收在線出版物訪問費用的理由是需要資金支持以便將期刊的印刷歸檔轉換為在線形式。這項工作需要完成，但我們需要尋找其他

的資金支持方式，使其不會造成阻礙訪問的結果。這項工作本身將不再是困難並且耗資巨大的。將印刷歸檔數字化但又通過限制訪問使其成果被浪費，這種做法是自相矛盾的。

美國憲法宣稱版權存在的理由是“促進科學進步”。而當版權阻礙了科學進步的時候，科學必須擺脫版權的束縛。

3.3.1 後期的發展

某些大學——例如麻省理工學院（MIT）¹——已經採納了某些政策以限制出版者的權力。然而，更加強有力的政策是必需的，因為 MIT 允許個人作者“退出”（屈服）。

美國政府對某些受資金支持的科研項目提出了稱為“公共訪問”的要求。這要求在一定時期內的出版物發表在某個允許任何人查看論文的網站上。這種要求是積極的一步，但並不足夠，由於它並不包含再分發該論文自由。

有趣的是，2002 年布達佩斯開放存取倡議（BOAI）所提出的“開放訪問”確實包含了再分發的自由。我還是簽署了該聲明，儘管我對“開放”一詞極其厭惡，但其立場是正確的。

然而，關於“開放”一詞還有一個笑話。具有影響力的“開放訪問”運動的支持者後來在其目標中放棄了再分發的自由。我站在 BOAI² 的立場上，但現在“開放訪問”已經是其他意思了。我將其稱為“可再分發出版物”或“自由鏡像出版物”。

¹“MIT Faculty Open Access Policy,” 於 2009 年三月 18 日經教學科研人員無記名投票通過。<http://libraries.mit.edu/scholarly/mit-open-access/open-access-at-mit/mit-open-access-policy/>（譯者注：目前（2019 年一月）該鏈接已失效，可於 <https://libraries.mit.edu/scholarly/mit-open-access/open-access-policy/> 訪問到該內容）。

²參見 <http://www.budapestopenaccessinitiative.org/> 以獲知 BOAI 指導思想。

3.4 計算機網絡時代的版權與社區之爭

本文是 Richard Stallman 所做主題演講之抄本，該主題演講於 2009 年十月在新西蘭基督城（克賴斯特徹奇）會展中心所舉行的新西蘭奧特亞羅瓦圖書館與信息協會（LIANZA）會議上進行。

Copyright © 2009 自由軟件基金會，感謝 Bookman 為此原始抄本所做的貢獻。

Brenda Chawner: 今天，本人榮幸介紹 Richard Stallman，他的主題演講由惠靈頓維多利亞大學信息管理學院所贊助。

Richard 致力於推進軟件自由已逾 25 年。他曾於 1983 年創始 GNU 計劃以開發一種自由的操作系統（GNU 操作系統），並且於 1985 年創立自由軟件基金會（FSF）。每當您讀取或發送一條 nz-libs 信息時，您就在使用 Mailman 軟件，它是 GNU 計劃的一部分。因此無論您是否意識到這一點，Richard 的創作已經深入我們每個人的生活。

我想將他描述為最有影響力卻不為大多數人所熟知的人物，儘管他曾對我說，這不可能是真的，由於它們的正確性不可能被證實。

RMS: 不能這樣說。

BC: 我是想說——我仍然喜歡這樣說。他關於軟件自由以及信息應當可自由獲取的理念被 Tim Berners-Lee 用於創建世界上第一臺網絡服務器，並且他在 1999 年對於一部自由的在線百科全書的深入思考啟發了 Jimmy Wales，後者創立了現在的 Wikipedia。

今天，Richard 將為我們帶來關於計算機網絡時代的版權與社區之爭，以及它們對於圖書館的啟示的演講。有請 Richard。

RMS：我來到新西蘭已有兩週時間，在北島的大部分時間都在下雨。現在我可以理解為何他們將長筒橡膠靴子稱為“惠靈頓靴”。而後我看到了一位使用銀葉蕨的木材製造桌椅的工匠，他將其稱為蕨類傢俱（fern-iture）。然後我們乘坐渡船來到此地，當我們下船時，人們立即開始嘲笑我們；但他們沒有任何惡意，只是想讓我們感受皮克頓當地的風情。

人們邀請我演講的原因通常是基於我為自由軟件所做的工作。而今天這場演講並非關於自由軟件；而是要回答這樣一個問題，即自由軟件的理念是否可以延伸至其他類型的作品。但為了使這個問題有意義，我最好還是簡要介紹一下自由軟件意味著什麼。

自由軟件關乎的是自由而非價格，因此請思考“自由言論”而非“免費啤酒”。自由軟件是尊重用戶自由的軟件，而這裡有四項特定的自由是用戶總是應當擁有的：

- 自由之零：以您所希望的任何方式運行該程序的自由；
- 自由之一：研究程序的源代碼並對其進行修改使程序能夠滿足您的需求的自由；
- 自由之二：幫助他人——即再分發該程序原始副本的自由；
- 自由之三：貢獻您的社區——即再分發您對該程序的改進版本的自由。

如果程序賦予您這四項基本自由，那麼它就是自由軟件，也就是意味著它的發佈和使用所構成的社會體系屬於一種尊重用戶自由以及用戶社區的社會協作倫理體系。但如果以上自由中的任意一條缺失或是不充分，則它稱為私有軟件、非自由軟件或者迫使用戶屈從的軟件。這是不符合倫理的，也不能稱其為對社會的貢獻，而是一種權力的攫取。這種不符合倫理的實踐不應當存在；自由軟件運動的最終目標是終結這種行徑。所有軟件應當是自由的，於是所有用戶也就因此獲得自由。

私有軟件使得用戶陷入孤立無援的困境：所謂孤立，是由於用戶被禁止分享；所謂無援，是由於用戶不能擁有源代碼，因此不能對其進行修改，他們甚至不能研究它以便確定它真正是在對他們做什麼，而且眾多私有軟件擁有惡意功能用於窺探用戶、限制用戶、甚至為對用戶的攻擊提供後門。

例如，微軟 Windows 操作系統擁有後門使得微軟可以強制安裝軟件更改而無需得到理應作為計算機的擁有者——用戶的許可。您可能認為它仍是屬於您的計算機，但如果您犯了用它運行 Windows 的錯誤，實際上是微軟擁有了您的計算機。這樣計算機應當被扔出窗外，意即要麼將 Windows 從計算機中扔出去，要麼將這臺計算機從房間的窗戶中扔出去。

但是，任何私有軟件都賦予了其開發者凌駕於用戶之上的不公權力。開發者們或多或少地濫用這種權力，但他們都不應該擁有這種權力。您理應擁有對您所進行的計算的控制權，並且不必屈從於任何一家特定的公司。因此，您應當使用自由軟件。

在關於自由軟件的演說結束時，人們有時會問，這些相同的自由和理念是否也適用於其他事物。如果您在自己的計算機上擁有一份已發表作品的副本，提問您是否應當擁有同樣的四項基本自由是有意義的——即您是否擁有這些自由在倫理上是否重要。這就是我今天將要著重論述的問題。

如果您擁有一份除了軟件以外的某種東西的副本，對於大多數情況，唯一可能拒絕您的任何一種自由的東西就是版權法。而對於軟件則不是這樣。使軟件成為私有的主要途徑是利用合同以及拒絕對用戶公開源代碼，而版權只是某種次要的備選方案。而對於其他東西，並沒有源代碼和可執行代碼這樣的區分。

例如，我們談論一篇文本，如果您能夠看到該文本以便閱讀它，那麼該文本就沒有您所不能看到的東西。因此這並不是存在於軟件中的同類問題。大多數情況下，只有版權可以拒絕您的這些自由。

於是這個問題可以被轉述為：“版權法應當允許您對已發佈的作品做什麼？版權法應當說些什麼？”

版權隨著複製技術一起發展，因此有必要回顧一下複製技術的發展史。在古代世界，只要您能在書寫表面上使用書寫工具，那裡就有複製技

術的發展。您可以閱讀一份副本並且抄寫另一份。

這種複製技術的效率相當低下，但它的一個特性就是沒有經濟規模效應。為了抄寫 10 份副本，您需要花費 10 倍於抄寫一份副本的時間，它除了書寫工具以外並不需要其他特殊設備，同時除了識字以外您也不需要其他的特殊技能。其結果是任何一本書的副本都是以一種去中心化的方式創作的。只要有一份副本，只要某人想要複製它，他就可以這樣去做。

古代世界沒有像版權這樣的東西的存在。如果您有一份副本並且想要複製它，無人會告知您您不被允許如此做——除非當地的首領不喜歡書中的內容，此時他可能會因為您複製了這本書而對您進行處罰。但這不是版權，而是與之緊密相關的其他東西，稱為審查。而如今版權常常被用於對人們進行審查的企圖。

這種情況持續了數千年，然而，此後複製技術發生了巨大進步，這稱為印刷機。印刷機使得複製過程更有效率，但這並不具有一致性。由於批量複製的效率得到巨大提升，但每次僅僅印製一份副本並不會由於印刷機的存在而變得高效。事實上，您最好還是自己進行手抄，這將會比用印刷機印製一份副本更快。

印刷機具有經濟規模效應：需要花費大量工作來進行排版，但您隨後就可以快速複製很多份副本。同時，印刷機和字模都屬於相對昂貴的設備，大多數人並不擁有它們；而且大部分識字的人們也不知道如何使用它們。使用印刷機是一種不同於書寫的技術。其結果是一種中心化的複製方式：任何給定的書本的副本只能在少數地方被印製，它們隨後可以被運輸到有人想要購買副本的任何地方。

版權始於印刷機的時代。從 16 世紀開始，英格蘭的版權制度開始成為一種審查體系。我相信其最初的本意只是審查新教徒，但其隨後改為審查天主教徒，並且很可能也被用於審查許多其他人。根據這一法律，為了發行一本書，您必須得到皇家許可，而這種許可是以永久壟斷權的形式被授予的。這種制度我相信一直被允許存在直到 17 世紀 80 年代（根據 Wikipedia 相關詞條，它於 1695 年被廢止）。出版商想要重新得到這項權利，但它們實際得到的是與之不同的東西。安娜法令賦予了作者一份版權，

並且僅持續 14 年，儘管作者可以續期一次。

這是一種完全不同的理念——出版商的永久壟斷權變成了作者的臨時壟斷權。這一理念使版權成為了一種促進創作的方式。

起草美國憲法時，一些人希望作者被授予版權權利，但這被否決了。與之相反，美國憲法聲明國會可以有選擇地採納一種版權法，並且如果真的要有一部版權法，其目標必須是促進進步。換言之，其目標不是版權持有人或者與他們進行交易的其他人的利益，而是為了公眾的利益。版權必須僅僅持續一段有限的時間；而出版商一直都在盼望我們忘記這一點。

這裡，我們有了關於版權的這樣一種理念，它是一種約束出版商的行業規範，由作者所控制，並且被設計為最終能夠為公眾帶來利益。它能夠以這種方式發揮作用，由於它並不限制讀者。

在印刷時代的最初幾個世紀，我相信直到 18 世紀 90 年代還是這樣，大部分讀者仍然採用手抄方式進行復制，由於他們買不起印刷的副本。從未有人期望版權法變成行業規範以外的東西，它的本意是約束出版商。正因為此，它容易被強制執行、不會引起爭議、並且可以認為是對社會有益的。

它是容易強制執行的，由於它只需針對出版商強制執行。並且想要找出未經許可的出版商也是容易的——您可以前往一家書店並且說：“這些副本來自哪裡？”您無需入侵某人的住宅或者某人的計算機以實現這一目的。

它不會引起爭議，由於讀者並未被限制，他們無需擔心什麼。從理論上說，他們被禁止從事出版，但由於他們不是出版商，並且沒有印刷機可用，他們無論如何不能從事出版。而在他們實際上可以做的事情當中，他們並未受到什麼限制。

它可以被認為是有益的，由於根據版權法的概念，公眾所出讓的是一種他們所不能行使的、僅存在於理論上的權利。作為回報，他們從更多的作品中得到了好處。

現在，如果您出讓的是一些您沒有任何辦法使用的東西，而得到的回報是一些您可以使用的東西，這是一筆有利的交易。不論您是否能夠通過

其他方式得到一筆更加划算的交易，那是另外一個問題，但至少這是一筆不虧的交易。

因此，如果現在仍然處在印刷機時代，我想我不會抱怨版權法。但是，印刷機時代正在逐步讓位於計算機網絡時代——複製技術的另一次革命，它使得複製更加高效，同時也更加不那麼具有一致性。

這是我們曾經在印刷機時代擁有的東西：批量複製非常高效，而一次複製一本時仍然和古代一樣慢。數字技術為我們帶來了這些：它們都得到了提升，然而一次一本的複製方式所得到的效率提升最大。

我們由此來到了一種更像古代世界的境地，彼時一次複製一本的行為相對於批量複製並不顯得非常低效（困難），它只是略微低效，但它的成本足夠低，以至於數以億計的人們都能夠進行這樣的複製操作。想想看有多少人可以不時地燒錄光盤（CD），即使是在欠發達的國家。您可能沒有一臺 CD 刻錄機，不過您可以前往一家可提供此服務的商店。

這意味著版權不再像過去那樣適應技術的發展。即使版權法中的詞句保持不變，它也不會再產生與之前相同的效果。此時的版權不再是一種由作者控制的針對出版商的行業規範，並且由此使公眾受益。它已經成為了主要由出版商控制，以作者之名對公眾的限制。

換言之，這是一種暴政，這是不可容忍的，我們決不能允許它以這種方式發展下去。

由於這種性質上的轉變，版權不再容易被強制執行、不再沒有爭議、也不再有益。

它不再容易強制執行，由於現在是那些出版商想要針對每個人強制執行。而要達到這一目的，需要嚴酷的措施、德拉古式的嚴厲刑罰、對隱私的侵犯以及廢除我們關於公平的基本理念。目前看來，他們將會在發動“消滅分享之戰”的道路上走多遠，一時還看不到界限。

它不再沒有爭議。在一些國家已經出現了這樣一些政黨，其基本立場是“分享自由”。

它也不再有益，由於我們曾經在概念上出讓（由於那時我們不能行使它們）的自由現在是我們可以行使的了。它們是那麼地重要，以至於我們

現在想要行使它們。

那麼，一個民主政府在這種情況下應當做些什麼呢？

它應當減少版權權力。它應當說：“我們在以公民之名進行的交易中出讓的那部分自由是現在他們所需要的，這是不可容忍的。我們必須改變這一切；我們不能輕易出讓這種至關重要的自由。”我們可以如此衡量民主的欠缺，通過考察世界各國政府做出與民主背道而馳的事情的傾向性，即它們在本應減少版權權力的時候反而不斷延伸版權權力。

一個例子是版權的時間尺度。放眼全世界，我們看到各種壓力使得版權持續時間更長、更長、更長。

一波此類事件於 1998 年發生在美國。過去和將來的作品的版權期限都延長了 20 年。我不明白他們怎樣才能指望使得那些活躍於 20 世紀 20 至 30 年代的已經去世或者年邁的作家相信，通過在今天延長他們的作品版權能夠促使他們在當時創作更多的作品。如果他們擁有一臺可用於在當時通知他們的時光機器，他們也從未使用過它。我們的歷史教材也從未說過，在 20 世紀 20 年代曾經發生過文科創作活力的爆發式增長，由於當時所有的藝術家預知他們的作品版權將會在 1998 年被延長 20 年。

為未來的作品延長 20 年的版權期限將會說服人們為創作這些作品付出更多努力。這在理論上是可信的，但這不能說服那些有理性的人們，由於在未來 75 年——如果這篇作品是出資聘人完成的——如果該作品擁有一位獨立版權持有人，可能還會更長一些，在此基礎上額外增加的 20 年版權期限的存在價值將會大打折扣，其存在價值小到不能說服任何有理性的人們為此去做一些不同的事情。任何想要宣稱事情並非如此的商業公司應當被要求出示它在未來 75 年中的財務狀況表，然而它們當然做不到這一點，由於它們當中沒有一家會真正看問題看得那麼遠。

這條法律的真正原因，以及驅使不同的商業公司向美國國會購買這條法律（國會在很大程度上有權決定法律）的原因在於它們已經擁有使其大大得利的壟斷權，並且想讓這種壟斷權持續下去。

例如，迪士尼意識到米老鼠首次出現的那部電影將會在幾年內進入公有領域，然後任何人都可以自由地繪製相同的角色以用於其他作品。迪士

尼不希望這件事發生。此前，迪士尼曾經從公有領域借鑑了很多東西供自己使用，但它決定永遠也不向公有領域做出哪怕是最微薄的回饋。於是迪士尼買來了這條法律，我們稱之為米老鼠版權法案。

電影公司說它們想要永久版權，但是美國憲法不會讓它們以官方的方式得到它。於是它們想出了一種方式以便以非官方形式達到同樣的結果：“永久版權無限續期方案”。每隔 20 年，它們會將版權期限再次延長 20 年。因此，在任意給定的時間，對於任意給定的作品，都存在這樣一個日期，它們應當在此日期進入公有領域。但是，這個日期就如同明日復明日，永遠不會到來。當您等到那一天的時候，它們又將其延後了，除非我們能夠在下次阻止它們這樣做。

以上這些是一個方面，即版權期限的方面。但是更重要的一個方面是寬度：即版權會覆蓋作品在哪些方面的使用。

在印刷機時代，版權不應覆蓋一篇作品的所有應用場景。由於版權所管制的某些應用同時也是一系列更寬泛的未受管制的應用中的一些例外。您很自然地被許可利用您的某本書的副本去做某些特定的事情。

現在，出版商有了這樣的理念，它們可以使得我們的計算機背叛我們，並且利用它們攫取對於已發佈作品的所有使用可能的絕對控制權力。它們想要建立一種按次付費點播的通用規則。它們正在通過數字限制管理（DRM）來實現這一點——軟件的功能被故意設計為限制用戶。而且計算機本身也通常被設計為限制用戶。

公眾首次見識到它的方式是通過數碼多功能影音光盤（DVD）。保存於 DVD 上的電影通常是加密的，並且其格式是私密的。DVD 陰謀集團保守這一祕密，因為它們宣稱任何人想要製造 DVD 播放器必須加入這一陰謀集團，並且承諾保守這一祕密，同時還要承諾設計出的 DVD 播放器必須根據規則限制用戶，也就是說，它必須阻止用戶做這件事、還有那件事、還有那件事——一系列精準的要求，對我們都是惡意的。

雖然花費了一些時間，但隨後某些人還是破解了這種私密的格式，並且發佈了可用於讀取並播放 DVD 電影的自由軟件。然後出版商說：“既然我們不能在事實上阻止他們，我們必須讓這件事成為罪行。”它們於 1998

年在美國利用數字千年版權法案（DMCA）開始這麼做，這一法案對能實現上述功能的軟件強制實施了審查。

因此，這一類特定的自由軟件被訴諸法庭案例。它們在美國的分發被禁止；美國對其實施了審查。

電影公司意識到它們不能真正使得那個程序徹底消失——它仍然很容易被搜索到。於是它們設計出了另一套加密系統，並且希望它更難被破解，它被稱為高級訪問信息系統（AACS），或者“戰斧”。

AACS 陰謀集團為所有播放器制定了細緻的規則。例如在 2011 年，它將會禁止模擬視頻輸出。因此所有視頻輸出將必須是數字式的，它們將會把信號以加密形式傳輸到一臺特別設計為使這些內容對用戶保持私密的顯示器上。那是一種惡意硬件。它們宣稱如此做的目的是“修補模擬漏洞”（Stallman 摘下眼鏡）。這裡有一個，那裡還有一個，而它們想讓這些漏洞永久消失¹。

我是如何知道這些陰謀的呢？其原因是它們並不是祕密——它們有其官方網站。AACS 網站驕傲地展示了生產商必須簽署的霸王條款，這就是我之所以會知道這些限制條件。它還驕傲地展示了共同建立了這一陰謀集團的商業公司的名字，包括微軟、蘋果、英特爾、索尼、迪士尼、IBM 等。

這些公司所設計的陰謀旨在限制公眾對技術的訪問，這應當作為一項重罪被起訴，就像操縱價格的陰謀那樣，只是這比操縱價格更壞而已。因此，這種罪行的刑期應當更長。但是，那些公司信心滿滿，由於政府站在它們那邊共同壓制我們。它們絲毫不擔心可能會因為這些陰謀而被起訴，這就是它們為何甚至不屑於掩飾這一點的原因。

通常，DRM 是由一些公司組成的陰謀集團所實施的。有時一家單獨的公司也能這麼做，但通常這需要在技術公司和出版商之間達成某種陰謀，因此它幾乎總是陰謀。

它們認為無人能夠破解 AACS，但是大約在 3 年半之前，某人發佈了

¹在 2010 年，關於數字視頻輸出的加密系統被正式破解。參見 Mark Hachman 的文章“HDCP Master Key Confirmed; Blu-Ray Content Vulnerable”（2010 年 9 月 16 日），位於 <http://pcmag.com/article2/0,2817,2369280,00.asp> 以獲得更多信息。

一款可以解密那種格式的自由軟件。然而，這是完全無用的，因為如果您想要運行它，您需要知道它的密鑰。

然後在 6 個月後，我看到了一幅照片，裡面有兩隻可愛的小狗，每隻小狗身上有 32 位的十六進制數，我當時覺得奇怪：“為何將這兩樣東西放在一起？我懷疑這些十六進制數是否可能是某個重要的密鑰，而某人可能是將這些十六進制數和小狗放在一起，希望人們複製這張小狗的照片，由於它們是那麼可愛。這也許能夠保護該密鑰免遭被抹除的厄運。”

事實上，它就是——破解“戰斧”的密鑰。人們發佈它，然後網站編輯刪除它，由於現在很多國家的法律動員他們對這類信息實施審查。它再次被髮布，他們又將其刪除；最後他們放棄了，這組密鑰在兩週之內被髮布到了 70 多萬個網站上。

這是公眾對 DRM 的不滿的一次大規模爆發。但這並未贏得這場戰爭，由於出版商更換了密鑰。不僅如此，對於高清 DVD (HD DVD)，這種方式足以破解其 DRM，但對於藍光 (Blu-ray) 光盤則不行。藍光光盤擁有一個額外層級的 DRM，至今沒有自由軟件可以破解它，這意味著您必須將藍光光盤看做某種與您自己的自由完全不兼容的東西。它們是您所不可能與之共存的敵人，至少對於我們當前的知識水平是如此。

永遠不要接受任何被設計為用於攻擊您自由的產品。如果您沒有可用於播放 DVD 的自由軟件，您必須不要購買或者租用任何 DVD，或者以禮物的形式接受它們，除非是稀有的未加密的 DVD，並且確實存在少數這樣的東西。我確實擁有幾片——但我沒有任何加密的 DVD，我不會設法獲得它們。

以上這些就是視頻方面的情況，但我們也已經遇到了音樂中的 DRM。

例如，大約 10 年前，我們開始見到一種形似音樂光盤 (CD) 的東西，但是它們的燒錄方式和 CD 並不相似。它們不遵守標準。我們稱之為“損壞的光盤”，它們所遵循的理念是它們可以用某種音頻播放器進行播放，但不可能在計算機上讀取。這些不同的方法具有不同的問題。

最終，索尼想出了一個奸詐的主意。它將一個程序存儲在盤片上，於是當您將盤片放入計算機中時，該盤片將會安裝該程序。這個程序被設計

為像病毒一樣取得系統的控制權。它被稱為“root kit”，這意味著它擁有破解系統安全措施的能力，使得它可以將軟件植入系統深處，並且修改系統。

例如，它修改了您可用於查看系統狀態以獲知某個軟件是否存在的命令，因此它隱藏了自己的存在。它修改了您可用於刪除它的某些文件的命令，因此這些命令並不能真正刪除它們。現在，所有這些都是嚴重的罪行，但這還不是索尼所犯下的唯一罪行，由於該軟件同時包含自由軟件代碼——這些代碼是以 GNU 通用公共許可證（GNU GPL）發佈的。

現在，GNU GPL 是一種左版（copyleft）許可證，這如同說：“是的，您擁有將此代碼整合到其他程序中的自由，但如果您決定這麼做，您將此代碼整合到其中的整個程序也必須以相同的許可證作為自由軟件發佈。並且您必須保證用戶可獲得源代碼，以及為了保證他們對於自己的權利擁有知情權，您必須在他們得到軟件的同時為他們提供一份此許可證的副本。”

索尼並未完全遵守這些。這屬於商業版權侵犯行為，是一種重罪。這兩種行為都是重罪，但索尼並未因此被起訴，由於政府明白，政府和法律的目的是維護這些公司凌駕於用戶之上的權力，而非以任何方式幫助用戶捍衛他們的自由。

不堪忍受的人們起訴了索尼。然而他們犯了一個錯誤。他們並未將譴責集中於這一陰謀的邪惡目的之上，而是僅僅侷限於索尼將其次要罪惡付諸實現的不同方法上。於是索尼化解了這些法律訴訟，並承諾在未來繼續侵犯我們的自由的時候，它將不會再次採用這些方法。

事實上，這種“損壞的光盤”的陰謀並不是非常地壞，由於只要您不用 Windows，它不會對您產生任何影響。甚至即使您正在使用 Windows，在您的鍵盤上有這樣一個鍵——如果您每次都記著按住它，該盤片就不會安裝那個軟件。但是，每次都記著按住那個鍵顯然不是一件容易做到的事情；您總有一天會疏忽。這展示了我們必須去設法應對的某種事情。

幸運的是，音樂所受的 DRM 威脅正在減少。即使是大型唱片公司也會銷售不帶 DRM 的下載版。但是，我們又看到了一波試圖為電子書強行施加 DRM 的攻勢。

您應該看到了，出版商想要剝奪讀者的傳統閱讀自由——諸如從公共

圖書館借閱；或者將書借給朋友；或者將書賣給舊書商店；或者使用現金匿名購書等自由（這是我購書的唯一方式——我們必須抵禦各種誘惑，不讓當權者知道我們所做的每一件事情）。

甚至是您想要保有該書任意長的時間，或者閱讀該書任意多次的自由，它們也計劃剝奪。

它們通過 DRM 的方式做到這一點。它們知道有那麼多的人讀書，如果直接剝奪他們的這些自由，他們將會強烈反對，因此它們不認為自己能夠簡單地通過購買一條法律來特定地廢除這些自由——這將會招致太多反對。民主是有缺陷的，但有時人們也能要求得到某些東西。於是它們想出了一種兩步走的方案。

首先，剝奪電子書的這些閱讀自由，然後，迫使用戶從紙版書轉向電子書。它們已經成功做到了第一步。

在美國，它們通過 DMCA 做到了這一點，而在新西蘭，這是版權法案（2008 年）的一部分，即對於可用於破解 DRM 的軟件實施審查。這是一種不公平的條款，它應當被廢除。

第二步是說服人們從紙版書轉向電子書；這項進展並不十分順利。

2001 年，一家出版商想出了一種方法，如果它能夠以我的傳記開頭，它的系列電子書將會變得非常流行。於是它們找到一位作者，後者詢問我是否願意合作，我說：“除非該電子書以未加密並且無 DRM 的形式發佈。”出版商不願意接受這一條，而我則強烈要求這樣——我最終拒絕了。最後我們找到了另一家出版商，它願意如此做——事實上是希望使用自由許可證發佈該書以賦予您四項基本自由——於是該書最終出版，並且售出了很多份紙版副本¹。

但不管怎麼說，電子書在這十年之初並未取得預期成果。人們並不十分願意閱讀它們。我曾說過：“它們將會捲土重來。”我們見到了數量可觀的關於類似電子墨水的新聞文章（或者稱為電子紙，我記不清是哪個了），我認為之所以會有那麼多文章，很可能是由於出版商想讓我們考慮這種載

¹此書即《Free as in Freedom》。中文譯名《若為自由故：自由軟件之父理查德·斯托曼傳》，鄭楠、李凡希翻譯，人民郵電出版社 2015 年出版。——譯者注。

體。它們希望我們迫切地想要成為下一代電子書的讀者。

現在，它們終於來了。諸如索尼碎紙機（Shreader，它的官方名字是 reader，但是如果您加上 sh 兩個字母，這就能很好地解釋它是被設計用於對您的書做什麼的了）和亞馬遜詐騙（Swindle）這樣的設備是被設計為用於在您毫無戒備的情況下騙走您的傳統閱讀自由的。當然，亞馬遜將其稱為 Kindle（焚燒）恰如其分地解釋了它將會對您的書做些什麼。

亞馬遜焚書機（Kindle）是一款極端惡意的設備，幾乎和微軟 Windows 操作系統一樣壞。它們都擁有間諜功能，都擁有 DRM，都擁有後門。

對於 Kindle，您唯一可能的購書方式就是從亞馬遜購買，它要求您提供所有個人信息，於是它們知道關於您所購買的東西的所有信息。

由於存在 DRM，您不能將書借出或是賣給舊書商店，圖書館也不能將書借出。

另外是存在後門，這是我們在大約 3 個月前因亞馬遜使用了它而得知的。亞馬遜向所有 Kindle 發送了一條指令以刪除一本特定的書，即喬治·奧威爾（George Orwell）的《一九八四》。是的，它們找不出另一本更具諷刺意味的書來刪除了。這就是我們如何知道亞馬遜擁有某種後門並且通過它遠程刪除電子書的。

它還能做出什麼事情來呢？誰知道。也許它就像微軟 Windows。也許亞馬遜可以遠程升級它的軟件，這意味著所有那些今天還沒有的惡意功能，它們都可能在明天被帶來。

這是不可容忍的——這其中的任何一條限制都是不可容忍的。它們想要創造這樣一個世界，再也無人能夠將書籍借給他人。

假設您前去拜訪一位朋友，他的書架上並沒有一本書，這並不是由於您的朋友從來不讀書，而是由於他的所有書都保存在某一臺設備中，他當然不能將那些書借給您。他所能夠借給您任意一本書的唯一方式是借給您他的整個圖書館。請求他人如此做顯然是荒唐的。因此，熱愛讀書的人們之間的友誼將不復存在。

請您確保告訴人們這種設備意味著什麼。它意味著其他讀者不再是您的朋友，因為您在他們面前表現得像個傻瓜。一定要先發制人地傳播這句

話。這款設備是您的敵人，它是任何讀者的敵人。那些尚未意識到這一點的人們是那些思想短視，未能看穿這一點的人們。我們的任務是幫助他們透過眼前的易用性看穿這款設備的本質。

我絕不反對以數字形式發佈書籍，如果它們並非被設計用於剝奪我們的自由。嚴格地說，要想擁有這樣一款電子書閱讀器是可能的：

- 不是被設計用於攻擊您；
- 運行自由軟件而非私有軟件；
- 不帶 DRM；
- 不要求用戶購書時提供身份信息；
- 不帶後門；
- 不限制您可以用它對您的計算機上的文件做什麼事情。

這是可能的，但是，真正推廣電子書的大型出版公司如此做的目的都是侵犯我們的自由，這是我們無論如何不能容忍和支持的。這正是政府和大型出版企業正在合謀所做的事情，通過使得版權變得越來越嚴酷和齷齪，越來越具有約束性來侵犯我們的自由。

但是，它們應該做些什麼呢？政府應該減少版權權力。以下是我的一些特別建議。

首先，是關於版權的時間尺度。我提議版權應該僅僅持續 10 年，從作品發佈的日期計算。

為何從作品發佈的日期開始計算？由於在此之前，我們並沒有任何副本。我們是否被允許複製那些我們尚未擁有的作品副本並不重要，因此我認為我們同樣可以允許作者擁有足夠的時間來安排作品的發行，然後再開始計時。

那麼為何是 10 年？我不知道這個國家（新西蘭）情況如何，但在美國，出版週期正在變得越來越短。現在幾乎所有的書都只在 2 年內被廉價

出售，並且會在3年內絕版。因此，10年的期限是通常出版週期的3倍有餘——這應當足以令人滿意。

但是並非所有人都同意這一點。我曾在一次同科幻作家的研討會上提出這一觀點，而坐在我身邊的一位獲獎科幻作家說道：“10年？不可能。超過5年的任何東西都是不可容忍的。”您可以看到，他與他的出版商之間進行了與法律有關的爭論。他的書似乎將要絕版，但是出版商並不同意。該出版商正在使用此人自己所著的書的版權來禁止此人自行分發其副本，而他則希望如此做以使得人們可以閱讀此書。

這正是每位作者正在開始要求的——他們想要分發自己的作品使其被人們閱讀和感激。極少有人能賺得盆滿鉢滿。這很小的一部分大發橫財的人正在面臨道德敗壞的風險，比如 JK Rowling。

在加拿大，JK Rowling 取得了一項針對那些從某家書店購買了她所著的書的人們的禁令，強行要求這些讀者不得閱讀該書。作為迴應，我號召發動了一場抵制《哈利波特》叢書的運動。但我並未說您不應該閱讀它們，我將這種抵制留給作者和出版商。我只是說，您不應該購買它們。

只有少數作者能夠賺到足以使得他們腐化墮落的錢。他們中的大多數遠未達到這種境地，並且繼續要求那些他們從一開始就在訴求的東西：他們想要自己的作品受到感激。

他想要分發自己的書，但版權禁止他如此。他意識到超過5年的版權期限不可能為他帶來任何好處。

如果人們更想讓版權僅僅持續5年，我不會反對。我只是提議將10年作為解決問題的第一步。首先讓我們將其縮短到10年，然後觀望一段時間，接下來我們就可以繼續調整它。我並沒有說我認為10年就剛好是恰當的數字——我也不知道到底應該是多少。

那麼，關於版權的寬度又如何呢？版權應當覆蓋哪些活動？我在此區分三大類作品。

首先，有一些功能性的作品是您可以在您的生活中將其用於執行一項實踐任務的。這包括軟件、菜譜、教材、參考文獻、字體、以及您所能想到的其他東西。這些作品應當是自由的。

如果您在生活中使用這樣一件作品來完成一項任務，那麼如果您不能修改該作品以使其適合您的需求，您就不能控制您的生活。一旦您更改了該作品以適應您的需求，那麼您應該擁有發佈它的自由——發佈您的版本——因為將會有其他人想要得到您所做的修改。

這很快就能推出結論，用戶必須擁有同樣的四項基本自由（對於所有功能性的作品）而非僅限於軟件。並且您將會注意到，對於菜譜，或者從實踐角度講，烹飪的行為總是在分享並且改進菜譜，如同菜譜是自由的一般。設想一下，人們將會作何反應，如果政府試圖禁止所謂的“菜譜盜版”。

“盜版”一詞完全是一種鼓吹的概念。當人們問我關於音樂盜版的看法時，我會說：“就我所知，當海盜們發動攻擊時，他們並非採取拙劣地演奏樂器的方式，而是使用武力。因此這不是音樂‘盜版’，由於海盜行為是攻擊船隻，而分享和攻擊船隻的行為在道德上相去甚遠。”攻擊船隻是壞事，而與他人分享則是好事，因此我們應當在我們聽到“盜版”這一鼓吹的概念的時候給予堅決的揭露。

早在 20 多年前，人們可能會反對這一點：“如果我們不放棄自己的自由，如果我們不讓這些作品的出版商支配我們，這些作品就沒有機會被創作出來，這是一種恐怖的災難。”現在，請看自由軟件社區，還有那些廣為流傳的菜譜，還有諸如 Wikipedia 這樣的參考文獻——我們甚至還開始見到自由的教科書的發佈——我們就可以知道這種恐懼是被誤導的結果。

完全沒有必要因為想到若非如此作品就不會被創作出來而感到絕望並且因此放棄我們的自由。還有很多方式可以鼓勵作者來創作這些作品，如果我們想要更多的作品——很多方式都可以維持並且尊重我們的自由。對於這一大類作品，它們都應當是自由的。

那麼，對於第二大類作品又如何呢？即對於那些包含人們的思想的作品，諸如回憶錄、議論文、科研論文¹以及其他不同形式的作品。發佈某些其他人對其自身思想的論述的修改版本將會曲解此人的本意，這絕不是對

¹在 2015 年，我加入了科研論文這一條，由於我認為發佈他人的論文的修改版本是有害的；然而，將物理和數學論文以創作共用：署名許可證發佈在 arxiv.org 或是眾多其他自由期刊上似乎並不會帶來什麼問題。於是，我隨後得出結論，科研論文應當是自由的。

社會的貢獻。

因此，創立這樣一種在一定程度上弱化的版權體系是可行並且可接受的，在此體系中，所有商業應用受版權限制，所有對原作的修改受版權限制，但是任何人都擁有對原始版本的副本進行非商業性再分發的自由。

這些自由是我們必須建立起來的針對所有已發佈作品的最低限度自由，因為正是對這些自由的否定發動了“消滅分享之戰”——它創造惡毒的鼓吹的概念：分享就是盜竊，分享（盜版）行為就像海盜攻擊船隻的行為。這是荒謬的，但是這種由金錢支撐的荒謬理念腐化了我們的政府。我們需要終結“消滅分享之戰”；我們需要使得分享任何已發佈作品的原始版本合法化。

對於這第二大類作品，以上這些就是我們所需要的；我們不必須使其成為自由的。因此我認為擁有一種覆蓋了商業應用以及任何修改的弱化的版權體系是可以接受的。這可以為作者提供一種收入來源（通常是不足以體現其應得收入的），以某種與現有體系或多或少地相同的方式。您應當留意在現有體系中，除了少數明星大腕，絕大多數作者完全沒有得到應得的收入。

那麼對於藝術娛樂作品又如何呢？在此，我花費了一些時間來決定如何看待對作品的修改。

您可以看到，一方面，一幅藝術作品可能具有某種藝術上的完整性，對其進行修改可能會破壞這種完整性。當然，版權並不會必然地防止作品以這種方式被糟蹋。好萊塢就一直在這樣幹。另一方面，修改一幅作品也可能成為對藝術的一種貢獻，它使得民間傳承的素材也能變得富有美感和意蘊。

即使我們僅僅著眼於著名的作家：比如莎士比亞，他從幾十年前發佈的作品中借鑑故事情節並且以其他方式進行呈現，最終創作出了文學史上的鉅著。如果今天的版權法在當時也存在，這種創作方式將被禁止，那些劇本也不可能被創作出來。

但是，我最終意識到修改一幅藝術作品可能成為對藝術的貢獻，然而這在大多數情況下並非急迫需求。如果您必須等上 10 年使得版權過期，您

是可以等上那麼長時間的。不像現在的版權法迫使您等上 75 年或是 95 年。在墨西哥，有時您可能不得不等上將近 200 年，由於墨西哥的版權將會在作者去世 100 年後過期。這是極其不理智的，但是如我所提議的 10 年，作為版權應該持續的時間，是人們可以在有生之年等到的。

因此我提議相同的弱化版權覆蓋商業應用和作品修改，但每個人都應當擁有對其原始版本進行非商業性再分發的自由。10 年之後，它將進入公有領域，人們可以通過發佈他們的修改版本來繼續為藝術發展做出貢獻。

還有一件事情，如果您想要從一系列作品中挑選一些片斷並且將其整合為某種全新的作品，這應當是合法的，由於版權的初衷是促進藝術發展，而非阻礙藝術發展。使用版權來禁止這種使用作品片斷的創作方式是愚蠢的——它是自相矛盾的。這是一種扭曲，只有當政府被那些現存成功作品的出版商所支配的時候才會發生這種事情，並且這已經完全迷失了其初衷。

這就是我所提議的，特別要指出的是，這意味著在互聯網上分享副本必須是合法的。分享是美好的。分享構建了社會的契約。反對分享就是反對社會。

因此，每當政府提議新的措施以制裁分享作品的人們並且強行禁止他們分享時，我們必須認識到這是一種罪惡。不僅僅是因為這些被提議的制裁措施無一例外地侵害了公平這一基本理念。但這並非巧合；其原因是這種提議的目的是罪惡的。分享是美好的，政府應當鼓勵分享。

但不管怎麼說，版權確實有其有用的目的。只是版權作為實現該目的的手段在當下遇到了問題，由於它不能適應當今我們所使用的技術。它干涉了讀者、聽眾、觀眾及其他人的所有基本自由，但它促進藝術發展的目標仍然是美好的。因此，版權系統將繼續作為版權系統而存在，除了在一定限度上減少它的權力。我提議兩種其他方式。

其一是（通過）徵稅（來實現）——直接將稅金分發給作者。這可以是一種特殊稅收，可以是對於互聯網連接服務的稅收，也可以來自一般性的收入，由於它的總量不會非常大，如果它是通過某種有效的方式進行分發的。以促進藝術發展為目的對其進行有效分發意味著不是按照作品流行度的線性比例進行分發。它應當基於流行度，由於我們不希望由官僚主義

者獨斷決定哪些作者應該得到支持而哪些作者將會被無視。但是，基於流行度並不必然意味著按照線性比例進行分發。

我所提議的是統計不同作品的流行度，您可以通過投票（樣本統計）做到這一點，並且不是每個人都被強制參與，然後取其立方根。立方根就像這樣：它基本意味著（收入）增加值將會逐漸變小。

如果某位明星大腕 A 的流行度是某位成功藝術家 B 的 1000 倍，按照這種體系，A 將得到 10 倍於 B 的收入，而非 1000 倍。

如果按照線性比例，A 將得到 1000 倍於 B 的收入，這意味著如果我們想讓 B 得到足以維持其生計的錢，我們必須使 A 賺得盆滿鉢滿。這是一種對稅金的浪費——不應該這樣做。

但如果我們使得收入增量逐漸變小，那麼是的，每位明星仍將得到比普通的成功藝術家多得多的收入，但所有這些明星的收入總和將只佔稅金（總額）的一小部分。大部分稅金將會被用於支持一大批相對成功、相對受歡迎、相對流行的藝術家。因此，相對於現有的體系，這一體系將能夠更有效地利用這筆稅金。

現有的體系是倒退的。它實際上，例如對每張唱片，給予明星大腕的錢遠遠超過給予其他任何人的。這種花錢的方式非常不合理。其結果是我們按照這種方式所實際花費的錢要少得多。我希望這足以安慰那些對稅收有著如同膝跳反應般的敵意的人們——我並不反對稅收，因為我信任一個福利國家。

我還有一個建議是關於志願者捐助的。假設每臺播放器都有這樣一個按鈕，您可以通過按下它來向您正在播放的作品或者已經播放的上部作品的作者發送 1 美元。這筆錢將會被匿名地發送給作者。我相信會有很多人比較經常地按下那個按鈕。

例如，我們當中的所有入應當能夠負擔得起每天按一次按鈕的費用，這不會花費我們那麼多錢。這筆錢對於我們是可接受的，我非常確信。當然，也會有貧困的人們不能負擔按一下按鈕的費用，然而即使他們不去按下按鈕也沒關係。我們不需要從窮人身上榨取錢財來支持作者。有足夠多的經濟狀況尚可的人們，他們可以很好地完成這件事。我相信您已經注意

到，有很多人確實喜愛某種藝術，並且非常樂意支持作者。

於是我又有了一個想法。播放器還可以給您一份證書，表示您曾經支持過某位無名作者，它甚至還可以統計您曾經送出多少次支持，並且給您這樣一份證書：“我為這些藝術家付出了這麼多。”總之，我們可以通過很多方式鼓勵那些想要這麼做的人們。

例如，我們可以開展一場友善的 PR 活動：“您今天向某位藝術家捐助 1 美元了嗎？為什麼不呢？這只是 1 美元——您將永遠不會失去它，您難道不喜愛他們所創作的作品嗎？現在就按下按鈕！”這將使人們產生好感，他們將會想：“是的，我喜愛我所觀看的作品，我將要捐助 1 美元。”

這在一定程度上已經開始產生效果。有一位曾經叫做 Jane Siberry 的加拿大歌手。她將自己的音樂放在自己的網站上，邀請人們下載並且捐助他們所願意支付的任意金額。她曾經報道收到了平均每份副本多於 1 美元的收入，這是有趣的事情，由於大型唱片公司的收費還不到每份副本 1 美元。通過讓人們自行決定是否捐助與捐助多少，她實際上得到了更多——如果按照實際下載了某些東西的訪問者來統計，她通過每位訪問者得到的甚至更多。但這可能還沒有考慮是否存在這樣一種效應，即吸引更多訪問，從而增大了這一平均值所對應的分母。

因此，這是可行的，但在當前環境下是一種令人頭痛的事。首先您必須擁有一張信用卡，這意味著您不能進行匿名捐助。並且您必須找到去哪裡進行支付，而對於小額支付，現有的支付系統並不非常高效，因此藝術家們只是成功了一半。如果我們能夠建立一種理想的支付系統，它將能夠更好地實現我們的目的。

在 mecenat-global.org 網站¹，您可以找到結合了這兩條建議思想的另一種方案，它由 Francis Muguet 發明，並且被設計為旨在更好地適應現存法律體系以使其更容易實施。

當心這種關於“補償版權持有人”的鬼話，由於當他們在說“補償”的時候，他們是在試圖假設如果您感激一篇作品，您已經欠下了某人一筆特

¹此網頁不再可訪問，請移步至 <https://stallman.org/mecenat/global-patronage.html>

殊的債務，並且您必須“補償”此人。當他們說“版權持有人”的時候，它應該是在迫使您認為這是在支持藝術家，而實際上這是在支持出版商——正是那些從根本上剝削了所有藝術家的出版商（除了少數各位都聽說過的藝術家，由於他們那麼出名，以至於他們擁有了某種勢力）。

我們並不欠任何人任何東西；我們沒有“補償”任何人的理由和義務。但是，支持藝術發展仍然是有益的事情。這曾經是版權的動機，彼時的版權仍然是適應那個時代的技術發展水平的。而今天，版權對於支持藝術發展成了一種壞的實施方式。但以尊重我們的自由的方式來實施它，這還是好的。

我們想要利用它們改變新西蘭版權法案中的兩大邪惡部分，它們不應取代“三振出局法”¹，由於分享是美好的，它們應當擺脫針對用於破解 DRM 的軟件的審查。當心反假冒貿易協定（ACTA），它們試圖在不同國家之間商定一種條約，使這些國家可以攻擊它們的國民，我們不知道如何攻擊，由於它們不會告訴我們。

¹新西蘭實施了一種未經審判的刑罰系統，對於被指控進行復制的互聯網用戶；然後，迫於大眾的抗議，政府並未立即實施，而是宣佈了一項計劃以實施一種修訂過的但仍然不公平的刑罰系統。本文此處的觀點是，它們不應實施一種替代方案——與之相反，它們不應擁有這類系統。然而，我的用詞未能清楚地表達這一點。新西蘭政府隨後實施了一種與原本計劃的或多或少相同的刑罰方案。

第 4 部分

軟件專利：對程序員的威脅

4.1 軟件專利和文學專利

Copyright © 2005, 2007, 2008 Richard Stallman 本文最初於 2005 年 6 月 23 日以標題 “Patent Absurdity” 發表於英國衛報，並且針對當時被提出的 “歐洲軟件專利指導意見”。

當政治家們考慮軟件專利的問題的時候，他們通常會盲目地進行表決；由於他們不是程序員，他們並不瞭解軟件專利真正在做什麼。他們通常認為專利法與版權法類似（“除了少數細節以外”）——然而事實並非如此。例如，當我公開詢問法國工業部長 Patrick Devedjian 關於法國將如何針對軟件專利問題進行表決的時候，Devedjian 用一段對版權法的充滿激情的辯護作為迴應，並且盛讚維克多·雨果為了使版權這一概念為人類所接納所做出的貢獻。（“知識產權” 這一帶有誤導性的短語加深了這種混淆——這是它永遠不應該被使用的原因之一）。

那些認為軟件專利將會產生與版權法類似效果的人們未能抓住並認清軟件專利所可能帶來的災難性後果。我們可以以維克多·雨果為例，說明這二者的區別。

一部小說和一個現代化複雜程序具有某些共同點：其一是大型化，以及在實現它的過程中需要將諸多設計思想融會貫通。於是，讓我們沿著這個類比，假設專利法已經於19世紀被應用於小說；並且假設諸如法國這樣的國家允許對文學創意進行專利保護。這將會對維克多·雨果的寫作帶來怎樣的影響？文學專利帶來的影響與文學版權帶來的影響相比又會如何呢？

考慮維克多·雨果的名著《悲慘世界》。由於他是作者，該書的版權屬於他本人。他完全無需擔心某些陌生人對他進行侵犯版權訴訟並且勝訴。那是不可能的，由於版權僅僅覆蓋一篇作品的作者身份的細節，而非蘊含在作品中的思想創意，並且僅僅限制對作品的抄襲。維克多·雨果並未抄襲《悲慘世界》，所以他不會受到版權的威脅。

專利發揮作用的方式則與之不同。專利將會覆蓋想法和靈感；每一項專利都是對實施某種想法的壟斷，這是描述在專利本身的內容之中的。以下是一項假想的文學專利案例：

- 要求1：一種在讀者腦海中表達了這樣一種概念的交流過程：某一角色在監獄中被監禁了很長時間，由此對社會和人性產生怨恨；
- 要求2：一種基於要求1的交流過程：該角色隨後在他人的關懷下得到了道德的救贖；
- 要求3：一種基於要求1和要求2的交流過程：該角色在故事發展過程中改名換姓。

如果這樣一項專利存在於1862年《悲慘世界》發表之時，該著作將會與上述全部三項專利要求相沖突，由於所有這些事情都發生在小說中的冉阿讓（Jean Valjean）¹一個角色身上。維克多·雨果可能會因此被起訴，而

¹這裡選用了人民文學出版社出版的《悲慘世界》（李丹、方於譯）對小說主人公的翻譯——冉阿讓。《悲慘世界》講述了主人公冉阿讓因偷竊麵包而獲刑長達十九年，出獄後被米里哀主教救贖，之後他多次改名換姓重獲新生，並試圖贖罪的故事。小說人物豐富，時間跨度大，涵蓋拿破崙戰爭和1832年巴黎共和黨人起義等政治現象。——譯者注

且一旦被起訴，他將會敗訴。這部小說可能會被禁止——事實上是被審查了——由專利擁有者。

現在考慮這樣一項假想的文學專利：

- 要求 1：一種在讀者腦海中表達了這樣一種概念的交流過程：一位角色在監獄中被監禁了很長時間，並且隨後改名換姓。

《悲慘世界》同樣會被這項文學專利禁止，由於這一描述同樣適用於冉阿讓的人生故事。現在還有這樣一項假想的文學專利：

- 要求 1：一種在讀者腦海中表達了這樣一種概念的交流過程：一位角色得到了道德的救贖並且隨後改名換姓。

冉阿讓同樣會被這項專利所禁止。

所有這三項專利都將覆蓋並且禁止這一角色的人生故事。它們相互重疊，但它們又都不是對其他專利的精確複製，因此它們可以共存；三項專利的持有人都可以起訴維克多·雨果，而他們中的任何一位都可以禁止《悲慘世界》。

這項專利同樣會被違反：

- 要求 1：一種交流過程表現了這樣一位角色，其名字與其姓氏的最後一個音節相同。

這是通過角色的姓名“冉阿讓”而產生的，然而至少這項專利是容易迴避的。

您可能認為這些想法過於簡單，以至於沒有任何一家專利機構會去批准它們。我們作為程序員通常也會對於真實的軟件專利所覆蓋的想法是如此地簡單而感到驚訝——例如歐洲版權廳（EPO）批准了一項關於進度條的專利，以及一項接受信用卡支付的專利。這些專利看起來將會顯得非常可笑，如果它們並非如此陰險。

《悲慘世界》的其他方面也可能與專利產生衝突。例如，可能有關於對滑鐵盧戰役進行小說化描寫的專利，或是關於在小說中使用巴黎俚語的專

利。這將帶來兩場新的法律訴訟。事實上，並沒有關於同時適用於對某一作品諸如《悲慘世界》的作者進行起訴的專利數量的限制，所有那些專利持有人都會宣稱他們理應得到獎勵，由於他們那些受專利保護的思想和靈感為文學發展所帶來的貢獻；然而，這些障礙並不能促進文學發展，反而只會阻礙文學的發展。

然而，一項非常寬泛的專利將會使得以上這些問題失去意義。考慮一項帶有寬泛要求的專利：

- 一種交流過程，帶有連續數頁的旁白結構；
- 一種旁白結構，有時類似於賦格曲或者即興作品；
- 一種圍繞特定角色之間的衝突所展開，隨後又依次為其他人佈下機關的陰謀；
- 一種表現了諸多社會階層的旁白；
- 一種揭示了隱藏的陰謀的旁白。

誰會是這些專利的持有人？可能是其他小說家，也許是大仲馬或者巴爾扎克，由於他們寫過這樣的小說——但並不一定。要想為軟件創意申請專利，其本人不需要進行編程，因此如果我們假想的文學專利符合真實的專利體系，其持有人不需要自己寫小說，或者故事，或者任何東西——除了專利申請。那些除了威脅和法律訴訟，沒有創造任何東西的專利寄生蟲公司、企業正在日漸繁榮。

如果有了這些寬泛的專利，維克多·雨果也許不會走到詢問他所使用冉阿讓這一角色將會受到哪些專利起訴這一步，由於他根本不會考慮去寫這樣一部小說。

這種類比可以讓非程序員看清軟件專利到底在做什麼。軟件專利覆蓋了功能，例如在文字處理器中定義縮寫，或是在電子表格中自然順序的重新計算。專利也覆蓋了程序設計所需要使用的算法。專利還覆蓋了文件格式的方方面面，例如微軟 Office 開放可擴展標記語言（OOXML），而 MPEG 2 視頻格式被多達 39 項美國專利所覆蓋。

正如一部小說可以同時與許多不同的文學專利相沖突，一個程序也可以同時被許多項軟件專利所禁止。想要鑑定所有可能與某個大型程序相關的專利的工作量如此之大，以至於只有一項這樣的研究被實施。在 2004 年關於 Linux，即 GNU/Linux 操作系統的內核，的研究發現 283 項不同的美國軟件專利可能對其進行覆蓋。也就是說，這 283 項專利中的任何一項都可能禁止可以在 Linux 的數千頁源代碼中的某處找到的計算過程。與此同時，Linux 內核只佔整個 GNU/Linux 操作系統的 1%。那麼，這樣一個系統的發佈者又將會受到多少項專利的起訴呢？

要想阻止軟件專利對軟件發展造成的危害的方法很簡單：不要批准它們。這應該是容易做到的，由於大部分專利法都擁有反對軟件專利的條款。它們通常會說“軟件本身”不能被專利保護。但是，全世界的版權機構正在試圖曲解其涵義並且為程序設計中實現的想法和創意提供專利保護。除非我們能夠阻止這一趨勢，這種趨勢將會將所有軟件開發者置於險境。

4.2 軟件專利的威脅

Copyright © 2009, 2010, 2014 Richard Stallman 此抄本最初於 2009 年發表於 <http://gnu.org>。

本文是 Richard Stallman 於 2009 年 11 月 8 日在位於新西蘭惠靈頓的惠靈頓維多利亞大學所做的演講的未編輯抄本。

我所做的最出名的事情是發起自由軟件運動以及領導開發 GNU 操作系統——儘管大部分使用它的人們錯誤地認為它是 Linux，並且認為它是由其他人在十年之後發起的。但是我今天並不打算談論與之相關的任何話題。我在此是想講述一種對所有的軟件開發者、發佈者和用戶的合法威脅：專利的威脅——不論是對於計算思想、計算技術還是對於您所能在計算機上進行的任何事情的想法。

現在，為了認識這個問題，您需要認識到的最重要的事情是，專利法與版權法沒有任何關係——它們是完全不同的。對於您所瞭解的關於它們之一的任何東西，您都可以確認它們並不適用於另一方。

於是，舉個例子，只要某人做出關於“知識產權”的論述，這就是在散播一種混淆。由於它不僅僅將專利法與版權法混為一談，並且還混入了至少數十種其他東西。它們都是不同的，其結果是，任何大意是關於“知識產權”的論述都是一種混淆——要麼做出這種論述的人本身處於混亂不清的狀態，要麼此人正在試圖使其他人混亂不清。但不論何種情況，不論是無意的還是惡意的，這都是一種混淆。

您必須拒絕接受任何使用那個短語的論述，才能使自己不被混淆。要想對於這些法律中的任何一個做出深刻的評價，並且進行清晰的思考，首先只有將其同其他東西區分開來，並且專注於討論或思考某一特別的法律。這樣，我們才能瞭解它真正在做什麼，並且對其得出結論。因此，我在此將會討論專利法，以及在那些允許專利法對軟件進行限制的國家發生了什麼。

那麼，專利到底在做什麼？一項專利是一種具體的、由政府批准的對

於某一種想法的使用權的壟斷。在專利中有一部分稱為權利要求，它精確地描述了您所不被允許做的事情（儘管它們被書寫為一種您可能不能理解的形式）。要想獲知那些禁令的真實涵義是一種費力的抗爭，它們可能擁有很多頁小字印刷的繁瑣細則。

專利的有效期一般會持續 20 年，這對於我們的領域是一段相當長的時間。20 年前還沒有萬維網——海量的計算機應用進入了一個在 20 年前甚至不可能被提出的領域。當然，人們在其上做的每一件事對於 20 年前都是新奇的——至少在某些方面是新奇的。因此，如果專利被申請了，我們會被禁止做所有這些事情，並且在那些蠢到擁有這樣的政策的國家，我們會被禁止做所有這些事情。

在大多數時間，當人們描述專利體系的功能時，他們會表現出對該體系的既定的興趣。他們可能是專利律師，或者他們在專利局工作，或者他們在某家行業大鱷的專利部門工作。總之，他們想讓您喜歡這種專利體系。

《經濟學人》雜誌曾將專利系統稱為“一種耗時的撞大運的事”。如果您曾經見過關於彩票的宣傳，您將理解它如何運作：它們反覆強調微不足道的獲勝機率，並且從不提及壓倒性的失敗機率。通過如此做，它們有意地、系統性地展示了關於什麼事情將會發生在您身上的帶有偏向性的場景，而它們事實上卻又並沒有針對任何特定的事實撒謊。

對於專利系統的宣傳與之類似：他們描述的是您走在街上，口袋裡裝著一紙專利證書的時候如何如何——或者先說獲得一項專利將會如何如何，然後您在口袋裡裝有一紙專利證書的時候如何如何，通常您可以從口袋裡拿出那張專利證書，用它指著別人說：“把你的錢給我。”

為了補償他們的偏見，我將會從另一方面進行描述——即受害者的角度——那些想要開發、發佈或者運行軟件的人們將會如何如何。您不得不擔心有朝一日，某人可能會向您走來，拿出一張專利證書指著您，說道：“把你的錢給我。”

如果您想要在一個允許軟件專利的國家開發軟件，並且您想要配合專利法，您將必須去做哪些事呢？

您可以嘗試列出一項關於您想要編寫的程序中可能找到的各種想法和

創意的清單。除此之外，您還需要面對這樣的現實：在您開始開始編寫這個程序的時候，您還不知道那個清單。但是，即使您已經完成編寫該程序，您也不可能列出這樣一份清單。

造成這種情況的原因是……在您以某種特定的方式想出一種計劃的過程中，您已經有了一種應用於您的設計的思想上的架構。正因為如此，它將阻止您去認識那些其他人可能用於理解同一程序的其他架構——因為您並不是首次想到它的；您已經在腦海中使用某種架構設計出了它。其他人在首次見到它的時候可能看到的是另一種架構，它涉及不同的想法和創意，並且對你來說看到那些其他的想法和創意是什麼是困難的。但是儘管如此，它們在您的程序中實現了，因此相關的專利將會禁止您的程序，如果那些想法是受專利保護的。

例如，假設有一些關於圖形創意的專利，並且您想要繪製一個正方形。當然，您可能會意識到如果有一項關於“底邊”的專利，它將阻止您繪製正方形。您可以將“底邊”置於您的繪畫中所可能實現的所有想法的清單之中。但是，您也許不會預料，如果某人持有一項“底角”專利，他仍然可以輕鬆地起訴您，由於他可以拿到您的繪畫並且將其旋轉 45 度。此時您的正方形看起來就有了底角。

因此，您不可能列出一項想法的清單，其中這些想法一旦受到專利保護就將禁止您的程序。

您可能會嘗試去做的是試圖找出所有可能在您的程序中使用的受專利保護的思想。然而，事實上您不可能做到這一點，由於專利申請在至少 18 個月內保密；其結果是專利局可以在這段時間內考慮是否批准一項專利，並且它們不會告知您。這不僅僅是學術上、理論上的可能性。

例如，一款名為 Compress 的程序於 1984 年被編寫出來，這是一款使用數據壓縮算法來壓縮文件的軟件，而當時並沒有關於那種文件壓縮算法的專利。該軟件的作者通過某家期刊上的一篇文章學到了該算法。當時，我們認為計算機科學期刊的目的應該是發佈算法以供人們使用。

此人編寫並且發佈了該軟件，並且該算法於 1985 年獲得專利。但是該專利擁有者非常精明，並不急於通告人們立即停止使用它。其專利擁有

者看到了這一點：讓每個人將自己的墳墓挖得更深一些。幾年之後，該專利擁有者開始威脅人們，很顯然，我們不能再使用 Compress，於是我請求人們推薦我們可用於壓縮文件的其他算法。

而後，有人在來信中說到：“我開發出了另一種效果更佳的數據壓縮算法，並且寫出了一段程序，我願意把它送給您。”於是，我們做好了發佈它的準備，就在它將要被髮布的一週之前，我特意閱讀了《紐約時報》週刊的專利專欄，我很少去閱讀它——可能每年只看幾次——但幸運的是，我看到了某人已經得到了一項關於“發明一種用於數據壓縮的新方法”的專利。於是我說我們最好看看這項專利。確實，它覆蓋了我們馬上就要發佈的軟件。但是事情還有可能變得更壞：這項專利可以再晚一年被批准，也許是兩年、三年，或者五年。

儘管如此，還是有其他人帶來了又一種更好的壓縮算法，這種壓縮算法被用於 gzip 軟件中，而幾乎每一個想要壓縮文件的用戶都轉向了 gzip，這看起來像是一個完美的結局。但是，您隨後將會知道，並非整個過程都是這樣美好。

因此，您不能獲知有關正在被評估的專利的任何信息，即使當它們粉墨登場的時候可能會禁止您的工作，但您可以獲知已被批准的專利的信息。它們會由專利局公佈，問題是您不可能全部閱讀它們，由於它們實在是太多了。

在美國，我相信總共有數十萬的軟件專利；對它們保持跟蹤是一項龐雜的工作。因此您將不得不隨時查詢相關專利。您將會找到大量相關專利，但您必然不可能找到它們的全部。

例如，在 20 世紀 80 到 90 年代，有一項關於在電子表格“重新計算自然順序”的專利。曾經有人向我索取一份它的副本，於是我在自己的列出了專利號的計算機文件中進行查詢，並且複印了一份發送給對方。對方收到後說：“我想您發給我的不是我想要的專利。這項專利是關於編譯器的。”於是我想可能是我們的文件中記錄了錯誤的專利號。我再次查詢它，並且確信它所描述的是“一種將公式編譯到對象代碼中的方法”。然後我開始仔細閱讀它，以便確認它是否真的並非我們想找的專利。我讀到了它的權

利要求，發現它確實是關於重新計算自然順序的專利，但它並未使用這些短語。它沒有使用“電子表格”這一短語。事實上，該專利禁止了數十種用於實現拓撲排序的方法——他們所能想到的所有方法。但是我並不認為它用到了“拓撲排序”這一短語。

因此，如果您正在編寫一份電子表格並且試圖通過搜索查詢相關專利，您可能會找到大量相關專利。但是，您並不會發現這樣一條專利，直到某一天您對某人說：“哦，我正在設計一份電子表格。”而對方說：“哦，你知道其他那些正在使電子表格軟件受到起訴的公司嗎？”然後您才會知道這一點。

您不能通過搜索找出全部相關專利，但可以找到它們中的許多。然後您必須獲知它們是什麼意思，這是很困難的，由於專利是用冗長並且含混不清的法律語言書寫的，很難理解其真正涵義。因此您將不得不花費大量時間和錢財向律師解釋您想要做什麼，以便從律師那裡獲知您是否被允許如此做。

即使是那些專利持有人，通常也不能完全認出他們的專利到底是什麼意思。例如，Paul Heckel 曾經發布了一個用於在小屏幕上顯示大量數據的程序，基於此程序中的一些思想，此人獲得了兩項專利。

我曾經試圖找出一種簡單的方式以描述其中一項專利的要求 1 所覆蓋的內容。然而，我發現我未能找到一種比其專利原文更簡單的解釋方式；而對於那條陳述，我不能試圖一次性將其全部裝入我的腦中，不論我多麼努力地進行嘗試。

然而，Heckel 也未能抓住其本質，由於當他看到 HyperCard 這一概念時，他所能發現的是這與他的程序全然不似。他並沒有想到他的專利書寫方式將會禁止 HyperCard；但是他的律師卻有這種想法，於是他威脅蘋果。然後他威脅了蘋果的用戶，最終，蘋果與此人達成了一項不為外人所知的和解方案，因此我們不知誰是真正贏家。因此，這一例子足以說明任何人想要獲知某項專利是否禁止某一事情到底有多麼困難。

事實上，我之前曾進行過這篇演講，並且 Heckel 也是聽眾之一。當演講進行到這個節點之時，他跳了起來，說道：“這不是真的，我只是不知道

我自己的專利所保護的範圍。”我於是說道：“這正是我所講的。”此時他坐下了，而這正是我被 Heckel 所詰問的經歷的結束。如果我當時說“不”，他很可能找到某種方式與我辯論。

儘管如此，在與律師進行過一段冗長並且昂貴的交談之後，律師很可能會給您這樣的解答：

如果您做了這個領域中的一些事情，您幾乎肯定會輸掉一場訴訟；如果您做了這個領域中的一些事情，您有相當大的機率輸掉一場訴訟；如果您真的想要確保安全，您必須避免涉足這個領域。但是，任何訴訟的結果都存在相當大的變數。

現在，您已經瞭解從事商業活動的清晰、可預測的規則。接下來，您實際上將會如何做？為了應對任何專利可能帶來的問題，您有三件事可以做。其一是迴避專利；其二是獲得它的使用許可；其三是使其作廢。接下來，我將會逐條討論。

其一，迴避該項專利是有可能的，也就是說，不要實施那些被它禁止了的東西。當然，如果很難區分什麼才是它所禁止的，可能也很難說出哪些方式足以迴避它。

兩年以前，柯達起訴 Sun，由於對方使用了一項與面向對象程序設計有關的專利，而 Sun 認為這並未侵犯該項專利。但最終法庭判決這構成專利侵犯；然而，其他人在看到那項專利之後，都難以得出有關該判決公正與否的哪怕是最微弱的想法。沒有人能夠區分什麼才是那項專利所覆蓋或者沒有覆蓋的，但是 Sun 不得不支付數億美元的賠款，由於它侵犯了一條完全不可理解的法律。

有時，您能夠區分什麼才是您需要迴避的；有時您需要迴避的是一種算法。

例如，我曾經見過某項專利是關於快速傅里葉變換（FFT）相關的東西，但它的運行速度快兩倍。當然，如果普通的 FFT 對於您的應用足夠快，那麼這將成為迴避這項專利的簡單方式。並且這將適用於大多數情況。而您一旦試圖做一些一直需要運行 FFT 的事情，並且那種快速算法勉強足夠

快，那麼您將不能迴避它，儘管您可以等上幾年以得到一臺更快的計算機。不過這畢竟是較為罕見的情形。對於大多數情形，那項專利還是容易迴避的。

另一方面，一項關於某種算法的專利也許是不可能迴避的。考慮諸如 LZW 數據壓縮算法。如我之前所述，我們發現了一種更好的數據壓縮算法，並且每個需要壓縮文件的用戶都轉向了用到更好算法的 gzip 程序。其原因是，如果您只是需要壓縮某些文件並且在以後進行解壓，您可以告知人們使用該程序進行解壓；然後您可以使用基於任何算法的任何程序，並且您只需關心該程序工作得如何。

但是 LZW 也被應用於其他事情；例如 PostScript 語言指定了用於 LZW 壓縮和 LZW 解壓的操作符。擁有其他更好的算法並沒有意義，由於這隻會產生一種新的數據格式，而它們將是不可互操作的。如果您使用 gzip 算法對其進行壓縮，您將不能使用 LZW 對其解壓。因此不論您的其他算法多麼好，不論它是什麼，它都不能允許您根據該算法的規範實施 PostScript。

但是，我注意到用戶極少會要求他們的打印機去壓縮什麼東西。一般情況下，他們需要打印機去做的只是解壓；而我也注意到，關於 LZW 算法的兩項專利都是以這種方式書寫，如果您的系統只能進行解壓，這並未被禁止。這些專利被這樣書寫使得它們覆蓋了數據壓縮，並且它們擁有同時覆蓋了壓縮和解壓的其他權利要求；但是並沒有僅僅覆蓋瞭解壓的權利要求。於是我意識到，如果我們僅僅實施 LZW 解壓，我們將會是安全的。儘管這樣做並不足以滿足標準規範，這已經足以取悅用戶；它將會只做那些它們實際上所需要去做的事情。於是這就是我們如何勉強僥倖地迴避了那兩項專利的經歷。

現在，有一種用於圖像的 GIF 格式，它也使用了 LZW 算法用於壓縮。人們並沒有花費太長時間就定義出了另一種圖像格式，它稱為 PNG，其涵義為 PNG 不是 GIF (PNG's Not GIF)。我認為它使用了 gzip 壓縮算法。並且我們開始對人們說：“不要使用 GIF，它是危險的。轉向使用 PNG。”而用戶則會說：“好吧，也許某天我們會這樣做，但是瀏覽器還沒有實施它。”並且瀏覽器開發者也會說：“我們也許會在某一天實施它，但是現在還沒

有太多的用戶需求。”

當然，事情的發展狀況是非常明顯的——GIF 是一種事實上的標準。實際上，要求人們從他們所習慣使用的事實標準格式轉向另一種不同的格式，就如同試圖要求每位新西蘭人改說匈牙利語。人們會說：“好吧，也許某一天我會去學著說匈牙利語，只要所有其他人都這樣做。”於是我們對於要求人們停止使用 GIF 的努力從未成功過。即使其專利持有者之一已經開始將矛頭對準網站操作者，並且威脅起訴他們，除非他們能證明其網站的所有 GIF 圖片都是使用經過認證和授權許可的軟件製作的。

於是，GIF 對於我們社區的大部分人來說都是危險的陷阱。我們認為我們有一種 GIF 格式的替代品，稱為 JPEG，但是其後就有人說：“我剛剛翻閱了我的專利公文包。”——我認為這是某些剛剛買了某項專利並且立即將其用於威脅眾人的人——並且他接著說：“然後我就發現其中一項專利覆蓋了 JPEG 格式。”

當然，JPEG 格式並非事實上的標準，它只是一種官方標準，由一個標準委員會頒佈；該委員會也有一位律師。他們的律師聲稱他並不認為那項專利真正覆蓋了 JPEG 格式。

那麼，究竟誰是正確的？這位專利持有人起訴了多家公司，並且如果有這樣的一種結論，它將會說出誰是正確的。但是，我未曾聽說過有這樣一種定論；我不確定是否真的有定論，我認為他們達成了某種和解，而最終的解決方案幾乎肯定是絕密的，這意味著它不能向我們傳達關於誰對誰錯的任何信息。

這些是相當輕量級的案例：關於 JPEG 的一項專利，關於 GIF 所使用的 LZW 算法的兩項專利。現在，您可能想知道為何關於同一種算法會有兩項專利。這本不應該發生，但卻就是發生了。其原因是專利審查者事實上不可能拿出時間來研究他們也許需要仔細研究和比對的每一對概念，這是由於他們不可能拿出那麼長的時間。並且由於算法實際上就是數學，您不可能定義清楚您需要比較哪些應用和專利。

您將會看到，在物理工程領域，他們可以使用將要發生的事物的物理本性來縮窄其適用範圍。例如，在化學工程中，他們可以說：“輸入的物質

是什麼，輸出的物質又是什麼。”如果兩項專利申請是以這種形式相互區別的，那麼它們並不是指的同一個過程，因此您不必為此擔心。但是，同一種數學原理可以被表現為看起來非常不同的形式，並且即使您將它們放在一起研究，您仍然不能意識到它們描述的是同樣的東西。並且由於這個原因，在軟件領域發現同一事物被多次賦予專利是非常普通的事情。

您還記得那個在我們準備發佈之前被一項專利扼殺的程序嗎？那種算法也被賦予了兩次專利。在一個小的領域中，我們已經看到這種情況發生在我們偶然遇到的兩起案例中——同一種算法被賦予兩次專利。好吧，我想我的解釋已經讓您明白這為什麼會發生。

但是，僅僅需要面對一兩項專利只是一種輕量級的案例。那麼，MPEG2 這種視頻格式的情況又如何呢？我曾見到過一份列出了超過 70 項覆蓋它的專利的清單，想要通過某種方式的談判獲得所有相關專利的使用授權許可所花費的時間超過了該標準本身的開發過程。JPEG 委員會想要開發一種後繼的標準，但他們還是放棄了。他們說這涉及的專利實在是太多了，以至於沒有可能實現、

有時，受專利保護的是某種功能特性，迴避該專利的唯一方式是不去實現那項特性。例如，文字處理器 Xywrite 的用戶曾經收到有關功能降級的郵件，這項降級移除了一項特性。該特性允許您定義一系列縮略語。例如，如果您將 exp 定義為 experiment 的縮略語，那麼當您輸入 exp 空格或者 exp 逗號時，exp 將會自動變為 experiment。

然後，此功能的專利持有者威脅他們，他們得出結論，所能做的唯一事情就是移除該功能，他們於是向所有用戶發送了降級。

但是，他們也聯繫了我，由於我編寫的 Emacs 文本編輯器自 20 世紀 70 年代後期就擁有一項類似功能。這是被寫在 Emacs 用戶手冊中的，於是他們認為我也許能夠幫助他們使那項專利作廢。我對於自己的一生之中能有至少一項可獲專利的想法而感到滿意，但我對其他人對它申請了專利並不高興。

事實上，幸運的是，那項專利最終被判無效，這在一定程度上得益於我先於該專利將其公開這一事實。然而在當時他們確實不得不移除了那項

功能。

現在看來，移除一兩項功能可能並不至於帶來一場災難。但是，如果您不得不移除 50 項功能呢？您自己可以這樣做，但人們很可能會說：“這個程序不怎麼樣，缺少我所需要的任何功能。”因此，這可能不能成為一種解決方案。並且，有時一項專利的覆蓋範圍是如此寬泛，以至於它橫掃了某一整個領域，例如關於公鑰加密的專利，它事實上使得公鑰加密在將近 10 年時間內幾乎被完全禁止。

以上說的是迴避專利的選項——這通常是可能的，但有時是不可能的，並且您最多能迴避多少專利是有限度的。

那麼，下一種可能性又當如何？即試圖獲得專利的使用授權許可。

首先，專利持有者也許根本不會考慮給您一份授權許可，這完全取決於他的意志。他可以說：“我就是要逼你停業。”我曾經收到某人來信求助，此人的家族企業當時正在經營博彩遊戲，當然是計算機上的，他被專利持有人威脅關閉他的整個企業。他將該項專利發送給我。其權利要求 1 是關於這樣的東西：“一種擁有多臺計算機的網絡，其中每臺計算機支持多種遊戲，並且允許多個遊戲會話在同時進行。”

現在，我能夠確信在 20 世紀 80 年代，有一所大學建立起了這樣一間計算機房，內有一系列聯網的工作站，每臺工作站都擁有某種窗口顯示功能。他們所需要做的全部事情就是去安裝幾種遊戲，那些計算機將會可能同時顯示多個遊戲會話。這是多麼地平淡無奇，以至於沒有人會閒得發佈一篇文章講述如何實現它。沒有人會有興趣來專門為此發表一篇文章，但是，這件事就是值得為其申請專利。如果這件事發生在您身上，您就將獲得對於這樣平淡無奇之事的壟斷權，然後您就可以用這項專利來迫使您的競爭對手停業。

但是，為何專利局會批准如此之多的在我們看來荒唐可笑而又窮極無聊的專利呢？

這並非由於專利審查者的愚蠢，而是由於他們必須遵守一種體系，這種體系有其規則，而這些規則必然引起這種結果。

您已經看到了，如果某人研究了一臺機器用於做一次某事，而其他人

研製了一臺能做同樣一件事的機器，但它會連續做 N 次。對我們來說這就是 for 循環，但對於專利局而言這就是一種創新。如果有某些機器可以做事 A，而有另一些機器可以做事 B，然後某人發明了一種機器可以做 A 或 B，對於我們來說，這就是 if-then-else 語句，但對於專利局而言這就是一種創新。也就是說，他們的標準如此之低，並且他們遵循這些低標準；其結果就是那些在我們看來荒唐可笑而又窮極無聊的專利。它們是否真正合法有效我不敢確定，但是每個程序員見了它們都會哈哈大笑。

在任何情況下，我都不能給他任何所能採取措施進行自救的建議，他最終被迫關閉了他的企業。但是，大部分專利持有者將會考慮提供給您一份使用授權許可。當然，這通常極其昂貴。

但是，也有一些軟件開發者覺得在大多數情況下都很容易獲得專利使用授權許可。它們就是行業大鱷。在任何領域，這些行業大鱷通常都會手握該領域的半數專利。它們之間可以進行交叉授權許可，它們也可以迫使任何人同它們進行交叉授權許可，如果此人真的打算生產任何東西。其結果是它們不費吹灰之力就可以獲得幾乎所有專利的使用授權許可。

IBM 曾在其自家雜誌，Think 雜誌上刊登一篇文章——我認為那是在 1990 年的第 5 期上——文中提到當時 IBM 通過其所擁有的約 9000 項美國專利（現在已有 45000 項甚至更多）所獲得的好處。他們說到其中一項好處就是他們藉此賺得盆滿鉢滿；然而，他們所認為的比上述好處還要大出一個數量級的最主要的好處是通過自身擁有的專利獲得其他公司所擁有的專利的使用權，即所謂的交叉授權許可。

這種情況意味著，由於 IBM 本身擁有如此之多的專利，它幾乎可以迫使任何人給它交叉授權許可。由此 IBM 得以迴避專利體系可能為任何其他人施加的幾乎所有災難。這就可以解釋為何 IBM 歡迎軟件專利。這也可以解釋為何行業大鱷普遍歡迎軟件專利。由於它們知道自己可以通過交叉授權許可獨佔整個行業的制高點，猶如在山頂上成立某種專屬俱樂部。而我們當中的所有其他人則位於山腳下，並且沒有辦法企及它們所在的高度。您應該明白，如果您是一位天才，您也許會創立一家小公司並且可能獲得一些專利，但您始終不可能進入 IBM 所在的行列，無論您做什麼。

現在，許多公司會對它的員工說：“請你為我們獲得專利，這樣我們就可以用這些專利保護自己。”它們的實際意思是“試圖用這些專利換取交叉授權許可”。但這並不能很好地解決問題。這也不是一種有效的策略，如果您僅僅擁有少數專利。

假設您擁有 3 項專利，其中一項指向那裡，另一項指向那裡，又一項指向那裡。而位於此 3 處以外的某處的某人將一項專利矛頭指向了您。此時您的 3 項專利不能為您帶來絲毫幫助，由於它們當中沒有任何一項指向此人。從另一個角度講，該公司裡的某些人遲早會注意到，這項專利實際上是針對某些人的，並且公司可以以此威脅這些人並且榨取其錢財——卻從未注意到這些人並未對公司構成威脅。

於是，如果您的僱主對您說：“我們需要專利來保護自己，請你幫助我們獲得專利。”我建議您如此回覆：

領導，我信任您並且確信您只會在公司受到威脅的時候使用這些專利來保護公司。但是，我不能確定 5 年以後誰將會是這家公司的首席執行官（CEO）。據我所知，這家公司可能會被微軟收購。因此我實在不能信任這家公司關於只會使用這些專利保護自己的口頭承諾，除非我能夠得到書面承諾。請您白紙黑字地保證我所為這家公司提供的任何專利都將只能被用於保護自己以及共同的安全，而非用於壓制他人，然後我才可能帶著良知去為公司獲得專利。

事情將會變得非常有趣，如果您不僅僅在同您的上司私下交談時提出這個問題；而是同時在公司的討論列表中提出。

另一種可能發生的事情是這家公司將會破產並且其資產將被拍賣，包括專利；而這些專利的買家將會蓄意使用這些專利去做一些齷齪的事情。

理解這種交叉授權許可的實踐是非常重要的，由於這種理解揭穿了軟件專利倡導者的論證，他們宣稱軟件專利是有必要的，這可以用於保護那些窮困潦倒的天才程序員。他們將您帶到了一種虛幻的場景，這裡有一些不太可能發生的事情。

現在讓我們著眼於這件事情。根據這種場景，有一位天才的設計者，他擅長設計任何東西，以他能夠以更好的方式去實現任何事情的天賦起家，獨立工作了若干年，現在時機成熟，他想要創辦一家企業以量產他的產品；並且由於他的創意是如此之高明，以至於他的公司不可避免地獲得成功——除了一件事情以外：行業大鱷將會與他競爭並且奪走他的全部市場份額。正由於此，他的企業幾乎肯定會破產，然後他將會窮困潦倒。

現在，讓我們來看看這裡的所有那些不太可能發生的假設。

首先，是關於此人以獨立工作的方式起家。這是一種不太可能發生的事情。在高科技領域，大部分進展是由在同一領域共同工作的人們共同取得的。但我並不想說這是不可能的，即任何一件事都不是由他獨立完成的。

但儘管如此，下一個假設是此人將會創立一家企業並且該企業將會獲得成功，這是不太可能的。這是因為此人是一位天才的工程師並不意味著此人在經營企業方面有任何優勢。絕大多數初創公司都失敗了；超過 95% 的初創公司，我想，將會在短短幾年內破產。因此這很可能就是將會發生在他身上的事情，不論他做什麼。

好吧，讓我們做一些附加的假設：一位天才的工程師以他自己的卓越的設計起家，並且此人也精通企業運營。如果他擁有關於經營企業的獨到本領，也許他的企業一時不會破產。不管怎麼說，並非所有的初創企業都破產了，確實有少數取得了成功。如果此人瞭解商業規則，那麼此人與其試圖與行業大鱷正面交鋒，不如試圖做一些小公司更加擅長的事情，這樣成功的機會更大。他也許能夠成功，但我們假設不管怎樣他還是失敗了。如果他真的如此有天賦並且有一套經營企業的本領，我相信他不會窮困潦倒，由於有些人可能願意給他一個職位。

這就是一系列不太可能的事情——這已經是一種不太現實的場景。但我們繼續著眼於它。

由於不管走到哪裡都會有這樣一種說法，專利體系將會“保護”那些窮困潦倒的天才，由於他可以獲得一項關於他的技術的專利。於是，當 IBM 想要同他競爭的時候，他會說：“IBM，你不能同我競爭，因為我擁有這項專利。”然後 IBM 會說：“哦，不，再也不會了！”

然而，這才是真正將會發生的事情：

IBM 將會說：“哦，真不錯，你擁有一項專利。可是我們擁有這項專利，還有這項專利，還有這項專利，還有這項專利，還有這項專利，所有這些專利覆蓋了你的產品中實施的其他思想。如果你認為你能夠對抗我們以上這些專利，我們將會拿出更多專利。這樣吧，咱們簽訂一份交叉授權許可協議，這樣誰都不會受到傷害。”現在，由於我們已經假設我們的天才懂得商業規則，他一定會意識到自己別無選擇。他將會簽訂交叉授權許可的城下之盟，如同 IBM 要求這個的時候每個人所做的。這意味著 IBM 將獲得他的專利的使用權，也就是 IBM 可以同他展開自由競爭，如同他沒有任何專利一般。這也就意味著他們所宣稱的他通過擁有這項專利所理應獲得的好處是不現實的，他不會得到這樣的好處。

專利也許確實能夠“保護”他不受來自您或者我的競爭威脅，但不能阻止來自 IBM 的競爭威脅——來自那些被此場景證實是對他的威脅的行業大鱷的競爭威脅。您已經事先知道，當那些效忠於行業大鱷的說客建議這樣一套政策，據說是由於它有利於保護小公司免於來自行業大鱷的競爭威脅時，其理由當中必然存在某種瑕疵。如果它真的將會產生其所宣稱的結果，他們不可能支持它。但這也解釋了為什麼軟件專利不可能達到這樣的目的。

即使是 IBM 也不能總是成功採取這樣的方式，由於還存在著一些我們稱之為專利流氓或者專利寄生蟲的公司，它們所從事的唯一業務就是在人們真正想要做一些事情的時候，跳出來使用其所掌控的專利來榨取他們的錢財。

專利律師對我們說：在你們的領域裡有專利的存在真的是一件美好的事情。但是，在他們的領域裡並沒有專利。那裡沒有關於如何書寫並寄出恐嚇信的專利，沒有關於如何發起一樁法律訴訟的專利，沒有關於如何說服法官或陪審團的專利。因此，即使是 IBM 也不能迫使專利流氓簽訂交叉授權許可協議。但是 IBM 算清了：“我們的競爭對手也必須向它們付錢；這只是進行商業活動的成本的一部分，我們能夠承受。”IBM 和其他行業大鱷得出這樣的結論：通過其所擁有的專利能夠獲得對於所有商業活動的

普遍統治地位，扣除付給專利流氓的保護費後仍然能夠承受。這就是它們歡迎軟件專利的理由。

確實也有軟件開發者發現他們很難獲得一項專利的使用授權許可，這些人是自由軟件的開發者。其原因是通常的專利授權許可協議包含我們完全不可能接受的條款：由於通常的專利授權許可協議要求按照再分發的副本數量付費。但是由於自由軟件賦予了用戶再分發和複製的自由，我們沒有辦法統計總共存在多少副本。

如果某人向我提供了一項專利使用授權許可，要求為每份副本支付一百萬分之一美元的費用，我需要支付的總金額現在也許能夠裝在我的口袋中，也許是 50 美元，但我不能確定到底是 50 美元，還是 49 美元，還是其他金額，由於我不可能確定人們所複製的副本數量。

還有些專利持有人不願意按照再分發的副本數量收費；他可以向您開出提供專利使用授權許可的一次性總價，不過這樣的總價通常是很高的，例如 10 萬美元。

我們之所以能夠開發出這麼多的尊重用戶自由的軟件，其原因是我們可以在沒有錢的條件下開發軟件，但我們不能在沒有錢的情況下支付一大筆專利授權許可費用。如果我們被迫花錢以換取為公眾編寫軟件的權利，我們不可能在這方面有所建樹。

以上就是關於獲得專利使用授權許可的可能性。還有一種可能性是試圖使專利作廢。如果國家將軟件專利視為大體有效的並且允許批准它們，那麼唯一的問題是某項特定的專利是否符合評估標準。只有當您擁有壓倒性的證據的時候，您前去法庭才有意義。

那麼，到底需要的是什麼樣的證據呢？您必須找到證據以證明早在數年之前，即那項專利被申請之前，人們已經瞭解相同的思想。您必須找到今天仍然存在的東西以證明當時人們已經普遍知道這種想法。因此，骰子已經於數年前被擲出，如果擲骰子的結果在今天看來仍然對您有利，並且您能夠在今天證實當時的事情，那麼您將擁有可能用於嘗試推翻該專利的證據，這也許能夠成功。

打完這場官司也許會花費您很多錢。其結果是，一項很可能是無效的

專利仍然是可用於威脅您的可怕武器，如果您沒有那麼多的錢。有人就付不起錢來捍衛他們的權利——非常多的人。當然，那些能夠付得起這筆錢的人除外。

以上三件事就是當任何專利禁止您的程序中的某些東西的時候，您所能採取的措施。問題的關鍵是，其中的任何一種方法是否可能，取決於不同的環境細節。因此在有些時候，它們都是不可能的；當這種情況發生時，您的項目已經死了。

但是，大多數國家的律師會這樣對我們說：“不要想著事先找到相關專利。”其理由是如果您已知某項專利，侵犯它的罰金會更高。於是他們所傳達給您的無外乎是：“閉上眼睛，不要試圖查找專利，只要盲目地實施你的設計決定，然後去撞大運。”

當然，對於每次單一的設計決定，您可能不會觸碰專利，也許您不會遇到任何麻煩。但是，您需要邁出那麼多步才能走出雷區，以至於您想要毫髮無傷全身而退是非常不現實的。另外顯而易見的是，那些專利持有人不會一下子全都現身，於是您不可能知道到底將會遇到多少專利持有人。

電子表格中的自然順序重新計算方法專利的持有人要求按照每份電子表格銷售總額的 5% 支付費用。您可以想象為少數幾項類似的專利使用授權許可付費，但是，當第 20 位專利持有人前來拜訪，要求您將最後剩下的 5% 的錢用於支付專利使用授權許可費用的時候又當如何呢？而第 21 位專利持有人前來拜訪的時候又當如何呢？

商務人士可能會說這種場景雖然有趣但卻是荒唐的，由於在您走到那種境地之前，您的企業早就破產了。他們告訴我，只要兩三項類似的專利使用授權許可費用就足以讓您的企業破產，於是您不會等到第 20 位專利持有人。由於他們一個接一個地現身，您不可能知道還會來多少位。

軟件專利是一團混亂，它對於軟件開發者是一團糟，但除此之外，它們是對每一位計算機用戶的限制，因為軟件專利限制了您可以用您的計算機去做什麼事情。

這與其他領域的專利是非常不同的，例如關於汽車引擎的專利，它們只限制汽車製造商，而不會限制您和我。但是，軟件專利確實在限制您和

我，以及所有使用計算機的人們。因此我們不能將其僅僅作為經濟概念考慮；我們不能僅僅從單純的經濟角度評估這個問題。這裡有更重要的，生死攸關的事情。

但是，即使只是在經濟方面，這種體系也是自相矛盾的。由於它的初衷是促進發展。據說它想要通過創造這種人為的激勵機制以鼓勵人們發表想法，它將會促進這個領域的進步。但是，它所產生的實際效果恰恰與之相反，因為軟件開發過程中的複雜工作不是隨著想法信手拈來的，它需要在一個程序中實施數千種想法。而軟件專利阻礙了這一過程，因此它們從經濟角度上講也是自相矛盾的。

甚至還有經濟學研究證實事實確實如此——這些研究結果顯示，在一個擁有大量增量創新的領域，專利體系確實會減少研發投入。當然，它也會通過其他方式阻礙發展。因此，即使我們無視軟件專利所帶來的不公，甚至如果我們僅僅狹隘地從它們通常被提議的經濟方面考察軟件專利，它仍然是有害的。

人們有時會以這種觀點作為迴應：“其他領域的人們已經與專利共存了幾十年，他們已經習慣了專利的存在，為什麼你就應該成為例外？”

在這裡，這個問題當中包含了一種荒唐的假設，這就如同說：“其他人都患了癌症，為什麼你就應該倖免？”我認為無論如何，人們不患癌症才是好的，不管其他人如何。這種問題之所以荒唐，是由於它預設了這樣的觀點：無論如何我們都必須有義務去忍受專利對我們造成的傷害。

但是，這其中又蘊含著一個合理的問題，這個合理的問題是：“在不同的領域之間，究竟存在著什麼方面的不同，以致於這種不同將會影響到專利政策在這些領域中是好的還是壞？”

在不同領域之間，確實存在著某種基本的重要差別，即任何一款產品的組成部分當中有可能被多少項專利禁止或覆蓋。

現在，我們的腦海中可能有這樣一種天真的想法，這是我正在盡力克服的，由於它不是真實的。這種想法是在任何一款產品的背後都只有一項專利，而這項專利覆蓋了這款產品的全部設計理念。因此如果您設計一款新產品，它不能是已獲專利的，並且您將有機會獲得關於該產品的“那項

專利”。

事情並不是這樣的。也許早在 19 世紀確實是這樣，但現在則不是。事實上，領域之間可以像光譜那樣按照每個產品對應多少項專利來劃分。這種光譜的起始點是 1，但如今已經沒有那樣的領域；當今的領域分佈在光譜上的不同位置。

最接近這種情況的領域是製藥。幾十年前，確實每種藥物只有一項專利，在任何時候都如此，由於該專利覆蓋了一種特定物質的全部化學分子式。在當時，如果您開發了一種新藥，您可以確認它沒有被任何其他人申請專利，並且您可以獲得該藥物的唯一專利。

但是現在的情況與之不同。現在有了更寬泛的專利，於是現在您可能開發一種新藥，但是您不被允許生產它，由於某人已經擁有一項覆蓋了它的寬泛的專利。

也許甚至會有幾項專利同時覆蓋了您的新藥，但不會是多達數百項。其原因是，我們進行生物工程研究的能力還相對有限，沒有人知道如何將這麼多的思想組合在一起以生產出在醫學方面有用的物質。如果您能夠組合其中的兩種，您的成就對於我們的知識水平已經非常了不起。但是，其他領域將會涉及將眾多想法組合起來以做成一件事。

這幅光譜的另一端是軟件領域。在這裡，我們比任何其他人都能將更多的想法融入一項有用的設計，由於我們的領域從根本上比任何其他領域都更簡單。我假設我們的領域中的人們的智力與物理工程領域的人們的智力相當。這不是說我們從根本上比他們更有能力；這只是在說我們的領域從根本上說更加簡單，因為我們是用數學來工作。

程序是由眾多數學的成分構成的，這些數學成分擁有某種定義，而物理對象是沒有定義的。物質會按其規律發生作用，這是由於物質的本性，而您的設計不一定會按照它們“應當”採取的作用方式發生作用。這只是一種困難。您不能說物質有錯誤，而物理宇宙應當受到修復。而我們程序員可以在一條數學中的沒有粗細的線上建起一座城堡，它能夠屹立不倒，因為它裡面的任何東西都沒有重量。

在物理工程中，您必須解決眾多複雜性，而我們無需為之擔心。

例如，當我將一個 if 語句置於一個 while 循環中時：

- 我無需擔心如果 while 循環以錯誤的頻率重複，其中的 if 語句可能將會開始振動，它也許將會由於發生共振而斷裂；
- 我無需擔心如果它們的共振頻率過快——您知道，大約每秒數百萬次——以至於它將會生成無線電頻率信號並由此導致程序中的其他部分產生錯誤的值；
- 我無需擔心環境中的腐蝕性液體可能會滲入 if 語句和 while 語句之間的縫隙並且開始侵蝕它們直到信號再也不能被傳遞；
- 我無需擔心 if 語句產生的熱如何才能傳導至 while 語句以外，以保證這不會使得 if 語句過熱燒燬；並且
- 我無需擔心我應當以何種方式移除受損的 if 語句，如果它的確斷裂、燒燬或者被腐蝕了，並且將其更換為另一個完好的 if 語句以使得程序再次能夠運行。

基於此原因，我無需擔心在我每次為程序複製一份副本的時候應當以何種方式將 if 語句插入到 while 語句中。我無需設計一座工廠來複製我的程序，由於幾個通用的命令就能用於複製任何東西。

如果我想要在光盤（CD）上製作副本，我只需刻錄一片母盤；有一種程序使得我可以將其用於為任何東西製作母盤或者燒錄任何我需要寫入的數據。我可以製作一片母盤，刻錄之後將其送至一座工廠，他們將會複製我所發送的任何東西。我無需為我想要複製的每件不同的東西設計一座不同的工廠。

而對於物理工程，您通常不得不去做這些事情：您必須基於可製造性來設計產品。設計工廠也許甚至比設計產品的任務更為艱鉅，並且而後您可能還必須花費數百萬美元建廠。由於以上這些困難，您將不能將如此之多的想法融入一款產品並且使之可用。

一項擁有一百萬項不重複的設計元素的物理設計是浩大的，而一個擁有一百萬項設計元素的程序則再普通不過。它只是數十萬行代碼，幾個人

可以在幾年之內完成，因此它不是什麼大事。其結果是，專利體系對我們造成的壓力相比之下更重，相對於那些在其他領域工作，被物質的本性所阻擋的人們。

一位律師曾經研究過一個特定的大型程序，也就是 Linux 內核。它與我所發起的 GNU 操作系統配合使用。這是在 5 年之前的事情；他發現共有 283 項不同的美國專利，其中每一項看起來都會禁止在 Linux 代碼中的某處進行某種計算。與此同時，我看到的一篇文章稱 Linux 約佔整個 GNU 操作系統的 0.25%。因此，將 300 乘以 400，我們便可預計出可能禁止了整個系統中的某些東西的專利數量約為 10 萬。這是一個非常粗略的估計，沒有更加精確的信息了，由於試圖弄清這個問題將會是一項過於龐雜的任務。

現在，這位律師並未公佈相關專利的清單，由於這將威脅 Linux 內核開發者，將其置於一種一旦被起訴將面臨更高罰金的境地。他並不想傷害他們；他只是想展示，關於專利困局，問題究竟有多麼嚴重。

程序員可以立即理解這些，但是政治家通常對編程知之甚少；他們通常想象專利大體上就像版權，只是略強一些。他們想象既然軟件開發者沒有受到關於他們的工作的版權的威脅，於是他們也不會受到關於他們的工作的專利的威脅。他們想象既然當您編寫一個程序的時候您可以獲得它的版權，與之相似地，當您編寫一個程序的時候也將獲得它的專利。這是不正確的——那麼我們怎樣才能給他們一條線索，使他們明白專利真正將會造成什麼後果呢？它們在像美國這樣允許軟件專利的國家裡究竟造成了什麼後果呢？

我發現將軟件和交響樂進行類比是有用的。以下是為什麼這是一種好的類比的理由：

一段程序或者一段交響樂都將集合諸多靈感創意。所不同的是，交響樂所彙集的是眾多音樂靈感。但是，您不能只是簡單地拎起一串靈感並且說：“這是我的靈感集合，你喜歡嗎？”由於為了使它們有意義，您必須去實現這些靈感。您不能只是挑選若干靈感，列出清單並且說：“嗨，你到底有多麼喜歡這種靈感組合？”您不能將那份靈感清單當成音樂來聽。您必

須寫出音符來將這些靈感一起實現。

而這項我們當中的大部分人可能完全不擅長的艱鉅任務，是寫出全部所需的音符以使得所有這一切悅耳動聽。當然，我們中的很多人都能從音樂靈感清單中挑選一些，但我們並不知道如何創作出動聽的交響樂來實現這些靈感。我們當中只有一些人擁有這樣的天賦。正是這件事限制了您。我也許可能發明少數音樂創意，但我不知道怎樣使用它們以產生任何效果。

假設現在是 18 世紀，歐洲各國政府決定它們想要通過創立一種音樂靈感專利體系來促進交響樂的發展，於是任何以文字形式描述的音樂靈感都可以被專利保護。

例如，將某一特定序列的音符用於主題可以被專利保護；或者某種和絃進程可以被專利保護；或者某種旋律結構可以被專利保護；或者由其本人使用某些特定樂器可以被專利保護；或者音樂行進過程中的某種重複格式可以被專利保護。總之，任何類型的音樂靈感只要能被文字描述皆可獲得專利。

假設現在是在 19 世紀，您是貝多芬，並且您想要創作一首交響樂。您將會發現想要創作出一首使您不會受到專利起訴的交響樂遠比創作出一首動聽的交響樂更加困難。由於您必須繞過所有已存在的專利。如果您對此不滿，專利持有者將會說：“哦，貝多芬，您看起來只是對我們先於您擁有這些靈感而感到嫉妒，您為何不走開，並且回去想出一些屬於您自己的靈感呢？”

現在，貝多芬有了自己的靈感。他被稱為一位偉大的作曲家是由於他所擁有並且實際使用的所有靈感。並且他還知道如何使用這些靈感，使得它們能夠發揮作用，即將他們與眾多為人們所熟知的靈感相結合。他可以将少數創新的靈感與眾多古老的並且不會引起爭論的靈感一同融入創作。其創作結果是一段可能引起爭議的樂章，但還沒有達到足以使得人們難以適應的程度。

對於我們來說，貝多芬的音樂聽起來並不那麼容易引起爭議；我聽說他的作品曾經是引起爭議的，在它仍然新穎的時候。但由於他將其創新的靈感與眾多已知的靈感相結合，他才給人們以延伸其境界的機會，他們也

能夠這樣做，這正是為什麼對我們來說那些靈感其實很好的原因。但是，沒有人能夠，即使是貝多芬也不能，成為這樣的天才，其天賦足以使他能夠從零開始重新發明音樂，而完全不去使用任何已經為人們所熟知的靈感，並且創造出人們想要欣賞的東西。而且也沒有人能夠成為這樣的天才，其天賦足以使他能夠從零開始重新發明計算機，而完全不用任何已經為人們所熟知的靈感，並且創造出人們想要使用的東西。

在這個技術環境變化如此頻繁的時代，您終將達到這樣一種境地，20 年之前所實現的東西現在完全不足以勝任需求。20 年前根本沒有萬維網。當然，人們在那之前也曾使用計算機做過很多事情，但他們今天想要做的事情是那些能夠與萬維網協同工作的事情。並且您不可能只用 20 年前為人們所知的想法去做這些事情。並且我假設技術環境將會持續發生變化，從而創造出嶄新的機會使得某些人可以得到那些欺騙了整個領域的專利。

行業大鱷甚至可以自己來做這件事情。例如幾年之前，微軟決定設計一種用於文檔的偽開放標準並且通過賄賂腐化國際標準化組織（ISO）使其被批准為一項標準，他們的確做到了。但是微軟在設計它的過程中使用了一些受微軟自家的專利所保護的東西。微軟足夠強大，使得它可以從一項專利起手，使用這種受專利保護的想法來設計一種格式或協議（不論其是否有益），這樣一種設計方式使得沒有任何方法可以與其兼容，除非您使用完全相同的設計思想。然後微軟可以使其成為事實上的標準，不論能否得到已經腐化墮落的標準化組織的助紂為虐。僅憑其自身的地位，微軟就可以迫使用戶使用那種格式，這就基本上宣告微軟已經扼住了全世界的咽喉。因此，我們需要向政治家說明這將會真正造成什麼後果。我們需要向他們證明為什麼這絕不是一件好事。

現在我聽說新西蘭正在考慮軟件專利的的原因是一家行業大鱷想要藉此被賦予某種壟斷地位。以限制國家裡的每一個人的自由的方式來讓一家商業公司大發橫財，這是完全違背治國理念的。

4.3 保護軟件領域免受專利困擾

Copyright © 2012, 2013 自由軟件基金會，本文的一個版本最初以標題 “Let’s Limit the Effect of Software Patents, Since We Can’t Eliminate Them” 發表於《連線》雜誌 Wired 網站 (Wired, 2012 年 11 月 1 日。 <http://wired.com/opinion/2012/11/richard-stallman-software-patents/>)。該版本於 2012 年發表於 <http://gnu.org>

另請參閱我的文章 “Patent Reform Is Not Enough” <http://gnu.org/philosophy/patent-reform-is-not-enough.html>。

專利會對每位軟件開發者構成威脅，我們長久以來一直擔憂的專利之爭終於打響了。軟件開發者和軟件用戶——我們社會中的大多數人——需要保證軟件不受專利困擾。

對我們構成威脅的那一類專利通常稱為“軟件專利”，但這個短語是帶有誤導性的。這些專利並非針對任何具體的程序。與之相反，每一項專利描述了某種實踐上的想法和創意，並且宣稱任何人試圖實現這樣的思想將會被起訴。因此，將它們稱為“計算思想專利”更清楚。

美國專利體系並不會為每一項專利貼標籤，稱這項專利屬於軟件專利而那項不屬於。事實上，是由軟件開發者來為我們區分那些將會對我們構成威脅的——那些覆蓋了可能在軟件開發過程中實現想法的專利——以及其他的專利。例如，如果一項專利所保護的想法是關於一種物理結構或者一種化學反應的，沒有程序可以實現那種想法；這樣的專利並不對軟件領域構成威脅。但如果一項專利所保護的想法是一種計算過程，那麼該專利的炮筒是對準軟件開發者和用戶的。

這並不意味著那些計算思想專利僅僅會禁止軟件。這些想法也會被實施在硬件中——並且它們中的很多已經被實施在硬件中。每項專利通常同時覆蓋關於某種想法的硬件和軟件實現。

4.3.1 關於軟件的特別問題

時至今日，軟件領域仍然是計算思想專利會造成特殊問題的地方。在軟件領域，在一個程序中實施數千種想法是很平常的。如果它們中的 10% 受到專利保護，這意味著該程序受到數百項專利的威脅。

當供職於公共專利基金會（PUBPAT）的 Dan Ravicher 於 2004 年對一款大型軟件（Linux，即 GNU/Linux 操作系統的內核）進行研究時，發現了 283 項美國專利可能覆蓋了該程序中實施的計算思想。同年，一家雜誌預計 Linux 約佔整個 GNU/Linux 操作系統的 0.25%。將 300 乘以 400，我們可以得到這樣的數量級估計，即該系統作為一個整體正在受到約 10 萬項軟件專利的威脅。

如果這些專利中的半數由於“品質不良”——即由於專利系統本身的問題——而被判無效，這並不能真正帶來改觀。不論是 10 萬項專利還是 5 萬項專利，都是同樣的災難。這是為什麼不應該將我們對於軟件專利的批評侷限於“專利流氓”或“劣質專利”的層面的原因。當今最壞的專利侵略者是蘋果，它不僅僅是通常意義上的“專利流氓”；我不清楚蘋果的專利是否屬於“優質”，但是那些專利的“質量”越好，其所帶來的威脅就越大。

我們需要解決全部的問題，而非部分。

關於解決這一問題的常見建議包括在立法層面修改專利授予的評價尺度——例如，禁止對計算實踐以及實現它們的系統進行專利保護。這種方式有兩個明顯的缺陷。

其一，專利律師精於將專利換一種方式表達，使其能夠適合任何規則；他們將任何旨在限制專利實質的嘗試化解為僅僅是形式上的要求。例如，美國的很多計算思想專利描述了這樣一種系統，它包括一個算術單元、一個指令排序器、一塊內存、加上用於執行特定計算過程的控制器。這是對於一臺計算機運行一個程序以進行某種計算的一種古怪描述；它是被設計用於使專利申請滿足那些美國專利體系在一段時間內要求滿足的評估標準。

其二，美國已經有了數千項計算思想專利，更改評估標準以阻止批准更多的專利並不能幫助我們擺脫已有的那些。我們不得不上將近 20 年，

才能等到這一問題隨著那些已有專利過期而徹底解決。我們可以展望立法廢除那些現存的專利，但這很可能是不符合憲法的（美國最高法院不顧一切地堅持要求國會有權以犧牲公共權利為代價擴展私人特權，而不是相反）。

4.3.2 另一種方式：限制專利的效力，而非專利的可獲得性

我的建議是改變專利的效力。我們應當立法規定，在通用計算硬件上開發、發佈、運程序並不構成專利侵犯。這樣有幾個好處：

- 不再需要區分專利或者專利申請到底屬於“軟件專利”還是“非軟件專利”；
- 可以對開發者和用戶提供保護，不論對於現存的還是將來潛在的計算思想專利；
- 專利律師不能通過以不同的方式編寫專利申請來抵抗這種法律

這種方式並不會使現存的計算思想專利完全失效，因為它們仍然可以被應用於使用特殊用途硬件的實現。這是一種好處，由於它打消了有關這一計劃的合法性的爭論。幾年之前，美國曾經通過一項法律，保護外科醫生不受專利起訴，因此即使一些外科手術流程受專利保護，外科醫生仍然是安全的。這樣的實踐為這種解決方案提供了先例。

軟件開發者和軟件用戶需要得到對於專利威脅的保護。這是唯一能夠提供對所有人全面保護的法律解決方案。然後我們可以重新回到競爭或合作……並且無需擔心某些陌生人會抹除我們的工作成果。

第 5 部分

自由軟件許可證

5.1 許可證簡介

Copyright © 2010 自由軟件基金會。由 Brett Smith 和 Richard Stallman 撰寫。

這部分包括了最新版的 GNU 主要許可證：GNU 通用公共許可證（GNU GPL），GNU 寬通用公共許可證（LGPL）以及 GNU 自由文檔許可證（FDL）。雖然都是法律文書，依然放到本書裡是因為這些是自由軟件理念的具體呈現形式。

GNU 操作系統的軟件開發始於 1984 年。當時理查德·斯托曼的一部分 GNU 系統組件已經可以發佈，它需要一份許可證來保護。當時很多自由軟件許可證已經存在，這些給了用戶修改和再分發的權限，但同時也允許代碼可以用於專有版本和程序。使用這些許可證，GNU 可能無法再向所有用戶帶去自由，因為中間人可以將 GNU 代碼放到專有軟件裡。

因此斯托曼設計了一款許可證來保證每個用戶都有修改和再分發軟件的自由。它通過一個關鍵條件來授予這些權限：無論誰發佈軟件都必須將修改和再分發的授權傳遞下去，同時包含源代碼以完成這個目標。斯托曼

創造了“Copyleft”（參見《什麼是 Copyleft?》(p. 222) 一文）來表述這個關鍵轉折，即使用法律權利，來保護所有用戶的自由。

GNU 的 Copyleft 許可證最初的發行是針對軟件，之後拓展到了相關領域比如軟件文檔。這些將自由軟件運動的原則放到了實踐領域，也都在本書中有解釋。這些許可證的每一次修訂都體現了自由軟件法律與實際障礙之間的鬥爭，並體現了自由軟件理念是如何變成法律條款的。

5.1.1 原始版本的 GPL

第一版 GNU 通用公共許可證發佈於 1989 年——然而早在 1985 年，斯托曼就已經在用 copyleft 許可證發佈 GNU 工程中的一部分軟件了。到 1989 年時，每一個已經發布的 GNU 程序都有其自己的許可證。比如 GNU CC 通用公共許可證，GDB 通用公共許可證等等，而不是統一的 GNU 通用公共許可證。這些許可證除了細微差別幾乎一樣：比如給用戶顯示許可證時提示的信息只是不同的程序不同而已，除非該程序只包含一個源文件，否則許可證都會包含該程序的名字。

到 1989 年，斯托曼已經在不同 GNU 軟件包的許可證上有足夠多的經驗，因此一個關鍵問題是將這些許可證集合在一起，以便能覆蓋到所有軟件包。他與在帕金斯史密斯和科恩律師事務所供職的律師傑裡·科恩（Jerry Cohen）一起，將之前所有這些許可證的理念彙集起來，並將之合併成一個許可證。這樣 1989 年 2 月 1 日，GNU 通用公共許可證誕生了。

第一版的許可證力求達到兩個成果：第一，所有派生軟件都可以使用相同許可證，第二，獲得該軟件的任何人都有機會獲得源代碼。這就要求實現強而有力的 Copyleft 以避免這三種使程序專有化的方式：版權，最終用戶許可協議以及不發佈源代碼。

與之前特定程序的許可證相比，GPL 第一版有很多實質性變化——GPL 是一種進化而不是革命——然而確實有實質性不同。之前，開發者若想發佈一個 copyleft 的程序需要修改既有的許可證，許多人都沒有這麼做。GPL 發佈以後，這些開發者有了直接就可以用的許可證，給了所有用戶分享和修改軟件的自由。這是個強而有力的工具。

5.1.2 第二版

1981 年美國最高法院對 Diamond 訴訟 Diehr 一案做出判決之後，美國專利和商標局開始為軟件發放專利。軟件專利既侵害了自由軟件也侵害了專有軟件（參見本書第四部分），斯托曼意識到需要將 GNU GPL 裡的 copyleft 做大幅修改。

通過選擇性地發放專利許可，專利持有人可以隨心所欲地控制如何根據它們的軟件分發或修改。專利持有人可以給一部分人轉售程序的權限，給另一部分人在其公司開放和使用修改版的權限，而給另一部分人符合 GPL 允許的活動。為換取這些權限他們會獅子大開口。他們將權力凌駕於任何想實現這個軟件專利的人，無論他們自己是否開發或分發軟件。而這個權力會損害自由軟件，因為專利持有的第三方可以給自由軟件的用戶和開發者強制增加限制。

如果一個專利持有者沒有分發或修改軟件，那麼一個使用 GPL 這種 copyleft 許可證的軟件是不能控制其行為的：他們沒有索要任何許可證上的權限。但是專利持有人可以通過在軟件許可證中增加限制性條款的方式停止每一個程序發行商的行為。而 GPL 第二版中增加了一個新的許可證章節（第七節），明確說如果要增加新的法律許可——比如專利許可證——與 GPL 條款衝突的話，那麼這個許可證必須在軟件發行時完全撤銷。結果就是，任何想發佈和修改軟件的人都會獲得一份專利許可，而條款必須保證與 GPL 條款完全兼容：這樣軟件的使用者必須收到相同的條款，沒有額外限制，同時也意味著獲得源代碼。

這一章節力圖保護 GPL 軟件的完整性。這個許可證的基本原則是希望覆蓋從卑微個體到大型企業都有相同的分享和修改軟件的權利。那些不分發軟件的專利持有者有選擇地提出專利許可，會潛在干擾這一目標的實現，雖然將許可條款分開，但他們卻認為是合適的。GPL 第二版的第七部分在力圖避免這種情況。

5.1.3 LGPL

在 GNU 工程早期，GPL 許可證在編程工具、實用程序和遊戲方面都可以很好地被使用。然而，斯托曼發現以同樣的方式發佈新開發的 GNU C 庫的時候會事與願違。除了一些擴展，GNU C 庫可以看成是 Unix C 庫的兼容性替代，因此任何 C 程序都可以與這兩個任何一個鏈接。而如果一個專有的 C 程序不能允許使用 GNU C 庫的話，他們就只能去使用 Unix 庫。如此嚴格卻沒有什麼好處。

斯托曼決定通過修改一個 copyleft 許可證讓步：既能保護庫本身的自由，又不能限制程序使用。這個想法最開始被稱為 GNU 庫通用公共許可證（GNU Library General Public License），於 1991 年六月首發 2.0 版。最初的 LGPL 和 GPL 相似，只有一處例外：如果其他人只是以庫的形式使用這個該庫，那麼程序代碼的許可證可以按照作者的選擇。然而，可執行程序與庫結合以後需要同時發佈 LGPL 許可證文本和庫的源代碼，並需要為用戶提供一些機制，讓用戶修改以後的庫可以更新可執行程序。

如何讓開發者在使用庫的時候享受 LGPLv2 的好處呢？若把計算機程序想成執行特定工作的一系列指令：程序與庫一起編譯或鏈接就相當於說，“當程序執行到這一點，其他指令可以去庫裡找，執行完後再回到這裡”。在軟件開發中，使用庫是非常普遍的，這樣可以減少的大量重複開發，同時也減少了錯誤的發生：程序員不需要重複造輪子——也可以減少在此過程中產生的問題。正是因為庫的廣泛創建和使用，開發者可以很容易地利用 LGPL 的額外權限。

2.0 版許可證的初衷是：在一些情況下，專有軟件開發者使用 LGPL 保護庫而不是專有庫，這樣用戶就可以自由地去分享和修改庫。而這並沒有產生“理想”的結果——用戶並沒有完全控制程序——當然 GPL 也不能達到理想的結果。LGPL 需要保障用戶在其他地方沒有的自由。

許可證名字裡的“庫”導致一些自由軟件開發者假設所有的庫原則上都應該用此許可證，但這不是設立此許可證的初衷——當一個自由的庫沒有專有競爭者的時候，將其以 GPL 許可證發佈會對自由軟件有益。為了避免造成的誤解，斯托曼將許可證改名為“寬通用公共許可證”，並增加版本

到 2.1 表示這只是文本上的小幅修改：許可證增加了一些前言，一小段聲明，允許程序通過特定的系統設備以共享庫的方式調用庫。寬通用公共許可證 2.1 版於 1999 年 2 月發佈。

5.1.4 FDL

進入新世紀以後，自由軟件的發展比最初還要快；然而文檔卻沒有受到重視。於是斯托曼認為“自由軟件需要自由的文檔”（參見本書《為什麼自由軟件需要自由的文檔》(p. 46) 一文)。

誠然軟件和文檔有一些相似——都是指導實踐的作品——但在使用方式上卻有很多重要的不同點。GPL 和 LGPL 對手冊並不適用。

一段時間以來，GNU 軟件包對每個手冊一直在使用一個未命名的，簡單的，臨時性的 copyleft 許可證。因為每一個手冊的許可證都不同，這樣無法在不同手冊之間複製文本。因此斯托曼寫了 GNU 自由文檔許可證，一個為軟件手冊和其它實踐作品設計的 copyleft 許可證。

FDL 首發於 2000 年 3 月。而 copyleft 則保持一致：每個收到作品副本的人都可以修改和再發布。FDL 與軟件許可證不同的地方在於實現細節：關於如何指明作品的原作者以及提供“源代碼”——可編輯的文檔——是不同的。

5.1.5 第三版

20 世紀 90 年代，因為自由軟件的迅速普及，GPL 成為社區首選的清晰的 copyleft 許可證，並應用在了主流的自由軟件項目裡；然而同時，專有軟件開發者想出了在不違反 GPL 的情況下，有效地否定 GPL 所保護的用戶的自由的方法。此外還有其他做法，且 GPL 處理起來並不方便。為了處理這種情況，一個新版本的許可證就呼之欲出了。

2002 年左右，斯托曼和其他自由軟件基金會（Free Software Foundation，以下簡稱 FSF）成員開始討論如何更新 GPL，還有 LGPL。FSF 建立了公眾複核流程，並與自由軟件法律中心（Software Freedom Law Center）的律

師合作，儘可能在發佈之前解決潛在的問題。社區的顧問委員參考了由公眾提交給斯托曼的觀點和爭論，並決定什麼需要採納。然後他基於律師的建議和意見寫了許可證的文本，關於其中的修改解釋，可參見本書《為何升級到 GPLv3》(p. 244) 一文。

第三版使用了新的術語來推動使用統一的司法解釋，並修改了一些需求以適應自由軟件社區新的實踐需要。除了這些，這一版還引入很多新條件以加強 copyleft，從而使自由軟件社區成為一個整體。比如：

- 禁用限制用戶修改硬件並拒絕用戶修改版的發行商（所謂 “tivoization”）；
- 允許代碼加入有限的額外需求，以兼容一些流行的自由軟件許可證；
- 並通過提供清晰的術語加強了對專利的要求來處理跨許可證的專利問題，這在大的專利持有公司是很常見的合約形式。

GPLv3 和 LGPLv3 都包括瞭解決這些問題的所有方法，並最終在 2007 年 6 月 29 日發佈。這些許可證都是 copyleft 藝術的一部分，以後會有更多其他的軟件許可證保護用戶的自由，並最終達到本書理想中的世界大同。

5.2 如何為你的作品選擇一份許可證

Copyright © 2011, 2013, 2014 自由軟件基金會。此文於 2011 年
首發於 <http://gnu.org>。

5.2.1 前言

人們經常問我們，為他們的項目向他們推薦什麼許可證。我們過去已經公開地寫了關於這個的文章，但信息在不同的文章、FAQ 條目和許可證註釋之間分散開了。這篇文章收集所有的信息至一個單一的來源中，讓人們更簡單地跟著做並可以參考。

這些推薦是用於那些被設計來做實用工作的作品的。這包括了軟件、文檔和一些其他的東西。藝術作品、以及表達觀點的作品，則不一樣；GNU 項目對它們該怎麼發佈沒有通用的立場，除了它們應該在沒有非自由軟件的情況下可以使用（特別是沒有 DRM¹）。然而，對於和某個程序在一起的藝術作品，你可能需要聽從這些推薦。

這些推薦可以用於許可一個你創建的作品——不論是已有作品的修改版還是原創作品。它們不能解決結合已有不同許可證的材料的問題。如果你在尋求那方面的幫助，請查看 GPL FAQ²。

在你看了我們的推薦之後，如果你想要建議，你可以寫郵件到 licensing@gnu.org。注意許可證小組的回信可能需要幾個星期；如果你在一個月內沒有得到回覆，請再寫一次。

5.2.2 貢獻到一個已有的項目

當你貢獻到一個已有的項目時，你通常應該在和原作品相同的許可證之下發布你修改過的版本。和項目的維護者合作是件好事，而為你的修改使用不同的許可證經常會讓合作變得困難。你只應在有強大理由支持的時

¹參見我們對數字限制管理 Digital Restrictions Management 的鬥爭活動 DefectiveByDesign.org。

²參見 <http://gnu.org/licenses/gpl-faq.html>。

候才那樣做。

一種可以合理使用不同許可證的情形是，你在非 copyleft 許可證下的作品上做出了主要修改。如果你創建的這個版本比原來的版本有用得多，那麼就值得讓你的作品 copyleft 化，就和我們通常推薦 copyleft 的原因一樣。如果你是在這種情況下，請遵循以下用於許可一個新項目的建議。

無論是什麼理由，如果你選擇以不同許可證發佈你的貢獻，你必須確保原有的許可證允許素材可以在你選擇的許可證之下使用。至少從良心上講，請明確地表示哪部分的作品是在哪種許可證之下。

5.2.3 軟件

對不同的項目我們推薦不同的許可證，這主要依據軟件的目的。通常來說，我們推薦使用最 copyleft 而不影響目的的許可證。我們的文章《什麼是 copyleft?》(p. 222) 更詳細地解釋了 copyleft 的概念，以及為什麼它通常是最好的許可方案。

對大多數程序，我們推薦你為你的項目使用最新版的 GPL。它強大的 copyleft 適合所有類型的軟件，並對用戶的自由有很多保護。也請允許使用 GPL 的未來版本——換句話說，在許可聲明中寫上你的程序被 GPL v3 或其更新版本覆蓋。

<http://gnu.org/licenses/gpl-howto.html> 有關於如何在 GNU GPL 之下發布程序的更多建議。

現在看看例外情形，這些情形下使用別的許可證比使用 GPL 更好。

小程序

對大多數小程序，使用 copyleft 是不值得的。我們用 300 行作為基準：當一個軟件包的源碼比這個短，copyleft 帶來的好處通常太小，不足以對抗確保許可證的複本總是伴隨軟件的不便。

對這些程序，我們推薦 Apache 2.0 許可證¹。這是一個 pushover（非

¹此許可證全文參見 <http://apache.org/licenses/LICENSE-2.0>

copyleft) 軟件許可證，它有用於避免貢獻者和分發者起訴專利侵權的條目。這並不會讓軟件避免來自專利的威脅（一個軟件許可證是做不到的），但它避免了專利持有者打著自由的幌子發佈軟件，這種情況下專利持有者會相當於做了一次“誘導轉向”，然後要求接受者同意專利證書中的非自由條目。

在不嚴格的 pushover 許可證中，Apache 2.0 是最好的；所以如果你要用一個不嚴格的 pushover 許可證，不論什麼原因，我們推薦用這一個。

庫

對於庫，我們分三種情形。

一些實現了自由標準的庫與那些限制性標準競爭，例如 Ogg Vorbis（和 MP3 音頻競爭）和 WebM（和 MPEG-4 視頻競爭）。對於這些項目，代碼的廣泛使用對於推進自由軟件事業非常重要，會比在項目代碼上的 copyleft 有更多的好處。

在這些特殊的情況，我們推薦 Apache 2.0 許可證。

對所有其他的庫，我們推薦某種 copyleft 許可證。如果開發者已經使用現有的以非自由或不嚴格的 pushover 許可證發佈的庫，那麼我們建議使用 GNU 寬通用公共許可證（LGPL）。

在第一種情形中，庫實現了一個倫理上更優秀的標準，但在這裡不像第一種情形，庫的廣泛使用並不會達成實際好處，所以完全沒有理由避免 copyleft。然而，如果你要求用庫的開發者在 copyleft 下發布整個程序，他們會簡單地用另一個可用的庫，那樣也不會推進我們的事業。較為寬鬆的 GPL（LGPL）是設計於填補這些情形中間地帶的，允許私有軟件開發者使用其保護起來的庫，但考慮到庫代碼本身，LGPL 提供了給用戶自由的弱 copyleft。

對於提供了特別設計，並且不會與現有非 copyleft 或非自由庫競爭的，我們推薦使用原始的 GNU GPL。要知道原因，請閱讀《為什麼你不應該在下一個函數庫中使用 LGPL》<http://gnu.org/licenses/why-not-lgpl.html>。

服務器軟件

如果其他人很有可能會給你在服務器上跑的軟件製作改進版並且不向其他人分發他們的版本，而且你擔心這將把你的版本置於一個不利的地位，我們推薦 GNU Affero 通用公共許可證 (AGPL¹)。AGPL 的條目和 GPL 幾乎相同；唯一實質的區別是它有一個額外的條件確保通過網絡用這個軟件的人們可以獲得它的源代碼。

對於用戶們而言，AGPL 的要求沒有解決當他們委託其計算或者數據給別人的服務器時，會產生的問題。例如，它不會制止作為軟件替代品的服務 (SaaS) 拒絕給予用戶的自由²——但大多數服務器不做 SaaS。要想知道更多關於這些的問題，請閱讀《為什麼要使用 GNU Affero GPL》<http://gnu.org/licenses/why-affero-gpl.html>。

5.2.4 文檔

對教程、參考手冊和其他大型文檔工作，我們推薦 GNU 自由文檔許可證。對教育型工作，這是個很強的 copyleft 許可證，最早為軟件文檔而編寫，並特別說明了當這些作品被分發或修改時的常見問題。

對於短的、次要的文檔工作，例如參考卡片，最好使用 GNU 全面容許性許可證³，因為一份 GFDL 的複本難以放進一張參考卡片裡面。不要用 CC-BY，因為它和 GFDL 不兼容。

對於 man 手冊頁面，如果頁面很長，我們推薦 GFDL，而如果它很短，則推薦 GNU 全面容許性許可證。

一些文檔包括了軟件源代碼。例如，一個編程語言的手冊可能包括讓用戶遵循的例子。你應當既在 FDL 之下在手冊中包含它們，又在另一個適合軟件的許可證下發布它們。這樣做使得在其他項目中用這些代碼變得簡單。我們推薦你用 CC0⁴ 貢獻小段的代碼到公有領域，並在有關軟件項

¹ 此許可證全文參見 <http://gnu.org/licenses/agpl.html>

² 有關 SaaS 可參見《服務器真正是在為誰服務?》(p. 302) 一文

³ 參見 <http://gnu.org/licenses/license-list.html/#GNUAllPermissive>

⁴ 關於此許可證可參見 <http://creativecommons.org/about/cc0>

目使用的相同的許可證之下分發大段的代碼。

5.2.5 其他用於程序的數據

這段討論你會包括在軟件中的所有其他實作品。例如圖表和其他功能性或有用的圖像、字體和地理數據等。你也可以在藝術中遵循這些，儘管你不這樣做我們也不會批評你。

如果你正特意為一個軟件項目創作這些作品，我們通常推薦你在和軟件使用相同的許可證下發布你的作品。用我們推薦的許可證這樣做不會有問題：GPLv3、LGPLv3、AGPLv3 和 GPLv2 都可以用於任何類型——不只是軟件——只需受版權保護並對修改版有清晰首選的形式。使用與軟件相同的許可證會讓分發者更容易遵守規定，並可以避免任何對潛在的兼容性問題的顧慮。如果提供了一些特別的實用的好處，比如和其他自由項目更好地合作，那麼使用一個不同的許可證可能是合適的。

如果你的作品不是為某個特定的軟件項目而創作的，或者使用和項目相同的許可證不合適，那麼我們只推薦你選擇一個適合你作品的 copyleft 許可證。有一些在我們的許可證列表中列出¹。如果沒有許可證看起來合適的，創作共用署名-相同方式共享許可證（CC-BY-SA²）是一個可以用於很多不同種類作品的 copyleft 許可證。

¹參見 <http://gnu.org/licenses/license-list.html/#OtherLicenses>

²關於使用此許可證可參見 <http://gnu.org/licenses/license-list.html/#ccbyssa>

5.3 X Window 系統的陷阱

Copyright © 1998, 1999, 2009 理查德·斯托曼 (Richard Stallman)。

本文最初於 1998 年發表在 <http://gnu.org>。

左版 (Copyleft)，還是不要左版？這是自由軟件社區中的主要分歧之一。左版的理念之一是我們可以以火滅火——即我們應當利用版權以保證我們的代碼持續自由。GNU 通用公共許可證 (GNU GPL) 就是一個左版許可證的例子。

一些自由軟件開發者更傾向於使用非左版進行分發。諸如 XFree86 和 BSD 這樣的非左版許可證基於這樣的理念，即永遠不對任何人說不——即使是對那些企圖用您的工作作為限制他人基礎的人。非左版許可證並沒有做錯事，但它錯失了對我們更改和再分發軟件的自由提供主動保護的機會。基於此原因，我們需要左版。

多年以來，X 聯盟一直是反對左版的主要力量。它通過施加道德勸說和壓力來阻止自由軟件開發者對其程序採用左版許可證。它通過暗示說“不”是不友善的來施加道德勸說。它利用它的“左版軟件不能出現在 X 發行版中”的規則來施加壓力。

X 聯盟為何採取這種政策？這必然和他們關於成功的概念有關。X 聯盟將成功定義為流行——這特指的是使得計算機公司採用 X Window 系統。這一定義將計算機公司置於主導地位：不論它們想要什麼，X 聯盟都必須設法幫助它們獲得。

計算機公司通常發佈私有軟件。它們要求自由軟件開發者捐獻他們的工作用於此用途。如果他們直接如此要求，人們會哈哈大笑。但是 X 聯盟在面對它們的時候可以將這種要求看做是無私的。“加入我們，將我們的工作捐贈給私有軟件開發者。”他們如是說，暗示這是一種高貴的自我犧牲。“加入我們，以獲取流行度。”他們如是說，暗示這甚至不是一種犧牲。

但是，自我犧牲還不是問題之所在：扔掉左版對於整個社區自由的保護，所犧牲掉的遠不止您自己。那些順應了 X 聯盟要求的人們將整個社區的未來託付給了 X 聯盟的一廂良願。

這種信任是錯位的。在它的最後一年，X 聯盟計劃限制即將發佈的 X11R6.4 版本使其不再是自由軟件。他們終於決定開始說不，但不僅僅是對私有軟件開發者，更是對我們的社區。

這裡有個諷刺。如果當 X 聯盟要求您不要使用左版許可證的時候，您說“是”，您便將 X 聯盟置於這樣一種境地，它可以將您的程序許可並限制為它自己的版本，對於 X 的核心部分的代碼也是如此。

X 聯盟最終沒有采取這種計劃。與之相反，它解散了，並且將 X 的開發權移交給了開放小組（Open Group），後者的員工正在實施一種類似的計劃。值得稱道的是，當我要求他們以 GNU GPL 發佈 X11R6.4 並且與他們所計劃的限制性許可證共存的時候，他們表示願意考慮這種想法（他們堅決反對墨守舊的 X11 發佈條款）。但是他們還未來得及肯定或是否定這一提議，它已經由於其他原因失敗了：XFree86 小組繼承了 X 聯盟的舊政策，並且拒絕接受任何左版軟件。

就在 1998 年九月，即 X11R6.4 以非自由分發條款發佈數月之後，開放小組逆轉了他們的決定，並且將其改為和 X11R6.3 相同的非左版自由軟件許可證重新發布。至此，開放小組終於做了一件正確的事，但遠未解決其基本問題。

但是，即使 X 聯盟或是開放小組本意上從未打算限制 X，其他人可能已經這麼做了。非左版軟件在任何方向都是脆弱的；它使得任何製造非自由軟件的人處於統治地位，如果此人投入足夠多的資源，利用私有代碼向其中添加重要功能。任何基於技術特性而非自由選擇軟件的人們很容易出於短期的易用性考慮而被引誘使用非自由版本。

X 聯盟和開放小組再也不能通過宣稱“說‘不’是不友善的”來施加道德勸說，這將會使得它更容易決定以左版發佈與 X 相關的軟件。

如果您貢獻於 X 核心或是諸如 X server、Xlib、Xt 等程序，確實有一種不使用左版的實踐上的原因。由於 X.org 小組在為整個社區維護這些程序的過程中已經做了重大貢獻，使用左版發佈我們的更改所帶來的好處少於分叉（fork）整體開發所帶來的負面影響。於是最好與他們協同工作並且不用左版發佈我們對這些程序的修改。與之相似的還有諸如 xset 和 xrdp 等

工具，它們與 X 核心很接近並且不需要大規模改進。至少我們知道 X.org 小組對於將這些程序作為自由軟件發佈有著堅定的承諾。

然而，X 核心外圍程序的問題與此不同：諸如應用程序、窗口管理器、附加的庫和部件工具箱。沒有理由不為其使用左版許可證，我們應當為其使用左版許可證。

如果任何人感受到了由 X 發行版的軟件包含準則所施加的壓力，GNU 計劃將會承擔宣傳能夠與 X 協同工作的左版軟件包的責任。如果您想要以左版許可證發佈什麼東西，但又擔心從 X 發行版中將其排除將會影響它的流行度，請向我們尋求幫助。

與此同時，如果我們不再那麼看重流行度，事情會變得更好。當一位商務人士以“更高的流行度”來引誘您的時候，此人可能是在試圖使您相信他對您的程序的使用對於它的成功至關重要。不要相信這種鬼話！如果您的程序確實出類拔萃，它總能得到眾多用戶；您無需為失去任何特別的用戶感到絕望，並且如果您不為所動，您將會更加強大。如果您能夠回覆“愛用不用——我才不會在乎所謂的流行度”。通常，那位商務人士將會改變主意並且接受該程序作為左版程序，只要您動了真格。

朋友們，自由軟件開發者們，不要再重複過去的錯誤！如果我們不以左版發佈我們的軟件，我們就會將它的未來置於任憑那些擁有的資源多於良知的人們擺佈的境地。只要有左版，我們就能捍衛自由，不僅為我們自己，更是為了我們的整個社區。

5.4 程序不得限制它們的自由運行

Copyright © 2012 自由軟件基金會。此文 2012 年首發於 <http://gnu.org>

自由軟件意味著軟件被其用戶所控制，而不是相反。具體的說，這意味著軟件在到達用戶手中時，就具有用戶應該得到的四大基本自由。列表¹中的第一條自由是自由之零：按照你的意願運程序來做你自己想做的事情。

一些開發者提議在軟件許可證中對某些使用加以限制，來禁止程序的某些用途，但這種做法是災難性的。這篇文章解釋了為何不得限制自由之零。限制程序用途的許可條件幾乎不能實現它們的目的，而且還可能破壞自由軟件社區。

首先，我們要明確自由之零的含義，即軟件的分發並不限制你的使用方式。但這並不能使你在法律上免責。例如，詐騙在美國法律中是犯罪行為——我認為這法律是正確且適當的。無論自由軟件許可證裡寫了些什麼，你如果使用自由程序行騙並不能保護你不被逮捕。

在詐騙是犯罪的國家，用許可證條款來打擊詐騙顯然是多餘的累贅。但在很多國家，“安全部隊”實施的酷刑可是常常被縱容的，為什麼不加入一項禁止酷刑的條款呢？

禁止酷刑的條款是不會有用的，因為任何自由軟件許可證的實施都是通過國家政權進行的。一個希望實施酷刑的國家將會忽略許可協議。當美國酷刑的受害者試圖起訴美國政府時，法院以涉密、威脅國家安全為由駁回了案件。如果一個軟件開發者試圖起訴美國政府，說他的程序被用來實施酷刑，違反了其許可證，這紙訴狀也會被駁回的。通常，國家對它們所做的任何惡毒的事情，總能“聰明”得找到法律上的藉口。遊說力量強大的企業也能這麼做。

那加入一些禁止某些特定私人行為的條款又如何呢？例如，善待動物組織（PETA）提議了禁止使用軟件於脊柱，造成動物痛苦的許可證。或者

¹參見《什麼是自由軟件？》(p. 1) 一文來了解自由軟件的完整定義

禁止使用某程序製作或發表穆罕默德畫像的條款又如何呢？又或者禁止將程序用於胚胎幹細胞實驗？亦或禁止利用程序未經授權複製音樂錄音？

這些條款能否實施是不明確的。自由軟件許可協議是基於版權法的，試圖以這樣的方式施加使用上的限制，延伸了版權法的允許範圍，這種延伸是非常危險的。你願意讓書籍限制你使用其信息的方式嗎？

那假如條款是合法可實施的，這又會有好處嗎？

事實是，人們對於使用軟件應當完成的活動有許多不同的道德見解。順便一說，我認為上述四類不尋常的社會活動是合法且不應當禁止的。具體說，我支持使用軟件進行動物醫學實驗或者加工肉類。我捍衛動物權利活動人士的人權，但我並不支持他們的觀點——我也不希望善待動物組織以它們的方式限制軟件的使用。

由於我不是和平主義者，我同樣反對“非軍用”的條款。我譴責侵略戰爭，但我不譴責反擊。事實上，我支持說服多支軍隊切換到自由軟件的努力，以便它們檢查危害國家安全的後門和監控特性。

由於我並不反對一切商業，我反對限制商用的條款。一個只能用於娛樂、愛好和學校的系統很大程度上限制了我們通過計算機能做的事情。

我剛才已經表述了我對許多其他政治問題，和某些政治活動是否正當的看法。你自己的觀點可能有所不同，而這恰恰準確表明了我的論點。如果我們接受這些帶有使用限制的程序作為一個自由操作系統，例如 GNU 的一部分，人們將最終制定出一大堆不同的使用限制。到時候會存在禁止用於肉類加工的程序；僅禁止加工豬肉的程序；僅禁止加工牛肉的程序；僅允許加工“潔淨”食物程序。討厭菠菜的人可能會編寫一個允許加工任何蔬菜，除了菠菜的食物；而《大力水手》的粉絲可能只允許程序用來加工菠菜。而只允許說唱音樂的音樂程序也會出現；而其他音樂程序則只允許古典音樂。

這麼做的結果是產生一個你不能指望它用於任何目的的系統。你每做一件事情，你就得檢查一大堆的許可證來看看系統的什麼部分對你的任務各有什麼限制。

用戶將會怎麼迴應呢？我想大部分用戶將轉而使用私有系統。允許在

自由軟件中對使用方式施加各自限制，總的來說將會驅使用戶使用非自由軟件。試圖利用自由軟件中的限制條款來阻止用戶做什麼，就好比使用一根又長又軟又直的意大利麵條推動一個物體一樣，沒有效果。

它甚至比“無效”更加糟糕：它同樣是錯誤的，因為軟件開發者不得在用戶想做的事情上行使這種權力。想象一下出售一支筆，筆上附有你能夠用它來寫什麼的條款吧。對通用的軟件而言也是一樣的道理。如果你製造了一樣非常有用的物品，例如一支筆，人們將用它做各種各樣的事情，包括那些最惡毒的事，例如對異見人士實施酷刑的命令。但你不應該擁有通過筆來控制他們行為的權力。這對於一個文本編輯器、編譯器或者內核來說也是同理。

你所擁有的是決定軟件能夠用來做什麼的機會——當你決定實現什麼功能的時候。你可以編寫一個主要被你認為行為正當的人使用的軟件，你也沒有任何義務去編寫任何可能被別人用於你不認同的行為的功能。

結論是明確的：程序不應當限制用戶利用它所完成的工作。自由之零必須完整無缺。我們需要禁止酷刑，但我們不能通過軟件許可協議來做到這點。軟件許可協議的正確職責是建立與保護用戶的自由。

5.5 什麼是 Copyleft?

Copyright © 1996–2009, 2013 自由軟件基金會。此文最初於 1996 年發表在 <http://gnu.org>。

Copyleft 是一種使一個程序或其它工作自由的通用方法，並要求該程序（或其他作品）的所有修改和衍生版本也是自由的。

使一個程序成為自由軟件最簡單的方法就是將它放到一個不受版權限制的公有領域。如果他們足夠明智，就應該允許人們共享該程序和他們的改進成果。但是這也使得那些不願合作的人有權利將該程序轉化為專有軟件。他們可以或多或少地做一些修改，並將其作為專有軟件發佈。而獲得修改版程序的人將不能擁有原作者賦予他們的自由，因為中間人已經將其剝奪了。

在 GNU 工程中，我們的目標是賦予所有用戶重新發布和修改 GNU 軟件的自由。如果中間人剝奪了這種自由，即便我們可能會有許多使用者，但他們將不能享有自由。因此我們使它“Copyleft”，來代替將 GNU 軟件放到公有領域。Copyleft 指明任何人在重新發布軟件時，不管有沒有修改，都必須將這種自由傳遞到下一個副本中並改變它，以使人們更多的複製和修改。Copyleft 保證了每一位用戶都擁有自由。

Copyleft 也對其他程序員發佈自由軟件提供了一種激勵，重要的自由程序如 GNU C++ 編譯器就是因為這個原因而存在。

Copyleft 也可以幫助那些想給自由軟件做改進的程序員獲得自由去改進自由軟件。這些程序員通常在那些為了賺更多錢，而幾乎什麼都乾的公司或大學裡工作。一個程序員可能想向社區貢獻他的改進，但是他的老闆可能反而想將其應用到專有軟件產品中。

當我們向老闆解釋不以自由軟件形式發佈改進版本是非法時，老闆常常決定與其將其捨棄不如將其作為自由軟件發佈。

要讓一個程序 Copyleft，我們首先應指出它是有版權的；隨後我們附加發布條款，這些條款有法律效力並且賦予每個人使用、修改、重新發布該程序，或任何基於該程序而派生出程序的源代碼的權利。但前提是這些

發佈條件沒有改變。因此，代碼和自由在法律上是不可分割的。

專有軟件開發者利用版權來剝奪用戶的自由，我們用版權來保證他們的自由。這正是為什麼我們顛倒這個名字，將“copyright”（版權）改為“copyleft”。

“Copyleft”是一種對程序享有版權的方式。它不意味著放棄版權；事實上，那樣做就不是 copyleft 了。Copyleft 裡的單詞“left”與動詞“to leave”（離開）沒什麼聯繫——只是一種與“right”（右邊）反向的說法。

Copyleft 是一個抽象的概念，而你不能直接使用抽象的概念；你只能使用該概念的一個具體實現。在 GNU 工程中，為大多數軟件所使用的具體發佈規則都包含在了 GNU 通用公共許可證（GNU General Public License）中。GNU 通用公共許可證經常被簡稱為 GNU GPL。還有一個關於 GNU GPL 常見問題的頁面 <http://gnu.org/licenses/gpl-faq.html>，你也可以閱讀為什麼 FSF 從貢獻者那得到版權轉讓 <http://gnu.org/copyleft/why-assign.html>。

Copyleft 的一種替代形式，是 GNU Affero 通用公共許可證（GNU Affero General Public License, AGPL），主要應用於服務器上的程序。它可以確保公開服務器上的修改版也公開發布源代碼。

Copyleft 的另一種替代形式，GNU 寬通用公共許可證（GNU Lesser General Public License, LGPL），應用於一小部分（並非全部）的 GNU 庫。想了解更多關於 LGPL 的正確使用，請閱讀文章《為什麼我們不應該在新的開發庫中使用 LGPL》<http://gnu.org/philosophy/why-not-lgpl.html>。

GNU 自由文檔許可證（GNU FDL）是 Copyleft 的一種形式，用於在手冊、教材或其它文檔上以保證任何人都可以自由地複製和發佈它們，不管是否對它們進行了修改，也不管是不是進行商業化使用。

相應的許可證被包含在眾多手冊和每個 GNU 源代碼的發佈中。

假如你是版權所有人，所有這些設計好的許可證能讓你很容易的接受並應用於自己的作品中。做這件事你不需要修改許可證，僅僅在作品中包含一份許可證的副本，並且在源文件中添加聲明，以示使用了適當的許可證。

對許多不同的程序使用相同的發佈條款，使得在不同的程序間複製代碼變得很容易。既然都有相同的發佈條款，這就沒有必要去考慮這些條款是否相容。LGPL 中有一項規定，它可以允許你把發佈條款改成普通的 GPL，因此你可以將代碼拷貝到另一個使用 GPL 的程序當中。第三版的 LGPL 被當作了 GPL 第三版的例外，所以自動具有兼容性。

如果你想用 GNU GPL 或 GNU LGPL 來 copyleft 你的程序，請查看許可證介紹頁面的建議 <http://gnu.org/copyleft/gpl-howto.html>。作為建議，請注意你必須使用我們許可證的全文。每篇許可證都是不可分割的整體，因此部分複製是不允許的。

如果你用 GNU FDL 來 copyleft 你的手冊，請查看 [FDL 文本結尾的說明](http://gnu.org/copyleft/fdl-howto.html)和 GFDL 說明頁面 <http://gnu.org/copyleft/fdl-howto.html>。同樣，部分複製是不允許的。

使用反轉的字母 C 外套一個圓圈作為版權符號是一個法律錯誤。Copyleft 是基於合法的版權，所以作品應該有版權聲明。版權聲明要求要麼使用版權符號（圓圈裡的字母 C），或使用單詞“Copyright”（版權所有）。

反轉的字母 C 外套一個圓圈沒有特別的法律意義，因此它不能成為一個版權聲明。也許放在書籍封面、海報上很有趣，但請注意其代表網頁時的情況。

5.6 為什麼使用 Copyleft

Copyright © 2003, 2007, 2008, 2013 自由軟件基金會。此文原載於 <http://gnu.org>

當涉及到為別人保護自由的時候，躺下什麼都不做是一種軟弱的表現，而不是謙卑。

在 GNU 工程中，我們通常建議人們使用 Copyleft¹ 許可證，比如 GNU GPL，而不是寬鬆的非 Copyleft 自由軟件許可證。我們不會極力反對非 Copyleft 許可證——事實上在特殊情況下我們還建議使用這些——然而這些許可證的倡導者往往極力反對 GPL。

比如一個人說，使用 BSD 許可證是“一種謙卑的舉動”：“我不會對使用我代碼的人索要任何權力，只要尊重我就行。”這種描述相當誇張，將一種法律訴求說成是“謙卑”，我們需要思考得更深刻一些。

謙卑是放棄你自己的利益，然而你和使用你代碼的人不是唯一受到你選擇這種自由軟件許可證影響的人。還包括另外一些將你的代碼使用在非自由程序的人，會試圖拒絕給別人以自由，如果你讓他這麼做了，你就沒能捍衛自己的自由。當涉及到為別人保護自由的時候，躺下什麼都不做是一種軟弱的表現，而不是謙卑。

以 BSD 或其他寬鬆的非 Copyleft 許可證發佈你的代碼，並不是錯的；程序依舊是自由軟件，對我們社區來講也是一份貢獻。然而這很弱，絕大多數情況下，這也不是推廣用戶分享和修改自由的最佳方式。

¹參看《什麼是 Copyleft?》(p. 222) 一文。

5.7 Copyleft：務實的理想主義

Copyright © 1998, 2003 自由軟件基金會

一個人做出的每個決定都取決於這個人的價值觀和目標。人們可以有很多種不同的目標和價值觀；聲譽、利益、愛情、生存、快樂和自由，其中有一些目標只有一些優秀的人才可能擁有。當這個目標是一種原則時，我們稱之為理想主義。

一個理想主義的目標激勵著我在自由軟件上的工作：傳播自由與協作的理念。我想鼓勵自由軟件的傳播，取代阻礙合作的專有軟件，從而使我們的社會更加美好¹。

這就是 GNU 通用公共許可證被寫成 Copyleft 的根本原因。所有代碼被添加到由 GPL 許可證保護的項目必須是自由軟件，即使它被放進一個單獨的文件裡。我使我的代碼在自由軟件裡可用，並且在專有軟件裡不可用，目的是為了鼓勵其他寫軟件的人也使它自由。我想既然專有軟件開發者們利用版權來阻止我們分享，我們合作的人可以利用版權給予其他合作者一些對於他們自身的好處：他們可以使用我們的代碼。

不是所有使用 GNU GPL 的人都有這樣的目標。許多年以前，我的一個朋友被要求在非 copyleft 的條款下重新發佈一個 copyleft 的程序。他大概是這樣迴應此事的：“有時我從事於自由軟件，而有時我從事於專有軟件——但當我從事於專有軟件時，我希望得到回報。”

他很樂意與一個分享軟件的社區來分享自己的成果，但似乎沒有理由向那些偏離我們社區限制的商業產品給出一個發佈版。他的目標不同於我的，但是他覺得 GNU GPL 對他的目標也有用。

如果你想完成某件事情，光靠理想主義是不夠的，你需要選擇一種實現你的目標的方法。換句話說，你需要更“務實”。GPL 許可證務實嗎？讓我們看看它的成果。

就拿 GNU C++ 來說吧。為什麼我們會有一個自由的 C++ 編譯器呢？僅僅因為 GNU GPL 規定它必須是自由的。GNU C++ 是由起源於 GNU C

¹參見《為什麼使用 Copyleft?》(p. 225) 一文

編譯器的一個工業協會 MCC 開發的。MCC 通常儘可能使它的作品專有化。但是他們讓 C++ 前端自始至終是自由軟件，因為 GNU GPL 規定這是發佈它的唯一方式。C++ 前端引入了很多新的文件，但是因為它們要鏈接到 GCC 上，所以 GPL 能約束它們。並且這樣做對我們社區的益處是顯而易見的。

再說說 GNU Object C。NeXT 公司最初想把這個前端變成專有的；他們打算以 .o 文件的形式發佈，並讓用戶使用 GCC 的其餘部分來鏈接他們，想這樣能繞過 GPL 的要求。但是我們的律師說這不能躲過這些要求，那是被禁止的。所以他們使 Objective C 前端成為了自由軟件。

這些例子發生在好多年前，但是 GNU GPL 一直持續帶給我們更多的自由軟件。

許多 GNU 庫遵循 GNU 寬通用公共許可證 (GNU Lesser General Public License)，但不全是，Readline 就是一個使用普通 GNU GPL 許可證保護的 GNU 庫，它實現了命令行編輯功能。我曾經發現一個非自由程序被設計為使用 Readline，並告知程序開發者這是不允許的。他本可以從程序中移除命令行編輯功能，但他實際所做的是讓這個程序基於 GNU GPL 許可證重新發布。現在這個程序是自由軟件了。

這些寫代碼改進 GCC（或者 Emacs、Bash、Linux 或其它遵守 GPL 的程序）的程序員們經常被公司或者大學僱傭。當程序員想把他們的改進回饋給社區，從而在下一次發佈能看到他的代碼時，老闆可能會說，“到此為止，你的代碼屬於我們！我們不想分享它；我們已經決定把你的改進版本添加到專有軟件產品裡了。”

這時候 GNU GPL 就會出來拯救你的成果了。程序員告訴老闆這樣的專有軟件產品是侵犯版權的，從而老闆會意識到他只有兩種選擇：把這些新代碼以自由軟件的形式發佈，或者什麼都不做。大部分時候老闆會讓程序員按照他一直所想的來做，把這些代碼加到下一個發佈版。

GNU GPL 不是“和事佬”，有時它也會對人們想要做的一些事情說“不”。有一些用戶說這是一件糟糕的事情——GPL 許可證會“拒絕”一些“需要被帶進自由軟件社區的專有軟件開發者”。

但是我們並沒有把他們拒絕在我們的社區之外；是他們選擇的不加入。他們使軟件專有化的決定也是在我們社區之外的決定。融入我們的社區就意味著與我們的合作；如果他們不想加入，我們就不能“把他們帶入我們的社區”。

我們能做的是提供給他們一個加入的動機。GNU GPL 許可證被設計用來從我們現有的軟件中產生一個動機：“如果你想讓你的軟件自由化，你就可以使用這個源代碼”。當然，雖然不會贏得一切，但會贏得一些時間。

專有軟件的開發不會對我們的社區有所貢獻，但它的開發者們經常想從我們這裡得到資料。自由軟件用戶可以為自由軟件開發者提供一些自我暗示——認可和感謝——但是當一個商人告訴你，“只要讓我們把你的軟件包放進我們的專有軟件中，你的程序會被成千上萬的人使用！”，這是非常誘人的。這個誘惑是很有力的，但從長遠來看，如果我們堅持這一點，最好離遠點。

當這種誘惑和壓迫通過迎合專有軟件策略的自由軟件組織間接到來時，它們就更加難以辨別。X 聯盟（與他的繼任者，Open Group）提供了一個例子：由開發專有軟件的公司建立，他們努力了十年來說服程序員們不要使用 Copyleft。Open Group 曾經試圖使 X11R6.4 變為非自由軟件¹時，我們當中很多人抵抗住了這個壓迫，很高興我們做到了。

在 1998 年 9 月，X11R6.4 以非自由發行條款發佈的幾個月之後，Open Group 推翻了自己的決定，並且用 X11R6.3 使用過的非 copyleft 的自由軟件許可證重新發布。感謝你，Open Group——但是這個後來的逆轉並不能改變我們已經得出的結論，他們依舊有可能加入限制。

從實際來講，考慮更長遠的目標會更加堅定你反抗這種壓迫的信念。如果你把你的思想專注於自由和可以待在公司建設的社區，你會找到這樣做的動力。“堅定信念，否則你會上當受騙”。

如果有憤世嫉俗的人嘲笑自由，嘲笑社區……如果“硬鼻子的現實主義者”說利潤是唯一的理想……忽略他們就好，繼續使用 Copyleft。

¹更多相關信息，請參見《X Window 系統的陷阱》(p. 216)一文

5.8 GNU 通用公共許可證

中文翻譯: bergwolf bergwolf@gmail.com, 源出處: https://sites.google.com/site/bergwolf02/gplv3_zh

5.8.1 聲明

This is an unofficial translation of the GNU General Public License into Chinese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL--only the original English text of the GNU GPL <http://www.gnu.org/licenses/gpl-3.0.html> does that. However, we hope that this translation will help Chinese speakers understand the GNU GPL better.

這是一份 GNU 通用公共授權的非官方中文簡體翻譯。它並非由自由軟件基金會發布，也不是使用 GNU 通用公共授權的軟件的法定發佈條款——只有 GNU 通用公共授權的英文原版 <http://www.gnu.org/licenses/gpl-3.0.html> 具有這樣的效力。然而，我們希望這份翻譯能夠幫助中文讀者更好的理解 GNU 通用公共授權。

第三版，2007 年 6 月 29 日

版權所有 © 2007 自由軟件基金會 <http://fsf.org/>

允許所有人複製和發佈本授權文件的完整版本

但不允許對它進行任何修改

5.8.2 導言

GNU 通用公共授權是一份針對軟件和其他種類作品的自由的 copyleft 授權文件。

大多數軟件授權申明被設計為剝奪您共享和修改軟件的自由。相反地，

GNU 通用公共授權力圖保護您分享和修改自由軟件地自由——以確保軟件對所有使用者都是自由的。我們，自由軟件基金會，對我們的大多數軟件使用 GNU 通用公共授權；本授權同樣適用於任何其作者以這種方式發佈的軟件。您也可以讓您的軟件使用本授權。

當我們談論自由軟件時，我們指的是行為的自由，而非價格免費。GNU 通用公共授權被設計為確保您擁有發佈自由軟件副本（以及為此收費，如果您希望的話）的自由，確保您能收到源代碼或者在您需要時能獲取源代碼，確保您能修改軟件或者將它的一部分用於新的自由軟件，並且確保您知道您能做這些事情。

為了保護您的權利，我們需要做出要求，禁止任何人否認您的這些權利或者要求您放棄這些權利。因此，如果您發佈此軟件的副本或者修改它，您就需要肩負起尊重他人自由的責任。

例如，如果您發佈自由軟件的副本，無論以免費還是以收費的模式，您都必須把您獲得的自由同樣的給予副本的接收者。您必須確保他們也能收到或者得到源代碼。而且您必須向他們展示這些條款，以使他們知道自己享有這樣的權利。

使用 GNU 通用公共授權的開發者通過兩項措施來保護您的權利：（1）聲明軟件的版權；（2）向您提供本授權文件以給您複製、發佈並且/或者修改軟件的法律許可。

為了保護軟件開發者和作者，通用公共授權明確闡釋自由軟件沒有任何擔保責任。如用戶和軟件作者所希望的，通用公共授權要求軟件被修改過的版本必須明確標示，從而避免它們的問題被錯誤地歸咎於先前的版本。

某些設備被設計成拒絕用戶安裝或運行其內部軟件的修改版本，儘管製造商可以安裝和運行它們。這從根本上違背了通用公共授權保護用戶能修改軟件的自由的宗旨。此類濫用本授權的系統模式出現在了最讓人無法接受的個人用戶產品領域。因此，我們設計了這個版本的通用公共授權來禁止那些產品的侵權行為。如果此類問題在其他領域大量出現，我們準備好了在將來的通用公共授權版本里擴展這項規定，以保護用戶的自由。

最後，每個程序都經常受到軟件專利的威脅。政府不應該允許專利權

限制通用計算機軟件的發展和使用，但是在政府確實允許這種事情的地區，我們希望避免應用於自由軟件的專利權使該軟件有效私有化的危險。為了阻止這樣的事情的發生，通用公共授權確保沒有人能夠使用專利權使得自由軟件非自由化。

以下是複製，發佈和修改軟件的詳細條款和條件。

5.8.3 條款和條件

0. 定義

“本授權”指 GNU 通用公共授權第三版

“版權”一詞同樣指適用於其他產品如半導體掩膜等的保護版權的法律。

“本程序”指任何在本授權下發布的受版權保護的作品。被授權人稱為“您”。“被授權人”和“版權接受者”可以是個人或組織。

“修改”作品是指從軟件中拷貝或者做出全部或一丁點兒的修改，這不同於逐字逐句的複製，是需要版權許可的。修改成果被稱為先前作品的“修改版本”或者“基於”先前作品的軟件。

“覆蓋程序”指未被修改過的本程序或者基於本程序的程序。

“傳播”程序指使用該程序做任何如果沒有許可就會在適用的版權法下直接或間接侵權的事情，不包括在電腦上執行程序或者是做出您不與人共享的修改。傳播包括複製，分發（無論修改與否），向公眾共享，以及在某些國家的其他行為。

“發佈”作品指任何讓其他組織製作或者接受副本的傳播行為。僅僅通過電腦網絡和一個用戶交流，且沒有發送程序拷貝的行為不是發佈。

一個顯示“適當的法律通告”的交互的用戶接口應包括這樣一個方便而顯著的可視部件，它具有以下功能：(1) 顯示一個合適的版權通告；(2) 告訴用戶對本程序沒有任何擔保責任（除非有擔保明確告知），受權人可以在本授權下發布本程序，以及如果閱讀本授權協議的副本。如果該接口顯示了一個用戶命令或選項列表，比如菜單，該列表中的選項需要符合上

述規範。

1. 源代碼

“源代碼”指修改程序常用的形式。“目標代碼”指程序的任何非源代碼形式。

“標準接口”有兩種含義，一是由標準組織分支定義的官方標準；二是針對某種語言專門定義的眾多接口中，在該類語言的開發者中廣為使用的那種接口。

可執程序的“系統庫”不是指整個程序，而是指任何包含於主要部件但不屬於該部件的部分，並且只是為了使能該部件而開發，或者為了實現某些已有公開源代碼的標準接口。“主要部件”在這裡指的是執程序的特定操作系統（如果有的話）的主要的關鍵部件（內核，窗口系統等），或者生成該可執程序時使用的編譯器，或者運行該程序的目標代碼解釋器。

目標代碼中的程序“對應的源代碼”指所有生成，安裝，（對可執程序而言）運行該目標代碼和修改該程序所需要的源代碼，包括控制這些行為的腳本。但是，它不包括程序需要的系統庫，通用目的的工具，以及程序在完成某些功能時不經修改地使用的那些不包括在程序中的普遍可用的自由軟件。例如，對應的源代碼包括與程序的源文件相關的接口定義文件，以及共享庫中的源代碼和該程序設計需要的通過如頻繁的數據交互或者這些子程序和該程序其他部分之間的控制流等方式獲得的動態鏈接子程序。

對應的源代碼不需要包含任何擁護可以從這些資源的其他部分自動再生的資源。

源代碼形式的程序對應的源代碼定義同上。

2. 基本的許可

所有在本授權協議下授予的權利都是對本程序的版權而言，並且只要所述的條件都滿足了，這些授權是不能收回的。本授權明確的確認您可以不受任何限制地運行本程序的未修改版本。運行一個本授權覆蓋的程序獲得的結果只有在該結果的內容構成一個覆蓋程序的時候才由本授權覆蓋。

本授權承認您正當使用或版權法規定的其它類似行為的權利。

只要您的授權仍然有效，您可以無條件地製作，運行和傳播那些您不發佈的覆蓋程序。只要您遵守本授權中關於發佈您不具有版權的資料的條款，您可以向別人發佈覆蓋程序，以要求他們為您做出專門的修改或者向您提供運行這些程序的簡易設備。那些為您製作或運行覆蓋程序的人作為您專門的代表也必須在您的指示和控制下做到這些，請禁止他們在他們和您的關係之外製作任何您擁有版權的程序的副本。

當下述條件滿足的時候，在任何其他情況下的發佈都是允許的。

轉授許可證授權是不允許的，第 10 節讓它變的沒有必要了。

3. 保護用戶的合法權利不受反破解法侵犯

在任何實現 1996 年通過的世界知識產權組織版權條約第 11 章中所述任務的法律，或者是禁止或限制這種破解方法的類似法律下，覆蓋程序都不會被認定為有效的技術手段的一部分。

當您發佈一個覆蓋程序時，您將放棄任何禁止技術手段破解的法律力量，甚至在本授權關於覆蓋程序的條款下執行權利也能完成破解。同時，您放棄任何限制用戶操作或修改該覆蓋程序以執行您禁止技術手段破解的合法權利的企圖。

4. 發佈完整副本

你可以通過任何媒介發佈本程序源代碼的未被修改過的完整副本，只要您顯著而適當地在每個副本上發佈一個合適的版權通告；保持完整所有敘述本授權和任何按照第 7 節加入的非許可的條款；保持完整所有的免責申明；並隨程序給所有的接受者一份本授權。

您可以為您的副本收取任何價格的費用或者免費，你也可以提供技術支持或者責任擔保來收取費用。

5. 發佈修改過的源碼版本

您可以在第 4 節的條款下以源碼形式發佈一個基於本程序的軟件，或者從本程序中製作該軟件需要進行的修改，只要您同時滿足所有以下條件：

- a) 製作的軟件必須包含明確的通告說明您修改了它，並給出相應的修改日期。
- b) 製作的軟件必須包含明確的通告，陳述它在本授權下發布並指出任何按照第 7 節加入的條件。這條要求修改了第 4 節的“保持所有通知完整”的要求。
- c) 您必須把整個軟件作為一個整體向任何獲取副本的人按照本授權協議授權。本授權因此會和任何按照第 7 節加入的條款一起，對整個軟件及其所有部分，無論是以什麼形式打包的，起法律效力。本授權不允許以其他任何形式授權該軟件，但如果您個別地收到這樣的許可，本授權並不否定該許可。
- d) 如果您製作的軟件包含交互的用戶接口，每個用戶接口都必須顯示適當的法律通告；但是，如果本程序包含沒有顯示適當的法律通告的交互接口，您的軟件沒有必要修改他們讓他們顯示。

如果一個覆蓋程序和其他本身不是該程序的擴展的程序的聯合體，這樣的聯合的目的不是為了在某個存儲或發佈媒體上生成更大的程序，且聯合體程序和相應產生的版權沒有用來限制程序的使用或限制單個程序賦予的聯合程序的用戶的合法權利的時候，這樣的聯合體就被稱為“聚集體”。在聚集體中包含覆蓋程序並不會使本授權應用於該聚集體的其他部分。

6. 發佈非源碼形式的副本

您可以在第 4、5 節條款下以目標代碼形式發佈程序，只要您同時以一下的一種方式在本授權條款下發布機器可讀的對應的源代碼：

- a) 在物理產品（包括一個物理的發佈媒介）中或作為其一部分發布目標代碼，並在通常用於軟件交換的耐用的物理媒介中發佈對應的源代碼。
- b) 在物理產品（包括一個物理的發佈媒介）中或作為其一部分發布目標代碼，並附上有效期至少 3 年且與您為該產品模型提供配件或客戶服務的時間等長的書面承諾，給予每個擁有該目標代碼的人（1）要麼在通常用於軟件交換的耐用物理媒介中，以不高於您執行這種源碼的發佈行為所花費的合理費用的價格，一份該產品中所有由本授權覆蓋的軟件的對應的源代碼的拷貝；（2）要麼通過網絡服務器免費提供這些對應源代碼的訪問。
- c) 單獨地發佈目標代碼的副本，並附上一份提供對應源代碼的書面承諾。這種行為只允許偶爾發生並不能盈利，且在您收到的目標代碼附有第 6 節 b 規定的承諾的時候。
- d) 在指定的地點（免費或收費地）提供發佈的目標代碼的訪問並在同樣的地點以不增加價格的方式提供對應源代碼的同樣的訪問權。您不需要要求接收者在複製目標代碼的時候一道複製對應的源代碼。如果複製目標代碼的地點是網絡服務器，對應的源代碼可以在另外一個支持相同複製功能的服務器上（由您或者第三方運作），只要您在目標代碼旁邊明確指出在哪裡可以找到對應的源代碼。無論什麼樣的服務器提供這些對應的源代碼，您都有義務保證它在任何有需求的時候都可用，從而滿足本條規定。
- e) 用點對點傳輸發佈目標代碼，您需要告知其他的節點目標代碼和對應的源代碼在哪裡按照第 6 節 d 的條款向大眾免費提供。

目標代碼中可分離的部分，其源代碼作為系統庫不包含在對應的源代碼中，不需要包含在發佈目標代碼的行為中。

“用戶產品”指（1）“消費品”，即通常用於個人的、家庭的或日常目的的有形個人財產；或者（2）任何為公司設計或銷售卻賣給了個人的東西。

在判斷一個產品是否消費品時，有疑點的案例將以有利於覆蓋面的結果加以判斷。對特定用戶接收到的特定產品，“正常使用”指該類產品的典型的或通常的使用，無論該用戶的特殊情況，或者該用戶實際使用該產品的情況，或者該產品要求的使用方式如何。一個產品是否是消費品與該產品是否具有實質的經濟上的、工業的或非消費品的用處無關，除非該用處是此類產品唯一的重要使用模式。

用戶產品的“安裝信息”指從對應源碼的修改版本安裝和運行該用戶產品中包含的覆蓋程序的修改版本所需要的任何方法、過程、授權密鑰或其他信息。這些信息必須足以保證修改後的目標代碼不會僅僅因為被修改過而不能繼續運行。

如果您在本節條款下在用戶產品中，或隨同，或專門為了其中的使用，發佈目標代碼程序，而在發佈過程中用戶產品的所有權和使用權都永久地或在一定時期內（無論此項發佈的特點如何）傳遞給了接收者，在本節所述的條款下發布的對應的源代碼必須包含安裝信息。但是如果您或者任何第三方組織都沒有保留在用戶產品上安裝修改過的目標代碼的能力（比如程序被安裝在了 ROM 上），那麼這項要求不會生效。

提供安裝信息的要求並沒有要求為接收者修改或安裝過的程序，或者修改或安裝該程序的用戶產品，繼續提供支持服務、擔保或升級。當修改本身實際上相反地影響了網絡的運行，或者違反了網絡通信的規則和協議時，網絡訪問可以被拒絕。

根據本節發佈的對應源代碼和提供的安裝信息必須以公共的文件格式發佈（並附加一個該類型文檔的實現方法以源碼形式向公眾共享），解壓縮、閱讀或複製這些信息不能要求任何密碼。

7. 附加條款

“附加許可”是通過允許一些本授權的特例來補充本授權的條款。只要它們在使用法律下合法，對整個程序都生效的附加許可就應當被認為是本授權的內容。如果附加許可只是對本程序的一部分生效，那麼該部分可以在那些許可下獨立使用，但整個程序是在本授權管理下，無論附加許可如

何。

當您發佈覆蓋程序的副本時，您可以選擇刪除該副本或其部分的任何附加許可。（當您修改程序時，附加許可可能要求在某些情況下將自身刪除）。您可以把附加許可放在材料上，加入到您擁有或能授予版權許可的覆蓋程序中。

儘管本授權在別處有提供，對於您加入到程序中的材料，您可以（如果您由該材料的版權所有者授權的話）用以下條款補充本授權：

- a. 拒絕擔保責任或以與本授權第 15 和 16 小節條款不同的方式限制責任；或者
- b. 要求保留特定的合理法律通告，或者該材料中或包含於適當法律通告中的該程序的作者貢獻；或者
- c. 禁止誤傳該材料的來源，或者要求該材料的修改版本以合理的方式標誌為與原版本不同的版本；或者
- d. 限制以宣傳為目的的使用該材料作者或授權人的姓名；或者
- e. 降低授權級別以在商標法下使用一些商品名稱，商標或服務標記；或者
- f. 要求任何發佈該材料（或其修改版本）的人用對接收者的責任假設合同對授權人和材料作者進行保護，避免任何這樣的假設合同直接造成授權人和作者的責任。

所有其他不許可的附加條款都被認為是第 10 節中的“進一步的約束”。如果您收到的程序或者其部分，聲稱自己由本授權管理，並補充了進一步約束，那麼您可以刪除這些約束。如果一個授權文件包含進一步約束，但是允許再次授權或者在本授權下發布，只要這樣的進一步的約束在這樣的再次授權或發佈中無法保留下來，您就可以在覆蓋程序中加入該授權文件條款管理下的材料。

如果您依據本小節向覆蓋程序添加條款，您必須在相關的源碼文件中加入一個應用於那些文件的附加條款的聲明或者指明在哪裡可以找到這些條款的通告。

附加的條款，無論是許可的還是非許可的條款，都可以寫在一個單獨

的書面授權中，或者申明為例外情況；這兩種方法都可以實現上述要求。

8. 終止授權

您只有在本授權的明確授權下才能傳播或修改覆蓋程序。任何其它的傳播或修改覆蓋程序的嘗試都是非法的，並將自動終止您在本授權下獲取的權利（包括依據第 11 節第三段條款授予的任何專利授權）。

然而，如果您停止違反本授權，那麼您從某個特定版權所有者處獲取的授權許可能夠以以下方式恢復（a）您可以暫時地擁有授權，直到版權所有者明確地終止您的授權；（b）如果在您停止違反本授權後的 60 天內，版權所有者沒有以某種合理的方式告知您的違背行為，那麼您可以永久地獲取該授權。

進一步地，如果某個版權所有者以某種合理的方式告知您違反本授權的行為，而這是您第一次收到來自該版權所有者的違反本授權的通知（對任何軟件），並且在收到通知後 30 天內修正了違反行為，那麼您從該版權所有者處獲取的授權將永久地恢復。

當您的授權在本節條款下被終止時，那些從您那獲取授權的組織只要保持不違反本授權協議，其授權就不會被終止。您只有在授權被版權所有者恢復了之後才有資格依據第 10 節的條款獲取該材料的新的授權。

9. 獲取副本不需要接受本授權

您不需要為了接收或運行本程序的副本而接受本授權協議。僅僅是因為點對點傳輸獲取副本引起傳播行為，也不要求您接受本授權協議。然而，除了本授權外，任何授權協議都不能授予您傳播或修改覆蓋程序的許可。因此，如果您修改或者傳播了本程序的副本，那麼您就默認地接受了本授權。

10. 下游接收者的自動授權

每次您發佈覆蓋程序，接收者都自動獲得一份來自原授權人的依照本授權協議運行、修改和傳播該程序的授權。依據本授權，您不為執行任何第三方組織的要求負責。

“實體事務”指轉移一個組織的控制權或全部資產，或者拆分組織，或者合併組織的事務。如果覆蓋程序的傳播是實體事務造成的，該事務中每一個接收本程序副本的組織都將獲取一份其前身擁有的或者能夠依據前面的條款提供的任何授權，以及從其前身獲取程序對應的源代碼的權利，如果前身擁有或以合理的努力能夠獲取這些源代碼的話。

您不可以對從本授權協議獲取或確認的權利的執行強加任何約束。比如，您不可以要求授權費用，版稅要求或對從本授權獲取的權利的執行收取任何費用。您不可以發起訴訟（包括聯合訴訟和反訴）聲稱由於製作、使用、銷售、批發或者引進本程序或其任何一部分而侵犯了任何專利權。

11. 專利權

“貢獻者”是在本授權下授予本程序或者本程序所基於的程序的版權的所有者。這樣的程序被成為貢獻者的“貢獻者版本”。

一個貢獻者的“實質的專利申明”是該貢獻者所佔有和控制的全部專利，無論已經獲得的還是在將來獲得的，那些可能受到某種方式侵犯的專利權。本授權允許製作、使用和銷售其貢獻者版本，但不包括那些只會由於對貢獻者版本進一步的修改而受到侵犯的專利的申明。為此，“控制”一詞包括以同本授權要求一致的方式給予從屬授權的權利。

每個貢獻者在該貢獻者的實質的專利申明下授予您非獨家的，全世界的，不需要版稅的專利授權，允許您製作、使用、銷售、批發、進口以及運行、修改和傳播其貢獻者版本內容。

在以下三個自然段中，“專利授權”指任何形式表達的不執行專利權的協議或承諾（例如使用專利權的口頭許可，或者不為侵犯專利而起訴的契約）。向一個組織授予專利授權指做出這樣的不向該組織提出強制執行專

利權的承諾。

如果您在自己明確知道的情況下發布基於某個專利授權的覆蓋程序，而這個程序的對應的源代碼並不能在本授權條款下通過網絡服務器或其他有效途徑免費地向公眾提供訪問，您必須做到：（1）使對應的源代碼按照上述方法可訪問；或者（2）放棄從該程序的專利授權獲取任何利益；或者（3）以某種與本授權要求一致的方法使該專利授權延伸到下游的接收者。“在自己明確知道的情況下”指您明確地知道除了獲取專利授權外，在某個國家您傳播覆蓋程序的行為，或者接收者使用覆蓋程序的行為，會由於該專利授權而侵犯一個或多個在該國可確認的專利權，而這些專利權您有足夠的理由相信它們是有效的。

在依照或者涉及某一次事務或安排時，如果您通過獲取發佈或傳播覆蓋程序的傳輸版本，並給予接收該覆蓋程序的某些組織專利授權，允許他們使用，傳播，修改或者發佈該覆蓋程序的特殊版本，那麼您賦予這些組織的專利授權將自動延伸到所有該覆蓋程序及基於該程序的作品接收者。

一份專利授權是“有偏見的”，如果它沒有在自身所覆蓋的範圍內包含，禁止行使，或者要求不執行一個或多個本授權下明確認可的權利。以下情況，您不可以發佈一個覆蓋程序：如果您與軟件發佈行業的第三方組織有協議，而該協議要求您根據該程序的發佈情況向該組織付費，同時該組織在你們的協議中賦予任何從您那裡獲得覆蓋軟件的組織一份有偏見的專利授權，要麼（a）連同您所發佈的副本（或者從這些副本製作的副本）；要麼（b）主要為了並連同某個的產品或者包含該覆蓋程序的聯合體。如果您簽署該協議或獲得該專利授權的日期早於 2007 年 3 月 28 日，那麼您不受本條款約束。

本授權的任何部分不會被解釋為拒絕或者限制任何暗含的授權或其他在適用專利權法下保護您的專利不受侵犯的措施。

12. 不要放棄別人的自由

如果您遇到了與本授權向矛盾的情況（無論是法庭判決，合同或者其他情況），它們不能使您免去本授權的要求。如果您不能同時按照本授權中的義務和其他相關義務來發布覆蓋程序，那麼您將不能發佈它們。比如，如果您接受了要求您向從您這裡或許本程序的人收取版稅的條款，您唯一能夠同時滿足本授權和那些條款的方法是完全不要發佈本程序。

13. 和 GNU Affero 通用公共授權一起使用

儘管本協議有其他防備條款，您有權把任何覆蓋程序和基於第三版 GNU Affero 通用公共授權的程序鏈接起來，並且發佈該聯合程序。本授權的條款仍然對您的覆蓋程序有效，但是 GNU Affero 通用公共授權第 13 節關於通過網絡交互的要求會對整個聯合體有效。

14. 本授權的修訂版

自由軟件基金會有時候可能會發布 GNU 通用軟件授權的修訂版本和/或新版本。這樣的新版本將會和現行版本保持精神上的一致性，但是可能會在細節上有所不同，以處理新的問題和情況。

每個版本都有一個單獨的版本號。如果本程序指出了應用於本程序的一個特定的 GNU 通用公共授權版本號“以及後續版本”，您將擁有選擇該版本或任何由自由軟件基金會發布的後續版本中的條款和條件的權利。如果本程序沒有指定特定的 GNU 通用公共授權版本號，那麼您可以選擇任何自由軟件基金會已發佈的版本。

如果本程序指出某個代理可以決定將來的 GNU 通用公共授權是否可以應用於本程序，那麼該代理的接受任何版本的公開稱述都是您選擇該版本應用於本程序的永久認可。

後續的授權版本可能會賦予您額外的或者不同的許可。但是，您對後續版本的選擇不會對任何作者和版權所有者強加任何義務。

15. 免責申明

在適用法律許可下，本授權不對本程序承擔任何擔保責任。除非是書面申明，否則版權所有者和/或提供本程序的第三方組織，“照舊”不承擔任何形式的擔保責任，無論是承諾的還是暗含的，包括但不限於就適售性和為某個特殊目的的適用性的默認擔保責任。有關本程序質量與效能的全部風險均由您承擔。如本程序被證明有瑕疵，您應承擔所有必要的服務、修復或更正的費用。

16. 責任範圍

除非受適用法律要求或者書面同意，任何版權所有者，或任何依前述方式修改和/或發佈本程序者，對於您因為使用或不能使用本程序所造成的一般性、特殊性、意外性或間接性損失，不負任何責任（包括但不限於，資料損失，資料執行不精確，或應由您或第三人承擔的損失，或本程序無法與其他程序運作等），即便該版權所有者或其他組織已經被告知程序有此類損失的可能性也是如此。

17. 第 15 和 16 節的解釋

如果上述免責申明和責任範圍不能按照地方法律條款獲得法律效力，複審法庭應該採用最接近於完全放棄關於本程序的民事責任的法律，除非隨同本程序的責任擔保或責任假設合同是收費的。

5.8.4 條款和條件結束

5.8.5 如何在您的新程序中應用這些條款？

如果您開發了一個新程序，並且希望能夠讓它儘可能地被大眾使用，達成此目的的最好方式就是讓它成為自由軟件。任何人都能夠依據這些條款對該軟件再次發佈和修改。

為了做到這一點，請將以下聲明附加到程序上。最安全的作法，是將

聲明放在每份源碼文件的起始處，以有效傳達無擔保責任的訊息；且每份文件至少應有「版權」列以及本份聲明全文位置的提示。

版權所有 (C) 本程序為自由軟件；您可依據自由軟件基金會所發表的 GNU 通用公共授權條款，對本程序再次發佈和/或修改；無論您依據的是本授權的第三版，或（您可選的）任一日後發行的版本。本程序是基於使用目的而加以發佈，然而不負任何擔保責任；亦無對適售性或特定目的適用性所為的默示性擔保。詳情請參照 GNU 通用公共授權。您應已收到附隨於本程序的 GNU 通用公共授權的副本；如果沒有，請參照 <http://www.gnu.org/licenses/>。

同時附上如何以電子及書面信件與您聯繫的資料。

如果程序進行終端交互方式運作，請在交互式模式開始時，輸出以下提示：

版權所有 (C) 本程序不負任何擔保責任，欲知詳情請鍵入 'show w'。這是一個自由軟件，歡迎您在特定條件下再發布本程序；欲知詳情請鍵入 'show c'。

所假設的指令 'show w' 與 'show c' 應顯示通用公共授權的相對應條款。當然，您可以使用 'show w' 與 'show c' 以外的指令名稱；對於圖形用戶界面，您可以用“關於”項代實現此功能。

如有需要，您還應該取得您的僱主（若您的工作為程序設計師）或學校就本程序所簽署的“版權放棄承諾書”。欲知這方面的詳情，以及如何應用和遵守 GNU 通用公共授權，請參考 <http://www.gnu.org/licenses/>。

GNU 通用公共授權並不允許您將本程序合併到私有的程序中。若您的程序是一個子程序庫，您可能認為允許私有的應用程序鏈接該庫會更有用。如果這是您所想做的，請使用 GNU 鬆弛通用公共授權代替本授權。但這樣做之前，請閱讀 <http://www.gnu.org/philosophy/why-not-lgpl.html>。

5.9 為何升級到 GPLv3

Copyright © 2007, 2009 理查德·斯托曼 (Richard Stallman) 本文
首發於 2007 年在 <http://gnu.org>。

GNU 通用公共許可證 (GNU GPL) 第三版已經發布，使得自由的軟件包可以從 GPL 第二版升級到第三版。本文章闡述了升級許可證的必要性。

首先要說明重要的一點，是否升級許可證完全是一個選擇上的考慮。GPL 第二版依然會是一份有效的許可證，就算一些程序依然留守 GPLv2 而另一些升級到 GPLv3，也完全不是什麼災難。這兩份協議固然是不兼容的，但這並不是根本性的問題。

當我們說 GPLv2 和 GPLv3 不兼容時，我們的意思是指沒有合法的方法將 GPLv2 發佈的代碼與 GPLv3 發佈的代碼合併到同一個程序中。這是由於兩份許可證都是 copyleft 許可證，都各自要求“如果你將以本協議發佈的代碼合併到一個更大的程序中，那麼這個程序也必須遵守該許可。”因此使它們兼容是不可能的。我們固然可以在 GPLv3 許可證中加入一項兼容 GPLv2 的豁免條款，但這是沒有用的，因為 GPLv2 也需要一項類似條款。

幸運的是，許可證兼容性問題僅僅在你希望鏈接、合併或組合兩個不同程序的代碼到一起時才是問題。在操作系統中 GPLv2 與 GPLv3 的程序共存是沒有任何問題的。例如，TeX 許可證與 Apache 許可證都和 GPLv2 不兼容，但是這並不能阻止我們把它們與 Linux、Bash 和 GCC 共同運行在系統中。這是因為它們都是獨立的程序，同理，如果 Bash 和 GCC 升級到了 GPLv3 但 Linux 依然使用 GPLv2，也不會有什麼衝突。

將程序保留 GPLv2 的許可不會創造新的問題。遷移到 GPLv3 的理由是，GPLv3 解決了目前現有的問題。

GPLv3 能夠阻止的一大危險就是“機頂盒化”(Tivoization，譯者注：Tivo 是美國一大有線電視機頂盒產品)，機頂盒化的意思是，某些“電器”(帶有計算機在內)包含了以 GPL 許可證發佈的軟件，但你實際上卻不可能修改它們，因為一旦設備發現了任何軟件修改，就會自動關機。通常，

廠商進行“機頂盒化”行為的動機是：廠商知道其軟件可以被人們修改，因此力圖避免這樣的修改。這些設備廠商享受了自由軟件帶來的自由，卻不允許你也這樣做。

有些人認為，自由市場中不同電器的自由競爭能夠使這種噁心的特性儘可能的少。單靠競爭大概也確實能夠避免像“每週二下午 1 點到 5 點強制關機”這樣的反特性出現，但即使這樣，主人的選擇依然不是自由的。自由意味著你自己控制你軟件的所作所為，而不僅僅是你能祈求或者威脅那個替你做出決定的人。

在那些被數字限制管理（DRM）——一個用來限制你使用自己電腦中數據的噁心功能——所控制的關鍵領域，市場競爭毫無作用，因為相關的競爭是被禁止的：《千年數字版權法案》（DMCA）和類似的法律下，在美國和許多國家，發佈 DVD 播放器是非法的，除非播放器遵守由 DVD 利益團體制定的規則去限制用戶（它們的網站是 <http://www.dvdcqa.org/>，但那些規則卻似乎沒有刊登其上）。公眾們沒有辦法購買沒有 DRM 的播放器將 DRM 拒之門外，因為市場上根本就沒有啊。無論有多少產品可供你選擇，它們都帶有相同的數字手銬。

GPLv3 則確保你具有去除這些手銬的自由。它本身並沒有禁止 DRM 或者任何類似的特性。它沒有對你可以加入到程序中的實質性功能做任何的限制，同樣也沒有限制你刪除這些功能。也就是說，發行商有權利加入這些噁心的功能，你也有權利去除這些功能。而“機頂盒化”則是廠商拒絕你自由的方式，而為了保護你的自由，GPLv3 禁止了“機頂盒化”。

對“機頂盒化”的取締包括了消費者所能購買到的任何產品，即使是不常見的產品。GPLv3 僅僅在產品幾乎完全被企業和機構獨佔時才對“機頂盒化”網開一面。

GPLv3 所能抵禦的另一個威脅就是像 Novell 和 Microsoft 這樣的專利交易。微軟希望通過它的數千項專利，被迫用戶為獲得運行 GNU/Linux 的特權而交費，而簽訂了這樣一份協議來實現這個目標。這筆交易給 Novell 的消費者在微軟的專利面前提供了相當有限的保護。

然而微軟在與 Novell 的交易中犯了一些錯誤，而 GPLv3 旨在利用它

們反擊微軟，排除這種對整個社區相當有限的專利保護。為了得益於這個保護措施，程序需要使用 GPLv3。

微軟的律師們當然也不是傻子，他們下次的時候肯定會設法避免錯誤。因此，GPLv3 中也明確表示，他們沒有“下次”了。將程序以 GPL 第三版發佈，將保護髮行商免於微軟向最終用戶的程序徵收專利費。

GPLv3 同樣提供了明確的專利保護，讓用戶免受程序的貢獻者與分發者的侵擾。在 GPLv2 中僅僅規定用戶可以從公司獲得一份程序拷貝，或者分發程序拷貝而免於起訴，專利保護僅僅是隱含在其中的。

然而，GPLv3 中的專利保護並沒有我們希望的那麼強力。在理想的狀態下，我們當然希望任何分發 GPL 許可軟件的人，和不使用 GPL 許可證份發軟件的人，都放棄他們全部的軟件專利，因為軟件專利根本就不應該存在。軟件專利的體制是惡毒的、荒謬的，將所有軟件開發者都置於被一個聞所未聞的公司起訴的危險之下，更別說那些大型集團了。大型程序往往包含了成百上千的思想，因此這些思想被上百個專利所涵蓋就不足為奇了。大型集團公司收集這些成千上萬的專利，然後利用它們欺壓中小開發者。專利已經對自由軟件的發展形成了阻礙。

使軟件開發變得安全的唯一方式就是廢除軟件專利，而我們希望在未來的某一天實現。然而，我們不能夠通過一份軟件許可證就能做到這一點。任何程序，無論自由與否，都能夠被八杆子打不著的一個人手中的專利殺死，而這是程序的許可協議本身所不能避免的。唯一能夠使軟件開發免於專利威脅的，只有法院的一紙判決，或者專利法律的修訂。如果我們試圖靠 GPLv3 來做到這點，這必然是會失敗的。

因此，GPLv3 力求限制與“疏導”危險，更具體的說，就是我們試圖將自由軟件從比死亡更加悲慘的命運中拯救，避免自由軟件通過專利都變為了事實上的私有軟件。GPLv3 中明確的專利協議確保了使用 GPL 的公司能給予用戶四大自由，而不能通過專利來改變這點，對用戶說“自由不屬於你”。這同時也避免了勾結其他專利持有人做這種事情的可能性。

GPLv3 的更多優點包括了更好的國際化、更溫和的權利終止、對 BitTorrent 的支持，以及對 Apache 許可證的兼容。所有的這一切都是升級

的眾多理由。

變化，在 GPLv3 發佈之後顯然是不會停止的。如果產生了對用戶的危險，我們將研發 GPL 第四版。因此，假如我們寫好了 GPLv4，確保程序能順利升級顯然就很重要了。

做到這一切的方式之一就是將程序以“GPL 第三版或任何更新的版本”發佈；另一個方法是讓項目所有的貢獻者指定一位可以解決是否升級 GPL 版本的代理人；而第三種方式則是讓所有的貢獻者將各自的版權統一移交給一位版權持有人，這位版權持有人將負責升級協議版本。無論採用何種方式，程序應當能靈活升級到未來的 GPL 版本。

5.10 GNU 寬通用許可證

中文翻譯 Leo leohca@yahoo.com 源網頁:<http://www.thebigfly.com/gnu/lgpl/lgpl-v3.php> 修改: IFRFSX 1079092922@qq.com

這是一份 GNU 寬通用許可證非正式的中文翻譯。它並非由自由軟件基金會所發表，亦非使用 GNU 寬通用許可證的文件的法定發佈條款——只有 GNU 寬通用許可證英文原文 <http://www.gnu.org/licenses/lgpl.html> 的版本始具有此等效力。然而，我們希望這份翻譯能幫助中文的使用者更瞭解 GNU 寬通用許可證。

This is an unofficial translation of the LGPL Free Documentation License into Chinese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documents that uses the GNU LGPL--only the original English text of the GNU LGPL <http://www.gnu.org/licenses/lgpl.html> does that. However, we hope that this translation will help Chinese speakers understand the GNU LGPL better

第 3 版，2007 年 6 月 29 日

版權所有 (C) 2007 自由軟件基金會 <http://fsf.org/>

允許每個人複製和發佈本許可證文件的完整副本，但不允許對它進行任何修改。

本版本的 GNU 較寬鬆公共許可證包含了 GNU 通用公共授權第三版裡的項目和條款，以下是複製，發佈和修改軟件的詳細條款和條件。

5.10.1 0. 附加定義

“本許可證”指 GNU 較寬鬆公共許可證第三版，“GNU GPL”是指通用公共授權第三版。

“庫”指在本許可證管理下的涉及的作品，有別於下列定義的應用軟件或組合軟件。

“應用程序”指任何通過“庫”提供用於接口使用的軟件，但非基於“庫”。用“庫”定義一個集的子集被認為是“庫”提供的使用接口的一種模式。

“組合軟件”是“庫”和“應用軟件”結合或連接而產生的一種軟件。這種通過“組合軟件”產生的特殊版本的“庫”也被稱之為“連接版本”。

“組合軟件”裡“最小對應源代碼”是指“組合軟件”基於“應用程序”，而非“連接版本”裡的“對應源代碼”，不包括被認為是孤立“組合軟件”部份裡的任何原代碼。

“組合軟件”裡“對應應用程序代碼”是指“應用程序”的目標編碼和/或原代碼，包括“應用程序”裡重新編制“組合軟件”所需的數據和有效程序，但不包括“組合軟件”裡的系統數據庫。

5.10.2 1. 通用公共授權第三版之例外

你可以發佈本許可證第三和四部份下的涉及的作品，不受 GNU GPL 第三部份的限制。

5.10.3 2. 發佈修訂版本

如果你修改了“庫”的副本，並且在你的修改版本中，一個組件引用了使用此組件的“應用程序”提供的函數或數據（有別於作為組件被調用時傳遞的參數），那麼你可以發佈修改版的副本：

a) 在此許可下，假如你盡力確保在“應用程序”不能提供這些函數和數據的情況下，該組件仍能工作，並執行其目標中仍有意義的任何部份，否則

- b) 在通用公共授權下，任何本授權中的額外許可都不適用於該副本。

5.10.4 3. 合併庫標頭文件裡的資料為目標代碼

以目標代碼形式的程序可以從庫裡標頭文件的資料合併而成是庫的一部分。你可以按照你選擇的條件上傳這樣的目標代碼，只要組合資料對數字參數，數據結構設計，存取器或小的宏，排列功能和模版不受限定（長度小於 10 排），你可以做下列兩項情況：

- a) 目標代碼的每一個副本要給出明確通告庫的使用要在本許可證下發布。
- b) 目標代碼要與 GNU 公用授權副本和本許可證文件一起使用。

5.10.5 4. 組合軟件

你可以根據你的選擇來傳送一個合併後的作品，並且有效地發佈但不限制修改庫（包含在組合軟件裡的庫）的一部分，還可逆向調試這種修改，只要你滿足以下的每一個條件：

- a) 目標代碼的每一個副本要給出明確通告庫的使用要在本許可證下發布。
- b) 目標代碼要與 GNU 公用授權副本和本許可證文件一起使用。
- c) 組合軟件在完成期間要顯示版權通告，在這些通告中要包括庫的版權通告，同時也要指出給使用者 GNU 公用授權的副本和本許可證文件作為參考。
- d) 做下列之一：
 - 0) 在本許可證下發布最小對應源代碼，對應應用程序代碼的形成，在此條件下允許用戶用修改版重新組合或聯接應用程序產生修改的組合程序，用此方式用以說明通過通用公共授權第六部分發布對應源代碼。
 - 1) 使用適當的共享庫機構聯接庫。適當機制是指 (a) 在運行時使用已經存在於用戶電腦系統的庫的副本。(b) 庫的修改版可以正常工作並且與連接版本兼容。

e) 提供安裝信息，但只有如果你，否則必須根據通用公共授權第六部份提供這些資料，範圍僅限於這些資料是必要的安裝和執行修改後的版本合併的工作所產生的重組或再重新聯接應用程序與聯繫版本的修改版本。（如果選擇 4d0，安裝信息必須與最小通訊資源和通訊應用軟件代碼一起使用。如果選擇 4d1，你必須在指定的方式下提供安裝信息，由通用公共授權第六部份傳輸通訊資源。）

5.10.6 5. 組合的庫

您可以將基於該庫的作品的庫組件與其他非應用程序且不在本許可範圍內的庫組件並排放置在一個單獨的庫中，並根據您的選擇，發佈這樣組合的庫，只要做到以下全部兩項：

a) 將組合的庫與基於該（按本協議發佈的）庫的同一作品的未與任何其他庫組件組合的副本一併發佈。

b) 對組合庫給出醒目的通知，說明其部分是基於該庫的作品，並解釋在哪裡找到一併發佈的同一作品的未組合形式。

5.10.7 6.GNU 寬通用許可證的修改版

自由軟件基金會有時候可能會發布 GNU 較寬鬆公共許可證的修訂版本和/或新版本。這樣的新版本將會和現行版本保持精神上的一致性，但是可能會在細節上有所不同，以處理新的問題和情況。

每個版本都有一個單獨的版本號。如果您收到的庫指出了應用於本程序的一個特定的 GNU 較寬鬆公共許可證版本號“以及後續版本”，您將擁有選擇該版本或任何由自由軟件基金會發布的後續版本中的條款和條件的權利。如果您收的庫沒有指定特定的 GNU 較寬鬆公共許可證版本號，那麼您可以選擇任何自由軟件基金會已發佈的版本。

如果您收到的庫指出某個代理可以決定將來的 GNU 較寬鬆公共許可證是否可以應用於本庫，那麼該代理的接受任何版本的公開稱述都是您選擇該版本應用於本庫的永久認可。

5.11 GNU 自由文檔許可證

此版本由 Leo-Hong (leohca (at) yahoo.com) 翻譯整理。源網頁：
<http://www.thebigfly.com/gnu/FDLv1.3/>。修改者：IFRFSX
1079092922@qq.com

這是一份 GNU 自由文檔許可證非正式的中文翻譯。它並非由自由軟件基金會所發表，亦非使用 GNU 自由文檔許可證的文件的法定發佈條款——只有 GNU 自由文檔許可證英文原文 <http://www.gnu.org/licenses/fdl.html> 的版本始具有此等效力。然而，我們希望這份翻譯能幫助中文的使用者更瞭解 GNU 自由文檔許可證。

This is an unofficial translation of the GNU Free Documentation License into Chinese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documents that uses the GNU FDL--only the original English text of the GNU FDL <http://www.gnu.org/licenses/fdl.html> does that. However, we hope that this translation will help Chinese speakers understand the GNU FDL better.

第 1.3 版，2008 年 11 月

Copyright © 2000,2001,2002,2007,2008 自由軟件基金會 <http://fsf.org/>

允許每個人複製和發佈本許可證文件的完整副本，
但不允許對它進行任何修改。

5.11.1 0. 導言

本許可證的目的在於作為一種手冊、教科書或其它的具有功能性的有用文件獲得自由：確保每一個人都具有複製和重新發布它的有效自由，而

不論是否作出修改，也不論其是否具有商業行為。其次，本許可證保存了作者以及出版者由於他們的工作而得到名譽的方式，同時也不被認為應該對其他人所作出的修改而擔負責任。

本許可證是一種“copyleft”，這表示文件的衍生作品本身必須具有相同的自由涵義。它補足了 GNU 公共通用許可證——一種為了自由軟件而設計的“copyleft”許可證。

我們設計了本許可證是為了將它使用到自由軟件的使用手冊上，因為自由軟件需要自由的文檔：一種自由的程序應該提供與此軟件具有相同的自由的使用手冊。但是本許可證並不被限制在軟件使用手冊的應用上；它可以被用於任何以文字作基礎的作品，而不論其主題內容，或者它是否是一個被出版的印刷書籍。我們建議本許可證主要應用在以使用說明或提供參考作為目的的作品上。

5.11.2 1. 效力與定義

本許可證的效力在於任何媒體中的任何的使用手冊或其它作品，只要其中包含由版權所有人所指定的聲明，說明它可以在本許可證的條款下被髮布。這樣的一份聲明提供了全球範圍內的，免版稅的和沒有期限的許可，在此所陳述條件下使用那個作品。以下所稱的文件，指的是任何像這樣的使用手冊或作品。公眾中的任何成員都是被許可人，並且稱作為你。如果你以一種需要在版權法下取得允許的方式進行復制、修改或發佈作品，你就接受了這項許可。

“修改版本”指的是任何包含文件或是它的其中一部份，不論是逐字的複製或是經過修正，或翻譯成其它語言的任何作品。

“次要章節”是一個具名的附錄，或是文件的本文之前內容的章節，專門用來處理文件的出版者或作者，與文件整體主題（或其它相關內容）的關係，並且不包含任何可以直接落入那個整體主題的內容。（因此，如果文件的部分內容是作為數學教科書，那麼其次要章節就可以不用來解釋任何數學。）它的關係可以是與主題相關的歷史連接，或是與其相關的法律、商業、哲學、倫理道德或政治立場。

“不變章節”是標題已被指定的某些次要章節，在一個聲明瞭是以本許可證加以發行的文件中，依此作為不變章節。如果一個章節並不符合上述有關於次要的定義時，則它並不允許被指定為不變。文件可以不包含不變章節。如果文件並沒有指出任何不變章節，那麼就是沒有。

“封面文字”是某些被加以列出的簡短文字段落，在一個聲明瞭是以本許可證加以發行的文件中，依此作為前封面文字或後封面文字。前封面文字最多可以包含 5 個單詞，後封面文字最多可以包含 25 個單詞。

文件的“透明”副本指的是一份機器可讀的副本，它以一种一般公眾可以取得其規格說明的格式來表現，適合於直接用一般文字編輯器、一般點陣圖像程序用於由圖元像素構成的影像或一些可以廣泛取得的繪圖程序用於由向量繪製的圖形直接地進行修訂；並且適合於輸入到文字格式化程式，或是可以自動地轉換到適合於輸入到文字格式化程序的各種格式。一份以透明以外的檔案格式所構成的副本，其標記或缺少標記，若是被安排成用來挫折或是打消讀者進行其後續的修改，則此副本並非透明。一種影像格式，如果僅僅是用來充斥文本的資料量時，就不是透明的。一個不是透明的副本被稱為混濁。

透明副本適合格式的例子包括有：沒有標記的純 ASCII、Texinfo 輸入格式、 \LaTeX 輸入格式、使用可以公開取得其 DTD 的 SGML 或 XML、合乎標準的簡單 HTML、PostScript 或 PDF。透明影像格式的例子有 PNG、XCF 和 JPG。混濁格式包括只能夠以私人文書處理器閱讀以及編輯的私人格式、DTD 以及或處理工具不能夠一般地加以取得的 SGML 或 XML、以及由某些文書處理器只是為了輸出的目的而做出的，由機器製作的 HTML、PostScript 或 PDF。

標題頁對一本印刷書籍來說，指的是標題頁本身，以及所需要用來容納本許可證必須出現在標題頁的易讀內容的，如此的接續數頁。對於並沒有任何如此頁面的作品的某些格式，標題頁指的是本文主體開始之前作品標題最顯著位置的文字。

出版者指的是把那些文本副本發佈給公眾的任何人或實體。

一個標題為 XYZ 的章節指的是文件的一個具名的次要單元，其標題

精確地為 XYZ 或是將 XYZ 包含在跟著翻譯為其它語言的 XYZ 文字後面的括號內——這裡 XYZ 代表的是名稱於下提及的特定章節，像是感謝、貢獻、背書或歷史。當你修改文件時，給像這樣子的章節保存其標題指的是，它保持為一個根據這個定義的標題為 XYZ 的章節。

文件可以在用來陳述本許可證效力及於文件的聲明後，包括擔保放棄。這些擔保放棄被考慮為以提及的方式，包括在本許可證中，但是隻被看作為放棄擔保之用：任何這些擔保放棄可能會有其它暗示都是無效的，並且也對本許可證的含義沒有影響。

以下是有關複製、發佈及修改的明確條款及條件。

5.11.3 2. 逐字的複製

你可以複製或發佈文件於任何媒介，而不論其是否具有商業買賣行為，其條件為具有本許可證、版權聲明和許可聲明，說明本許可證適用於文件的所有副本，並且你沒有增加任何其它條件到本許可證的條件中。你不可以使用技術手段，來妨礙或控制你所製作或發佈的副本閱讀或進一步的發佈。然而，你可以接受補償以作為副本的交換。如果你發佈了數量足夠大的副本，你也必須遵循第三條的條件。

你也可以在上述的相同條件下借出副本，並且你可以公開地陳列副本。

5.11.4 3. 大量地複製

如果你出版文件的印刷副本或者通常具有印刷封面的媒體的副本，數量上超過一百個單位，而且文件許可聲明要求有封面文字，那麼你必須將這些副本附上清楚且易讀的文字：前封面文字於前封面上、後封面文字於後封面上。這兩種封面必須清楚易讀地辨認出，你是這些副本的出版者。前封面文字必須展示完整的標題，而標題的文字應當同等地顯著可見。你可以增加額外的內容於封面上。僅在封面作出改變的複製，只要它們保存了文件的標題，並且滿足了這些條件，可以在其它方面被看作為逐字的複製。

如果對於任意一個封面所需要的文字，數量過於龐大以至於不能符合易讀的原則，你應該在實際封面的最前面列出所能符合易讀原則的內容，然後將剩下的接續在相鄰的頁面。

如果你出版或發佈數量超過一百個單位文件的混濁副本，你必須與此混濁副本一起包含一份機器可讀的透明副本，或是與一份混濁副本一起或其陳述一個計算機網絡位置，使一般的網絡使用公眾具有存取權，可以使用公開標準的網絡協定，下載一份文件的完全透明副本，此副本中並且沒有增加額外的內容。如果你使用後面的選項，當你開始大量地發佈混濁副本時，你必須採取合理的審慎步驟，以保證這個透明副本將會在發佈的一開始就保持可供存取，直到你最後一次發佈那個發行版的一份混濁副本給公眾後，至少一年為止。以保證這個透明副本，將會在所陳述的位址保持如此的可存取性，直到你最後一次發佈那個發行版直接或經由你的代理商或零售商的一份混濁副本給公眾後，至少一年為止。

你被要求，但不是必須，在重新發布任何大數量的副本之前與文件的作者聯絡，給予他們提供你一份文件的更新版本的機會。

5.11.5 4. 修改

你可以在上述第二條和第三條的條件下，複製和發佈文件的修改版本，其條件為你要精確地在本許可證下發布修改版本，且修改版本補足了文件的角色，從而允許修改版本的發佈和修改權利給任何擁有它副本的人。另外，你必須在修改版本中做這些事：

第一款、在標題頁或在封面上使用，如果有與先前版本不同的文件，應該被列在文件的歷史章節不同的標題。如果版本的原始出版者允許，你可以使用與某一個先前版本相同的標題。

第二款、在修改版本的標題頁上列出擔負作者權的一個或多個人或實體作為作者，並且列出至少五位文件的主要作者。如果少於五位，則列出全部的主要作者，除非他們免除了你這個要求。

第三款、在標題頁陳述修改版本的出版者的名稱作為出版者。

第四款、保存文件的所有版權聲明。

第五款、為你的修改增加一個與其它版權聲明相鄰的適當的版權聲明。

第六款、在版權聲明後面，以許可證附錄所顯示的形式，包括一個給予公眾在本許可證條款下使用修改版本的許可聲明。

第七款、在那個許可聲明中保存固定章節和文件許可聲明中必要封面文字的全部列表。

第八款、包括一個未被改變的本許可證的副本。

第九款、保存標題為歷史的章節和其標題，並且增加一項至少陳述如同在標題頁中所給的修改版本的標題、年份、新作者和出版者。如果在文件中沒有標題為歷史的章節，則製作出一個陳述如同在它的標題頁中所給的文件的標題、年份、新作者和出版者，然後增加一項描述修改版本如前面句子所陳述的情形。

第十款、如果有的話，保存在文件中為了給公眾存取文件的透明副本，而給予的網絡位址，以及同樣地在文件中為了它所根據的先前版本，而給予的網絡位址。這些可以被放置在歷史章節。你可以省略一個在文件本身之前，已經至少出版了四年的作品的網絡位址，或是如果它所參照的那個版本的原始出版者給予允許的情形下也可以省略它。

第十一款、在任何標題為感謝或貢獻的章節，保存章節的標題，並且在那章節保存到那時候為止，每一個貢獻者的感謝以及或貢獻的所有聲色。

第十二款、保存文件的所有固定章節，於其文字以及標題皆不得變更。章節號碼或其同等物並不被認為是章節標題的一部份。

第十三款、刪除任何標題為背書的章節。這樣子的章節不可以被包括在修改版本中。

第十四款、不要重新命名任何現存的章節，而使其標題為背書，或造成與任何固定章節相沖突的標題。

第十五款、保存任何的擔保放棄。

如果修改版本包括新的本文之前內容的章節，或合乎作為次要章節的附錄，並且沒有包含複製自文件的內容，則你具有選擇可以指定一些或全部這些章節為恆常的。要這樣做，將它們的標題增加到在修改版本許可聲明中的固定章節列表中。這些標題必須可以和任何其它章節標題加以區別。

你可以增加一個標題為背書的章節，其條件為它僅只包含由許多團體所提供的你的修改版本的背書——舉例來說，同儕評審的說明，或本文已經被一個機構認可為一個標準的權威定義。

你可以增加一個作為前封面文字的最多五個字的段落，以及一個作為後封面文字的最多二十五個字的段落，到修改版本的封面文字列表的後面。前封面文字和後封面文字都只能有一個段落，可以經由任何一個實體，或經由任何一個實體所作出的安排而被加入。如果文件已經在同樣的封面包括了封面文字——先前由你或由你所代表的相同實體所作出的安排而加入，則你不可以增加另外一個；但是你可以在先前出版者的明確允許下替換掉舊的。

文件的作者和出版者並不由此授權，而給予允許使用他們的名字以為了或經由聲稱或暗示任何修改版本背書為自己所應得的方式而獲得名聲的權利。

5.11.6 5. 組合文件

你可以在上述第四條的條款中對於修改版本的定義之下，將文件與其它在本許可證下發行的文件組合起來，其條件是你要在組合品中，包括所有原始文件的所有固定章節，不做修改，同時在組合作品的許可聲明中將它們全部列為固定章節，並且你要保存它們所有的擔保放棄。

組合作品只需包含本許可證的一份副本，並且重複的固定章節可以僅以單一個副本來取代。如果名稱重複但內容不同的固定章節，則將任此章節的標題，以在它的後面增加的方式加以獨特化，如果已知的話，於括號中指出那個章節的原始作者或出版者的名稱，或是指定一個獨特的號碼。在此組合作品許可聲明中固定章節的列表中，對其章節標題也作出相同的調整。

在組合品中，你必須組合在不同原始文件中，標題為歷史的任何章節，形成一個標題為歷史的章節；同樣組合任意標題為感謝或貢獻的章節。你必須刪除標題為背書的所有章節。

5.11.7 6. 文件的收集

你可以製作含有文件以及其它以本許可證發行文件的收集品，並且將本許可證對不同文件中的個別副本，以單一個包括在收集品的副本取代，其條件是你要遵循在其它方面，給予一個文件逐字複製的允許本許可證的規則。

你可以從這樣的一個收集品中抽取出一份單一的文件，並且在本許可證下將它單獨地發佈，其條件是你要在抽取出的文件中插入本許可證的一份副本，並且在關於那份文件的逐字的複製的所有其它方面，遵循本許可證。

5.11.8 7. 獨立作品的聚集

一個文件的編輯物，其中或附加於儲存物或發佈媒體的一冊的，具有其它分別且獨立的文件或作品的衍生品，如果經由編輯而產生的版權，並沒有用來限制此編輯物使用者的法律權力，而超過了個別的作品所允許的，則被稱為一個聚集品。當文件中包括一個聚集品，本許可證的效力並不僅在於此聚集品中的，於其本身並非文件的衍生作品的其它作品。

如果第三條的封面文字要求效力於這些文件的副本，並且文件的篇幅少於整個聚集品的一半，則文件的封面文字可以被放在只圍繞著文件，並於聚集品內部的封面或是電子的封面同等物上，如果文件是以電子的形式出現的話。否則它們必須出現在繞著整個聚集品的印刷封面上。

5.11.9 8. 翻譯

翻譯被認為一種修改，因此你可以在第四條的條款下發布文件的翻譯。用翻譯更換固定章節需要取得版權所有者的特別允許，但是你可以包括部份或所有固定章節的翻譯，使其附加到這些固定章節的原始版本之中。你可以包括本許可證、文件中的所有許可聲明和任何的擔保放棄的翻譯，其條件為你也必須包括本許可證的原始英文版本，以及這些聲明與放棄的原始版本。如果發生翻譯與本許可證、聲明或放棄的原始版本有任何的不同

意時，將以原始版本為準。

如果在文件中的章節被標題為感謝、貢獻或歷史，則保存標題第一條的必要條件第四條，典型上將會需要去更動實際的標題。

5.11.10 9. 終止

除了本許可證明確規定外，你不可以複製、修改、在本許可證下再設定額外條件的次授權或發佈文件。任何其它的複製、修改、在本許可證下再設定額外條件的次授權、或發佈文件的意圖都是無效的，並且將會自動地終止你在本許可證下所被保障的權利。

然而，如果你終止所有違反本許可證的行為，特定版權所有人會暫時恢復你的授權直到此版權所有者明確並最終地終止你的授權。或者特定版權所有人永久地恢復你的授權如果此版權所有人在停止違反許可證後的 60 天內沒有通過合理的方式通知你違反許可證。

此外，此版權所有者用一些合理的方式通知你違反了許可證規定，也是你第一次從此版權所有者收到違反許可證的通知，並且你在收到通知後的 30 天內終止了這種行為，那麼此版權所有者會永久地恢復你的授權。

你的權利在此章節的終止並不代表終止得到你的副本和權利的當事人的授權。如果你的權利被終止而沒有被永久性地恢復，那麼你將沒有任何權利去使用此章節的全部或部分資料和資料的副本。

5.11.11 10. 本許可證的未來改版

自由軟件基金會可能會不定期地出版自由文檔許可證新修訂過的版本。這種新版本在精神上將會與現在的版本相似，但在細節上可能會有不同，以對應新的問題或相關的事。請見 <http://www.gnu.org/copyleft/>。

本許可證的任何版本都被指定一個可供區分的版本號碼。如果文件指定一個效力於它的特定號碼版本的本許可證或任何以後的版本，你就具有選擇遵循指定的版本，或任何已經由自由軟件基金會出版的後來版本並且不是草稿的條款和條件。如果文件並沒有指定一個本許可證的版本號碼，

你就可以選擇任何一個曾由自由軟件基金會所出版的不是草稿的版本。如果文件指定一個代理能夠決定本許可證未來的那一種版本可以用，而且還指定代理公開聲明如果你接受一種版本你將會永久地被授權為文本選擇此版本的權利。

5.11.12 11. 重新授權

“MMC 網站”是指任何發佈有版權作品的網站服務器，也為任何人提供卓越的設施去編輯一些作品。任何人都可編輯的一種公眾維基就是這種服務器的一個例子。包含在這個網站的“MMC”是指任何一套在 MMC 網站上發佈的具有著作版權的作品。

“CC-BY-SA”是指 Creative Commons Attribution-Share Alike 3.0 許可證，它是被“知識共享組織”頒佈的，“知識共享組織”是一家非贏利性的，在加利福尼亞州聖弗朗西斯科具有重要的商業地位的組織。而且未來的“copyleft”版本的許可證也是被同一個組織發佈的。

“合併”是指以整體或作為另一個文本的部分發布或重新發佈一個文本。

MMC 有重新授權的資格，如果它是在 MMC 下授權；或者所有的作品首次發佈並非在 MMC 下授權，後來以整體或部分合併到 MMC 下，它們沒有覆蓋性文本或不變的章節並且在 2008 年 11 月 1 日之前合併的。

那麼 MMC 網站的操作者會在 2009 年 8 月 1 日之前的任何時間在同一網站重新發布包含在這一網站的 MMC 是經過 CC-BY-SA 授權的，只要那個 MMC 有資格重新授權。

5.11.13 許可證附錄：如何使用本許可證用於你的文件

為使用本許可證成為你撰寫成的一份文件，必須在文件中包括本許可證的一份複本，以及標題頁的後面包括許可聲明：

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 1.2 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".

如果你有固定章節、前封面文字和後封面文字，請將 with... Texts 這一行以這些文本取代：

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

如果你有不具封面文字的固定章節，或一些其它這三者的組合，將可選擇的二項合併以符合實際情形。

如果你的文件中包含有並非微不足道的程序碼範例，我們建議這些範例平行地在你的自由軟件授權選擇下，比如以 GNU General Public License 的自由軟件授權來發布，從而允許它們作為自由軟件而使用。

因為本書是基於 **GNU FPL** 許可證發佈的，所以下面附加 **FDL** 許可證的英文原版，同時亦可與前面中文翻譯版共同幫助理解。特此註明。

5.11.14 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful 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.

2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

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. (Thus, 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. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

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 Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document 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, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. 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, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include 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, PostScript or PDF 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.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

3. 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.

4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) 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 computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. 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.

5. 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 fewer than five), unless they release you from this requirement.
- (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", Preserve 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) For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, 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 to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- (o) Preserve any Warranty Disclaimers.

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.

6. 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, and that you preserve all their Warranty Disclaimers.

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 sec-

tions Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

7. 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.

8. 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, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which 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 half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

9. 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, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the

violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

11. 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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

12. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multi-

author Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: 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 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts,

and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

with the Invariant Sections being list their titles, with the Front-Cover Texts being list, and with the Back-Cover Texts being list.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

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.

5.12 關於出售例外對 GNU GPL 的影響

Copyright © 2009, 2010 理查德·斯托曼 (Richard Stallman)

當我參與聯合簽署知識生態國際 (Knowledge Ecology International) 發起的公開信，警告 Oracle 收購 MySQL (加上 Sun 公司剩下的部分) 可能不利於 MySQL 的時候，出售許可證例外便成為了一個熱門話題。

就像以下文章解釋的那樣，我對出售許可證例外的感覺是混雜的。明顯的是，不出售例外在 GNU GPL 下開發強大而複雜的軟件包是可能的，我們在做這個。MySQL 也可以這樣的方式開發。然而，MySQL 開發者已經用到出售例外的做法了。誰應該決定是否繼續這樣做呢？我認為把關於一個自由軟件項目的主要決定交給一個大型私有競爭者是不明智的，它會自然地使項目縮減，而不是更多地開發。

一件不合理的事情是更改 MySQL 的許可證到某種非 copyleft 許可證的想法。那樣會消除出售例外的可能，但允許所有種類的私有修改版本。無論 MySQL 應該走到哪裡，都不應選擇非 copyleft 許可證。

當我共同簽署了反對 Oracle 計劃好的 MySQL 收購¹ (還有 Sun 公司剩下的部分) 的信件的時候，一些自由軟件支持者很驚訝我會贊成 MySQL 開發者用過的出售許可證例外的行為。他們期望我完全地譴責這種行為。這篇文章解釋了我對這種行為的想法，以及原因。

出售例外的意思是代碼的版權所有者以自由軟件許可證將代碼發佈至公眾，然後讓顧客為在不同條款下使用同樣代碼的許可付款，例如允許它

¹James Love and Malini Aisola (Knowledge Ecology International), Richard Stallman (FSF), Jim Killock (Open Rights Group), letter to Neelie Kroes (Commissioner for Competition, European Commission), 19 October 2009, http://keionline.org/sites/default/files/ec_letter_mysql_oct19.pdf.

包括在私有應用裡面。

我們必須把出售例外的做法和另一件至關重要的不同事情區分開：一個自由程序的完全私有的擴展或版本。這兩種行動，即使是同時被一個公司所實行，都是不同的問題。在出售例外中，例外所用於的相同代碼是以自由軟件的形式對公眾可用的。只在私有許可證下的一個擴展或者修改後的版本是完全的私有軟件，並且不會優於任何其他私有軟件。這篇文章關心的是嚴格地並且只涉及例外出售的情形。

我從 20 世紀 90 年代開始就認為出售例外是可以接受的，並且有時我已經對公司建議過。有時這種做法已經讓重要的程序成為自由軟件成為可能。

KDE 桌面在 90 年代基於 Qt 庫開發。Qt 那時是私有軟件，並且 TrollTech 公司對把它嵌入在私有應用中的許可收費。TrollTech 允許 Qt 在自由應用中免費使用，但這不會讓它成為自由軟件。因此完全自由的操作系統不會包括 Qt，所以它們也不能用 KDE。

1998 年，TrollTech 的管理層認識到他們可以讓 Qt 成為自由軟件並且繼續對把它嵌入在私有軟件中的許可收費。我不記得這個建議是不是從我來的，但看到了這個改變我當然高興，它讓在自由軟件世界中使用 Qt 和 KDE 成為可能。

開始的時候，他們用自己的許可證，Q 公共許可證（QPL）——就自由軟件許可證來說是很有限制性的，並且和 GNU GPL 不兼容。之後，他們轉到了 GNU GPL；我想我已經給他們解釋了這可以為這種目的起作用。

出售例外從根本上依賴於為自由軟件的發佈使用一個 copyleft 許可證，如 GNU GPL。一個 copyleft 許可證允許嵌入一個更大的程序僅當整個合併後的程序是以那個許可證發佈的；這是它確保擴展版本也自由的做法。因此，想讓合併後的程序私有化的用戶需要特殊的許可。只有版權所有者可以授予那個許可，而出售例外就是這樣做的一種方式。其他人，在 GNU GPL 或者另一個 copyleft 許可證下收到代碼的，不能授予例外。

當我第一次聽說出售例外的做法時，我問自己這種行為是否符合倫理。如果某人買一個例外來把一個程序嵌入一個更大的私有程序，他正在

做錯誤的事情（換句話說，製作私有軟件）。是否出售例外的開發者也在做錯誤的事情呢？

如果那個含義是有效的，那麼它也能用於發佈同樣軟件於非 copyleft 自由許可證下的行為，如 X11 許可證，那也允許了這樣的嵌入。所以我們要麼不得不得出結論，以 X11 許可證發佈任何東西都是錯誤的，或者拒絕這種含義。使用一個非 copyleft 許可證是不好的，並且通常是一個次等的選擇，但它不是錯誤的。

換句話說，出售例外允許私有軟件中有限地嵌入代碼，但 X11 許可證走得更遠，允許無限制地在私有軟件中使用代碼（及其修改後的版本）。如果這不讓 X11 許可證不可接受，那麼它不能讓出售例外不可接受。

FSF 不實行出售例外有三個原因。一個是它不會通往 FSF 的目標：確保我們軟件的每個用戶是自由的。那是我們寫 GNU GPL 的目的，最徹底地實現這個的方法就是在 GPL 第三版或後續版本下發布，並且不允許嵌入於私有軟件。出售例外不能實現這個，就像在 X11 許可證下發布不能做到一樣。所以正常我們不會做那兩種事情。我們只在 GPL 下發布。

我們只在 GPL 下發布的另一個原因是為了不允許在我們自由程序上出現會呈現實用優勢的私有擴展。不把自由看作價值的用戶可能會選擇那些非自由的版本，而不是它們所基於的自由的版本——從而失去他們的自由。我們不想鼓勵那種情況。

但會有偶然的情況，為了特定的戰略原因，我們判定在一個特定的程序用更寬鬆的許可證對自由的事業更為有利。在那些情形，我們在那個寬鬆許可證下給每個人發佈這個程序。

這是因為另一個 FSF 跟隨的倫理準則：無差別地對待所有用戶。為了自由的理想主義運動不應當歧視，所以 FSF 致力於給所有用戶同樣的許可證。FSF 從來不出售例外；無論我們以什麼許可證發佈一個程序，它都是所有人能夠獲得的。

但我們不需要堅持公司們遵循那條原則。對於公司我認為出售例外是可以接受的，並且我會在合適的時候建議這樣做，作為讓程序自由化的一種方式。

第 6 部分

陷阱和挑戰

6.1 您能夠信任您的計算機嗎？

Copyright © 2002, 2007, 2014, 2015 Richard Stallman 本文首先於
2002 年發表於 <http://gnu.org>。

您的計算機應該聽從誰的命令？大多數人認為他們自己的計算機應當聽從他們自己的命令而非他人的。但是，大型媒體公司（包括電影公司和唱片公司）正在聯合大型計算機公司，諸如英特爾和微軟，試圖通過一項它們稱之為“信任計算”的陰謀使得您的計算機聽命於它們而非您（這種陰謀的微軟版本稱為 Palladium）。以前的私有軟件本來就包含惡意功能，但現在這項陰謀將會使它們變為無惡不作的惡意軟件。

從本質上說，私有軟件意味著您不能控制它們的所作所為；您不能研究它們的源代碼也不能修改它們。精明的商業人士總會找到辦法以便利用他們的控制權將您置於不利地位，這並不令人感到絲毫驚訝。微軟曾經多次這樣做：某個版本的 Windows 被設計為能夠向微軟報告您的硬盤驅動器上安裝的所有軟件；最近的一項用於 Windows Media Player 的“安全”更新強制用戶接受新的限制。但是，微軟並不是唯一一家正在如此做的：例如

KaZaa 音樂共享軟件被設計為使得 KaZaa 的業務合作伙伴可以將您的計算機使用狀況出賣給它們的客戶。這些惡意的功能通常是隱祕的，但即使您能夠知道它們的存在，也難以移除它們，由於您不能訪問它們的源代碼。

在過去，這些行為只是孤立事件。然而，所謂的“信任計算”將會使得這種行為無處不在。“背叛計算”無疑是一個更為貼切的名字，由於這個陰謀被設計的初衷是確保您的計算機系統性地不再聽命於您。事實上，它被設計為使您的計算機不再成為通用目的計算機。任何一項操作都可能要求得到它們的明確授權。

背叛計算的基本技術思想是使計算機包含一個數字加密與簽名設備，而它所用的密鑰對您是絕密的。私有軟件將會利用這一設備來控制您可以運行哪些其他程序，您可以訪問哪些文檔或數據，以及您可以把它們傳給什麼程序。這些程序將會不斷從互聯網上下載更多的認證限制規則，並且自動將這些限制強加到您的工作中。如果您不想讓您的計算機定期從互聯網上獲取新的限制規則，一些功能將會自動停止工作。

當然，好萊塢和唱片公司計劃將背叛計算用於數字限制管理（DRM），這樣，您下載到的視頻和音樂將只能在某一臺特定的計算機上播放。分享將會完全成為不可能，至少是對於那些您從這些公司獲得的經過認證的文件。您，作為公眾的一員，理應擁有分享這些內容的自由和能力（我期望某些人能夠找到一種方式來製作未加密的版本並且上傳分享它們，這樣 DRM 將不能完全得逞，但是，這不能成為我們原諒它的理由）。

禁止分享已經足夠惡劣了，但是事情正在向著更壞的方向發展。已經有計劃將這種機制用於郵件和文檔——使得郵件將會在兩週之後消失，或者文檔只能被某家公司的計算機讀取。

假設您收到上司的郵件，對方要求您去做某些您認為過於激進冒險的事情。一個月之後，當事情產生相反的效果時，您不能利用那封消失了的郵件證明這一決策並非您所做出。此時的白紙黑字並不能證明您的清白，如果它是用會褪色的墨水寫的。

假設您收到上司的郵件，郵件中說明了一種非法的或者通常是令人無法容忍的政策，例如銷燬您的公司的審計檔案，或者放任某種對您的國家

的威脅發展坐大。今天，您還可以將其發送給記者並曝光這種行為。但是，一旦有了背叛計算，記者將不能讀取您的文檔；由於記者的計算機拒絕聽命於本人。背叛計算將會成為腐敗和犯罪的淵藪。

諸如微軟 Word 等文字處理器可能會使用背叛計算來保存您的文檔，以使得競爭對手的文字處理器不能讀取它們。如今，我們必須試圖通過耗時費力的試驗來破解 Word 格式的祕密，以使得自由的文字處理器能夠讀取 Word 文檔。但如果 Word 在保存文檔時使用背叛計算加密文檔內容，自由軟件社區將毫無機會開發出能夠讀取它們的軟件——即使我們能夠開發出來，這樣的軟件也可能甚至會被數字千年版權法案（DMCA）判定為非法。

使用背叛計算的軟件將會持續從互聯網下載新的認證限制規則，並且將其強加於您的工作。如果微軟或者美國政府不喜歡您在文檔中所寫的內容，它們可以發送新的指令通知所有計算機拒絕任何人讀取該文檔。每臺計算機在下載到新的指令時都會執行它。這樣，您所寫的內容將會遭受《一九八四》式的反動的抹除，甚至您自己都可能再也無法讀取它。

您可能認為您能夠通過某種方式發現一款背叛計算的應用程序會做出哪些齷齪的事情，瞭解那些行為是多麼地讓人痛苦，然後決定是否接受它們。但即使您能夠發現這些，對您來說接受這些協議也是愚蠢的，您甚至不能指望這些條款會維持現狀而不是變得更壞。您一旦依賴於使用這些軟件，您已經對此成癮並且它們知道這一點；然後它們可以更改條款使其對您更加不利。一些應用程序會自動下載更新，而這些更新會在暗中做一些其他事情——而它們不會給您是否接受更新的選擇權。

如今，您仍然可以通過拒絕使用私有軟件來避免受其限制。如果您運行 GNU/Linux 或者其他的自由操作系統，並且您避免在其上安裝私有軟件，那麼您仍然掌控著您的計算機的行為。如果一款自由軟件包含惡意功能，社區中的其他開發者將會將其移除，您將能夠使用修正後的版本。您也可以在不自由操作系統上運行自由軟件和工具；這並不能完全賦予您自由，但很多用戶確實如此做。

背叛計算將自由操作系統和自由軟件置於絕境，因為您甚至可能完全

不能運行它們。某些版本的背叛計算將會要求操作系統必須是由某家特定公司所具體認證的。這時自由操作系統將不被允許安裝。某些版本的背叛計算可能會要求每個應用程序都要經過操作系統開發廠商具體認證。這時，您不能在這樣的操作系統上運行自由軟件。如果您設法找到破解方法並且告知他人，這將被視為犯罪行為。

事實上，美國法律已經有提案要求所有計算機支持背叛計算，並且禁止老舊計算機接入互聯網。美國消費者寬帶和數字電視推廣法案（CBDTPA，我們稱之為 Consume But Don't Try Programming Act，即“消費但不要嘗試編程法案”）就是其中之一。但即使它們還沒有從法律層面完全強制您轉向背叛計算，這也會為您施加巨大壓力以迫使您最終接受。如今，人們通常在通訊中使用 Word 文檔格式，儘管這會導致諸多問題¹。如果只有支持背叛計算的機器才能讀取最新的 Word 文檔，那麼很多人將會轉向它，如果他們僅僅將這種情況看作個人行為（即“愛用不用”）。為了反對背叛計算²，我們必須聯合起來，並且作為一種集體的抉擇來勇敢面對這一困境。

阻止背叛計算將會需要大量的公民來組織參與。我們需要您的幫助！請支持 DefectiveByDesign.org，自由軟件基金會反對數字限制管理的運動。

6.1.1 補篇

1. 計算機安全領域以另一種方式使用短語“信任計算”——注意不要混淆二者的涵義。
2. GNU 計劃發佈了 GNU 隱私衛士（GPG），這是一款能夠實施公鑰加密和數字簽名的軟件，您可以使用它發送安全私密的郵件。您需要認清 GPG 和背叛計算的本質區別，並且知道為何前者是有益的而後

¹參見我的文章“[We Can Put an End to Word Attachments](http://gnu.org/philosophy/no-word-attachments.html),” 位於 <http://gnu.org/philosophy/no-word-attachments.html>，以獲知 Word 文檔所能導致的一系列問題的描述以及我們關於如何解決它們的一些建議。

²如需獲得更多信息，參見“[Trusted Computing](http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html)’ Frequently Asked Questions,” 一文，位於 <http://www.cl.cam.ac.uk/users/rja14/tcpa-faq.html>。

者是極度陰險的。

當某人使用 GPG 向您發送一份加密文檔，並且您使用 GPG 對其解密，將會得到一份未加密的文檔。您可以讀取、複製、回覆甚至重新對其加密並且安全地發送給他人。而背叛計算應用程序只是讓您可以讀取屏幕上的單詞，但並不允許您生成一份未加密的文檔副本用於其他用途。GPG，作為一款自由軟件，使安全特性對用戶可用，即“他們利用它”。而背叛計算被設計為向用戶施加限制，即“它利用他們”。

3. 背叛計算的支持者總是著重論述其好處。他們所說的往往是正確的，但並不重要。

同大多數硬件一樣，背叛計算硬件也可以被用於非惡意用途，但這些功能也可由不帶背叛計算的硬件以其他方式實現。基本的區別在於背叛計算對用戶所做的是這樣齷齪的事情：讓您的計算機以對您不利的方式運行。

他們說的是事實，而我說的也是事實。將二者放在一起考慮將會得出什麼結論？背叛計算是一套剝奪我們的自由的陰謀，它提供了一些小恩小惠以吸引我們的注意力，使我們忽視我們將會因此失去的更為重要的東西。

4. 微軟將 Palladium 作為一項安全措施推出，並宣稱它將提供反病毒保護，但這是確鑿的謊言。由微軟研究院於 2002 年十月提供的一份演示文稿指出 Palladium 的特性之一是現存的操作系統和應用程序仍然能夠運行；因此，病毒自然可以繼續做它們現在所能做的一切。

當微軟員工提及與 Palladium 相關聯的“安全”概念時，它並非指的是我們通常理解的安全：保護您的機器免受您不想要的東西的危害。相反，他們指的是保護您的計算機上的數據，並且阻止您以其他人不喜歡的方式訪問它們。演示文稿的一頁幻燈片列出了 Palladium 可用於保守的幾類機密，包括“第三方機密”和“用戶機密”——但它將

“用戶機密”置於引號之中，指出這在 Palladium 的語境中是一種荒謬的東西。

該演示文稿頻繁使用一些我們經常與安全語境相關聯的短語，諸如“攻擊”、“惡意代碼”、“冒名頂替”，當然還有“信任”。它們在這裡的涵義都不是其通常的涵義。例如，“攻擊”不是指某人試圖傷害您，而是指您試圖複製受保護的音樂；“惡意代碼”指您自行安裝的代碼，用於使您的機器去做那些其他人不希望您做的事情；“冒名頂替”不是指其他人愚弄您，而是指您愚弄 Palladium，等等。

5. 先前由 Palladium 開發者發出一份聲明提出了一種基本的前提條件，即信息的任何製造者和收集者對於您如何使用該信息應當擁有完全控制權。這意味著對已有的倫理和法律系統理念的徹底顛覆，並且將會創建一個史無前例的控制體系。這些系統的具體問題絕非偶然，因為它是基於上述基本目標的。這種基本目標是我們必須堅決反對的。
6. 直至 2015 年，背叛計算在個人計算機（PC）中以“信任平臺模塊”（TPM）的形式實現；然而由於實踐上的原因，TPM 被證實完全未能實現其目標：為數字限制管理（DRM）提供遠程認證平臺。因此，計算機公司以其他方式實現 DRM。現在，TPM 完全沒有被用於實現 DRM，並且有理由懷疑將它們用於 DRM 是根本不可行的。諷刺的是，這意味著只有當前的 TPM 應用是無害的非重要應用——例如，證實沒有人偷偷地更改了計算機中的系統。

因此，我們得出結論：PC 中的 TPM 並不危險，沒有理由不在 PC 中安裝一塊 TPM，或者不對其提供任何系統軟件支持。

這並不意味著所有事情都是美好的。其他禁止計算機的所有者改變其中的軟件的硬件系統正在某些 ARM PC 以及移動電話、汽車、電視機和其他設備的處理器上使用，這與我們所預期的一樣壞。

這也不意味著遠程認證是無害的。如果一旦有一款設備成功實施了它，它將對用戶的自由造成毀滅性的影響。當前的 TPM 無害，僅僅

是因為它暫時未能實現其使遠程認證成為可能的企圖。我們一定不要假設它在未來的所有企圖都將失敗。

6.2 JavaScript 陷阱

Copyright © 2009–2013 Richard Stallman

您可能每天都在您的計算機上通過您的瀏覽器運行非自由軟件而沒有意識到這一點。

在自由軟件社區，非自由軟件虐待它們的用戶這一觀念為人們所熟知。我們中的一些人完全拒絕安裝任何私有軟件，眾多其他人將沒有自由視為不採用這種軟件的一大理由。很多用戶注意到這一準則同樣適用於瀏覽器提供安裝的插件，因為它們可以是自由的，也可以是非自由的。

但是，瀏覽器會運行其他非自由軟件而不會詢問甚至不會告知您——那些包含在網頁中或者鏈接到網頁的程序。這些程序通常由 JavaScript 編寫，儘管其他編程語言也會被用到。

JavaScript（官方名稱為 ECMAScript，但是這個名字很少被使用）曾經僅僅用於網頁中的非主要修飾部分，例如精巧但無關緊要的導航或是顯示特性。將這些次要功能僅僅看作超文本標記語言（HTML）的擴展而非真正意義上的軟件是可以接受的；它們並不會構成一個主要問題。

很多網站仍然以這種方式使用 JavaScript，但有些網站將其用於主要的程序以執行大型任務。例如谷歌 Docs 將會向您的計算機下傳一個大約 0.5 MB 的 JavaScript 程序，這個程序用一種我們稱之為混淆腳本（Obfuscrypt）的緊密格式編寫，由於其中沒有註釋，也極少使用空白字符，而方法的名稱只有一個字母。程序的源代碼應當採用有利於修改它的形式；因此這種緊密的代碼不能稱之為源代碼，而此程序的真正源代碼仍然對其用戶不可用。

瀏覽器通常不會告知您它們何時加載 JavaScript 程序。大多數瀏覽器提供了一種完全關閉 JavaScript 的方式，但是沒有瀏覽器可以檢測出哪些 JavaScript 程序是非平凡、非自由的。即使您注意到了這個問題，要想識別並阻止這些程序可能會給您帶來相當可觀的麻煩。然而，即使是在自由軟件社區內，也有很多用戶並未覺察到這一問題；瀏覽器的沉默傾向掩蓋了

這個問題的存在。

可以將一段 JavaScript 程序作為自由軟件發佈，通過在自由軟件許可證下發布其源代碼。但是，即使程序的源代碼可獲得，並沒有一種簡易的方法來使用您的修改版本替代原始版本。當前的自由瀏覽器並沒有提供這樣一種機制來使用您自己修改過的版本替代來自網頁的版本。這種影響可以和“TiVo 化”（tivoization）相比，儘管並不是那麼非常難以克服。

JavaScript 並非網站向用戶推送的程序所使用的唯一編程語言。Flash 支持通過 JavaScript 的一種擴展變體進行編程。我們需要研究 Flash 的問題來做出合理的建議。Silverlight 看起來將會帶來和 Flash 相似的問題，唯一的區別在於它比 Flash 更壞，由於微軟將其用作非自由編碼解碼器的平臺。一款 Silverlight 的自由軟件替代品並不能為自由軟件世界解決問題，除非它也能正常地提供自由的編碼解碼器替代品。

Java 小程序也是在瀏覽器中運行的，它們也會帶來類似的問題。從一般意義上來說，任何類型的小程序系統都會帶來這類問題。然而，只要擁有一個自由的小程序執行平臺，就足以使我們能夠克服這個問題。

已經有一場聲勢浩大的運動號召網站僅僅使用自由的（有些人稱之為“開放的”）格式和協議進行通訊；這些格式和協議分別是指用於發佈文檔的格式和任何人均可自由實現的協議。由於程序在網頁中的存在，這樣的標準尺度是必需的，但並不足夠。JavaScript 作為一種格式，其本身是自由的，並且在網絡中使用 JavaScript 程序並非絕對地壞。然而，如同我們在上文所見，這也不一定就沒問題。當網站向用戶傳送某個程序時，如果程序是用一種有文檔說明的、不受專利限制的編程語言所編寫的，僅僅滿足這一點並不足夠。該程序本身也必須是自由的。“只有自由軟件才能被傳送到用戶”必須成為網絡良好行為準則的一部分。

靜默地加載並運程序是“網頁應用”所帶來的諸多問題之一。“網頁應用”這一短語是被設計為故意忽視交付給用戶的軟件與在服務器上運行的軟件之間的根本區別的。它可以指代在瀏覽器中運行的專門的客戶端程序；也可以指代專門的服務器軟件；還可以指代與專門的服務器軟件共同工作的專門的客戶端程序。客戶端和服務器端的不同將會引起不同的倫理

問題，即使它們整合得如此緊密，以至於它們可以被認為是同一個軟件的組成部分。本文僅僅討論客戶端軟件帶來的問題。我們將會在另一篇文章中討論服務器端的問題。

在實踐層面，我們怎樣才能解決由網絡中的私有 JavaScript 程序帶來的問題呢？第一步當然是避免運行它們。

那麼，我們之前提到的“非平凡”是指什麼呢？這指的是一種程度，因此與之相關的問題是設計出一種簡單的標準尺度使其能夠帶來好的結果，而非試圖尋求唯一正確的答案。

我們試著提出了一種策略，這種策略認為一段 JavaScript 程序如果是非平凡的，那麼它需要滿足：

- 它會提交異步 JavaScript 和可擴展標記語言（AJAX）請求，或者與提交 AJAX 請求的腳本同時被加載；
- 它會動態加載外部腳本，或者與動態加載外部腳本的腳本同時被加載；
- 它定義了這樣的函數或方法：這些函數或方法要麼從 HTML 加載外部腳本，要麼作為外部腳本被加載；
- 它使用了不對程序進行轉譯就難以對其進行分析的動態 JavaScript 結構，或者與使用了這樣的結構的腳本一同被加載。這些結構包括：
 - 使用 eval 函數；
 - 使用方括號標記調用方法；
 - 使用除了帶有某些方法的字符串字面量（例如 Obj.write、Obj.createElement）以外的任何其他結構。

那麼，我們又該如何判斷一段 JavaScript 代碼是否自由呢？我們在本文末尾提出了一種常用約定。根據這種約定，網頁中的一段非平凡 JavaScript 程序可以指出其源代碼所位於的統一資源定位符（URL），並且使用形象化的註釋指出其所使用的許可證類型。

最後，我們還需要改變自由的瀏覽器來檢測並阻止網頁中的非平凡非自由 JavaScript 代碼。一款名為 LibreJS 的自由軟件能夠檢測並阻止您所訪問的網頁中的非平凡非自由的 JavaScript 代碼¹。LibreJS 是一款適用於 IceCat 和 IceWeasel（還有 FireFox）瀏覽器的插件。

瀏覽器用戶還需要一款簡便的工具以指定用戶想要使用的 JavaScript 代碼，而非直接使用某一特定網頁提供的 JavaScript 代碼。（用戶所指定的代碼可以是基於網頁自帶的自由 JavaScript 程序的完全替代品或者修改版本。）Greasemonkey 幾乎可以實現這一點，但仍稍有欠缺，由於它不能完全保證自己能夠在網頁中的 JavaScript 程序開始被執行之前完成對其代碼的修改。使用本地代理能夠解決這個問題，但其便捷程度還遠未達到能夠使其成為一種真正的解決方案的要求。我們需要構建一種兼具可靠性與便捷性的解決方案，以及能夠用於分享這些更改的網站。GNU 計劃只會推薦那些專注於自由更改的網站。

這些特性將會使得網頁中的 JavaScript 程序在事實上有可能成為自由的。JavaScript 將不再是實現我們的自由的一種特別的障礙——現在只有 C 語言和 Java 是這樣的。我們將能夠拒絕甚至替換網頁中的非平凡非自由 JavaScript 程序，正如我們能夠拒絕或替換通過常規方式提供安裝的私有軟件那樣。我們要求網站將其 JavaScript 代碼自由化的運動就可以更好地展開。

與此同時，確實有一種情況是允許運行非自由 JavaScript 程序的：向網站維護者發送投訴消息，告訴他們應該移除其網站中的 JavaScript 代碼或使其成為自由的。不要猶豫，馬上暫時允許 JavaScript 以便進行投訴——然後記得重新關閉 JavaScript。

6.2.1 附件 A：用於發佈自由 JavaScript 程序的一種常用約定

如需引用相關程序的源代碼，我們建議採用這種格式：

¹LibreJS 工程 (<http://gnu.org/software/librejs/>) 需要 JavaScript 程序員（的幫助）。如果您掌握了必備的技能，請幫助我們維護這個價值連城的瀏覽器擴展。

```
// @source:
```

後面跟著 URL。這種風格滿足 GNU 通用公共許可證 (GNU GPL) 用於發佈源代碼的要求。如果源代碼位於另一站點，您必須妥善處理這種情形。要想使軟件成為自由的，源代碼是必須提供的。

如需指出嵌入本頁面的 JavaScript 代碼所使用的許可證類型，我們建議將許可證文本置於如下所示的兩段文字之間：

```
The following is the entire license notice for  
the JavaScript code in this page.
```

```
...
```

```
The above is the entire license notice for the  
JavaScript code in this page.
```

當然，所有這些內容應當位於多行註釋之間。

同許多其他自由軟件許可證一樣，GNU GPL 要求連同程序的源代碼和二進制形式一起發佈一份許可證的副本。然而，由於 GNU GPL 過於冗長，將其同 JavaScript 程序一起包含在頁面中可能會帶來某種不便。對於您所擁有著作權的代碼，您可以移除這條要求，代之以類似下面這樣的許可證聲明：

```
Copyright (C) YYYY Developer
```

```
The JavaScript code in this page is free software:  
you can redistribute it and/or modify it under the  
terms of the GNU General Public License (GNU GPL)  
as published by the Free Software Foundation, either  
version 3 of the License, or (at your option) any
```

later version. The code is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU GPL for more details.

As additional permission under GNU GPL version 3 section 7, you may distribute non-source (e.g., minimized or compacted) forms of that code without the copy of the GNU GPL normally required by section 4, provided you include this license notice and a URL through which recipients can access the Corresponding Source.

感謝 Jaffar Rumith 提醒我關注這一問題。

6.2.2 附件 B：以網絡管理員身份發佈自由的 JavaScript 程序

如果您是一位網絡管理員並且在您的網站上部署自由的 JavaScript 軟件，明晰並且一致地發佈這些程序文件的許可證信息和源代碼有助於您的網絡訪問者確定他們正在運行自由軟件，並且有助於您指導您自己遵守這些許可條款。

發佈這些許可證信息的一種方式如上文中的附件 A。另一種方法，稱為 JavaScript 許可證網絡標籤，可能對於精簡的 JavaScript 代碼更為便捷，特別是對於那些並非您編寫的代碼。

6.3 如果您在大學工作，請發佈自由軟件

Copyright © 2002, 2014 理查德·斯托曼 (Richard Stallman)。本文最初於 2002 年發表於 <http://gnu.org>。

在自由軟件運動中，我們堅信用戶應當擁有對他們所使用的軟件進行修改和再分發的自由。自由軟件中的“free”是指“freedom”中的自由：即用戶擁有運行、修改和再分發的自由。自由軟件能夠為人類知識做貢獻，而私有軟件則不能。因此，大學應當以推進人類知識發展之名鼓勵自由軟件，如同它們應當鼓勵科學家和其他學者發表其工作成果。

嗚呼，很多大學的管理人員對軟件（以及對科學）持有一種貪婪的態度；他們將軟件視為贏利的機會而非為人類知識做貢獻的機會。自由軟件開發者同這種傾向進行鬥爭已有將近 20 年了。

在 1984 年，當我開始著手開發 GNU 操作系統時，我的第一步是先辭去我在麻省理工學院 (MIT) 的工作。我特意選擇這麼做，這樣 MIT 許可證機構就不能再對我將 GNU 作為自由軟件發佈這一行為進行干涉。我計劃了這樣一種方式對 GNU 中的程序進行授權許可，這種方式可以保證它的所有修改版本必須仍是自由的——這種授權許可方式後來發展成為了 GNU 通用公共許可證 (GNU GPL) ——而且我不希望乞求 MIT 管理人員的允許才能使用它。

這些年來，一些大學的分支機構經常前來自由軟件基金會 (FSF) 尋求幫助，關於如何應對那些僅僅將軟件視為用於販賣之物的管理人員。有一種方法甚至對於那些受到特別資助的項目也是適用的，就是使您的工作基於一款已經在 GNU GPL 下發布的現有自由軟件。然後您可以告知管理人員：“我們不能發佈此修改版本，除非使用 GNU GPL 許可證——任何其他方式都將侵犯版權。”當他們眼中的美元符號逐漸褪色時，他們通常將會不再反對將其作為自由軟件發佈。

您也可以求助於資金贊助商。當紐約大學 (NYU) 的一個開發小組在美國空軍資助下開發出 GNU Ada 編譯器時，合同中明確提出將其代碼捐贈自由軟件基金會。首先同贊助商達成某種默契，然後禮貌地對大學管理

人員說明這不允許重新談判。他們將會想要簽訂一紙合同來開發自由軟件，而不想失去它，於是他們很可能會同意。

不論您採取哪種方式，一定要儘早提出關於自由軟件的問題——例如程序開發進程達到一半以前。此時，大學仍然需要您的工作，因此您可以採取強硬措施：告知管理人員您將會完成該程序並且使之可用，僅當他們白紙黑字地同意使其成為自由軟件（並且同意由您選擇使用何種自由軟件許可證）。否則您為這個項目所能繼續做的只是寫一篇關於它的論文，並且不會做出一個適合於發佈的良好版本。當管理人員意識到他們只能在得到一個可以為大學爭得榮譽的自由軟件與什麼也得不到之間二選一，他們通常會選擇前者。

FSF 可能能夠說服您的大學接受 GNU GPL，或者 GPL 第三版。如果您難以獨立完成這一任務，請給我們幫忙的機會。請發送郵件至 licensing@fsf.org，並且在主題中加上“緊急”（“urgent”）標記。

並非所有大學都實行貪婪政策。德州大學有一項寬鬆的政策使得在那裡開發的軟件在 GNU GPL 下作為自由軟件發佈變得容易。巴西 Univates 大學和印度海德拉巴信息技術國際學院都有有利於以 GPL 發佈軟件的政策。在首先得到了開發團隊的支持後，您也可以在大學實行類似的政策。將這個問題作為一種原則性的問題提出：大學是否應該以推進人類知識發展為己任，還是僅僅滿足於使自己長期存在？

在試圖說服大學的過程中，最好能夠從倫理的視角出發並且下定決心，如同我們在自由軟件運動中所做的那樣。為了能夠以符合倫理的方式對待大眾，軟件應當是自由的——如同“freedom”中的自由——對於全體大眾。

很多自由軟件開發者狹隘地宣稱如此做的一些實踐上的原因：他們倡導允許他人分享和修改軟件只是一種使得軟件變得更加強大且可靠的權宜之計。如果是這樣的價值觀驅使您開發自由軟件，也罷，感謝您和您的貢獻。但是，這樣的價值觀並不能給予您堅定的立場，一旦大學管理人員威逼利誘您將您的軟件變成私有的。

例如，他們可能會爭論“我們能夠讓它變得更加強大且可靠，只要我們能夠得到錢”。這樣的宣言最後也許能夠實現，也許不能，但它很難在事

先被證偽。他們可能會提出這樣一種分發副本的許可證：“免費，僅限學術應用”，這將會向普通大眾暗示他們不應獲得自由，他們還會論證這種行為將會得到學術界的支持，而這一切是（他們宣稱）您所需要的。

如果您的出發點只是易用性的價值，這將很難說服人們拒絕這些沒有出路的提議。而如果您的立場是基於倫理和政治方面的價值，就很容易做到。如果犧牲了用戶的自由，一款軟件做得再強大再可靠又有何益？難道自由不是應該普惠大眾而不僅限於學術界內嗎？如果自由和社區是您的目標，這些問題的答案是很明瞭的。自由軟件尊重用戶的自由，而私有軟件否定用戶的自由。

凡是能夠使您堅定信心的，莫強如瞭解什麼才是社區的自由所依賴的。舉個例子，比如您。

6.4 GNU/Linux 上帶有數字限制管理 (DRM) 的非自由遊戲：是老是壞？

Copyright © 2013 自由軟件基金會

Valve，一家知名的遊戲公司，主要發佈各種帶有數字限制管理 (DRM) 的非自由計算機遊戲，最近它宣稱將會為 GNU/Linux 平臺發佈遊戲。這會帶來哪些好的或者壞的影響？

我假設流行的非自由軟件在 GNU/Linux 上的可獲得性，將會推動 GNU/Linux 操作系統的普及。然而，GNU 的終極目標並非僅限於獲得“成功”；它的目的是賦予用戶自由¹。因此，更大的問題是，這種發展趨勢將會怎樣影響用戶的自由。

這些遊戲的問題並不在於它們是商業的²（我們不認為商業化有什麼問題）。也不在於其開發者販賣其副本³，這也沒有問題。它們的真正問題在於這些遊戲包含了非自由軟件（當然，指的是“freedom”中的自由）⁴。

（同其他非自由軟件一樣）非自由遊戲軟件是不符合倫理的，由於它們拒絕用戶的自由（遊戲藝術是另一個問題，由於它們不是軟件）。如果您渴求自由，一個起碼條件是不要擁有或者在自己的計算機上運行非自由軟件。這一點是很清楚的。

然而，如果您一定要玩這些遊戲，您最好還是在 GNU/Linux 上進行而非在微軟 Windows 上進行。至少您可以藉此避開 Windows 對您自由的踐踏⁵。

因此，從直接的實踐角度看，這種發展趨勢既有好的影響又有壞的影響。這可能會鼓勵 GNU/Linux 用戶去安裝這些遊戲，也可能會鼓勵這些遊

¹參見《如今自由軟件更加重要》(p. 32) 一文以獲得更多信息。

²參見《應避免使用（或慎用）的詞語》一文的“商業化”一節 (p. 110) 以獲得關於“商業化”一詞所帶來的困惑的解釋。

³參見《售賣自由軟件》(p. 49) 一文以獲得更多信息。

⁴參見《什麼是自由軟件？》(p. 1) 以獲知自由軟件的完整定義。

⁵參見我們的運動位於 <http://upgradefromwindows8.org/> 以獲得更多信息。

戲的玩家使用 GNU/Linux 來取代 Windows。我猜想直接的好處大於直接的壞處。但是，這裡還有一種間接的影響：這些遊戲的玩家將會向我們社區裡的人傳授什麼？

任何自帶軟件以提供這些遊戲的 GNU/Linux 發行版將會暗示用戶：自由並不是關鍵。GNU/Linux 發行版中的非自由軟件¹已經在破壞自由這一終極目標。向發行版中添加這類遊戲將會加深這種危害。

自由軟件的實質是自由而非價格。一款自由的遊戲不必是免費的。商業化開發自由的遊戲軟件是可行的，同時尊重您修改所使用軟件的自由。由於遊戲中的藝術不屬於軟件，它也不必是免費的。事實上，確實有由商業公司開發自由的遊戲軟件，與那些由志願者以非商業化方式開發的自由的遊戲軟件並存。眾籌式的開發只會讓開發變得更容易。

但是，如果我們假設在當前情況下開發某類自由遊戲軟件是不可行的——將會發生什麼？將其作為一款非自由遊戲軟件開發並無任何益處。想要在您的計算中擁有自由，必須拒絕非自由軟件，就是這麼簡單直白。您，作為熱愛自由者，不會由於一款非自由遊戲的存在而去玩它，因此，您也不會由於它的不存在而失去任何東西。

如果您想要推進計算中的自由事業，請注意不要將這類遊戲在 GNU/Linux 上的可獲得性說成是對我們事業的支持。與之相反，您可以向人們介紹 LibreGameWiki²，那裡試圖列出所有自由的遊戲軟件，以及 FreeGameDev 論壇³，還有 LibrePlanet Gaming Collective 的自由遊戲之夜⁴。

¹參見 <http://gnu.org/distros/common-distros.html> 以獲知為何我們不推薦某些（通常是流行的）發行版。

²參見 https://libregamewiki.org/Main_Page.

³參見 <http://forum.freegamedev.net/index.php>.

⁴參見 http://libreplanet.org/wiki/Group:LibrePlanet_Gaming_Collective.

6.5 電子書的威脅

Copyright © 2011, 2014 Richard Stallman

在這樣一個由企業支配我們的政府並且制定法律的時代，任何技術上的進展都會為企業提供對公眾施加新的限制條件的機會。本應賦予我們權利的技術現在被用於束縛我們。

對於紙版書：

- 您可以用現金匿名購買；
- 然後您擁有它；
- 您不需要簽訂限制您使用它的霸王條款；
- 它的格式是已知的，不需要藉助私有技術就能閱讀；
- 您可以贈送、借閱或轉賣給其他人；
- 您可以物理地掃描或者複製整本書，有時這也是符合版權法的；
- 任何其他人無權銷燬您的書。

與之相反，對於亞馬遜電子書（比較普遍）：

- 亞馬遜要求用戶提供身份信息才能購書；
- 在某些國家，包括美國，亞馬遜宣稱用戶不能擁有其發行的電子書；
- 亞馬遜要求用戶接受限制其對電子書使用的霸王條款；
- 電子書的格式是私密的，必須藉助限制用戶自由的私有軟件才能閱讀；
- 在一段限定時間內，代用詞“借閱”被允許用於某些電子書，但僅限使用相同系統的實名用戶。禁止贈與或轉賣。

- 在閱讀器中複製電子書是不可能的，由於數字限制管理（DRM¹），這也被霸王條款所禁止，它比版權法更加苛刻。
- 亞馬遜可以利用後門遠程刪除電子書，例如它曾於 2009 年利用此後門遠程刪除了數千份喬治·奧威爾（George Orwell）所著的《一九八四》。

這些侵害行為只需其一便可使得電子書相對於紙版書成為一種倒退。我們必須拒絕電子書，直到他們尊重我們的自由。

電子書公司宣稱它們拒絕我們的傳統閱讀自由是必需的，唯有如此它們才能繼續向作者支付稿酬。現行的版權系統對這些公司給予了慷慨的支持，但卻未能給予大多數作者起碼應有的支持。我們可以用其他方式對作者提供更好的支持，而無需犧牲自己的自由，甚至使分享合法化。我在此建議兩種方式：

- 根據每位作者受歡迎程度的立方根²將稅金分發給作者；
- 開發出這樣的閱讀器，使得用戶可以向作者匿名捐款。

電子書不應該侵害我們的自由（古騰堡計劃的電子書不會這樣），但是，一旦商業公司決定如此做，它們就會侵害我們的自由。我們有責任阻止這種行為。

加入這場鬥爭，在這裡簽名：<http://DefectiveByDesign.org/ebooks.html>。

¹參見《閱讀的權利》（p. 129）一文以獲得更多信息。

²參見《計算機網絡時代的版權與社區之爭》（p. 154）一文和我在 2012 年致巴西參議院主席 Senator José Sarney 的公開信，位於 <https://stallman.org/articles/internet-sharing-license.en.html>，以獲得更多信息。

6.6 電子書必須增進我們的自由而非限制我們的自由

Copyright © 2012 理查德·斯托曼 (Richard Stallman)。本文最初於 2012 年 4 月 17 日以標題 “Technology Should Help Us Share, Not Constrain Us”，發表於英國衛報 <http://guardian.co.uk>，並帶有一些特別的修訂。此版本整合了部分修訂，同時恢復了部分原文。

我喜歡 The Jehovah Contract 這本書，也希望每個人都喜歡它。這些年來，我已經至少借出它六次了。紙版書允許我們這麼做。

我不能對大部分商業電子書如此，因為這是“不被允許的”。如果我們試圖違反，閱讀器中的軟件具有一種稱為數字限制管理 (DRM) 的惡意功能，它將會限制用戶閱讀，因此不能分享。由於電子書是加密的，只有具有惡意功能的私有軟件才能顯示它們。

我們作為讀者所習慣的很多其他閱讀方式對於電子書是“不被允許的”。對於亞馬遜 Kindle（更合適它的名字是 Swindle¹），作為一個例子，用戶不能使用現金匿名購買電子書。Kindle 的電子書通常僅可以從亞馬遜獲得，而亞馬遜要求用戶提供身份信息。由此亞馬遜知道每位用戶讀了哪些書。在某些國家，例如英國，您可能會由於擁有一本禁書而被起訴²，這比假想中的奧威爾主義 (Orwellian)³ 更加殘酷。

參閱《電子書的威脅》(p. 297) 一文，並且考慮加入我們關於電子書威脅的郵件列表，位於 <http://defectivebydesign.org/ebooks.html>。

¹參見《為何稱之為詐騙 (Swindle)?》(p. 104) 一文以獲得更多信息。

²Ben Quinn, “Man in London Charged with Terrorism Offences over Al-Qaida Document,” 4 April 2012, <http://www.theguardian.com/world/2012/apr/04/al-qaida-terrorism>.

³指現代專制政權藉由嚴厲執行政治宣傳、監視、故意提供虛假資料、否認事實（雙重思想）和操縱過去（包括製造“非人”，意指把一個人過去的存在從公共記錄和記憶中消除）的政策以控制社會，就如《一九八四》（作者喬治·奧威爾）中的世界觀一樣。——譯者注，摘自維基百科

更壞的是，您不能在讀完一部電子書之後將其轉賣（如果亞馬遜按其意志行事，那些我可以在一個下午光顧好幾家的舊書店將成為歷史）。您也不能將其贈與他人，由於根據亞馬遜的霸王條款，您不能真正擁有它。亞馬遜要求用戶簽訂的最終用戶許可協議（EULA）如是說。

您甚至不能保證某部電子書明天還存在於您的機器中。人們在使用 Kindle 閱讀《一九八四》的時候親歷了奧威爾主義：他們的電子書在眼皮底下消失了，由於亞馬遜使用了一種帶有後門的惡意軟件將其遠程刪除（虛擬世界的焚書行為，這就是 Kindle 的涵義嗎？）。但是請不必擔心，亞馬遜承諾不再如此，除非有國家命令。

對於軟件，要麼用戶控制程序（稱這類軟件為自由的¹），要麼程序控制用戶（非自由）。亞馬遜的電子書政策模仿私有軟件的發佈政策，但這並非二者之間的唯一關聯。上述惡意軟件特性²是通過私有軟件強加給用戶的。如果一款自由軟件擁有類似的惡意特性，一些具有編程技能的用戶可以移除它們，然後將修正後的版本提供給其他用戶。用戶不能更改私有軟件，這使得它們成為其開發者對公眾行使不公權力的理想工具³。

以上這些對我們自由的侵害任何一種都足以作為我們說“不”的理由。如果這些政策僅限於亞馬遜，我們可以跳過它們，但是其他電子書發行商的政策大致相似。

最讓我們擔心的是未來可能失去紙版書的選項。英國衛報宣稱“僅限數字閱讀”，換言之，書籍只能靠犧牲自由去換取。我不會以此為代價閱讀任何書籍。從現在起的五年以後，對於大部分書籍，未經許可的副本會不會成為僅存的倫理上可接受的版本呢？

事情不必須朝著那個方向發展。通過互聯網匿名支付，購買並下載無 DRM、無 EULA 的電子書將會尊重我們的自由。實體書店可以接受現金購

¹參見《什麼是自由軟件？》(p. 1) 一文以獲知自由軟件的完整定義。

²參見 <http://gnu.org/proprietary/proprietary.html> 以獲知一系列日益增長的威脅。

³參見我的文章《如今自由軟件更加重要》(p. 32) 和 “The Problem Is Software Controlled by Its Developer,” 位於 <http://gnu.org/philosophy/the-root-of-this-problem.html> 以獲得更多信息。

買這樣的電子書，如同光盤（CD）上的數字音樂仍然可能買到，即使是現在這個音樂產業極具侵略性地推廣帶有 DRM 限制服務（諸如 Spotify）的時代。實體 CD 商店不得不面對昂貴的庫存成本負擔，而實體電子書商店可以將電子書副本複製到您的 USB 存儲器上，這是唯一可以賣給您的載體，如果您需要。

出版商關於它們限制電子書使用自由所給出的解釋是為了阻止用戶分享電子書副本。它們宣稱這是為了作者著想；但即使它們真的服務於作者的利益（對於非常著名的作者，也許是這樣），這並不能成為剝奪讀者自由的 DRM、EULA 和數字經濟法案的辯護理由。事實上，版權系統對於支持作者，除了那些最著名的作者以外，做出了極壞的事情。其他作者的基本利益應當被更好地為人們所瞭解，因此分享他們的作品對於他們和讀者同樣有益。為何不轉向一種能夠更好地解決這個問題而又與分享相容的體系呢？

對存儲器和互聯網連接徵稅，與大多數歐盟國家正在實施的一般準則放在一起，可以更好地完成這項任務，如果以下三點得到滿足。這筆資金應該由國家收集並依法發放，而非給予私人的集資組織；它應當分給所有作者，我們不能允許商業公司侵佔其中的任何一分錢；並且這筆錢的分配應當基於一種變化的尺度而非按照流行度的線性比例。我建議使用作者流行度的立方根作為依據：如果 A 的流行度是 B 的八倍，則 A 得到 B 的兩倍的錢（而不是八倍）。這將給予眾多普通作家足夠的支持，而非僅僅使得少數明星大腕賺得盆滿鉢滿。

另一種體系是在閱讀器上設置一個按鈕以便向作者匿名支付一小筆錢（例如在英國，25 便士可能是合適的）。

分享是美好的，並且有了數字技術，分享應該變得更容易。（我指對原始版本進行非商業的再分發）因此，分享理應合法，而阻止分享並不能成為將電子書變為讀者數字手銬的理由。如果電子書意味著必須在增進與限制讀者自由中二選一，我們必須要求它們增進讀者的自由。

6.7 服務器真正是在為誰服務？

Copyright © 2010, 2013, 2015 理查德·斯托曼 (Richard Stallman)。
本文最初於 2010 年三月 8 日以標題 “What Does That Server Really Serve?” 發表於 Boston Review 在線版。

在互聯網環境中，私有軟件並非剝奪您的自由的唯一方式。“作為軟件替代品的服務” (**Service as a Software Substitute**)，或者 SaaS，是另一種賦予其他人支配您的計算的權力的途徑。

SaaS 指的是使用某種由其他人實現的服務來替代您所運行的軟件副本。這是我們創造的短語，文章和廣告並不會使用它，它們也不會告知您某種服務是否為 SaaS。與之相反，它們很可能蔽之以一種含混不清並且旨在轉移您的注意力的短語：雲。這將 SaaS 與其他實現，不論是惡意的還是有益的，混為一談。不過，看了本文的解釋和示例之後，您將能夠區分某種服務是否為 SaaS。

6.7.1 背景知識：私有軟件是如何剝奪您的自由的

數字技術可以賦予您自由，也可以剝奪您的自由。對於我們掌控自己計算的首要威脅來自私有軟件：私有軟件是用戶所不能控制的，由於它受其擁有者（例如蘋果或微軟這樣的商業公司）控制。其擁有者通常利用這種不公的權力向其中植入惡意功能，諸如間諜軟件、後門、數字限制管理 (DRM，它們將其鼓吹為數字版權管理)¹。

如需獲得關於此問題的更多信息，參見 “The Bug Nobody Is Allowed to Understand,” 一文，它位於 <http://gnu.org/philosophy/bug-nobody-allowed-to-understand.html>。

我們針對這個問題的解決方案是開發自由軟件並且拒絕私有軟件。自由軟件意味著您作為用戶擁有四項基本自由：(0) 以任何目的運行該軟件

¹請加入我們的運動一起反對 DRM，它位於 DefectiveByDesign.org。

的自由；(1) 研究和修改源代碼使其以您所期望的方式工作的自由；(2) 再發布相同副本的自由；(3) 再發布您的修改版本的自由（參見《什麼是自由軟件？》(p. 1) 一文中有關自由軟件定義）。

只要有了自由軟件，我們作為用戶就能奪回我們計算的控制權。私有軟件仍然存在，但我們可以將其排除在生活之外，並且我們中的很多人已經在這麼做。然而現在，我們對自己的計算的控制權面臨新的威脅：作為軟件替代品的服務（SaaS）。為了捍衛我們的自由，我們必須像拒絕私有軟件那樣拒絕 SaaS。

6.7.2 作為軟件替代品的服務是如何剝奪您的自由的

作為軟件替代品的服務（SaaS）指的是使用某種服務作為運行您的軟件副本的替代品。具體地說，它指的是由某人架設一臺網絡服務器以實現某種計算任務——例如，修改一幅照片，或者將文本翻譯為另一種語言等——然後邀請用戶通過那臺服務器進行計算。服務器的用戶會將其本人的數據上傳至服務器，後者對用戶所上傳的數據進行用戶自己的計算，再將計算結果返回用戶，或者直接以用戶的名義做什麼事情。

假設滿足這樣的條件，這種計算可以稱之為用戶自己的，即用戶原則上能夠在其本人的計算機上通過運行某軟件可以達到相同的目的（不論當時能夠達到此目的的軟件對於用戶是否可獲得）。當這種假設不成立時，這就不是 SaaS。

這些服務器比私有軟件更加殘酷地剝奪用戶對其計算的控制權。對於私有軟件，用戶一般會得到一份可執行文件而非源代碼。這使得用戶難以研究其所運行的代碼，因而難以判定此程序真正在做什麼，並且難以對其進行修改。

對於 SaaS，用戶連用於計算的可執行文件都得不到：它位於某些其他人的服務器上，用戶不能企及。因此用戶完全不可能確認它們到底在做什麼，也完全不可能對其進行修改。

進一步講，SaaS 將會自然而然地引起與某些私有軟件中的惡意功能相當的結果。

例如，一些私有軟件是“間諜軟件”：它們將會發送與用戶的計算活動相關的數據¹。微軟 Windows 向微軟發送用戶活動相關信息。微軟 Windows 媒體播放器（Windows Media Player）向微軟彙報每位用戶播放過的內容。亞馬遜 Kindle 彙報用戶於何時閱讀哪本書的第幾頁。憤怒的小鳥（Angry Birds）將會彙報用戶地理位置的歷史記錄。

與私有軟件不同的是，SaaS 並不依賴隱祕的代碼以獲取用戶數據。而是用戶被強制要求將其數據上傳到服務器以便使用 SaaS。這與間諜軟件具有相同的效果：服務器運營商可以不費吹灰之力得到用戶數據——正因為 SaaS 的天性。Amy Webb 曾經刻意避免發佈她女兒的任何照片，但她犯了使用 SaaS（Instagram）的錯誤，用它來編輯自己的照片。這些照片最終通過這條途徑洩露²。

一些私有操作系統擁有通用的後門，允許某人遠程進行軟件更改。例如，微軟可以通過 Windows 的通用後門強制更改用戶計算機上的軟件。此類後門也存在於幾乎所有的移動電話中。一些私有應用程序也擁有通用後門，例如 Steam 的 GNU/Linux 客戶端允許其開發者遠程安裝修改版本。

對於 SaaS，服務器的運營商可以更改服務器上所用的軟件。他應該可以這麼做，因為那是他自己的計算機；然而，使用這樣的 SaaS 與使用帶有通用後門的私有應用程序，在結果上並無區別：某人擁有偷偷改變如何完成用戶計算的權利。

因此，使用 SaaS 相當於使用帶有間諜軟件和通用後門的私有應用程序，這賦予了其服務器運營商凌駕於用戶之上的不公權力，這種權力是我們必須堅決反對的。

¹如需獲知監控正在通過哪些日益增加的方式在業界蔓延，參見 <http://gnu.org/philosophy/proprietary/proprietary-surveillance.html>。

²Amy Webb, “Congratulations, You Found a Photo of My Daughter Online,” 12 September 2013, http://slate.com/articles/technology/data_mine_1/2013/09/privacy_facebook_kids_don_t_post_photos_of_your_kids_on_social_media.html.

6.7.3 SaaS 和 SaaS

最初，我們將這種有問題的實踐稱為“SaaS”，意為 Software as a Service（軟件即服務）。這個短語通常用於指代在服務器上部署軟件而非向其用戶提供副本。我們當時認為，在這些問題發生時，這一短語對實際情況做出了精確描述。

然後，我們開始意識到，短語 SaaS 有時用於通訊服務——而這一名稱並不適用於這些活動。此外，短語“軟件即服務”並不能解釋為何這是一種壞的實踐，於是我們打造了這一短語“作為軟件替代品的服務”，這一稱謂對這種壞的實踐給出了更為清晰的定義，並且解釋了它為什麼是壞的。

6.7.4 理清 SaaS 帶來的問題和私有軟件帶來的問題

SaaS 和私有軟件將會帶來相似的後果，但二者的作用機制並不相同。對於私有軟件，其機制是您擁有一份副本，但對其進行修改是困難或者非法的。而對於 SaaS，其機制是您不能擁有一份您所用於計算的軟件副本。

這兩類問題通常被混淆，而混淆不僅僅是一種偶然。網絡開發者使用含混不清的短語“網頁應用”將服務器軟件和您在計算機瀏覽器中運行的程序混為一談。一些網頁會在您的瀏覽器中安裝非平凡甚至是大型的 JavaScript 程序而不會告知您。如果這些 JavaScript 程序是私有的¹，它們將會造成與任何其他私有軟件類似的不公。然而，我們在此將會主要討論使用服務本身帶來的問題。

很多自由軟件支持者設想：SaaS 帶來的問題將可以通過開發自由的服務器軟件來解決。對於服務器運營商而言，其服務器上的軟件最好是自由的；否則如果它們是私有的，其擁有者便可擁有控制服務器的權力。這對於服務器運營商是不公的，並且根本不會使用戶受益。但是，即使服務器上運行的軟件是自由的，這也不能保護服務器的用戶免受 SaaS 的影響。這些軟件賦予了服務器運營商自由，但卻不能賦予服務器用戶自由。

公開發布服務器軟件的源代碼確實會使社區受益：這使得具有適當技

¹參見《JavaScript 陷阱》(p. 286)一文以獲得更多信息。

能的用戶能夠架設類似的服務器，也許還能更改軟件本身。我們建議對通常在服務器上運行的軟件使用 GNU Affero 通用公共許可證（GNU Affero GPL）¹。

但是，這些服務器都不能使您能夠控制由它進行的計算，除非那是您自己的服務器。也許，對於某些工作，您可以信任您的朋友的服務器，如同您可能會請求朋友維護您計算機上的軟件。對於除此之外的任何情況，這些服務器對您來說都是 SaaS。SaaS 總是迫使您屈從於服務器運營商的權力。唯一的解決方式是：堅決不要使用 SaaS！不要使用任何其他人的服務器來進行您自己的計算以處理由您自己提供的數據。

這個問題解釋了“開放”和“自由”之間有著多大程度的區別。“開源”概念中的源代碼幾乎都是自由的²。但是“開源軟件”服務³的理念並不能解決 SaaS 的問題，由於它只意味著這種服務所依賴的服務器軟件是開源的，並且/或者是自由的。

服務與軟件有著本質區別，因而由服務引起的倫理問題與那些由軟件引起的問題有著本質區別。為了避免這種混淆，我們避免將一項服務描述為“自由”或“私有”的⁴。

6.7.5 區分 SaaS 和其他網絡服務

哪些在線服務是 SaaS？最清楚的例子是翻譯服務，例如將英文文本翻譯為西班牙文。為您翻譯一段文本是一種完全屬於您自己的計算。您可以在您自己的計算機上運行相應軟件進行翻譯（為了符合倫理，這款軟件應該是自由的）。翻譯服務替代了那款程序，因此它是一種作為軟件替代

¹參見《如何為你的作品選擇一份許可證》(p. 211) 一文以獲得我們關於許可證選擇的建議。

²參見 “How Free Software and Open Source Relate as Categories of Programs,” <http://gnu.org/philosophy/free-open-overlap.html> 一文以獲得更多信息。

³如需獲知 “Open Software Service Definition,” 參見 <http://opendefinition.org/ossd/index.html> 一文。

⁴如需獲得更多信息，參見我的文章 “Network Services Aren’t Free or Nonfree; They Raise Other Issues,” 位於 <http://gnu.org/philosophy/network-services-arent-free-or-nonfree.html>。

品的服務，即 SaaS。由於它拒絕您對您的計算的控制權，它對您作了惡。

另一個清楚的例子是使用諸如 Flickr 或者 Instagram 來修改照片。人們在自己的計算機上修改照片已有幾十年時間了；在服務器而非您自己的計算機上進行這些工作是一種 SaaS。

拒絕 SaaS 並不意味著必須拒絕使用任何由他人而非您所運行的網絡服務器。大部分服務器並非 SaaS，由於它們所進行計算並非用戶自己的計算。

設立網絡服務器的初衷並非為您做計算，而是發佈信息以供您訪問。即使在今天，這也是大部分網站正在做的，這並不會帶來 SaaS 的問題，由於訪問由某人發佈的信息並非是在做您自己的計算。通過博客網站或者諸如 Twitter 或 StatusNet 等微博服務發佈您的個人材料也都不是（當然，這些服務可能有其他問題）。其他並非有意成為私密的通訊，例如群聊，也是如此。

從本質上說，社交網絡是一種形式的通訊和信息發佈，並非 SaaS。然而，一項以社交網絡為主要功能的服務可能擁有屬於 SaaS 的特性或擴展。

如果一項服務不屬於 SaaS，並不意味著它就是沒有問題的。服務可能存在其他倫理問題。例如，Facebook 以 Flash 格式發佈視頻，這將迫使用戶運行私有軟件；它要求運行私有 JavaScript 代碼；它給用戶一種關於隱私的錯誤印象，同時引誘用戶在 Facebook 上暴露他們的生活。那些都是重要的問題，但不屬於 SaaS 的問題。

諸如搜索引擎的服務會從網絡上搜集數據以供您檢索。瀏覽它們所收集的數據在通常意義上並不屬於您自己的計算——由於您並不提供那些數據——因此使用這樣的服務檢索互聯網並非 SaaS。然而，使用他人的服務器來實現用於您自己網站的搜索工具是一種 SaaS。

在線購物不是 SaaS，由於這種計算不是您自己的；相反，這是由您和賣家為了共同的目的而共同實現的。在線購物的真正問題是您能否將您的錢財和其他個人信息（首當其衝的是您的名字）託付給對方。

倉庫網站諸如 Savannah 或 SourceForge 本質上不是 SaaS，由於它們

的職責是發佈那些提供給它的數據。

使用聯合項目的服務器不是 SaaS，由於您以這種方式所做的計算不是您自己的。例如，如果您編輯 Wikipedia 上的頁面，您並非在進行您自己的計算；與之相反，您正在參與 Wikipedia 的計算。Wikipedia 控制其自己的服務器，但是如果任何個人或組織使用他人的服務器進行他們自己的計算，他們將會遇到 SaaS 問題。

有些網站提供多種服務，如果其中之一不是 SaaS，其他服務有可能是 SaaS。例如，Facebook 的主要服務是社交網絡，那不是 SaaS；然而，它提供了第三方應用，其中的某些是 SaaS。Flickr 的主要服務是發佈照片，這不是 SaaS，但它也會提供照片編輯功能，這就是 SaaS。類似地，使用 Instagram 發佈這些照片並非 SaaS，但使用它轉換照片則是。

Google Docs 向我們展示要評估單一服務到底可以有多麼複雜。它邀請用戶通過運行大量私有 JavaScript 程序¹以編輯文檔，這無疑是壞的。然而，它還提供了用於標準格式上傳或下載文檔的應用程序接口（API）。自由的文檔編輯器可以通過這個 API 工作，這種應用場景不是 SaaS，由於它僅僅把 Google Docs 用作倉庫。將您的所有數據展示給一家商業公司不是好事，但這屬於隱私範疇，而非 SaaS；依賴這樣的服務來訪問您的數據不是好事，但這屬於風險範疇，而非 SaaS。另一方面，使用這種服務轉換文檔格式是一種 SaaS，因為您可以通過在您的計算機上運行適當的軟件（最好是自由的）來實現這一目的。

當然，通過自由的文檔編輯器使用 Google Docs 的情形並不常見。更為常見的情形是，人們通過私有 JavaScript 程序使用它，這類私有 JavaScript 程序和任何其他私有軟件一樣壞。而這種應用場景也可能涉及 SaaS；這取決於文檔編輯工作的哪些部分由 JavaScript 程序完成而哪些部分由服務器完成。我們並不清楚這個問題的答案，但既然 SaaS 和私有軟件都以相似的方式對用戶作惡，我們並不急迫地想知道這個問題的答案。

通過他人的倉庫發佈信息並不會引起隱私洩露的問題，但是通過 Google Docs 發佈則會遇到一個特別的問題：如果不借助運行私有

¹參見《JavaScript 陷阱》(p. 286) 一文以獲得更多信息。

JavaScript 代碼，甚至不可能在瀏覽器中顯示一段 *Google Docs* 中的文本。因此，您應該避免使用 *Google Docs* 發佈任何信息——然而其原因不屬於 SaaS 範疇。

信息技術（IT）業界試圖阻止用戶作出這樣的區分，這正是他們用行話“雲計算”來表達的用意所在。這個短語是那麼地空泛，以至於它可以指代有關互聯網的幾乎所有應用。它包括了 SaaS 以及許多其他網絡應用實踐。在任何給定的上下文語境中，一位作者（如果是一位技術工作者）在寫出“雲”這個詞的時候，他的腦海中可能確實有一種確切的涵義，但通常並不會指出這個詞在其他文章中可能具有的其他確切的涵義。這個短語將會誘導用戶將那些他們本來應該分開單獨考慮的實踐概念一般化。

如果“雲計算”確實有一種涵義，這種涵義並不是指一種進行計算的方式，與之相反，它應該指的是一種對計算的思考方式：一種不顧一切的極端思考方式可能會主張：“什麼也不要問；不要關心誰控制您的計算，也不要關心誰掌控您的數據；在像魚餌一樣吞下我們的服務之前也不要去看檢查裡面是否藏著魚鉤……什麼都不要猶豫，只要信任我們的商業公司就好。”換言之，就是“做一個沒有獨立思想的傀儡任人擺佈”。思想中的陰雲是進行清晰思考的障礙。為了能夠理清關於計算的思考，必須避免使用“雲”這個概念。

6.7.6 應對 SaaS 所帶來的問題

只有少數的網站提供 SaaS；大部分並不會帶來這樣的問題。但是，對於那些帶來了 SaaS 問題的網站，我們又該如何應對？

對於簡單的情形，如果您使用您的雙手進行您的計算，解決方案很簡單：使用您自己的自由軟件副本。例如編輯文本時，使用您自己的自由文本編輯器，比如 GNU Emacs，或者自由的文字處理器的副本。編輯照片時，使用您自己的自由軟件的副本，比如 GIMP。如果沒有可用的自由軟件又當如何呢？由於私有軟件和 SaaS 都會剝奪您的自由，您不應當使用它們。您可以貢獻您的時間或錢財來支持開發自由軟件替代品。

與他人一起作為一個小組進行協作的情況又如何呢？直到現在，不借

助服務器可能難以實現這一目的，並且您的用戶組可能不知道如何架設自己的服務器。如果您使用他人的服務器，至少不要信任由商業公司運營的服務器。對於消費者而言，僅憑一紙合同並不能帶來任何保護，除非您能夠抓住其中的漏洞並且真正能夠進行訴訟。即使如此，商業公司很可能在其編寫的合同條款中允許其在較寬的範圍內恣意妄為。假如那些公司不像之前那些為布什實施非法監聽其客戶的美國通訊公司那樣主動就範，國家仍然可以從商業公司那裡傳喚您和任何其他人的數據，如同奧巴馬對通訊公司所做的。因此，如果您不得不使用服務器，選擇那些運營商能夠給予您信任基礎而非僅僅是一層商業關係的服務器。

然而，在長遠的時間尺度上，我們可以開發出替代服務器的解決方案。例如，我們可以開發一種端對端的程序使得協作者可以分享加密數據。自由軟件社區應該試圖為關鍵的“網頁應用”開發出端對端的分佈式替代品。並且將它們在 GNU Affero GPL 許可證下發布可能是明智的，由於它們可以作為候選者供其他人轉換為基於服務器的軟件¹。GNU 計劃正在尋找志願者以開發這些替代品。我們可以邀請其他自由軟件項目在它們進行設計時考慮這一問題。

與此同時，如果一家商業公司邀請您使用它的服務器來進行您自己的計算任務，不要屈服；也不要使用 SaaS。不要購買或安裝所謂的“瘦客戶機”，它們只不過是廠商為了使您在其服務器上進行真實工作而提供的低性能計算機，除非您將會在您自己的服務器上使用它們。使用一臺真實的計算機來工作並且保存您的數據。使用您自己的自由軟件副本來進行您自己的計算，這是為了您的自由。

¹參見 “Why the Affero GPL” <http://gnu.org/licenses/why-affero-gpl.html> 以獲得完整解釋。

第 7 部分

珍視社區和你的自由

7.1 避免破壞性的妥協

Copyright © 2008, 2009, 2014, 2015 理查德·斯托曼 (Richard Stallman)。本文最早於 2008 年發表於 <http://gnu.org>

1983 年 9 月 27 日，我發起了一項計劃，致力於創造一個完全自由的操作系統，它稱作 GNU——意為 GNU 不是 Unix (GNU's Not Unix)。為了紀念 GNU 操作系統 25 週年，我寫出了這篇文章，旨在闡述我們的社區怎樣才能避開破壞性的妥協。除了避開這類妥協之外，您還可通過很多其他方式幫助 GNU 和自由軟件。其中一條基本方式是以合作成員的身份加入自由軟件基金會 (FSF) ¹。

在生活中的另一個領域存在這樣一種類似的觀點，例如文章 “‘Nudge’ Is Not Enough, It’s True. But We Already Knew That” <http://guardian.co.uk/commentisfree/2011/jul/19/nudge->

¹您可以在此通過加入會員的方式支持自由軟件基金會：<http://my.fsf.org/join>.

[is-not-enough-behaviour-change](#) (Jonathan Rowson, 2011 年 7 月 19 日) 所指出的觀點：除了溫和的勸說，我們還需要態度和視角方面的轉變。

自由軟件運動致力於帶來一場社會變革：讓所有軟件成為自由的¹，這樣所有軟件用戶將獲得自由，並且將會成為協作化社區的一部分。與之相反，任何一個非自由的程序賦予了其開發者凌駕於用戶之上的不公權力。我們的目標是終結這種不公。

自由之路，路漫漫兮²。這需要經歷很多步驟，花費很多年的時間才能達到這樣一種境地：軟件用戶擁有自由成為一種常態。這些步驟當中的一些可能會非常艱難，甚至需要為之付出一些犧牲。而其中有些步驟可能會變得簡單，如果我們與其他擁有不同最終目標的人們達成某種妥協。

因此，自由軟件基金會做出了一些妥協——甚至是重大妥協。例如，我們在 GNU 通用公共許可證 (GNU GPL) 第三版的專利提供條款中做出了妥協，使得主要的商業軟件公司可以貢獻併發布基於 GPL v3 的軟件，並藉此將部分軟件專利置於此條款的效力之下。

GNU 寬通用公共許可證 (GNU LGPL) 的目的在於一種妥協，我們將它用於一些選定的自由程序庫，以便允許將它們用於非自由軟件中。這是由於我們認為如果在法律層面禁止這樣做只會迫使開發者轉向使用私有程序庫。我們在 GNU 軟件中接受並安裝某種代碼使得它們能夠與常見的非自由軟件協同工作，並且我們對這種行為以這樣的方式進行描述和宣傳，鼓勵後者的用戶安裝並使用 GNU 軟件而不是相反。我們也會支持我們所認同的一些特定的運動，即使我們並不完全認同那些支持這些運動的組織。

但是，我們會堅決拒絕某些妥協，即使我們的社區中的很多其他人願意做出這樣的妥協。例如，我們僅僅支持並推廣有這樣的發行政策的

¹這裡的“自由”應理解為 freedom 中的自由，參見《什麼是自由軟件？》(p. 1) 一文以獲得自由軟件的完整定義。

²參見自由軟件基金會執行董事 John Sullivan 於 2008 年發表的文章：“The Last Mile Is Always the Hardest,” 位於 <http://fsf.org/bulletin/2008/spring/the-last-mile-is-always-the-hardest>，所提及的他本人對於這一問題的看法。

GNU/Linux 發行版¹：它們不會在其中包含非自由軟件，也不會引導用戶安裝非自由軟件。支持並推廣非自由的發行版將會是一種破壞性的妥協。

我們稱某種妥協是破壞性的，如果從長期來看，它們將會對實現我們的終極目標起到反作用。這種破壞性的妥協可以發生在想法的層面，也可以發生在行動的層面。

在想法層面，破壞性的妥協將會增強而非削弱那些我們所致力於改變的不利條件。我們的最終目的是實現這樣一種境地，即軟件用戶成為自由的。但是時至今日，大部分計算機用戶甚至不認可“自由”是一件重要的事。他們僅僅重視“消費者”價值。也就是說，他們評估任何軟件時，僅僅基於實踐上的特性，諸如價格和易用性。

戴爾·卡耐基的經典自立書籍《怎樣贏得朋友並且影響人們》(How to Win Friends and Influence People) 所建議的說服某人做某事最有效的方式是提供那些滿足對方價值觀的論證。我們可以通過多種方式滿足我們社會中典型的消費者價值觀。很多自由軟件也是兼具易用性與可靠性的。通過援引這些實踐層面的好處，我們已經成功地說服諸多用戶採納多種自由軟件，其中的一些已經大獲成功。

但是，如果長遠目標僅僅限於使更多的人使用某些自由軟件，您可能會選擇對“自由”這一概念保持沉默，並且將目光僅僅集中在那些實踐上對消費者價值有意義的好處上。而這正是“開源”這一概念以及與之關聯的一些術語所處的狀態。

然而，這樣一條路徑只能將我們帶到通往自由的終極目標的半路上。如果人們僅僅因為易用性而去使用某個自由軟件，那麼這些人將會僅僅在這款自由軟件仍然易用的情況下繼續堅持使用它。接下來，他們就再也找不到理由不去使用與這款自由軟件共存但更加易用的私有軟件。

“開源”的哲學滿足於消費者價值並且以此為前提，肯定並且強化這些消費者價值，這就是我們不能支持“開源”的原因。

為了完全地、長久地建立一個自由社區，我們要做的絕不僅限於讓人

¹您可以在這裡找到自由操作系統發行版準則 (GNU FSDG)：<http://gnu.org/philosophy/free-system-distribution-guidelines.html>。

們使用某些自由軟件。我們需要宣傳按照公民價值評估軟件（及其他事物）的理念，即基於它們是否尊重用戶的自由和社區，而非僅限於所謂的“易用性”。這樣人們就不會落入以吸引人、易用的特性作為誘餌的私有軟件的陷阱之中。

為了推廣公民價值，我們必須討論它們並且說明它們怎樣才能成為我們的行為準則。我們必須拒絕戴爾·卡耐基式的妥協，因為這將通過強調人們的消費者價值而影響他們的行為。

不過，這並不意味著我們完全不能援引實踐上的好處——我們可以適度地這樣做，並且我們也確實正在這樣做。只有當這種實踐上的好處佔據了用戶的注意力並且讓“自由”淪為背景時，這才成為問題。因此，當我們援引自由軟件實際的好處時，我們必須不厭其煩地反覆強調這些實踐上的好處只是用戶應該傾向於使用自由軟件附加的、次要的原因。

僅僅讓我們的口號與我們的理念保持一致並不夠，我們的行動也必須與理念保持一致。因此，我們必須避免做出這樣的妥協，它們會做那些我們所致力於終結的事情，或者使之合法化。

例如，經驗已經表明，您可以通過包含某些非自由軟件以吸引某些用戶使用 GNU/Linux。這可能是一個會吸引一些用戶目光的漂亮的非自由的應用程序，或者一種非自由的編程平臺，例如 Java¹（曾經是這樣）或者 Flash 運行時的環境（仍然是這樣），或者某種用於支持特定硬件型號的非自由設備驅動程序。

這些妥協無疑是誘人的，但它們會在無形之中毀掉我們的終極目標。如果您發佈非自由軟件或者引導用戶使用非自由軟件，您將會覺得自己沒有底氣說出“非自由軟件是一種不公，一種社會問題，我們必須終結它們”這樣的話。以至於即使您嘴上能夠繼續這樣說，您的行動也會漸漸破壞掉這些信念。

這裡的問題不在於用戶是否應該能夠或者被允許安裝非自由軟件；一個通用目的的系統應當支持並且允許用戶能夠去做他們所想要去做的任何事情。問題在於我們是否應當引導用戶投向非自由軟件。用戶獨立自主地

¹參見 <http://gnu.org/philosophy/java-trap.html> 獲取更多細節。

去做什麼事情是他們的責任，而我們為他們做了什麼，以及我們引導他們去做什麼則是我們的責任。我們無論如何不應該將用戶導向私有軟件，即使它看起來像是一種解決方案。因為私有軟件正是問題的根源。

破壞性的妥協不僅僅是對他人的不良影響，它還會通過認知上的不一致扭曲您自己的價值觀。如果您抱有某種價值觀，但是您的行動表現出來的是與之相沖突的其他價值觀，您很可能將會試圖改變您的價值觀或者行動，以便解決這種矛盾。因此，那些僅僅爭論實踐上的好處的，或者引導用戶轉向某些非自由軟件的項目，幾乎不可避免地羞於哪怕只是暗示“非自由軟件是不符合倫理的”。對於它們的參與者，以及對於公眾，它們只能強化消費者價值。我們必須拒絕這些妥協，如果我們想要堅持我們自己的價值觀不動搖。

如果您想要轉向自由軟件並且是為了自由的終極目標而不會做出妥協，您可以查閱自由軟件基金會的相關資源，它們位於 <http://www.fsf.org/resources>。這裡列出了能夠與自由軟件協同工作的硬件和機器配置列表，可供安裝的完全自由的 GNU/Linux 發行版的列表，以及能夠在百分之百自由軟件環境下工作的數千款自由軟件包¹。如果您想要幫助社區走在通往自由的正確道路上，很重要的一點就是公開支持公民價值。當人們正在討論什麼是好或者壞，或者要做什麼事情的時候，您可以援引自由和社區的價值觀並與他們爭論。

一條能夠讓您更快前行的道路有時並不是一條更好的道路，如果它通往錯誤的目的地。為了成就一個充滿雄心壯志的目標，做出一些妥協是必需的。但是，一定要警惕那些將您引離最終目標的妥協。

¹自由軟件目錄，位於 <http://directory.fsf.org>，這裡列出了我們所知的所有自由軟件。

7.2 克服社會慣性

Copyright © 2007, 2009 理查德·斯托曼 (Richard Stallman)，本文最初於 2007 年發佈於 <http://gnu.org>

自從 GNU 和 Linux 的結合使得自由地使用個人計算機 (PC) 成為可能已過去將近 20 年時間。我們也由此走過了一段漫長的歷程。現在，您甚至可以從不止一家硬件供應商那裡買到一臺預裝了 GNU/Linux 的便攜計算機——儘管它自帶的操作系統發行版可能不是完全由自由軟件組成的。那麼，究竟是什麼阻止了我們獲得完全的成功呢？

通往軟件自由的勝利之路上的主要障礙是社會慣性。它有多種存在形式，而且您一定曾見識過它們當中的某些形式。例如某些設備只能在 Windows 操作系統下使用，以及某些商業網站只能在 Windows 下訪問等。如果您相對於自由更加看重短期的易用性價值，您可能認為這些理由足以說服您使用 Windows。很多公司目前正在使用 Windows，因此那些思考短期價值的學生將會想要學習怎樣使用 Windows 並且要求學校講授 Windows。然後學校就會講授 Windows，從而培養出一屆又一屆只習慣於使用 Windows 的畢業生，而這又會鼓勵公司使用 Windows。

微軟一直在積極培養這種社會慣性：它鼓勵學校反覆灌輸對 Windows 的依賴性，並且約定企業建立起了眾多經後來證明只能從 Internet Explorer 瀏覽器訪問的網站。

幾年以前，微軟的廣告試圖論證運行 Windows 比運行 GNU/Linux 更加低成本。這些廣告裡做出的比較已經被揭穿。但是仍然值得注意的是，它們論證中的深層次瑕疵。一條暗含的前提條件援引一種形式的社會慣性，即“目前更多的技術人員相對於 GNU/Linux 更加熟悉 Windows”。真正珍視自由的人們不會為了節約成本而放棄自由，但是很多企業的執行官在思想意識上仍然堅信他們所擁有的任何東西都應該是可以買賣的，甚至是他們的自由。

社會慣性由那些屈從於社會慣性的人們構成。當您屈從於社會慣性時，您也成為了這種施加於他人的無形壓力的一部分。而當您抵抗這種社會慣

性的時候，您已經削弱了它的存在。我們可以通過識別社會慣性來克服它，並且堅定地拒絕成為它的一部分。

這是一個阻礙我們的社區發展的弱點：大部分 GNU/Linux 用戶甚至從未聽說過曾經推動了 GNU 運動的自由理念，因此他們仍然傾向於根據短期的易用性來評價事物，而非基於他們的自由。這使得他們更易被社會慣性牽住鼻子，從而也成為這種社會慣性的一部分。

為了建立起我們的社區力量以抵抗社會慣性，我們需要談論自由軟件和自由理念——而非僅僅談論那些開源支持者們所援引的實際好處。隨著更多人認識到他們為了克服社會慣性需要做什麼，我們將會取得更大進展。

7.3 自由還是權力？

Copyright © 2001, 2009 Bradley M. Kuhn 和 Richard Stallman 本文最初於 2001 年發表於 <http://gnu.org>。

本文由 Bradley M. Kuhn 和 Richard Stallman 撰寫。

對自由的愛是對他人的愛；對權力的愛是對自己的愛。——William Hazlitt

在自由軟件運動中，我們主張的是軟件用戶的自由。我們通過觀察哪些自由是美好的生活方式所必需的來闡述我們的觀點，並且讓實用的程序能夠幫助一個友善、合力、協作的社區成長。我們對於自由軟件的評價尺度¹指出了軟件用戶賴以同社區協作所需的自由。

我們主張程序員和其他用戶的自由。我們中的大部分人是程序員，在賦予您自由的同時也需要擁有我們自己的自由。但我們中的每個人都需要使用由他人編寫的軟件，並且我們需要在使用那些軟件的時候擁有自由，而不僅僅是使用我們自己編寫的代碼的自由。我們代表所有用戶的自由，不論他們經常、偶爾還是從不進行編程。

然而，有一種所謂的自由是我們絕不提倡的，即“為您所編寫的軟件隨意選擇授權許可的自由”。我們拒絕承認這一條的原因在於它實際上是一種權力而非自由。

這種時常被忽視的差別是至關重要的。自由是指能夠做出主要是影響自己的決定；而權力是指能夠做出影響他人甚於自己的決定。如果我們將權力和自由相混淆，我們將不能支持真正的自由。

使一個程序成為私有軟件的行為是一種行使權力的行為。當今的版權法律賦予了軟件開發者這樣的權力，因此他們，也只有他們，可以選擇施加於他人的規則——相對少數人替所有用戶做出了關於軟件的基本決定，通常是否定了用戶的自由。當用戶缺少定義了自由軟件的那些基本自由的時候，他們不能獲知自己所用的軟件到底在做什麼，不能檢查軟件的後門，

¹關於自由軟件的評價標準可參見《什麼是自由軟件？》(p. 1) 一文。

不能監視可能存在的病毒或蠕蟲，不能找出哪些個人信息正在被洩露（或者即使能夠知道也無法阻止洩露）。如果軟件不能正常工作，他們不能修復錯誤，而只能等待開發者行使他們的權力來進行修復。如果軟件本身不能根據用戶所需要的那樣工作，他們將受困於此。他們不能幫助他人改進這些軟件。

私有軟件的開發者往往也是商業公司。在自由軟件運動中，我們並不反對商業行為，但我們也看到了，當一家軟件公司擁有對其用戶施加任意規則的“自由”的時候將會發生什麼。關於拒絕用戶的自由是如何導致直接危害的，微軟是一個極壞的例子，但還不是唯一。即使是在沒有壟斷的情況下，私有軟件也會對社會造成危害。被迫在眾多主宰者中選擇一個的權利並不能稱之為自由。

關於軟件權利和規則的討論往往專注於程序員的利益。世界上只有少數人經常編寫程序，更少的人是私有軟件公司的擁有者。但是如今整個發展著的世界都需要並且使用軟件，因此現今的軟件開發者實際控制了整個世界的生活、經營、通訊和娛樂的方式。倫理和政治上的問題並不是一句口號“選擇的自由（僅對開發者適用）”就能解決的。

如果“代碼就是法律”¹，那麼我們所面對的真正問題就是：您所使用的代碼究竟應該由誰來控制——您，還是少數菁英人士。我們相信您應該有權控制您所使用的軟件，並且賦予您這樣的控制權正是自由軟件的終極目標。

我們堅信您應該有權決定您要使用軟件做什麼；然而，這不是當今的法律所表達的。當前的版權法律將我們置於這樣的境地：權力凌駕於代碼用戶之上，不論我們是否喜歡這樣。對於這種情況的倫理迴應是宣示每一位用戶的自由，如同權利法案要求以保證每位公民的自由為前提來行使政府的權力。這正是 GNU 通用公共許可證所致力於實現的：它使您能夠控制您對軟件的使用，同時保護您不被他人剝奪您做出決定的權利。²

¹William J. Mitchell, *City of Bits: Space, Place, and the Infobahn* (Cambridge, Mass.: MIT Press, 1995), p. 111, as quoted by Lawrence Lessig in *Code and Other Laws of Cyberspace, Version 2.0* (New York, NY: Basic Books, 2006), p. 5.

²參見《為什麼使用 Copyleft?》(p. 225) 獲得更多信息。

隨著越來越多的人意識到代碼就是法律，並且開始感受到他們也應當擁有自由，他們將會看到我們所主張的自由是多麼重要，如同越來越多的用戶開始感激我們所開發的自由軟件帶來的實際價值。

7.4 缺陷並不同於壓迫

Copyright © 2014 自由軟件基金會

當一款自由軟件缺少用戶想要的某些功能的時候，這是一件不幸的事情；我們需要人們添加缺失的功能。某些人可能會更進一步，並且宣稱一款軟件甚至不能稱之為自由軟件，如果它缺少某些功能——由於它拒絕了其所不能支持的用戶或應用場景的自由之零（以任何方式運行軟件的自由）。這一論述是理解錯誤的，因為它將功能判定為自由，而將缺陷等同於壓迫。

任何軟件在擁有特定功能的同時，不可避免地缺少某些可能是值得期待的其他功能。它可被用於執行某些任務，而它需要今後的進一步開發完善才能執行其他一些任務，這正是軟件的本質。

缺少某些關鍵的功能可能意味著特定用戶將會認為該軟件完全不可用。例如，如果您只懂得使用圖形界面，一款命令行界面的軟件可能對您而言完全不可用；如果您不能看清屏幕，一款不帶屏幕讀取器的軟件可能對您而言完全不可用；如果您只會講希臘語，一款帶有英語菜單和提示信息的軟件可能對您完全不可用；如果您的源程序是使用 Ada 語言編寫的，一款 C 語言編譯器可能對您完全不可用。要求您自己克服這樣的障礙是不合理的。自由軟件確實應該提供您所需要的功能。

自由軟件確實應該提供這些功能，然而缺少這些功能並不會使得這款軟件成為非自由的，因為這只是一種缺陷而非一種壓迫。

使一款軟件成為非自由的是由拒絕用戶自由的開發者做出的不公行為。這樣的開發者應該因此而受到譴責。譴責這樣的開發者是重要的，因為只要開發者一直這麼做，其他人就不能夠解除這種不公。我們可以，並且確實在嘗試通過開發自由的替代品來挽救受害者。但我們不能使那些非自由軟件變成自由的。

開發一款自由軟件而沒有添加某些重要功能並不是對任何人作錯事。與之相反，這在某種程度上是在做好事，只是尚未實現人們所需要的好處。並沒有某些特定的人應該因為未能開發出某些缺失的功能而受到譴

責，因為任何有能力進行彌補的人都可以做。單獨指出自由程序的作者們並且指責他們未能完成某些額外的工作是不通情理的，並且這樣做只會自討苦吃。

我們所能做的是指出完成這項任務尚需進行哪些額外的開發工作。這將是建設性的，因為這有助於我們說服某些有能力的人去從事那些開發工作。

如果您認為某個自由軟件中的某個特定擴展程序十分重要，請以一種尊重我們貢獻的方式提出。不要批判那些已經為我們貢獻了有用代碼的人。與之相反，您應該試著找到一種方式以完成任務。您可以敦促該程序的開發者在他們有時間進行更多工作的時候關注所缺失的功能；您還可以為他們提供幫助；您也可以通過招募人員或者籌集資金的方式支持這項工作。

7.5 民主可以承受多少監控？

Copyright © 2015 理查德·斯托曼 (Richard Stallman)。本文最初以相同的標題發表於《連線》雜誌 (2013 年 10 月 14 日) <http://www.wired.com/opinion/2013/10/a-necessary-evil-what-it-takes-for-democracy-to-survive-surveillance>

多虧了愛德華·斯諾登的揭露，我們才知道當今社會中的普遍監控級別已經與人權不相容。在美國和世界其他地區持續發生的針對持不同政見者、信息來源和新聞工作者的騷擾和指控確認了這一點。我們需要適當降低普遍監控的級別，但是到底應當在多大程度上？我們所必須保證其不被超越的，可接受的最大監控級別到底是多少？這種監控級別應當這樣定義：一旦超過了這樣的級別，監控行為將會干涉民主的運行。此時揭露者（例如斯諾登）很可能因此而被逮捕。

面對政府的保密政策，我們作為人民只能依靠揭露者告知國家正在做什麼¹。然而，當今的監控對潛在的揭露者進行了威迫，也就是說監控程度過高。為了重獲我們對於國家民主的控制，我們必須設法降低監控級別使得揭露者能夠確信他們是安全的。

如同我們已經倡導了 30 多年的，使用自由軟件是掌控我們自己的數字生活的第一步。這也包括了避免被監控。我們不可能信任任何私有軟件；由於美國國家安全局 (NSA) 通過利用²甚至還有製造³私有軟件中的安全漏洞來入侵我們自己的計算機和路由器。自由軟件賦予了我們控制自己計算機的權利，但僅憑這一點並不足以在我們涉足互聯網的時候保護我們的

¹Maira Sutton, “We’re TPP Activists: Reddit Asked Us Everything,” 21 November 2013, <https://www.eff.org/deeplinks/2013/11/reddit-tpp-ama>.

²Glyn Moody, “How Can Any Company Ever Trust Microsoft Again?” 17 June 2013, <http://www.computerworlduk.com/blogs/open-enterprise/how-can-any-company-ever-trust-microsoft-again-3569376/>.

³James Ball, Julian Borger and Glenn Greenwald, “Revealed: How US and UK Spy Agencies Defeat Internet Privacy and Security,” 6 September 2013, <http://theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security>.

隱私¹。

美國國內“限制國內監控權力”²的兩黨立法已經被提出，但它依賴於限制政府對我們的虛擬檔案的使用。這不足以保護揭露者，如果“逮捕揭露者”是獲取足夠信息以確認此人身份的基礎。我們需要做得更多。

7.5.1 民主社會中的監控程度上限

如果揭露者不敢揭發罪行和謊言，我們將會失去對政府和制度僅存的有效控制。這可以解釋為何那種使得國家可以知道是誰同記者進行交談的監控行為是過分的——超出了民主所能承受的範圍。

在 2011 年，一位匿名的美國政府官員不懷好意地告訴新聞工作者：美國政府在傳喚記者時，並不會以“我們知道你同誰交談”作為理由³。有時，記者的通話記錄將會被傳喚以獲知這個問題的答案⁴。但是，斯諾登已經向我們展示了，它們實際上將會在任何時刻傳喚美國境內的任何人的所有通話記錄，通過 Verizon⁵或其他通訊公司⁶。

反對者和持不同政見者的活動需要對國家保密，由於國家將會主動對

¹Bruce Schneier, “Want to Evade NSA Spying? Don’t Connect to the Internet,” 7 October 2013, <http://www.wired.com/2013/10/149481/>.

²Dan Roberts, “Patriot Act Author Prepares Bill to Put NSA Bulk Collection ‘Out of Business,’” 10 October 2013, <http://theguardian.com/world/2013/oct/10/nsa-surveillance-patriot-act-author-bill>.

³Lucy Dalglish, “Lessons from Wye River,” *The News Media & the Law* (Summer 2011): p. 1, <http://www.rcfp.org/browse-media-law-resources/news-media-law/news-media-and-law-summer-2011/lessons-wye-river>.

⁴Washington Agencies, “Yemen leak: former FBI man admits passing information to Associated Press,” 24 September 2013, <http://www.theguardian.com/media/2013/sep/24/yemen-leak-sachtleben-guilty-associated-press>.

⁵參見 “Verizon forced to hand over telephone data—full court ruling” (6 June 2013) 位於 <http://www.theguardian.com/world/interactive/2013/jun/06/verizon-telephone-data-court-order>，以獲得關於美國外國情報監控法庭（FISA）之下美國政府採集 Verizon 旗下數以百萬計的美國客戶的通話記錄的信息。

⁶Siobhan Gorman, Evan Perez, and Janet Hook, “NSA Data-Mining Digs into Networks Beyond Verizon,” 7 June 2013, <http://www.marketwatch.com/story/nsa-data-mining-digs-into-networks-beyond-verizon-2013-06-07>.

他們施展陰謀詭計。美國公民自由聯盟（ACLU）已經指出美國政府對和平的持不同政見者和組織所進行的系統性滲透行為，理由是他們當中可能潛伏著恐怖分子¹。

7.5.2 信息，一旦被採集，將會被濫用

當人們認識到普遍監控的級別過高的時候，他們的第一反應可能會是提議限制對採集到的數據的訪問。這聽起來很好，但這甚至絲毫不能解決問題，即使假定政府遵守這條規則。（美國國家安全局（NSA）曾經欺騙美國外國情報監控法庭（FISA），後者聲稱它不能有效地證實 NSA 有義務對其監控行為作出說明²。）犯罪的嫌疑將成為訪問這些數據的理由，於是，一旦揭露者被指控從事間諜活動，試圖找到這個間諜將會為訪問所採集到的數據提供理由。

此外，國家監控系統的工作人員會出於個人原因濫用數據。一些 NSA 特工使用美國國家監控系統來跟蹤他們的情人——不論是過去的、現在的還是正在追求的——並將這種做法稱為 LOVEINT³。NSA 宣稱它已經發現了幾次這樣的行為並對其進行了處罰；但是我們不知道還有多少次這樣的行為沒有抓到。然而，這些事件並不會讓我們感到驚訝，由於警方已經長期利用他們可訪問的駕照記錄以跟蹤那些迷倒了他們的對象。這種行為稱為“查詢車牌獲得約會”（running a plate for a date）⁴。

監控數據總會被用於其他目的，即使這是被禁止的。只要數據被採集，國家就有可能訪問它們。國家可以用極壞的方式濫用這些數據，發生在歐

¹ACLU, “Policing Free Speech: Police Surveillance And Obstruction of First Amendment-Protected Activity,” 29 June 2010, https://www.aclu.org/files/assets/Spyfiles_2_0.pdf.

²David Kravets, Kim Zetter, Kevin Poulsen, “NSA Illegally Gorged on U.S. Phone Records for Three Years,” 10 September 2013, <http://www.wired.com/2013/09/nsa-violations/>.

³Adam Gabbatt and agencies, “NSA Analysts ‘Wilfully Violated’ Surveillance Systems, Agency Admits,” 24 August 2013, <http://theguardian.com/world/2013/aug/24/nsa-analysts-abused-surveillance-systems>.

⁴M. L. Elrick, “Cops Tap Database to Harass, Intimidate,” 31 July 2001, <http://sweetliberty.org/issues/privacy/lein1.htm/#/.VeQixucpDow>.

洲¹和美國²的一些例子證實了這一點。

國家所採集的個人信息同樣可能由於服務器的安全措施被攻陷而被境外駭客獲得，甚至是被那些效力於敵對國家的駭客獲得³。

政府可以輕鬆地利用大規模監控能力直接顛覆民主制度⁴。

國家對監控數據的完全訪問權力使得國家可能對任何人發動大規模的徹底搜查。為了保護新聞業和民主的安全，我們必須限制對信息的採集，而這些信息是國家可以輕鬆訪問的。

7.5.3 對隱私的有效保護必須是技術層面的

電子前哨基金會（EFF）和一些其他組織提出了一系列法律準則以期阻止濫用大規模監控⁵。這些關鍵性準則包括了對揭露者明確的法律保護。其結果是這些準則將足以保護民主自由——如果它們能夠永久地得到完全採納，並且被沒有例外地嚴格強制執行。

然而，這樣的法律保護是不牢靠的：如同最近多年的歷史事實所展示的，它們可以被廢除（如同在美國外國情報監控法案 FISA 修正案中）、被架空或者被無視⁶。

同時，一些蠱惑民心的政客將會援引常見的理由作為支持完全監控的基礎；任何恐怖襲擊，即使只是造成了極少的人員傷亡，也可以被誇大作

¹Rick Falkvinge, “Collected Personal Data Will Always Be Used against the Citizens,” 17 March 2012, <http://falkvinge.net/2012/03/17/collected-personal-data-will-always-be-used-against-the-citizens/>.

²考慮二戰期間被收容的日裔美國人。

³Mike Masnick, “Second OPM Hack Revealed: Even Worse Than the First,” 12 June 2015, <https://www.techdirt.com/articles/20150612/16334231330/second-opm-hack-revealed-even-worse-than-first.shtml>.

⁴Joanna Berendt, “Macedonia Government Is Blamed for Wiretapping Scandal,” 21 June 2015, http://www.nytimes.com/2015/06/22/world/europe/macedonia-government-is-blamed-for-wiretapping-scandal.html?_r=0.

⁵“International Principles on the Application of Human Rights to Communications Surveillance,” last modified May 2014, <https://en.necessaryandproportionate.org/text>.

⁶Eric Lichtblau and James Risen, “Officials Say U.S. Wiretaps Exceeded Law,” 15 April 2009, <http://nytimes.com/2009/04/16/us/16nsa.html>.

為證據以支持上述理由。

如果對數據訪問的限制被擱置一旁，就如同這些限制從未存在過一般：多年來的有價值的檔案將會立即被國家和它的特工所濫用。如果這些檔案被商業公司獲得，也會被這些商業公司出於其私有目的而被濫用。然而，如果我們能夠阻止對個人檔案的採集，這些檔案將不復存在，並且國家不能追溯既往地編制這些檔案。新的反自由的政體將會不得不從頭實施監控，並且它自成立之初只會採集新的數據。至於架空或者立即無視這條法律，這種想法幾乎沒有任何意義。

7.5.4 最重要的是，不要犯傻

如果您想要擁有隱私，您一定不要主動將其放棄：有義務保護您的隱私的最重要的人就是您自己！您必須避免向網站暴露自己的身份，您可以使用 Tor 同它們聯絡，或者使用那些能夠阻止網站用於跟蹤訪問者的陰謀詭計的瀏覽器。您可以使用 GNU 隱私衛士（GnuPG）加密您的郵件內容。您可以使用現金支付任何費用。

保護您自己的數據；不要將您的數據儲存在某家商業公司的“簡便易用”的服務器上。然而，在下面這種情況下，將數據存儲在商業服務之上是安全的：只要您將文件置於歸檔文件中，並且加密整個歸檔文件，包括文件名。務必使用自由軟件在您自己的計算機上進行操作，然後再上傳。

出於隱私考慮，您必須避免使用私有軟件，這是由於私有軟件賦予他人控制您的計算機使用的權力，它們很可能在監控您¹。您還應該拒絕使用“作為軟件替代品的服務”（SaaS）²，同樣由於這賦予了他人控制您如何計算的權力，它要求您將全部相關數據提交到它們的服務器。

您也需要保護您的朋友或熟人的隱私。除了聯繫方式以外，不要洩露

¹幾十年來，自由軟件運動一直致力於揭露私有軟件公司諸如微軟和蘋果的專權的監控機制。監控行為已經在各行業之間蔓延，而不再限於軟件產業，並且——離開鍵盤的限制——進入移動計算領域，在辦公室、家庭、交通工具、教室內等。如需獲知監控行為進入這些領域的不斷增加的方式，參見 <http://gnu.org/philosophy/proprietary/proprietary-surveillance.html>。

²參見《服務器真正是在為誰服務？》（p. 302）一文以獲取更多信息。

他們的任何個人信息¹。並且不要向任何網站洩露您的郵件列表或者電話聯繫人。不要將您的朋友的任何信息告訴諸如 Facebook 這樣的公司，因為您的朋友也許並不想在報紙上公佈他們的名字。如果可能，根本不要被 Facebook 所利用。拒絕使用任何要求用戶提供真實姓名的通訊系統，即使您願意供出您的名字，由於這些通訊系統會向他人施壓以迫使他們交出隱私。

自我保護是至關重要的，但即使是最嚴密的自我保護也不足以保護您的隱私免於被不屬於您的系統所洩露。當我們與他人通訊或者在城市內出行的時候，我們的隱私取決於社會的實踐。我們可以避開一些但不是全部的可以監控我們的通訊或行蹤的系統。顯然，最佳的解決方案是讓所有這些系統停止監控任何人，除了法律意義上的嫌犯。

7.5.5 我們需要基於隱私原因設計每一個系統

如果我們不想要一個全面監控的社會，我們必須將監控視為一種社會汙染，並且限制監控對每一個新系統的影響，如同我們要限制實體建設工程對環境的影響。

例如，“智能”電錶宣稱它能夠向電力公司持續發送每位用戶的電力消耗，包括與普通用戶相比較的情況。這種統計是基於普遍監控而實現的，但實際上卻又不需要任何監控行為。電力公司可以容易地計算出某一居住區的平均電力消耗，通過將總消耗除以購電者的數量，並且將這個平均值發送至電錶。每個用戶的電錶可以將本人的電力使用情況同任何時期的平均電力使用相比較，這樣就可以不用監控而實現所有好處！

我們需要將這樣的隱私設計融入我們所有的數字系統。

¹Nicole Perlroth, “In Cybersecurity, Sometimes the Weakest Link Is a Family Member,” 21 May 2014, <http://bits.blogs.nytimes.com/2014/05/21/in-cybersecurity-sometimes-the-weakest-link-is-a-family-member/>.

7.5.6 針對數據採集的補救：讓數據分散開來

使監控行為不危害隱私的一種方式是使數據保持分散狀態從而難於訪問。老式的安保攝像頭對隱私很少構成威脅¹，由於錄像數據存儲在安裝這些設備的營業場所內，並且至多被保存幾周。由於訪問這些數據相對困難，這種數據採集行動從未被大規模部署；只有當某人報導一起犯罪行為時，這些影響數據才會被訪問。每天對數以百萬計的數據卡帶進行人工採集再進行觀看或複製幾乎是不現實的。

如今，老式的安保攝像頭已經變成了監控攝像機：由於它們被連接到互聯網，採集的影像可以傳輸到一所數據中心並被永久保存。這已經是很危險的了，但事情正在變得更壞。面部識別技術的發展使得這樣的事情成為可能：對可疑的新聞記者在街道上進行不間斷跟蹤以便察看他們同何人交談。

通常，聯網攝像機的自身數據安全措施極差，使得幾乎任何人都能察看它所記錄的內容²。為了重獲隱私，我們應當禁止在針對公眾的場合使用聯網的攝像機，除非它由人來操作。每個人必須被允許偶爾發佈照片或視頻記錄，但對於互聯網上相關數據的系統性採集行為必須受到限制。

7.5.7 針對互聯網商業監控的補救

大部分數據採集行為來自於人們自身的數字活動。數據通常首先由商業公司進行採集。但是，當討論監控行為對隱私和民主的威脅的時候，監控行為直接由國家進行或者由商業公司代為進行並無本質區別，由商業公司所採集的數據，國家也可以系統性地獲取。

¹我在此假定安保攝像頭用於諸如商店內部或大街上。任何由其他人架設的對準某人私人空間的攝像機侵犯了隱私，但這是另一個問題。

²Ms. Smith, "CIA Wants to Spy On You through Your Appliances," 18 March 2012, <http://networkworld.com/article/2221934/microsoft-subnet/cia-wants-to-spy-on-you-through-your-appliances.html>.

NSA 通過稜鏡計劃 (PRISM) 進入了多家大型互聯網公司的數據庫¹。AT&T 自 1987 年起保存了所有通話記錄並且允許美國緝毒局 (DEA) 搜索其所有數據²。嚴格地說，美國政府並不直接擁有這些數據，但它實際上確實擁有了這些數據。

因此，為了保證新聞業和民主的安全，我們必須減少由任何組織採集的關於個人的數據，而不僅僅減少由國家採集的數據。我們必須這樣重新設計各種數字系統以使其不再採集其用戶的數據。如果它們確實需要關於我們的重要數字資料，當超出處理數據所需的基本時間以後，這些機構和個人就不應該被允許繼續保留那些數據。

當前，互聯網監控級別的動機之一是經濟支持，即網站通過跟蹤用戶的活動和偏好而進行的廣告行為。這使得廣告，作為一種我們可以學會無視之的行為，從一種僅僅是惱人的行為變成了一種對我們造成傷害的監控系統，不論我們是否瞭解它的內情。互聯網購物同樣會跟蹤用戶，我們都已經意識到，所謂的“隱私條款”與其說是維護用戶隱私的承諾，不如說是它們用於侵犯用戶隱私的藉口。

我們可以通過採用一種匿名支付系統——即隱藏付款人的身份——來解決以上兩個問題（我們並不想協助收款人避稅）。比特幣不是匿名的³，儘管有人試圖開發出允許使用比特幣進行匿名支付的方式。然而，數字貨幣技術的開發始於 20 世紀 80 年代⁴；我們只需要對商業規則進行適當調整，並且使得國家不會阻止它們。

網站對個人數據的採集的更大的威脅在於駭客可以攻陷安全措施，獲

¹Jon Queally, “Latest Docs Show Financial Ties between NSA and Internet Companies,” 23 August 2013, <http://www.commondreams.org/news/2013/08/23/latest-docs-show-financial-ties-between-nsa-and-internet-companies>.

²Scott Shane and Colin Moynihan, “Drug Agents Use Vast Phone Trove, Eclipsing N.S.A.’s,” 1 September 2013, http://www.nytimes.com/2013/09/02/us/drug-agents-use-vast-phone-trove-eclipsing-nsas.html?_r=0.

³Dan Kaminsky, “Let’s Cut through the Bitcoin Hype: A Hacker-Entrepreneur’s Take,” 3 May 2013, <http://wired.com/2013/05/lets-cut-through-the-bitcoin-hype/>.

⁴Steven Levy, “E-Money (That’s What I Want),” Wired, 2.12 (December 1994), http://archive.wired.com/wired/archive/2.12/emoney_pr.html.

取並濫用個人數據。這可能還包括用戶的信用卡信息。而匿名支付系統可以終結這種威脅：如果網站不知道您的任何信息，那麼網站的安全漏洞就不會危害到您。

7.5.8 針對旅行監控的補救

我們必須將數字收費系統改為匿名支付系統（例如使用數字貨幣）。車輛牌照識別系統將會識別各種牌照，而這些數據可以被無限期保存¹；應該由法律要求它們僅僅記錄那些由法庭命令要求追查的牌照號碼。另一種不太安全的措施是在本地記錄所有車輛牌照，但僅保存幾天時間，並且不允許從網絡訪問所有數據；對數據的訪問應該限於搜索一系列由法庭命令要求追查的牌照號碼之中。

美國“禁飛黑名單”必須被廢除，由於這是一種未經審判的刑罰²。

要求對黑名單上某個乘客的行李進行額外的搜查是可以接受的，國內航班上的匿名乘客可以視為在此黑名單上。禁止非某國公民登上飛往該國的航班也是可以接受的，如果他們根本沒被批准入境。這些措施對於任何法律目的都是足夠的。

很多公共交通系統使用某種智能卡或者射頻識別（RFID）設備進行支付。這些系統將會採集個人數據：只要您錯誤地使用現金以外的任何方式進行支付，它們將會將此卡片和您的姓名永久關聯起來。接下來，它們將會記錄與每塊卡片相關聯的所有出行信息。這些行為加起來已經構成了大規模監控，這樣的數據採集必須被限制。

導航服務也會進行監控：用戶的計算機將用戶的所在地和目的地告知地圖服務；而後服務器確定路線，返回用戶的計算機並且顯示出來。現在，服務器很可能會記錄用戶的位置信息，由於沒有什麼措施能夠阻止它們這

¹Richard Bilton, “Camera Grid to Log Number Plates,” last updated on 22 May 2009, http://news.bbc.co.uk/2/hi/programmes/whos_watching_you/8064333.stm.

²Nusrat Choudhury, “Victory! Federal Court Recognizes Constitutional Rights of Americans on the No-Fly List,” 29 August 2013, <https://www.aclu.org/blog/victory-federal-court-recognizes-constitutional-rights-americans-no-fly-list>.

樣做。這種監控行為本質上並不必要，並且可以通過重新設計來解決：用戶計算機中的自由軟件將會下載相關地區的地圖數據（如果之前從未下載），計算出最佳路線並且顯示出來，而無需告知任何人用戶的所在地或目的地。

用於諸如自行車租借等目的的系統可以這樣設計：租借者的身份僅在其借出物品的站點內可知。物品借出時，將會通知所有站點某件物品處於借出狀態，這樣，當用戶將物品返還回任何站點（通常是另一處站點）的時候，該站點將會獲知該物品被借出的時間和地點，同時將會告知所有其他站點該物品不再處於借出狀態。站點還會計算用戶的賬單並將賬單信息（等待隨機長度的時間之後）沿著一系列環形拓撲結構的站點之間的線路發送至總部。這樣，總部將不會獲知賬單信息來自哪個站點。當這一操作完成後，歸還站點將會忘記所有與這筆已完成的業務有關的信息。如果某一物品長時間處於借出狀態，借出該物品的站點將會告知總部。此時，它可以立即發送借用者的身份信息。

7.5.9 關於通訊檔案的補救

互聯網服務供應商（ISP）和電信公司保存著海量的用戶聯繫人信息（瀏覽、通話記錄等）。對於移動電話，還會記錄用戶的物理位置¹，例如 AT&T 已保存了超過 30 年。不久以後它們甚至還會記錄用戶的身體活動²。並且 NSA 很可能正在大規模採集移動電話的物理位置數據³。

只要通訊系統創建這樣的通訊檔案，不受監視的通訊就不可能實現。因此創建或記錄這些通訊檔案應該被判定為非法。ISP 和電信公司必須不被允許長期保存這些信息，或者在沒有法庭命令的情況下長期監控某一特

¹Kai Biermann, “Betrayed by Our Own Data,” 26 March 2011, <http://www.zeit.de/digital/datenschutz/2011-03/data-protection-malte-spitz>.

²Sara M. Watson, “The Latest Smartphones Could Turn Us All into Activity Trackers,” 10 October 2013, <http://wired.com/2013/10/the-trojan-horse-of-the-latest-iphone-with-the-m7-coprocessor-we-all-become-qs-activity-trackers/>.

³Patrick Toomey, “It Sure Sounds Like the NSA Is Tracking Our Locations,” 30 September 2013, <https://aclu.org/blog/it-sure-sounds-nsa-tracking-our-locations>.

定人群。

這種解決方案並不完全令人滿意，由於這實際上並不能阻止政府在通訊信息生成的時候立即對其進行採集——這正是美國政府對部分或全部電信公司所做的¹。我們可能必須依靠法律禁止這種行為。但是，這種假設比現實的狀況好得多，現實中的相關法律（美國愛國者法案，我稱之為 PATRIOT Act，即“鎮壓暴動法案”）並不明確禁止這種行為。此外，如果政府重啟這種監控，它不應得到重啟監控的時間點之前發生的每位用戶的通話記錄數據。

為了保護您的電子郵件聯繫人的隱私，一種簡單的方式是您和他人都在使用某個不會與您所在國家政府進行合作的國家提供的郵件服務，並且在通訊過程中使用加密。然而，Ladar Levison（Lavabit 的擁有者，美國監控系統試圖對其郵件服務實現完全控制）提出了一種更為高級的加密系統設想：您向我的郵件服務的某位用戶發送郵件，我的郵件服務所知道的只是我收到了來自您所使用的郵件服務的某位用戶的郵件，但難以確認是您向我發送了郵件。

7.5.10 但是，適度的監控是必需的

國家為了緝捕罪犯，它需要能夠在法庭命令下調查特定的犯罪行為或者疑似犯罪預謀。在互聯網時代，監聽通話的權力自然延伸到監聽互聯網連接的權力。這種權力容易出於政治原因而被濫用，但這也是必需的。幸運的是，這並不會使得在案件發生之後找到揭露者變為可能，如果（如我所建議的）能夠阻止數字系統在事件發生之前進行大規模檔案信息採集。

擁有國家賦予權力的個人，例如警察，將被收回個人的隱私權並且必

¹Glenn Greenwald, “NSA Collecting Phone Records of Millions of Verizon Customers Daily,” 6 June 2013, <http://www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order>.

須被監視（事實上，警察擁有屬於他們自己的偽證罪別名 “testilying”¹。由於他們經常做出這樣的事情，特別是對抗議者和攝影師²）。加州的某個城市要求警察隨時隨身攜帶攝像機之後，他們的武力使用下降了 60%³。ACLU 對此表示歡迎。

商業公司不是自然人，因此不應被賦予自然人的權利⁴。要求商業公司公開其行為的細節是正當合理的，這些行為可能會造成化學、生物、核、財政、計算機相關（例如數字限制管理 DRM⁵）或者政治（例如遊說拉票行為）等方面對社會的危害，這些危害必須被控制在公眾福祉所要求的範圍以內。這些行為所造成的危害（考慮諸如墨西哥灣漏油事件、福島核電站事故、2008 年財政危機等）更甚於恐怖主義。

然而，新聞業必須被保護免遭監控，即使這種監控行為是作為某項事務的一部分而被執行的。

數字技術的發展極大地提高了我們的出行、活動和通訊所受的監控水平。這種監控水平遠遠超過了 20 世紀 90 年代我們所經歷過的，也遠遠超過了 20 世紀 80 年代生活在鐵幕籠罩之下的人們所經歷的⁶。而提議國家對使用採集到的數據進行法律限制並不能改變這種狀況。

商業公司正在設計更具侵略性的監控設施。一些充斥著監控行為的項目依附於諸如 Facebook 之類的公司，它們可能對人們的思考方式產生深

¹一些例子 “Testilying: Cops Are Liars Who Get Away with Perjury” (Nick Malinowski, 3 February 2013, <http://vice.com/read/testilying-cops-are-liars-who-get-away-with-perjury>) 以及 “Detective Is Found Guilty of Planting Drugs” (Tim Stelloh, 1 November 2011, <http://nytimes.com/2011/11/02/nyregion/brooklyn-detective-convicted-of-planting-drugs-on-innocent-people.html?pagewanted=all&r=0>)

²關於這一點參見 “攝影不是犯罪” 網站 <http://photographyisnotacrime.com/>

³Kevin Drum, “Ubiquitous Surveillance, Police Edition,” 22 August 2013, <http://motherjones.com/kevin-drum/2013/08/ubiquitous-surveillance-police-edition>.

⁴Public Citizen, “Call Your Representative: Tell Her or Him to Co-Sponsor a Constitutional Amendment to Overturn Citizens United and Restore Democracy to the People,” August 2015, http://action.citizen.org/p/dia/action3/common/public/?action_KEY=12266.

⁵參見《應避免使用（或慎用）的詞語》一文中有關 DRM 的敘述 (p. 115)。

⁶James Allworth, “Your Smartphone Works for the Surveillance State,” 7 June 2013, <https://hbr.org/2013/06/your-iphone-works-for-the-secret-police>.

遠的影響¹。這樣的可能性是不可預測的；然而它對民主的威脅已經不是推測。這種威脅無處不在，隨時可見。

除非我們堅信我們自由的國家之前的監控行為嚴重欠缺，並且我們理應受到更甚於蘇聯和民主德國（東德）那樣的監控，否則我們必須逆轉這種監控升級的趨勢。這依賴於阻止對民眾的大數據進行大規模採集。

¹Evan Selinger and Brett Frischmann, “Will the Internet of Things Result in Predictable People?” 10 August 2015, <http://theguardian.com/technology/2015/aug/10/internet-of-things-predictable-people>.

附錄 A

關於軟件的基礎知識

Copyright © 2002 Richard E. Buckman and Joshua Gay. 此文最早發佈於 2002 年。

由 Richard E. Buckman 和 Joshua Gay 撰寫

本節內容是特意为對計算機科學技術瞭解不多的人準備的。這節內容對於理解書中的文章和演講並不是必須的，不過，它可以幫助那些不熟悉編程和計算機的人理解一些術語。

計算機程序員來寫軟件或者計算機程序。而程序可以認為是告訴計算機如何完成特定任務的一系列指令。你應該熟悉許多不同類型的應用程序：比如你的網頁瀏覽器、你的文字處理器和你的郵件客戶端等等。

程序最初的形態通常是源代碼。這一系列高級指令由編程語言（比如 C 或者 Java）編寫而成。之後會被一個名為編譯器的工具編譯為一種更底層的語言——彙編語言。另一種被稱為彙編器的工具會將彙編代碼分解為最終的機器語言——計算機可以原理解的最底層代碼。

例如，“Hello World” 這個程序，通常是人們學習 C 語言時的第一個程

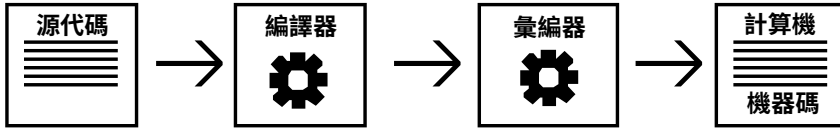


圖 A.1: 計算機程序是如何從源代碼生成二進制可執行文件的

序，編譯和執行後會在屏幕上打印出 “Hello World”¹。

```

int main(){
    printf("Hello World!");
    return 0;
}
  
```

在 Java 語言中，同樣的程序會是這麼寫：

```

public class hello {
    public static void main(String args[]) {
        System.out.println("Hello World!");
    }
}
  
```

然而用機器碼錶示，其中的一小部分可能會類似這樣：

1100011110111010100101001001001010101110

011010101001100000111100101101010111101

¹對於其他的編程語言，比如 Scheme，通常不從 Hello World 程序開始入門。在 Scheme 中，你通常會從這樣的程序開始：

```

(define (factorial n)
  (if (= n 0)
      1
      (* n (factorial (- n 1)))))
  
```

這段程序是求一個數的階乘；也就是運行 (factorial 5) 將會輸出 120，即 5 乘以 4 乘以 3 乘以 2 乘以 1。

```

01001111111111110010110110000000010100100
0100100001100101011011000110110001101111
0010000001010111011011110111001001101100
0110010000100001010000100110111101101111

```

上面這段機器碼就是基本的二進制形式。計算機中所有的數據都是由 0 或 1 組成的，但是人們很難理解這些數據。如果想要對二進制的值進行一個簡單的更改，就不得不對特定計算機編譯機器語言的原理了如指掌。這對於類似於上面這種小程序來說或許是可行的，但是對於任何有趣的程序，做一個簡單的修改就需要耗費大量的精力。

比如說，如果我們想讓讓 C 語言編寫的“Hello World”程序輸出的英語“Hello World”變成法語。只需要做出簡單的修改；新的程序如下：

```

int main() {
    printf("Bonjour, monde!");
    return 0;
}

```

可以肯定地說，我們可以很容易的推斷出如何修改 Java 代碼以達到同樣的效果。然而更多的程序員不懂如何修改二進制形式。當我們說“源代碼”的時候，不是指只有機器可以理解的機器語言——我們指的是一些高級語言比如 C 和 Java。還有一些其他的比較流行的編程語言，比如 C++、Perl 和 Python。或許有些在理解或編程時比較難以理解，但都比理解編譯和彙編以後的機器語言簡單多了。

另一個重要概念是理解什麼是操作系統。操作系統是控制輸入輸出，內存分配和任務調度的軟件。通常點說，一些更常見且更有用的程序，比如圖形用戶界面（*Graphical User Interface*, GUI），也是操作系統的一部分。GNU/Linux 操作系統包括 GNU 和非 GNU 軟件，以及一個被稱為 *Linux* 的內核。內核負責處理底層任務比如輸入輸出和任務調度。GNU 軟件包含了操作系統其餘的大部分，比如 GCC，一款支持多種語言的通用編譯器；GNU Emacs，一款可擴展的並且有很多很多特性的文本編輯器；GNOME，

GNU 桌面環境；GNU libc，一個程序庫，除內核外所有其他程序與內核通訊時都必須使用的庫；以及 Bash，GNU 命令行解釋器可以讀取你的命令行。這些程序很多都是早期由理查德·斯托曼在 GNU 工程裡開發，併成為現代 GNU/Linux 操作系統的組成部分。

重要的是理解即使你沒有修改程序源代碼或直接使用所有這些工具的能力，找到一個可以做到的人也是相對容易的。因此，有源代碼的程序你就有權力去修改、修復、定製和學習編程——而如果得不到源代碼就沒有這些權力。源代碼是讓一個軟件變自由的必要條件之一，而其他必要條件可從本書的哲學和理想中找到答案。

附錄 B

不同語言對“自由軟件”和 “免費軟件”的翻譯

Copyright © 1999, 2000, 2004, 2006–2015 自由軟件基金會 更新
的翻譯列表可參見 <http://gnu.org/philosophy/fs-translations.html>。增加新的語種翻譯可聯繫 web-translators@gnu.org。

以下列出的是對“自由軟件”和“免費軟件”不同語種的翻譯，拆分成不同的列以便比較。個別條目中的拉丁字母表示的是音譯（相關位置增加了元音）。

	語言名稱	自由軟件	免費軟件
en	英語	free software	gratis software
af	南非荷蘭語	vrye sagteware	gratis sagteware
ar	阿拉伯語	برمجيات حرة (<i>barmajiyat ḥorrah</i>)	

	語言名稱	自由軟件	免費軟件
be	白俄羅斯語	свабоднае праграмае забесьпячэньне (<i>svabodnae programae zabes'pjachen'ne</i>)	
bg	保加利亞語	свободен софтуер (<i>svoboden softuer</i>)	безплатен софтуер (<i>bezplaten softuer</i>)
bn	孟加拉語	স্বাধীন সফটওয়্যার (<i>swadhin software</i>)	
ca	加泰羅尼亞語	programari lliure	programari gratuït
cs	捷克語	svobodný software	bezplatný software
cy	威爾士語	meddalwedd	rydd
da	丹麥語	fri software 或 frit programmel	gratis software
de	德語	freie Software	Gratis-Software 或 kostenlose Software
el	希臘語	ελεύθερο λογισμικό (<i>elefthero logismiko</i>)	δωρεάν λογισμικό (<i>dorean logismiko</i>)
eo	世界語	libera programaro 或 programo	
eu	巴斯克語	software librea	doako softwarea
es	西班牙語	software libre	software gratuito
et	愛沙尼亞語	vaba tarkvara	tasuta tarkvara
fa	波斯語	نرم افزار آزاد (<i>narmafzar azad</i>)	نرم افزار رایگان (<i>narmafzar raygan</i>)
fi	芬蘭語	vapaa ohjelmisto	ilmainen ohjelmisto
fr	法語	logiciel libre	logiciel gratuit

	語言名稱	自由軟件	免費軟件
ga	愛爾蘭語	saorbhogearraí	bogearraí saora in aisce
he	希伯來語	חופשית תוכנה (<i>tochna chofshit</i>)	חינמית תוכנה (<i>tochna chinamit</i>)
hi	印地語	मुक्त सॉफ्टवेयर (<i>mukt software</i>)	मुफ्त सॉफ्टवेयर (<i>muft software</i>)
hr	克羅地亞語	slobodan softver	besplatani softver
hu	匈牙利語	szabad szoftver	ingyenes szoftver 或 ingyen szoftver
hy	亞美尼亞語	ազատ ծրագիր/ծրագրեր (<i>azat tsragir/tsragrer</i>)	
ia	國際語	libere programmation 或 libere programmario	
id	印度尼西亞語	perangkat lunak bebas	
io	伊多	libera programaro	
is	冰島語	frjáls hugbúnaður	
it	意大利語	software libero	software gratuito
ja	日語	自由ソフトウェア (<i>jiyū-sofutouea</i>)	無料ソフトウェア (<i>muryō-sofutouea</i>)
ka	格魯吉亞語	თავისუფალი პროგრამები (<i>tavisupali programebi</i>)	უფასო პროგრამები (<i>upaso programebi</i>)
ko	韓語	자유 소프트웨어 (<i>ja-yu software</i>)	
lt	立陶宛語	laisva programinė įranga	nemokama programinė įranga

	語言名稱	自由軟件	免費軟件
lv	拉脫維亞語	brīva programmatūra	bezmaksas programmatūra
mk	馬其頓語	слободен софтвер (<i>sloboden softver</i>)	бесплатен софтвер (<i>besplaten softver</i>)
ml	馬拉雅拉姆語	സ്വതന്ത്രസോഫ്റ്റ്‌വെയർ (<i>svatantrasophṭṭveyar</i>)	സൗജന്യസോഫ്റ്റ്‌വെയർ (<i>soujanyaṣophṭṭve- yar</i>)
ms	馬來語	perisian bebas	
nl	荷蘭語	vrije software	gratis software
no	挪威語	fri programvare	
pl	波蘭語	wolne oprogramowanie	darmowe oprogramowanie
pt	葡萄牙語	software livre	
ro	羅馬尼亞語	programe libere	programe gratuite
ru	俄語	свободные программы (<i>svobodnie programmi</i>)	бесплатные программы (<i>besplatnie programmi</i>)
sc	撒丁語	software liberu	
si	僧伽羅語	නිදහස් මෘදුකාංග (<i>nidahas mṛdukāṅga</i>)	
sk	斯洛伐克語	slobodný softvér	
sl	斯洛文尼亞語	prosto programje	
sq	阿爾巴尼亞語	software i lirë	software falas
sr	塞爾維亞	слободни софтвер 或 slobodni softver	бесплатни софтвер 或 besplatni softver
sv	瑞典語	fri programvara 或 fri mjukvara	

	語言名稱	自由軟件	免費軟件
sw	斯瓦希里語	software huru 或 programu huru za kompyuta	
ta	泰米爾文	கட்டற்ற மென்பொருள் (<i>kaṭṭarra menpoṇā!</i>)	இலவச மென்பொருள் (<i>illavasa menporul</i>)
th	泰語	ซอฟต์แวส์ (<i>sofotwerseri</i>)	
tl	他加祿語（菲律賓）	malayang software	
tr	土耳其語	özgür yazılım	
uk	烏克蘭語	вільне програмне забезпечення (<i>vil'ne prohramne zabezpechennja</i>)	
ur	烏爾都語	آزاد سافٹ ویئر (<i>azad software</i>)	مفت سافٹ ویئر (<i>muft software</i>)
vi	越南語	phần mềm tự do	
zh-cn	中文（簡體）	自由軟件 (<i>zi-you ruan-jian</i>)	免費軟件 (<i>mian-fei ruan-jian</i>)
zh-tw	中文（繁體）	自由軟體 (<i>zih-yo</i>)	免費軟體 (<i>mien-fei</i>)
zu	祖魯語	isoftware ekhululekile	

附錄 C

自由軟件之歌

Copyright © 2010 Richard Stallman. 理查德·斯托曼於 1991 年寫出此歌詞。

《自由軟件之歌》取材自保加利亞民歌《Sadi moma bela loza》的旋律。可以到這裡聆聽由保加利亞樂器演奏的傳統民歌版自由軟件之歌 <http://gnu.org/music/FreeSWSong.ogg>。

這首歌是 7/8 拍的；這種不尋常的不均勻古怪節奏常常被看成是一種錯誤。該部分可拆分為三個單拍子，以慢—快—快或 3—2—2 拍子劃分。保加利亞音樂的這種節奏，經常可以拉長，有些音樂家分析這首歌曲可以替代為 3—2—3 拍子；然而，最後的 3 拍並不如第一個 3 拍那麼長。蒐集並教授這種舞蹈的伊夫·莫羅，傾向於 7 拍節奏。

譯者注：讀者可下載 *Musescore* 格式的樂譜文件¹，通過自由的樂譜軟件 *Musescore*² 聆聽。

¹樂譜文件下載：<https://github.com/beijinglug/fsfs-zh/blob/master/docs/song-book-jutta-scrunch-crop-zh.mscx>

²可前往其官方網站下載 <https://musescore.org/>。GNU/Linux 發行版可通過包管理器搜索 *musescore* 來安裝。

自由軟件之歌

加入我們分享軟件，得自由，黑客們，
 貪婪之人家財萬貫，確實如此，黑客們，
 充實我們的自由軟件，齊努力，黑客們，
 加入我們分享軟件，得自由，黑客們，

5
 得自由；加入我們分享軟件，
 確實如此；他不幫助他的鄰居，
 齊努力；拋棄那些肮脏授權，
 得自由；加入我們分享軟件，

9
 得自由，黑客們，得自由。
 這不好，黑客們，很不。
 到永遠，黑客們，到永遠。
 得自由，黑客們，得自由。

圖 C.1: 自由軟件之歌樂譜