

EMACS 26.2 中文手册

作者: YuZJ

编译日期: September 16, 2019

【简介】



【版本号】

简明目录

1 屏幕的组织结构	7
A GNU GENERAL PUBLIC LICENSE	9
1.1 Preamble	9
1.2 TERMS AND CONDITIONS	10
1.3 How to Apply These Terms to Your New Programs	16
B GNU 通用公共许可协议	17
2.1 译者声明	17
2.2 引言	17
2.3 关于复制、分发和修改的术语和条件	18
2.4 附录：如何将上述条款应用到你的新程序	22
C The GNU Manifesto	23
3.1 What's GNU? Gnu's Not Unix!	23
3.2 Why I Must Write GNU	24
D GNU 宣言	29
4.1 GNU 为何? GNU 并非 UNIX!	29
4.2 为什么我必须编写 GNU	29
4.3 为什么 GNU 将会兼容 Unix	30
4.4 如何获取 GNU	30
4.5 为什么许多程序员想要提供帮助	30
4.6 你该如何做出贡献	30
4.7 为什么所有计算机用户都会受益	30
4.8 一些容易驳斥的、反对 GNU 目标的观点	31

目录

1 屏幕的组织结构	7	2.3.1 ○、定义	18
A GNU GENERAL PUBLIC LICENSE	9	2.3.2 一、源码	18
1.1 Preamble	9	2.3.3 二、基本许可	18
1.2 TERMS AND CONDITIONS	10	2.3.4 三、保护用户的合法权益免受反破解 法限制	18
0. Definitions.	10	2.3.5 四、转发完整副本	19
1. Source Code.	10	2.3.6 五、转发修改过的源码版本	19
2. Basic Permissions.	10	2.3.7 六、以非源码形式转发	19
3. Protecting Users' Legal Rights From Anti-Circumvention Law.	11	2.3.8 七、附加条款	20
4. Conveying Verbatim Copies.	11	2.3.9 八、终止授权	20
5. Conveying Modified Source Versions.	11	2.3.10 九、持有副本无需接受协议	20
6. Conveying Non-Source Forms.	11	2.3.11 十、对下游接收者的自动授权	20
7. Additional Terms.	12	2.3.12 十一、专利	21
8. Termination.	13	2.3.13 十二、不得牺牲他人的自由	21
9. Acceptance Not Required for Hav- ing Copies.	13	2.3.14 十三、和 GNU Affero 通用公共许可 协议一起使用	21
10. Automatic Licensing of Down- stream Recipients.	13	2.3.15 十四、本协议的修订版	21
11. Patents.	14	2.3.16 十五、不提供品质担保	21
12. No Surrender of Others' Freedom.	14	2.3.17 十六、责任范围	22
13. Use with the GNU Affero General Public License.	15	2.3.18 十七、第十五条和第十六条的解释	22
14. Revised Versions of this License.	15	2.4 附录：如何将上述条款应用到你的新程序	22
15. Disclaimer of Warranty.	15		
16. Limitation of Liability.	15		
17. Interpretation of Sections 15 and 16.	15		
1.3 How to Apply These Terms to Your New Pro- grams	16		
B GNU 通用公共许可协议	17		
2.1 译者声明	17		
2.2 引言	17		
2.3 关于复制、分发和修改的术语和条件	18		
		C The GNU Manifesto	23
		3.1 What's GNU? Gnu's Not Unix!	23
		3.2 Why I Must Write GNU	24
		D GNU 宣言	29
		4.1 GNU 为何? GNU 并非 UNIX!	29
		4.2 为什么我必须编写 GNU	29
		4.3 为什么 GNU 将会兼容 Unix	30
		4.4 如何获取 GNU	30
		4.5 为什么许多程序员想要提供帮助	30
		4.6 你该如何做出贡献	30
		4.7 为什么所有计算机用户都会受益	30
		4.8 一些容易驳斥的、反对 GNU 目标的观点	31

屏幕的组织结构

在一个图形界面（例如，使用 X Windows 系统的 GNU/Linux 操作系统）中，Emacs 占用了一个图形窗口（Window）。在一个文本终端中，Emacs 占用了一整个终端屏幕。我们将会使用“窗体”（Frame）表示 Emacs 占用的图形窗体或终端屏幕。Emacs 在这两种窗体下表现十分相似¹。一般情况下 Emacs 在启动时创建一个窗体，但是你可以按照你的意愿创建另外的窗体（参见第 18 章【窗体】）。

Emacs 窗体包括多个区分明显的区域。在窗体的最顶端是菜单栏（Menu Bar），它允许你通过一系列菜单访问命令。在一个图形界面上，菜单栏下方是工具栏（Tool Bar）。它由一系列图标构成，当你单击它是它将执行编辑命令。回显区（Echo Area）窗体的最底端，用于显示 Emacs 的提示信息或者当 Emacs 需要你输入时键入信息。

在一个 Emacs 窗体中，位于工具栏（如果存在的话）以及回显区的大块区域称为窗格（Window）。此后我们将会使用“窗格”来表示这个意思。图形界面系统一般会使用“窗口”（Window）来表示不同的意思；但是，正如已经声明的一样，我们将称这些图形界面窗口为“窗体”（Frame）。

缓冲区（Buffer）——文本、图像以及其他你正在编辑或浏览的东西所显示的地方——位于 Emacs 窗口。在一个图形界面上，窗格²拥有一个纵向的滚动条用于滚动显示整个缓冲区。窗格的最底端一栏（在终端界面下是一行文字）称为状态栏（Mode Line），用于显示关于缓冲区中正在进行的操作的信息，比如说是否有未保存的改变，正在使用的编辑模式（Editing Mode），当前的行号等等。

通常情况下，当你启动 Emacs 的时候，一个窗体中只有一个窗格。但是，你可以将这个窗格水平或竖直地分为多个窗格，它们各自独立地显示一个缓冲区（参见第 17 章【窗格】）。

在任何时候，有且仅有一个窗格是活动窗格（Selected Window）。在一个图形界面下，活动窗格显示一个更加明显的光标（Crusor）（一般是闪动的黑色方块“■”）；其它窗格显示一个不明显的光标（一般是空心方块“□”）。终端界面只将在活动窗格显示一个光标。在活动窗格显示的缓冲区称为活动缓冲区（“Selected Buffer”）并且就是正在被编辑的缓冲区。大部分 Emacs 命令含蓄地作用于活动缓冲区；在非活动的缓冲区中显示的文字仅为参考而显示。如果你使用多窗体的图形界面，选择一个窗体再选择一个窗格。

¹译注：虽然我不这么认为

²译注：是“Window”。此后窗格一律称“Window”，窗体一律称“Frame”。窗口？那是什么东西？

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright ©2007 Free Software Foundation, Inc. <<https://fsf.org/>>

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

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

第 1.2 章

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program.

The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work. A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product. “Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when

you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

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, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction,

each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying

from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public 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.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

第 1.3 章

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box” .

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<https://www.gnu.org/licenses/why-not-lgpl.html>>.

GNU 通用公共许可协议

第三版, 2007 年 6 月 29 日

版权所有 ©2007 自由软件基金会 <<http://fsf.org/>>

任何人皆可复制和发布本协议的完整副本, 但不得修改

译者声明

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 does that. However, we hope that this translation will help Chinese speakers understand the GNU GPL better.

这是 GNU 通用公共许可协议的一份非官方中文翻译, 并非自由软件基金会所发表, 不适用于使用 GNU 通用公共许可协议发布的软件的法律声明——只有 GNU 通用公共许可协议英文原版才具有法律效力。不过我们希望本翻译能够帮助中文读者更好地理解 GNU 通用公共许可协议。

You may publish this translation, modified or unmodified, only under the terms at <https://www.gnu.org/licenses/translations.html>.

引言

GNU 通用公共许可协议是一份面向软件及其他类型作品的, 自由的版权共享协议。

就多数软件而言, 许可协议被设计用于剥夺你分享和修改软件的自由。相反, GNU 通用公共许可协议力图保障你分享和修改某程序全部版本的权利——确保自由软件对其用户来说是自由的。我们自由软件基金会将 GNU 通用公共许可协议用于我们的大多数软件, 并为一些其他作品的作者效仿。你也可以将本协议用于你的程序。

所谓自由软件, 强调自由, 而非免费。本 GNU 通用公共许可协议设计用于确保你享有分发自由软件的自由 (你可以为此服务收费), 确保你可以在需要的时候获得这些软件的源码, 确保你可以修改这些软件或者在新的自由软件中复用其中某些片段, 并且确保你在这方面享有知情权。

为保障你的权益, 我们需要作一些限定: 禁止任何人否认你的上述权利, 或者要求你放弃它们。因此, 当你分发或修改这些软件时, 你有一定的责任——尊重他人的自由。如果你分发这种程序的副本, 无论收费还是免费, 你必须给予与你同等的权利。你还要确保他们也能收到源码并了解他们的权利。

采用 GNU 通用公共许可协议的开发者通过两步保障你的权益: 其一, 申明软件的版权; 其二, 通过本协议使你可以合法地复制、分发和修改该软件。

为了保护每一位作者和开发者, GNU 通用公共许可协议指明一点: 自由软件并没有品质担保。为用户和作者双方着想, GNU 通用公共许可协议要求修改版必须有标记, 以免其问题被错误地归到先前版本的作者身上。

某些设备设计成拒绝用户安装运行修改过的软件, 但厂商不受限。这和我们保护用户享有修改软件的自由的宗旨存在根本性矛盾。该滥用协议的模式出现于个人用品领域, 这恰是最不可接受的。因此, 我们设计了这版 GNU 通用公共许可协议来禁止这类产品。如果此类问题在其他领域涌现, 我们时刻准备着在将来的版本中把规定扩展到相应领域, 以保护用户的自由。

最后, 每个程序都持续受到软件专利的威胁。政府不应该允许专利限制通用计算机软件的开发和应用, 在做不到这点时, 我们希望避免专利应用有效地使自由软件私有化的危险。就此, GNU 通用公共许可协议保证专利不能使程序非自由化。

下文是关于复制、分发和修改的严谨描述和实施条件。

关于复制、分发和修改的术语和条件

第 2.3 章

O、定义

第 2.3.1 节

“本协议”指 GNU 通用公共许可协议第三版。

“版权”也指适用于诸如半导体掩模的其他类型作品的类似法律。

“本程序”指任何在本协议保护下的有版权的作品。每个许可获得者称作“你”。“许可获得者”和“接收者”可以是个人或组织。

“修改”一个作品指需要版权许可的复制及对作品全面的或部分的改编行为，有别于制作副本。所产生的作品称作前作的“修改版”，或“基于”前作的作品。

“受保护作品”指程序或其派生作品。

“传播”作品指那些未经许可就会在适用版权法律下构成直接或间接侵权的行为，不包括在计算机上运行和私下的修改。传播包括复制、分发（无论修改与否）、向公众公开，以及在某些国家的其他行为。“转发”作品指让他方能够制作或者接收副本的行为。仅仅通过计算机网络和用户交互，没有传输副本，则不算转发。

一个显示“适当的法律声明”的交互式用户界面应包括一个便捷而醒目的可视化特性：(1) 显示适当的版权声明；(2) 告知用户没有品质担保（提供了品质担保的情况除外），许可获得者可以在本协议约束下转发该作品，及查看本协议副本的途径。如果该界面提供一个命令列表，如菜单，其表项应符合上述规范。

第 2.3.2 节

一、源码

作品的源码指其可修改的首选形式，目标码指所有其他形式。

“标准接口”指标准化组织定义的官方标准中的接口，或针为某种编程语言设定的接口中为开发者广泛使用的接口。

可执行作品中的“系统库”不是指整个程序，而是涵盖此等内容：(a) 以通常形式和主部件打包到一起却并非后者一部分，且(b) 仅为和主部件一起使作品可用或实现某些已有公开实现源码的接口。“主部件”在这里指可执行作品运行依赖的操作系统（如果存在）的必要部件（内核、窗口系统等），生成该作品的编译器，或运行所需的目标码解释器。

目标码形式的作品中“相应的源码”指所有修改作品及生成、安装、运行（对可执行作品而言）目标码所需的源码，包括控制上述行为的脚本。可是，其中不包括系统库、通用工具、未修改直接用于支持上述行为却不是该作品一部分的通常可得的自由软件。例如，相应的源码包含配合作品源文件的接口定义，以及共享库和作品专门依赖的动态链接子程序的源码。这里的依赖体现为频密的数据交换或者该子程序和作品其他部分的控制流切换。

相应的源码不必包含那些用户可以通过源码其他部分自动生成的内容。

源码形式作品的相应源码即其本身。

第 2.3.3 节

二、基本许可

本协议的一切授权都是对本程序的版权而言的，并且在所述条件都满足时不可撤销。本协议明确批准你不受限制地运行本程序的未修改版本。受保护作品的运行输出，仅当其内容构成一个受保护作品时，才会为本协议所约束。如版权法所赋予，本协议承认你正当使用或与之等价的权利。

只要你获得的许可仍有效，你可以制作、运行和传播那些你并不转发的受保护作品。只要你遵守本协议中关于转发你不占有版权的材料的条款，你可以向他人转发，仅仅以求对方为你做定制或向你提供运行这些作品的工具。那些为你制作或运行这些受保护作品的人，应该在你的指引和控制下，谨代表你工作，即禁止他们在双方关系之外制作任何你提供的受版权保护材料的副本。

仅当满足后文所述条件时，其他各种情况下的转发才是允许的。不允许再授权行为，而第十条的存在使再授权变得没有必要。

第 2.3.4 节

三、保护用户的合法权益免受反破解法限制

在任何满足 1996 年 12 月 20 日通过的 WIPO 版权条约第 11 章要求的法律，或类似的禁止或限制技术手段破解的法律下，受保护作品不应该视为有效技术手段的一部分。

当你转发一个受保护作品时，你将失去任何通过法律途径限制技术手段破解的权力，乃至通过行使本协议所予权利实现的破解。你即已表明无心通过限制用户操作或修改受保护作品来确保你或第三方关于禁止技术手段破解的法定权利。

第 2.3.5 节

四、转发完整副本

你可以通过任何媒介发布你接收到的本程序的完整源码副本，但要做到：为每一个副本醒目而恰当地发布版权；完整地保留关于本协议及按第七条加入的非许可性条款；完整地保留免责声明；给接收者附上一份本协议的副本。

你可以免费或收费转发，也可以选择提供技术支持或品质担保以换取收入。

第 2.3.6 节

五、转发修改过的源码版本

你可以以源码形式转发基于本程序的作品或修改的内容，除满足第四条外还需要满足以下几点要求：a) 该作品必须带有醒目的修改声明及相应的日期。

b) 该作品必须带有醒目的声明，指出其在本协议及任何符合第七条的附加条件下发布。这个要求修正了第四条关于“完整保留”的内容。

c) 你必须按照本协议将该作品整体向想要获得许可的人授权，本协议及符合第七条的附加条款就此适用于整个作品，即其每一部分，不管如何建包。本协议不允许以其他形式授权该作品，但如果你收到别的许可则另当别论。

d) 如果该作品有交互式用户界面，则其必须显示适当的法律声明。然而，当本程序有交互式用户界面却不显示适当的法律声明时，你的作品也不必。

一个在存储或分发媒介上的受保护作品和其他分离的单体作品的联合作品，在既不是该受保护作品的自然扩展，也不以构筑更大的程序为目的，并且自身及其产生的版权并非用于限制单体作品给予联合作品用户的访问及其他合法权利时，称为“聚合体”。在聚合作品中包含受保护作品并不会使本协议影响聚合作品的其他部分。

第 2.3.7 节

六、以非源码形式转发

你可以如第四条和第五条所述那样以目标码形式转发受保护作品，同时在本协议规范下以如下方式之一转发机器可读的对应源码：

a) 目标码通过实体产品（涵盖某种实体分发媒介）转发时，通过常用于软件交换的耐用型实体媒介随同转发相应的源码。

b) 目标码通过实体产品（涵盖某种实体分发媒介）转发时，伴以具有至少三年且与售后服务等长有效期的书面承诺，给予目标码的持有者：(1) 包含产品全部软件的相应源码的常用于软件交换的耐用型实体媒介，且收费不超过其合理的转发成本；或者 (2) 通过网络免费获得相应源码的途径。

c) 单独转发目标码时，伴以提供源码的书面承诺。本选项仅在你收到目标码及 b 项形式的承诺的情况下可选。

d) 通过在指定地点提供目标码获取服务（无论是否收费）的形式转发目标码时，在同一地点以同样的方式提供对等的源码获取服务，并不得额外收费。你不以要求接收者在复制目标码的同时复制源码。如果提供目标码复制的地点为网络服务器，相应的源码可以提供在另一个支持相同复制功能的服务器上（由你或者第三方运营），不过你要在目标码处指出相应源码的确切路径。不管你用什么源码服务器，你有义务要确保持续可用以满足这些要求。

e) 通过点对点传输转发目标码时，告知其他节点目标码和源码在何处以 d 项形式向大众免费提供。

“面向用户的产品”指 (1) “消费品”，即个人、家庭或日常用途的个人有形财产；或者 (2) 面向社会团体设计或销售，却落入居家之物。在判断一款产品是否消费品时，争议案例的判断将向利于扩大保护靠拢。就特定用户接收到特定产品而言，“正常使用”指对此类产品的典型或一般使用，不管该用户的身份，该用户对该产品的实际用法，以及该产品的预期用法。无论产品是否实质上具有商业上的，工业上的，及非面向消费者的用法，它都视为消费品，除非以上用法代表了它唯一的重要使用模式。

“安装信息”对面向用户的产品而言，指基于修改过的源码安装运行该产品中的受保护作品的修改版所需的方法、流程、认证码及其他信息。这些信息必须足以保证修改过的目标码不会仅仅因为被修改过而不能继续工作。

如果你根据本条在，或随，或针对一款面向用户的产品，以目标码形式转发某作品，且转发体现于该产品的所有权和使用权永久或者在一定时期内转让予接收者的过程（无论其有何特点），根据本条进行的源码转发必须伴有安装信息。不过，如果你和第三方都没有保留在该产品上安装修改后的目标码的能力（如作品安装在 ROM 上），这项要求不成立。

要求提供安装信息并不要求为修改或安装的作品，以及其载体产品继续提供技术支持、品质担保和升级。当修改本身对网络运行有实质上的负面影响，或违背了网络通信协议和规则时，可以拒绝其联网。

根据本条发布的源码及安装信息，必须以公共的文件格式（并且存在可用的空开源码的处理工具）存在，同时不得对解压、阅读和复制设置任何密码。

第 2.3.8 节**七、附加条款**

“附加许可”用于补充本协议，以允许一些例外情况。合乎适用法律的对整个程序适用的附加许可，应该被视为本协议的内容。如果附加许可作用于程序的某部分，则该部分受此附加许可约束，而其他部分不受其影响。

当你转发本程序时，你可以选择性删除副本或其部分的附加条款。（附加条款可以写明在某些情况下要求你修改时删除该条款。）在你拥有或可授予恰当版权许可的受保护作品中，你可以在你添加的材料上附加许可。

尽管已存在本协议的其他条款，对你添加到受保护作品的材料，你可以（如果你获得该材料版权持有人的授权）以下条款补充本协议：

- a) 表示不提供品质担保或有超出十五、十六条的责任。
- b) 要求在此材料中或在适当的法律声明中保留特定的合理法律声明或创作印记。
- c) 禁止误传材料的起源，或要求合理标示修改以别于原版。
- d) 限制以宣传为目的使用该材料的作者或授权人的名号。
- e) 降低约束以便赋予在商标法下使用商品名、商品标识及服务标识。
- f) 要求任何转发该材料（或其修改版）并对接收者提供契约性责任许诺的人，保证这种许诺不会给作者或授权人带来连带责任。

此外的非许可性附加条款都被视作第十条所说的“进一步的限制”。如果你接收到的程序或其部分，声称受本协议约束，却补充了这种进一步的限制条款，你可以去掉它们。如果某许可协议包含进一步的限制条款，但允许通过本协议再授权或转发，你可以通过本协议再授权或转发加入了受前协议管理的材料，不过要同时移除上述条款。

如果你根据本条向受保护作品添加了调控，你必须在相关的源文件中加入对应的声明，或者指出哪里可以找到它们。

附加条款，不管是许可性的还是非许可性的，可以以独立的书面协议出现，也可以声明为例外情况，两种做法都可以实现上述要求。

第 2.3.9 节**八、终止授权**

除非在本协议明确授权下，你不得传播或修改受保护作品。其他任何传播或修改受保护作品的企图都是无效的，并将自动中止你通过本协议获得的权利（包括第十一条第 3 段中提到的专利授权）。

然而，当你不再违反本协议时，你从特定版权持有人处获得的授权恢复：(1) 暂时恢复，直到版权持有人明确终止；(2) 永久恢复，如果版权持有人没能在 60 天内以合理的方式指出你的侵权行为。

再者，如果你第一次收到了特定版权持有人关于你违反本协议（对任意作品）的通告，且在收到通告后 30 天内改正，那你可以继续享此有授权。

当你享有的权利如本条所述被中止时，已经从你那根据本协议获得授权的他方的权利不会因此中止。在你的权利恢复之前，你没有资格凭第十条获得同一材料的授权。

第 2.3.10 节**九、持有副本无需接受协议**

你不必为接收或运行本程序而接受本协议。类似的，仅仅因点对点传输接收到副本引发的对受保护作品的辅助性传播，并不要求接受本协议。但是，除本协议外没有什么可以授权你传播或修改任何受保护作品。如果你不接受本协议，这些行为就侵犯了版权。因此，一旦修改和传播一个受保护作品，就表明你接受本协议。

第 2.3.11 节**十、对下游接收者的自动授权**

每当你转发一个受保护作品，其接收者自动获得来自初始授权人的授权，依照本协议可以运行、修改和传播此作。你没有要求第三方遵守该协议的义务。

“实体事务”指转移一个组织的控制权或全部资产、或拆分或合并组织的事务。如果实体事务导致一个受保护作品的传播，则事务中各收到作品副本方，都有获得前利益相关者享有或可以如前段所述提供的对该作品的任何授权，以及从前利益相关者处获得并拥有相应的源码的权利，如果前利益相关者享有或可以通过合理的努力获得此源码。

你不可以对本协议所授权利的行使施以进一步的限制。例如，你不可以索要授权费或版税，或就行使本协议所授权利征收其他费用；你也不能发起诉讼（包括交互诉讼和反诉），宣称制作、使用、零售、批发、引进本程序或其部分的行为侵犯了任何专利。

第 2.3.12 节**十一、专利**

“贡献人”指通过本协议对本程序或其派生作品进行使用认证的版权持有人。授权作品成为贡献人的“贡献者版”。

贡献人的“实质专利权限”指其拥有或掌控的，无论是已获得的还是将获得的全部专利权限中，可能被通过某种本协议允许的方式制作、使用或销售其贡献者版作品的行为侵犯的部分，不包括仅有修改其贡献者版作品才构成侵犯的部分。“掌控”所指包括享有和本协议相一致的专利再授权的权利。

每位贡献人皆其就实质专利权限，授予你一份全球有效的免版税的非独占专利许可，以制作、使用、零售、批发、引进，及运行、修改、传播其贡献者版的内容。

在以下三段中，“专利许可”指通过任何方式明确表达的不行使专利权（如对使用专利的明确许可和不起诉专利侵权的契约）的协议或承诺。对某方“授予”专利许可，指这种不对其行使专利权的协议或承诺。

如果你转发的受保护作品已知依赖于某专利，而其相应的源码并不是任何人都能根据本协议从网上或其他地方免费获得，那你必须(1)以上述方式提供相应的源码；或者(2)放弃从该程序的专利许可中获得利益；或者(3)以某种和本协议相一致的方式将专利许可扩展到下游接收者。“已知依赖于”指你实际上知道若没有专利许可，你在某国家转发受保护作品的行为，或者接收者在某国家使用受保护作品的行为，会侵犯一项或多项该国认定的专利，而这些专利你有理由相信它们的有效性。

如果根据一项事务或安排，抑或与之相关，你转发某受保护作品，或通过促成其转手以实现传播，并且该作品的接收方授予专利许可，以使指可以使用、传播、修改或转发该作品的特定副本，则此等专利许可将自动延伸及每一个收到该作品或其派生作品的人。

如果某专利在其涵盖范围内，不包含本协议专门赋予的一项或多项权利，禁止行使它们或以不行使它们为前提，则该专利是“歧视性”的。如果你和软件发布行业的第三方有合作，合作要求你就转发受保护作品的情况向其付费，并授予作品接收方歧视性专利，而且该专利(a)与你转发的副本（或在此基础上制作的副本）有关，或针对包含该受保护作品的产品或联合作品，你不得转发本程序，除非参加此项合作或取得该专利早于2007年3月28日。

本协议的任何部分不应被解释成在排斥或限制任何暗含的授权，或者其他在适用法律下对抗侵权的措施。

第 2.3.13 节**十二、不得牺牲他人的自由**

即便你面临与本协议条款冲突的条件（来自于法庭要求、协议或其他），那也不能成为你违背本协议的理由。倘若你不能在转发受保护作品时同时满足本协议和其他文件的要求，你就不能转发本程序。例如，当你同意了某些要求你就再转发问题向你的转发对象收取版税的条款时，唯一能同时满足它和本协议要求的做法便是不转发本程序。

第 2.3.14 节**十三、和 GNU Affero 通用公共许可协议一起使用**

尽管已存在本协议的一些条款，你可以将任何受保护作品与以GNU Affero通用公共许可协议管理的作品关联或组合成一个联合作品，并转发。本协议对其中的受保护作品部分仍然有效，但GNU Affero通用公共许可协议第十三条的关于网络交互的特别要求适用于整个联合作品。

第 2.3.15 节**十四、本协议的修订版**

自由软件联盟可能会不定时发布GNU通用公共许可协议的修订版或新版。新版将秉承当前版本的精神，但对问题或事项的描述细节不尽相同。

每一版都会有不同的版本号，如果本程序指定其使用的GNU通用公共许可协议的版本“或任何更新的版本”，你可以选择遵守该版本或者任何更新的版本的条款。如果本程序没有指定协议版本，你可以选用自由软件联盟发布的任意版本的GNU通用公共许可协议。

如果本程序指定代理来决定将来那个GNU通用公共许可协议版本适用，则该代理的公开声明将指导你选择协议版本。新的版本可能会给予你额外或不同的许可。但是，任何作者或版权持有人的义务，不会因为你选择新的版本而增加。

第 2.3.16 节**十五、不提供品质担保**

本程序在适用法律范围内不提供品质担保。除非另作书面声明，版权持有人及其他程序提供者“概”不提供任何显式或隐式的品质担保，品质担保所指包括而不仅限于有经济价值和适合特定用途的保证。全部风险，如程序的质量和性能问题，皆由你承担。若程序出现缺陷，你将承担所有必要的修复和更正服务的费用。

十六、责任范围

第 2.3.17 节

十七、第十五条和第十六条的解释

第 2.3.18 节

除非适用法律或书面协议要求，任何版权持有人或本程序按本协议可能存在的第三方修改和再发布者，都不对你的损失负有责任，包括由于使用或者不能使用本程序造成的任何一般的、特殊的、偶发的或重大的损失（包括而不仅限于数据丢失、数据失真、你或第三方的后续损失、其他程序无法与本程序协同运作），即使那些人声称会对此负责。

附录：如何将上述条款应用到你的新程序

第 2.4 章

如果你开发了一个新程序，并希望它能最大限度地为公众所使用，最好的办法是将其作为自由软件，以使每个人都能在本协议约束下对其再发布及修改。

为此，请在附上以下声明。最安全的做法是将其附在每份源码的开头，以便于最有效地传递免责信息。同时，每个文件至少包含一处“版权”声明和一个协议全文的链接。

<用一行来标明程序名及其作用>

版权所有 (C) <年份> <作者姓名>

本程序为自由软件，在自由软件联盟发布的 GNU 通用公共许可协议的约束下，你可以对其进行再发布及修改。协议版本为第三版或（随你）更新的版本。

我们希望发布的这款程序有用，但不保证，甚至不保证它有经济价值和适合特定用途。详情参见 GNU 通用公共许可协议。

你理当已收到一份 GNU 通用公共许可协议的副本，如果没有，请查阅

<<http://www.gnu.org/licenses/>>

同时提供你的电子邮件地址或传统的邮件联系方式。

如果该程序是交互式的，让它在交互模式下输出类似下面的一段声明：

<程序名> 第 69 版，版权所有 (C) <年份> <作者姓名>

本程序从未提供品质担保，输入'show w' 可查看详情。

这是款自由软件，欢迎你在满足一定条件后对其再发布，输入'show c' 可查看详情。

例子中的命令'show w' 和'show c' 应用于显示 GNU 通用公共许可协议相应的部分。当然你也可以因地制宜地选用别的方式，对图形界面程序可以用“关于”菜单。

如果你之上存在雇主（你是码农）或校方，你还应当让他们在必要时为此程序签署放弃版权声明。详情参见 <<http://www.gnu.org/licenses/>>。

本 GNU 通用公共许可协议不允许把你的程序并入私有程序。如果你的程序是某种库，且你想允许它被私有程序链接而使之更有用，请使用 GNU 较宽松通用公共许可协议。决定前请先查阅 <<http://www.gnu.org/philosophy/why-not-lgpl.html>>。

翻译：阮坤良 <peterrk@pku.edu.cn> 参考：[繁体中文译本] <https://www.gnu.org/licenses/translations.html>

The GNU Manifesto

The GNU Manifesto (which appears below) was written by Richard Stallman in 1985 to ask for support in developing the GNU operating system. Part of the text was taken from the original announcement of 1983. Through 1987, it was updated in minor ways to account for developments; since then, it seems best to leave it unchanged.

Since that time, we have learned about certain common misunderstandings that different wording could help avoid. Footnotes added since 1993 help clarify these points.

If you want to install the GNU/Linux system, we recommend you use one of the 100% free software GNU/Linux distributions. For how to contribute, see <http://www.gnu.org/help>.

The GNU Project is part of the Free Software Movement, a campaign for freedom for users of software. It is a mistake to associate GNU with the term “open source”—that term was coined in 1998 by people who disagree with the Free Software Movement’s ethical values. They use it to promote an amoral approach to the same field.

What’s GNU? Gnu’s Not Unix!

GNU, which stands for Gnu’s Not Unix, is the name for the complete Unix-compatible software system which I am writing so that I can give it away free to everyone who can use it.¹ Several other volunteers are helping me. Contributions of time, money, programs and equipment are greatly needed.

So far we have an Emacs text editor with Lisp for writing editor commands, a source level debugger, a yacc-compatible parser generator, a linker, and around 35 utilities. A shell (command interpreter) is nearly completed. A new portable optimizing C compiler has compiled itself and may be released this year. An initial kernel exists but many more features are needed to emulate Unix. When the kernel and compiler are finished, it will be possible to distribute a GNU system suitable for program development. We will use TeX as our text formatter, but an nroff is being worked on. We will use the free, portable X Window System as well. After this we will add a portable Common Lisp, an Empire game, a spreadsheet, and hundreds of other things, plus online documentation. We hope to supply, eventually, everything useful that normally comes with a Unix system, and more.

GNU will be able to run Unix programs, but will not be identical to Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer file names, file version numbers, a crashproof file system, file name completion perhaps, terminal-independent display support, and perhaps eventually a Lisp-based window system through which several Lisp programs and ordinary Unix programs can share a screen. Both C and Lisp will be available as system programming languages. We will try to support UUCP, MIT Chaosnet, and Internet protocols for communication.

GNU is aimed initially at machines in the 68000/16000 class with virtual memory, because they are the easiest machines to make it run on. The extra effort to make it run on smaller machines will be left to someone who wants to use it on them.

To avoid horrible confusion, please pronounce the g in the word “GNU” when it is the name of this project.

¹The wording here was careless. The intention was that nobody would have to pay for permission to use the GNU system. But the words don’t make this clear, and people often interpret them as saying that copies of GNU should always be distributed at little or no charge. That was never the intent; later on, the manifesto mentions the possibility of companies providing the service of distribution for a profit. Subsequently I have learned to distinguish carefully between “free” in the sense of freedom and “free” in the sense of price. Free software is software that users have the freedom to distribute and change. Some users may obtain copies at no charge, while others pay to obtain copies—and if the funds help support improving the software, so much the better. The important thing is that everyone who has a copy has the freedom to cooperate with others in using it.

第 3.2 章

Why I Must Write GNU

I consider that the Golden Rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. For years I worked within the Artificial Intelligence Lab to resist such tendencies and other inhospitalities, but eventually they had gone too far: I could not remain in an institution where such things are done for me against my will.

So that I can continue to use computers without dishonor, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free. I have resigned from the AI Lab to deny MIT any legal excuse to prevent me from giving GNU away.(2) Why GNU Will Be Compatible with Unix

Unix is not my ideal system, but it is not too bad. The essential features of Unix seem to be good ones, and I think I can fill in what Unix lacks without spoiling them. And a system compatible with Unix would be convenient for many other people to adopt. How GNU Will Be Available

GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free. Why Many Other Programmers Want to Help

I have found many other programmers who are excited about GNU and want to help.

Many programmers are unhappy about the commercialization of system software. It may enable them to make more money, but it requires them to feel in conflict with other programmers in general rather than feel as comrades. The fundamental act of friendship among programmers is the sharing of programs; marketing arrangements now typically used essentially forbid programmers to treat others as friends. The purchaser of software must choose between friendship and obeying the law. Naturally, many decide that friendship is more important. But those who believe in law often do not feel at ease with either choice. They become cynical and think that programming is just a way of making money.

By working on and using GNU rather than proprietary programs, we can be hospitable to everyone and obey the law. In addition, GNU serves as an example to inspire and a banner to rally others to join us in sharing. This can give us a feeling of harmony which is impossible if we use software that is not free. For about half the programmers I talk to, this is an important happiness that money cannot replace. How You Can Contribute

(Nowadays, for software tasks to work on, see the High Priority Projects list and the GNU Help Wanted list, the general task list for GNU software packages. For other ways to help, see the guide to helping the GNU operating system.)

I am asking computer manufacturers for donations of machines and money. I'm asking individuals for donations of programs and work.

One consequence you can expect if you donate machines is that GNU will run on them at an early date. The machines should be complete, ready to use systems, approved for use in a residential area, and not in need of sophisticated cooling or power.

I have found very many programmers eager to contribute part-time work for GNU. For most projects, such part-time distributed work would be very hard to coordinate; the independently written parts would not work together. But for the particular task of replacing Unix, this problem is absent. A complete Unix system contains hundreds of utility programs, each of which is documented separately. Most interface specifications are fixed by Unix compatibility. If each contributor can write a compatible replacement for a single Unix utility, and make it work properly in place of the original on a Unix system, then these utilities will work right when put together. Even allowing for Murphy to create a few unexpected problems, assembling these components will be a feasible task. (The kernel will require closer communication and will be worked on by a small, tight group.)

If I get donations of money, I may be able to hire a few people full or part time. The salary won't be high by programmers' standards, but I'm looking for people for whom building community spirit is as important as making money. I view this as a way of enabling dedicated people to devote their full energies to working on GNU by sparing them the need to make a living in another way. Why All Computer Users Will Benefit

Once GNU is written, everyone will be able to obtain good system software free, just like air.(3)

This means much more than just saving everyone the price of a Unix license. It means that much wasteful duplication of system programming effort will be avoided. This effort can go instead into advancing the state of the art.

Complete system sources will be available to everyone. As a result, a user who needs changes in the system will always be free to make them himself, or hire any available programmer or company to make them for him. Users will no longer

be at the mercy of one programmer or company which owns the sources and is in sole position to make changes.

Schools will be able to provide a much more educational environment by encouraging all students to study and improve the system code. Harvard's computer lab used to have the policy that no program could be installed on the system if its sources were not on public display, and upheld it by actually refusing to install certain programs. I was very much inspired by this.

Finally, the overhead of considering who owns the system software and what one is or is not entitled to do with it will be lifted.

Arrangements to make people pay for using a program, including licensing of copies, always incur a tremendous cost to society through the cumbersome mechanisms necessary to figure out how much (that is, which programs) a person must pay for. And only a police state can force everyone to obey them. Consider a space station where air must be manufactured at great cost: charging each breather per liter of air may be fair, but wearing the metered gas mask all day and all night is intolerable even if everyone can afford to pay the air bill. And the TV cameras everywhere to see if you ever take the mask off are outrageous. It's better to support the air plant with a head tax and chuck the masks.

Copying all or parts of a program is as natural to a programmer as breathing, and as productive. It ought to be as free. Some Easily Rebuted Objections to GNU's Goals

“Nobody will use it if it is free, because that means they can't rely on any support.”

“You have to charge for the program to pay for providing the support.”

If people would rather pay for GNU plus service than get GNU free without service, a company to provide just service to people who have obtained GNU free ought to be profitable.(4)

We must distinguish between support in the form of real programming work and mere handholding. The former is something one cannot rely on from a software vendor. If your problem is not shared by enough people, the vendor will tell you to get lost.

If your business needs to be able to rely on support, the only way is to have all the necessary sources and tools. Then you can hire any available person to fix your problem; you are not at the mercy of any individual. With Unix, the price of sources puts this out of consideration for most businesses. With GNU this will be easy. It is still possible for there to be no available competent person, but this problem cannot be blamed on distribution arrangements. GNU does not eliminate all the world's problems, only some of them.

Meanwhile, the users who know nothing about computers need handholding: doing things for them which they could easily do themselves but don't know how.

Such services could be provided by companies that sell just handholding and repair service. If it is true that users would rather spend money and get a product with service, they will also be willing to buy the service having got the product free. The service companies will compete in quality and price; users will not be tied to any particular one. Meanwhile, those of us who don't need the service should be able to use the program without paying for the service.

“You cannot reach many people without advertising, and you must charge for the program to support that.”

“It's no use advertising a program people can get free.”

There are various forms of free or very cheap publicity that can be used to inform numbers of computer users about something like GNU. But it may be true that one can reach more microcomputer users with advertising. If this is really so, a business which advertises the service of copying and mailing GNU for a fee ought to be successful enough to pay for its advertising and more. This way, only the users who benefit from the advertising pay for it.

On the other hand, if many people get GNU from their friends, and such companies don't succeed, this will show that advertising was not really necessary to spread GNU. Why is it that free market advocates don't want to let the free market decide this?(5)

“My company needs a proprietary operating system to get a competitive edge.”

GNU will remove operating system software from the realm of competition. You will not be able to get an edge in this area, but neither will your competitors be able to get an edge over you. You and they will compete in other areas, while benefiting mutually in this one. If your business is selling an operating system, you will not like GNU, but that's tough on you. If your business is something else, GNU can save you from being pushed into the expensive business of selling operating systems.

I would like to see GNU development supported by gifts from many manufacturers and users, reducing the cost to each.(6)

“Don't programmers deserve a reward for their creativity?”

If anything deserves a reward, it is social contribution. Creativity can be a social contribution, but only in so far as society is free to use the results. If programmers deserve to be rewarded for creating innovative programs, by the same

token they deserve to be punished if they restrict the use of these programs.

“Shouldn’t a programmer be able to ask for a reward for his creativity?”

There is nothing wrong with wanting pay for work, or seeking to maximize one’s income, as long as one does not use means that are destructive. But the means customary in the field of software today are based on destruction.

Extracting money from users of a program by restricting their use of it is destructive because the restrictions reduce the amount and the ways that the program can be used. This reduces the amount of wealth that humanity derives from the program. When there is a deliberate choice to restrict, the harmful consequences are deliberate destruction.

The reason a good citizen does not use such destructive means to become wealthier is that, if everyone did so, we would all become poorer from the mutual destructiveness. This is Kantian ethics; or, the Golden Rule. Since I do not like the consequences that result if everyone hoards information, I am required to consider it wrong for one to do so. Specifically, the desire to be rewarded for one’s creativity does not justify depriving the world in general of all or part of that creativity.

“Won’t programmers starve?”

I could answer that nobody is forced to be a programmer. Most of us cannot manage to get any money for standing on the street and making faces. But we are not, as a result, condemned to spend our lives standing on the street making faces, and starving. We do something else.

But that is the wrong answer because it accepts the questioner’s implicit assumption: that without ownership of software, programmers cannot possibly be paid a cent. Supposedly it is all or nothing.

The real reason programmers will not starve is that it will still be possible for them to get paid for programming; just not paid as much as now.

Restricting copying is not the only basis for business in software. It is the most common basis(7) because it brings in the most money. If it were prohibited, or rejected by the customer, software business would move to other bases of organization which are now used less often. There are always numerous ways to organize any kind of business.

Probably programming will not be as lucrative on the new basis as it is now. But that is not an argument against the change. It is not considered an injustice that sales clerks make the salaries that they now do. If programmers made the same, that would not be an injustice either. (In practice they would still make considerably more than that.)

“Don’t people have a right to control how their creativity is used?”

“Control over the use of one’s ideas” really constitutes control over other people’s lives; and it is usually used to make their lives more difficult.

People who have studied the issue of intellectual property rights(8) carefully (such as lawyers) say that there is no intrinsic right to intellectual property. The kinds of supposed intellectual property rights that the government recognizes were created by specific acts of legislation for specific purposes.

For example, the patent system was established to encourage inventors to disclose the details of their inventions. Its purpose was to help society rather than to help inventors. At the time, the life span of 17 years for a patent was short compared with the rate of advance of the state of the art. Since patents are an issue only among manufacturers, for whom the cost and effort of a license agreement are small compared with setting up production, the patents often do not do much harm. They do not obstruct most individuals who use patented products.

The idea of copyright did not exist in ancient times, when authors frequently copied other authors at length in works of nonfiction. This practice was useful, and is the only way many authors’ works have survived even in part. The copyright system was created expressly for the purpose of encouraging authorship. In the domain for which it was invented—books, which could be copied economically only on a printing press—it did little harm, and did not obstruct most of the individuals who read the books.

All intellectual property rights are just licenses granted by society because it was thought, rightly or wrongly, that society as a whole would benefit by granting them. But in any particular situation, we have to ask: are we really better off granting such license? What kind of act are we licensing a person to do?

The case of programs today is very different from that of books a hundred years ago. The fact that the easiest way to copy a program is from one neighbor to another, the fact that a program has both source code and object code which are distinct, and the fact that a program is used rather than read and enjoyed, combine to create a situation in which a person who enforces a copyright is harming society as a whole both materially and spiritually; in which a person should not do so regardless of whether the law enables him to.

“Competition makes things get done better.”

The paradigm of competition is a race: by rewarding the winner, we encourage everyone to run faster. When capitalism really works this way, it does a good job; but its defenders are wrong in assuming it always works this way. If the runners

forget why the reward is offered and become intent on winning, no matter how, they may find other strategies—such as, attacking other runners. If the runners get into a fist fight, they will all finish late.

Proprietary and secret software is the moral equivalent of runners in a fist fight. Sad to say, the only referee we've got does not seem to object to fights; he just regulates them ("For every ten yards you run, you can fire one shot"). He really ought to break them up, and penalize runners for even trying to fight.

"Won't everyone stop programming without a monetary incentive?"

Actually, many people will program with absolutely no monetary incentive. Programming has an irresistible fascination for some people, usually the people who are best at it. There is no shortage of professional musicians who keep at it even though they have no hope of making a living that way.

But really this question, though commonly asked, is not appropriate to the situation. Pay for programmers will not disappear, only become less. So the right question is, will anyone program with a reduced monetary incentive? My experience shows that they will.

For more than ten years, many of the world's best programmers worked at the Artificial Intelligence Lab for far less money than they could have had anywhere else. They got many kinds of nonmonetary rewards: fame and appreciation, for example. And creativity is also fun, a reward in itself.

Then most of them left when offered a chance to do the same interesting work for a lot of money.

What the facts show is that people will program for reasons other than riches; but if given a chance to make a lot of money as well, they will come to expect and demand it. Low-paying organizations do poorly in competition with high-paying ones, but they do not have to do badly if the high-paying ones are banned.

"We need the programmers desperately. If they demand that we stop helping our neighbors, we have to obey."

You're never so desperate that you have to obey this sort of demand. Remember: millions for defense, but not a cent for tribute!

"Programmers need to make a living somehow."

In the short run, this is true. However, there are plenty of ways that programmers could make a living without selling the right to use a program. This way is customary now because it brings programmers and businessmen the most money, not because it is the only way to make a living. It is easy to find other ways if you want to find them. Here are a number of examples.

A manufacturer introducing a new computer will pay for the porting of operating systems onto the new hardware.

The sale of teaching, handholding and maintenance services could also employ programmers.

People with new ideas could distribute programs as freeware(9), asking for donations from satisfied users, or selling handholding services. I have met people who are already working this way successfully.

Users with related needs can form users' groups, and pay dues. A group would contract with programming companies to write programs that the group's members would like to use.

All sorts of development can be funded with a Software Tax:

Suppose everyone who buys a computer has to pay x percent of the price as a software tax. The government gives this to an agency like the NSF to spend on software development.

But if the computer buyer makes a donation to software development himself, he can take a credit against the tax. He can donate to the project of his own choosing—often, chosen because he hopes to use the results when it is done. He can take a credit for any amount of donation up to the total tax he had to pay.

The total tax rate could be decided by a vote of the payers of the tax, weighted according to the amount they will be taxed on.

The consequences:

The computer-using community supports software development. This community decides what level of support is needed. Users who care which projects their share is spent on can choose this for themselves.

In the long run, making programs free is a step toward the postscarcity world, where nobody will have to work very hard just to make a living. People will be free to devote themselves to activities that are fun, such as programming, after spending the necessary ten hours a week on required tasks such as legislation, family counseling, robot repair and asteroid prospecting. There will be no need to be able to make a living from programming.

We have already greatly reduced the amount of work that the whole society must do for its actual productivity, but only a little of this has translated itself into leisure for workers because much nonproductive activity is required to accompany productive activity. The main causes of this are bureaucracy and isometric struggles against competition. Free software will greatly reduce these drains in the area of software production. We must do this, in order for technical gains in productivity to translate into less work for us.

GNU 宣言

GNU 宣言¹（如下所示）由Richard Stallman在 1985 年撰写，用来请求大家支持 GNU 操作系统的开发。其部分文本摘自 1983 年撰写的初始声明。直到 1987 年，因为开发的原因它时时小有更改；那时起，看起来最好是保持它不再改变。

时过境迁，我们认识到使用不同的措辞可以避免一些常见的误解。从 1993 年起，我们添加了脚注来澄清这些问题。

如果你想安装 GNU/Linux 系统，我们建议你使用100% 自由的 GNU/Linux 发行版之一。如果你想做出贡献，请参看<http://www.gnu.org/help/help.html>。

GNU 工程是自由软件运动的一部分，该运动旨在捍卫软件用户的自由。把 GNU 和“开源”一词联系在一起是错误的—该词汇是 1998 年由一些不赞同自由软件运动之道德价值的人士发明的。他们使用该词汇来推动同一领域的非道德方案。

GNU 为何？ GNU 并非 UNIX！

GNU，代表的是 Gnu's Not Unix（GNU 并非 UNIX），是我正在编写的一个完全兼容 Unix 的软件系统，这样我就可以把它自由地交给想要使用它的人。²还有几个志愿者在帮助我。我们非常需要大家在时间、金钱、程序和设备方面的贡献。

目前，我们有一个可以用 lisp 编写编辑命令的 Emacs 文本编辑器、一个源代码级别的调试器、一个兼容 yacc 的分析器生成工具、一个链接器和大约 35 个应用程序。shell（命令解释器）也接近完成。一个新的可移植的优化 C 编译器已经可以自我编译，可能会在年内发布。现有一个初始的内核，不过还需要增加很多功能才可以模拟 Unix。当内核和编译器完成后，我们就有可能发布一个适合开发程序的 GNU 系统。我们会使用 Tex 作为文本排版工具，不过 nroff 还需要一些工作。我们还会使用自由的、可移植的 XWindow 系统。此后，我们还会加入一个可移植的 CommonLisp、一个 Empire 游戏、一个电子表格和数百个应用以及在线文档。最终，我们希望提供 Unix 系统常规带有的一切有用之物，以及更多。

GNU 将能够运行 Unix 的程序，但是它不完全和 Unix 一样。我们会根据我们在其他操作系统上的感受做出所有合理的改进。特别地，我们计划使用更长的文件名、文件版本号、防崩溃的文件系统、也许带有文件名填充、终端无关的显示支持、最后可能有一个基于 Lisp 的窗口系统，此时 Lisp 程序和普通 Unix 程序可以共享一个屏幕。C 和 Lisp 都将作为系统编程语言。我们会支持 UUCP、MITChaosnet 和 Internet 等通信协议。

GNU 最初的目标是 68000/16000 之类的带虚拟内存的机器，因为它们是最容易跑起来的机器。让 GNU 在更小的机器上运行的额外努力就留给那些需要使用这些机器的人吧。

为了避免可怕的混淆，请在指示本工程时，发出“GNU”中 g 的音。

为什么我必须编写 GNU

我认可的黄金法则是如果我喜欢一个程序，我就必须把它分享给喜欢它的人。软件销售商通过让每个用户保证不和其他人分享来分化用户并控制他们。我拒绝以这种方式打破和其他用户组成的统一体。我的良知让我无法签署这样的保密协议或软件许可证协议。几年来，我在人工智能实验室都在反抗这种趋势以及其他冷漠，但是最终他们还是走得太远了：我无法再呆在一个为我做违背我意愿之事的机构。

为了能够继续不失颜面地使用计算机，我决定把一些必要的自由软件集合在一起，这样我就能够继续下去而不需要任何非自由软件。我从人工智能实验室辞了职，这样就可以在我发布 GNU 时避免和 MIT 产生法律纠葛。³

¹注意，有些脚注是由 GNU CTT 加的。

²此处用词不当。其初衷是人们不必为使用 GNU 系统而支付许可费。但是用词却没有清楚地说明此事，而人们经常理解为这是说 GNU 的拷贝总是免费或廉价地发行。这不是本意；后来，宣言指出公司提供有偿发行服务的可能性。之后，我也了解到认真区别自由中的“free（自由）”和价格中的“free（免费）”。自由软件是用户有自由修改和发布的软件。有些用户可能得到免费拷贝，而有些用户付费得到拷贝—如果这些资金帮助到软件的改善，善莫大焉。重要的一点是拥有拷贝的用户有自由和其他人一起使用自由软件。

³2.“赠送”是另一个不妥的表达，它再次说明我那时还没有清楚地分开价格和自由的问题。我们现在建议在谈论自由软件时避免这一表达。请参看“不清楚的词汇和短语”了解更多解释

第 4.3 章

为什么 GNU 将会兼容 Unix

Unix 并不是我理想中的系统，但是它还不算太差。Unix 的主要功能看来是好的，而我认为我可以在不破坏这些好功能的情况下填补 Unix 缺少的东西。而且和 Unix 兼容可以让许多人能够方便地接纳它。

第 4.4 章

如何获取 GNU

GNU 不属于公共领域。GNU 允许任何人修改和再发布，但是任何发布者都不能限制它的继续发布。就是说，它不允许专有性的修改。我想让 GNU 的所有版本都保持自由。

第 4.5 章

为什么许多程序员想要提供帮助

我发现许多程序员看到 GNU 很兴奋并想要提供帮助。

许多程序员对系统软件的商业化并不高兴。这可能使他们赚到更多的钱，不过这一般要求他们和其他程序员之间是对立关系，而不是伙伴关系。程序员之间的友谊的基本方式是分享程序；而现在典型的市场活动基本上是禁止程序员互相成为朋友。软件买家必须在友谊和守法之间抉择。自然地，许多人认为友谊更重要。但是许多守法的人通常会感到选哪个都不自在。他们变得愤世嫉俗并且认为编程只是一个挣钱的手段。

开发 GNU 和使用 GNU 而不是专有软件，我们就能够变得友善并守法。另外，GNU 成为一个激励和团结其他人加入分享行列的榜样和旗帜。这给予我们一种和谐的感觉，它是使用非自由软件不可能有的。就和我讨论过的程序员来说，大约一半人认为这是一个重要的幸福感，而它是金钱无法替代的。

第 4.6 章

你该如何做出贡献

(现今，软件帮助任务请看高优先级项目列表和 GNU 帮助需求列表，这是 GNU 软件包的一般任务列表。其他帮助，请看帮助 GNU 操作系统的指南。)

我请求计算机制造商捐助机器和金钱。我请求个人捐助程序和作品。

如果你捐助机器，你可以期待的结果就是 GNU 将会早一天在该机器上运行。捐助的机器应该是完备的、可用的系统，它应该适用于居家的环境，并无需复杂的冷却或供电系统。

我已经找到相当多的程序员，他们热切地想要为 GNU 贡献闲暇时的工作。就大多数项目而言，这些工作很难协调；这些独立完成的部分凑在一起会不工作。但是就替代 Unix 的特定任务而言，没有这个问题。一个完整的 Unix 系统包含数百个应用程序，每个都有独立的文档。大多数的接口规格都由 Unix 兼容性所限定。如果每个贡献者能够编写一个单一的兼容性 Unix 应用，并使之在原始的 Unix 系统中正常工作，那么这些应用放在一起就会正常工作。即使出现一些意外的墨菲问题⁴，联合这些部件也是可以完成的任务。（内核将需要更密切的沟通，它将会由一个小的、紧凑的小组来进行。）

如果我得到金钱上的捐助，我也许能够雇佣一些全职或兼职的人。薪水按照程序员的标准来看的话不高，但是我要找的人要和看重金钱一样看中社区精神的建设。我把这当作一种方法，它让一些人能够全身心地为 GNU 工作而不用寻求其他谋生的手段。

第 4.7 章

为什么所有计算机用户都会受益

一旦 GNU 完成，任何人都能够自由地得到一个好用的系统，正如得到空气一样。⁵

其意义远远超出了只是为每个人省去一份 Unix 许可证费用。这意味着避免了大量重复的系统编程工作造成的浪费。这些努力就可以用于推进技术的进步。

完整的系统资源将向每个人开放。其结果是，如果有用户需要更改系统，他总可以自由地自己修改或雇用其他程序员或公司来改。用户就用不再祈求拥有源代码的那家公司或那一个程序员来帮他修改，没有人再处于独断的地位。

⁴1.Murphy，墨菲效应。是指事情如果有变坏的可能，不管这种可能性有多小，它总会发生。

⁵这是又一个我没有认真区别“free”一词的两种意思的地方。该陈述并没有错—你是可以免费获得 GNU 软件，从朋友那里或从网上下载。但是它在提倡错误的理念。

通过鼓励学生学习和改进系统代码，学校能够提供多得多的教育环境。哈佛大学的计算机实验室曾有一个政策：如果程序的源代码不能公开显示在屏幕上，那么就不能安装该程序，这就是坚持拒绝安装某些程序。我受此启发良多。

最后，考虑谁是系统软件的所有者以及谁应该做或不做什么的开销也被化解了。

筹划人们为一个程序付费，包括许可证费用，因为要通过麻烦的机制来搞清楚一个人应该为该程序支付多少费用，总是会导致大量的社会成本。而且只有管制的国家才能强制每个人都遵守付费制度。举例来说，空间站的空气要花大量成本来制造：为每次呼吸的容量计费是公平的，但是时时都带着测量面具即使是对负担得起呼吸费用的人也是无法忍受的事。加之随处可见的、监控人们是否脱掉面具的摄像头也令人无法容忍。所以，支持空气工厂的最好办法还是只收人头税并摆脱掉面具。

复制全部或部分程序对程序员来说和呼吸一样自然，一样有生产力。它也应该一样自由。

第 4.8 章

一些容易驳斥的、反对 GNU 目标的观点

“如果免费，就没有人会用了，因为用户没有可靠的技术支持。”

“你必须对程序收钱才能提供技术支持。”

如果人们宁愿免费获得没有服务的 GNU，而不是付费给 GNU 获得服务，那么为免费 GNU 提供技术服务的公司应该是有利可图的。⁶

我们必须区别对待真正的编程和仅仅是手把手服务这两种形式的技术支持。前者是你不能依赖一个软件供应商来解决的。如果你的问题没有被足够多的人共同体会，那么供应商会告诉你：快走开。

如果你的业务需要依赖于技术支持，那么唯一的办法是拥有所有必要的源代码和工具。然后，你就可以雇佣任何有能力的人为你解决问题；你就不必祈求某个特定的人。对 Unix，源代码的价格使大多数人都不会考虑。对 GNU，这就简单了。还会有找不到能人的时候，但这个问题不是发行策划的问题。GNU 并没有解决世界上所有的问题，只是其中一些问题。

同时，对计算机知之甚少的用户需要手把手服务：为他们做些很容易但他们真的不知道怎么做的事。

这些服务可以由那些只销售手把手服务和修复服务的公司提供。如果用户愿意花钱买带服务的产品，那么他们也应该会为免费的产品购买服务。服务公司竞争的是质量和价格；用户不会绑定在某个服务商上。同时，像我们这样的不需要服务的人可以不用购买服务来使用程序。

“不打广告，不可能有很多人知道，所以你必须对程序收费才能够支付广告费。”

“对免费可得的程序打广告是做无用功。”

有很多免费或极其廉价的宣传形式可以用来通知计算机用户关于 GNU 的消息。但是使用广告可能会通知到更多的计算机用户。如果真是这样，那么通过广告收费寄送 GNU 拷贝的业务应该可以赚回广告费及更多。这样的话，只有从该广告获利的用户才付费。

另一方面，如果许多人从朋友处获得 GNU，而此类业务并不成功，那么说明靠广告传播 GNU 并无实际必要。为什么自由市场的倡导者不能让自由市场决定这件事呢？⁷

“我公司需要专有操作系统来在竞争中取胜。”

GNU 将把操作系统软件从竞争的王国中移除。你不能在此取胜，你的对手也不能。你们将在其他方面竞争，但同时在操作系统领域获利。如果你的业务是销售操作系统，那么你不会喜欢 GNU，但这对你来说是困难的事。如果你的业务是其他，GNU 能够把你从昂贵的操作系统售价中解救出来。

我很想看到许多制造商和用户会捐助 GNU 的开发，这样会降低他们的花费。⁸

“难道程序员不该因为他们的创造力得到回报吗？”

值得回报的东西应该是对社会的贡献。创造力可以是一种社会贡献，但只有在社会能够自由使用其结果时才是。如果程序员应该由于创新程序而得到回报，同理，他们也应该由于限制程序的使用而得到惩罚。

“难道程序员不能为自己的创造力要求回报吗？”

工作获得报酬或追求更高的薪酬并没有什么不对，只要我们不使用破坏性的手段。但是今天，软件领域的常规手段就是建立在破坏之上的。

因为限制减少了程序使用的方法和人数，所以通过限制程序的使用来从用户身上榨取钱财是破坏性的。它限制了人类可以从该程序中获得财富的总量。当限制是故意为之，伤害的结果就是故意破坏。

优秀公民不会使用这种破坏手段来致富的原因是，如果每人都这样，我们都会被相互破坏搞得更穷困。这是康德伦理⁹；或者叫黄金定律。因为我不喜欢这样的结果，所以如果每个人都囤积信息，我就有义务说这样做是不对的。特别地，

⁶现在就有几个这样的公司。

⁷虽然它不是公司而只是慈善机构，自由软件基金会有 10 年是靠发行服务来获得其大部分资金的。你可以通过从 FSF 订购东西来支持它的工作。

⁸一组公司在 1991 年左右集资来支持 GNU C 编译器的维护。

⁹Kantian Ethics，康德伦理。是指德国哲学家康德的义务论伦理思想，其基本观点是，世界上只有一个东西是无条件的善，不但它自身是无条件善的，而且也是使一切其他东西成为善的条件，这个东西就是理性，即善良意志。

希望个人的创造力有回报并不能证明剥夺其他人的这种创造力就是对的。

“程序员不就饿死了吗?”

我可能会回答没人被迫成为程序员。我们大多数人无法靠沿街乞讨过活。但结果是，我们并没有被迫沿街乞讨并挨饿。我们会去做其他事情。

然而，这个回答是错的，因为它承认了提问者隐含的假设：没有软件的所有权，程序员就可能不会收到任何报酬。据此，报酬不是全部、就是没有。

程序员不被饿死的真正原因是他们还有能从编程谋生的方法；只是不如现在赚得多罢了。

限制拷贝不是软件行业唯一的办法。它是最常见的基础¹⁰因为它收获了最多的金钱。如果它被禁止或被客户拒绝，软件行业会迁移到那些现在不常用的基础结构之上。总是有多种方式来组织经营活动的。

也许在新基础之上的编程工作不再象现在一样可以赚大钱。可是那并不是反驳该变化的论据。现在销售人员按劳取酬并无不妥。如果程序员这样，那么也是正当的。（实际上，他们也许还能赚更多。）

“难道人们没有权利控制自己的创造力如何被使用？”

“控制自己想法的应用”真的构成对其他人生活的控制；而且通常是使他人的生活更困难。

认真研究过知识产权问题¹¹的人（比如律师）会说知识产权并非天生的权利。政府确认的那些知识产权种类是有具体目的的特定法律活动的产物。

比如，专利体系是为了鼓励发明家公开其发明详情而建立的。其目的是帮助社会而不是帮助发明家。那时，17年的专利期相比技术进步的速度是短暂的。由于专利只是制造商之间的问题，对他们来说，专利协议的花费比生产建设要小，所以专利通常没有太大的害处。专利没有限制使用它们的大多数用户。

版权的概念在古代并不存在，那时作者们经常互相大量拷贝非文学类作品。这是很实用的活动，也是许多作者的作品能够哪怕只有一部分流传下来的唯一方法。版权系统为鼓励作者权益而特意创建。在其创建的发明领域—书籍，只有用印刷机才能有效拷贝—版权没什么害处，也没有限制大多数读者。

所有知识产权都只是社会发放的许可证，因为人们曾经认为，不管是对还是错，发放这样的许可证可以使整个社会受益。但是就任何具体情况来说，我们都要问：发放该许可证真的让我们受益了吗？获得授权的人能够从事什么活动呢？

今天的软件和一百年前的书籍有很大的不同。软件最容易的拷贝是人传人，软件有源代码和目标代码两种不同形式，软件是来使用而不是阅读和欣赏的，这些事实结合在一起就构成了一种情形。在此情形下，加强版权对整个社会在物质和精神上都是伤害；无论法律是否允许，我们此时都不应该再维护版权。

“竞争使东西变得更好。”

赛跑是竞争的典范：通过回报优胜者，我们鼓励人们跑得更快。当资本主义真的这样运作时，它做得很好；但是其辩护者做的这个假设并不总是对的。如果竞争者忘记了回报的原因而只想着胜利，不计方法，那么他们就可能使用其他的策略—比如攻击别的竞争者。如果竞争者在互相打架，大家就都跑不快。

专有软件和保密软件在道德上等同于互相打架的竞争者。令人沮丧的是，我们唯一的裁判看来并不反对打架；他只是规范打架者（“每跑10米，你们可以打一下”）。他真的应该把他们分开，并严惩试图打架的竞争者。

“没有金钱刺激，人们不就不再编程了吗？”

实际上，许多人在绝对没有金钱刺激的情况下也会编程。编程对一些人有不可抗拒的魔力，这些人往往是最擅长编程的那些人。从来也不缺少坚持音乐的职业音乐家，即使他们毫无希望靠音乐谋生。

但是这个问题，虽然经常被问到，并不是指这种情况。程序员会得到报酬，只是变少。所以问题是，金钱减少时，还有人编程吗？我的经验是：有。

10多年来，许多世界上最好的程序员在人工智能实验室工作，这里的收入要比他们到其他地方工作少得太多。他们获得了许多非金钱的回报：比如，名望和感谢。而创造力本身也是快乐，也是回报。

然后，当有机会做同样有趣的工作并赚大钱时，大多数人离开了。

这说明人们会为致富之外的理由编程；如果有同时也能赚到大钱的机会，他们也会选择它。薪水低的企业在和薪水高的企业竞争时表现不佳，但是如果薪水高的企业被禁止，低薪水的企业不应该再表现差劲吧。

“我们迫切需要程序员。如果他们要求我们不要帮助友邻，我们不得不那样做。”

你永远也不会绝望到去遵守这样的命令。请记住：宁为玉碎，不为瓦全！¹²

“程序员也需要谋生啊。”

短期来看，是这样的。不过，程序员有很多不用出卖程序的使用权利就可以谋生的方法。出卖权利现在成了惯例，是因为它带给程序员和生意人最多的钱财，而不是因为它是谋生的唯一手段。如果想要，我们能够轻易找到其他的方法。这里举几个例子。

¹⁰ 我觉得我说专有软件是软件行业最常见的赚钱基础是个错误。看起来，定制软件开发过去和现在实际上都是最常见的商业模式。这个商业模式不提供收取租金的可能性，所以它必须不断地做事来维持收入。在自由软件的世界，软件定制行业还会继续存在，基本没什么变化。因此，我不再预期程序员在自由软件的世界里收入会变少。

¹¹ 在20世纪80年代，我还没有意识到谈论“知识产权”的“问题”多么令人困惑。该术语明显是倾向性的；较不明显的事是，它把针对非常不同的多种互不相干的法律纠结在一起。现在，我敦促人们彻底拒绝“知识产权”这一术语，免得它导致其他人以为这些法律构成一个相关的问题。明确的方法应该是独立讨论专利、版权和商标。请参看关于该术语如何散布混乱和偏见的进一步解释。

¹² Millions for defense, but not a cent for tribute! 原意是宁可战斗，也不乞和！

制造商新引进新计算机需要雇人来把操作系统移植到新硬件上。

教育培训、手把手服务和维护服务也可能雇佣程序员。

有新想法的人可以发布免费软件¹³，并向对此满意的用户寻求捐助，或者是销售手把手服务。我就碰到一些成功这样做的。

需求相关的用户可以组建用户组，并支付会费。用户组就可以和程序公司签约让公司定制组内成员需要的程序。

所有开发费用都可以由软件税来支付：

假设每个购买计算机的用户都要按价格支付一定比例的软件税。政府可以让诸如 NSF¹⁴之类的代理使用该税收支持软件开发。

但是如果购买者自己向软件开发做了捐助，那么他可以减税。他可以自己选择捐助项目—通常，他会选择他希望能够用到的项目。减税额度最高是免税。

税率可以由交税的人投票决定，票的权重可以按大家的应税额来算。

结果：

计算机使用社群支持软件开发。

该社群决定应该支持到什么程度。

用户可以根据自己的需要来选择要支持的项目。

长远来看，让软件自由是通往富足世界的一小步；在富足世界里，人们不必辛苦工作来谋生。人们在每周 10 小时的法律活动、家庭咨询、机器人维修和流星观察等规定任务之外，能够自由投入到象编程这样的有趣活动中。那时，就没有必要再以编程为谋生手段了。

我们已经把整个社会要维持生产力的工作大大减少了，但是只有很少一部分转化为劳动者的闲暇，因为生产活动需要夹杂很多的非生产活动。其主要原因是官僚主义和对竞争的抗拒。自由软件会大大减少在软件生产领域的生产力流失。我们必须做这件事，为了使技术进步带来的生产力提高能够转化为人们工作的减少。全书结束。

¹³后来，我们了解到要区别“自由软件”和“免费软件”。“免费软件”是指你可以自由再发布的软件，但是你并没有自由来学习和修改其源代码，所以大部分免费软件不是自由软件。请参看“不清楚的词汇和短语”了解更多解释。

¹⁴NSF, National Science Foundation: 美国国家科学基金会。