

开源协议

在开源协议方面，本人认为自己的对于开源协议的解释并不算太好，所以参考前辈对于开源协议的解释，并标注内容来源，表示尊重作者著作权。

作者: pdai

开源不等于免费！为了加速我们的开发，我们会使用开源的软件和源码； 为避免商业风险，需要在使用时了解第三方如软件协议，版本，和已知 CVE 风险等；本文旨在从开源软件再发布过程使用权限的角度入手，总结各个常见开源协议的异同，方便理解。

大部分人都希望作品能够被多数人分享查阅。这样不仅提高自己业界的知名度，同时也方便了需要的人为开源做出了贡献。但是代码一旦被贴出来，任何人都可以看到并获取，之后发生的事情你就无法控制了。

所以为了公开分享你的代码，同时又让你对代码保留一定权利，在作品中声明一个许可协议是非常有必要的。有协议和没声明协议的裸代码是有非常重要区别的，一般作品当中没声明协议的默认为 Copy right 的，也就是版权保留。此种情况表明他人没有任何授权，不得复制分发修改使用等等。有了协议的声明，在未来你的维权上面会方便很多，让你的作品在分享的同时保留了自身的一些权利。

License 是软件的授权许可，里面详尽表述了你获得代码后拥有的权利，可以对别人的作品进行何种操作，何种操作又是被禁止的。

软件协议可分为开源和商业

- 对于商业协议，或者叫法律声明、许可协议，每个软件会有自己的一套行文，由软件作者或专门律师撰写。因为涉及到以后侵权打官司这种事情，这种商业条款的行文是非常严谨而讲究的，读起来很晦涩难懂。
- 对于开源协议，要知道开源不等于免费，也不等于没有约束。虽然相对商业协议要更加简明，但对于很多人来说还是像在看天书一样。

协议列表

首先一共有哪些公开的协议: <https://opensource.org/licenses/alphabetical>

常用协议

最流行的六种---GPL、BSD、MIT、Mozilla、Apache 和 LGPL。

乌克兰程序员 Paul Bagwell，画了一张分析图，说明应该怎么选择，只用两分钟，你就能搞清楚这六种许可证之间的最大区别。 下面是阮一峰中文翻译版本：



1. Apache 许可协议

Apache 许可证(Apache License)，是一个在 Apache 软件基金会发布的自由软件许可证，最初为 Apache [http 服务器](#)而撰写。Apache 许可证要求被授权者保留版权和放弃权利的申明，但它不是一个反版权的许可证。

此许可证最新版本为“版本 2”，于 2004 年 1 月发布。Apache 许可证在 Apache 社区内外被广泛使用。Apache 基金会下属所有项目都使用 Apache 许可证，许多非 Apache 基金会项目也使用了 Apache 许可证：据统计，截至 2008 年 4 月，在 sourceforge 上有超过 3000 个项目使用了 Apache 许可证。

Apache 许可协议, 2.0 版本, 授予了用户大量的权利。这些权利可以应用于拷贝权，也可以用于专利权。因为很多许可协议只能适用于拷贝权，不适用于专利权，所以这个灵活性就成了让有专利的开发者们选择许可协议时的一个显著参考因素 (要想明白两者之间的不同，请参考 [How Stuff Works](#) 上的这篇文章)。

下面是关于 Apache 许可协议所允许的事项的详细说明：

- **权利永恒。** 一旦被授权，权利永久不失。
- **权利无疆界。** 在一个国家里被授权，形同于在所有国家被授权。例如，你在美国，但许可权最初在印度被授予，你同样可以使用这个被授权的程序。
- **授权无需付费和支付酬劳。** 你既不需要在使用之前支付任何的费用，也无需在每次使用时支付任何的费用，或者其它类似情况。
- **权利不排他。** 使用这种许可协议下的软件时，不妨碍你使用其它软件。

- **权利不可变更。** 权利一旦授予，不可剥夺。也就是说，你在使用这个软件的过程中，你无需担心这种情况：当你开发出了令人羡慕的基于这种授权软件的衍生产品时，有人突然跳出来对你说，抱歉，你将不再被允许使用这个程序。

(在这个协议里有个条款声明：如果你控告别人在这个许可协议下的产品有侵犯专利的行为，那你的授权将会自动终止，但这只是适用于有专利权的作品。只要你不搞有专利作品的诉讼，你永远无需担心这种问题。)

- **对再分发的作品还有个特殊要求，总的就是说要给予这些程序的作者和许可协议的维护者适当的名誉。**

2. MIT 许可协议

MIT 协议应该是在流行的开源协议中最简短的、使用最广泛的一种协议。它的条款非常的宽松，而且跟其它协议相比更自由。MIT 协议是目前最少限制的协议。

它基本上就是任何人可以对这个协议下的软件的做任何的事情，只要你能认可这个协议。这种协议最基本的条款 (the information that it is provided without warranty, which comprises the final paragraph)如下：

特此授权，任何人都可免费获得这个软件以及相关文档 (the Software) 的拷贝，可以无限制的使用这个软件，包括无限制的权利去使用、复制、修改、合并、发布、附加从属协议，以及/或者出售软件的拷贝，同时，为了让软件的提供者有权利做到这些，下面的条件必须遵守：

上面的拷贝权声明和许可声明必须包含在所有的这个软件拷贝里和实际分署部分里。

这也就是说：

- 你可以随意使用，复制，修改这个软件。没有人能够阻止你在任何工程里使用它，你可以复制任意次数、以任何形式，或按你的愿望修改它。
- 你可以向外免费发放，或出售。你可以随意的分发它，没有任何限制。
- 唯一的限制是你必须接受协议条款。

3. BSD 许可协议

BSD 协议有很多分支，它们都代表了一种宽松的自由软件协议，相对其它协议，例如 GPL，来说，它们对软件的传播给予了更少的限制。

在这种协议的各种版本中，有两个版本格外的重要：新 BSD 协议/修订版 BSD 协议和简化 BSD 协议/FreeBSD 协议。这两类协议都实现的对 GPL 兼容的自由软件协议，而且被 Open Source Initiative 认可为开源软件协议。

新 BSD 协议(3-clause license)无任何限制的允许你以任何目的二次分发这种软件，唯一的要求是必须保留拷贝权的声明和协议里的软件权利放弃条款。这种协议还

有一个限制，未经许可不得使用这个作品的所有曾经捐助者的署名。新 BSD 协议和简化 BSD 协议的最主要的区别是后者删除了署名条款。

BSD 开源协议是一个给予使用者很大自由的协议。基本上使用者可以“为所欲为”，可以自由的使用，修改源代码，也可以将修改后的代码作为开源或者专有软件再发布。

但“为所欲为”的前提当你发布使用了 BSD 协议的代码，或则以 BSD 协议代码为基础做二次开发自己的产品时，需要满足三个条件：

- 如果再发布的产品中包含源代码，则在源代码中必须带有原来代码中的 BSD 协议。
- 如果再发布的只是二进制类库/软件，则需要在类库/软件的文档和版权声明中包含原来代码中的 BSD 协议。
- 不可以用开源代码的作者/机构名字和原来产品的名字做市场推广。
- BSD 代码鼓励代码共享，但需要尊重代码作者的著作权。BSD 由于允许使用者修改和重新发布代码，也允许使用或在 BSD 代码上开发商业软件发布和销售，因此是对商业集成很友好的协议。而很多的公司企业在选用开源产品的时候都首选 BSD 协议，因为可以完全控制这些第三方的代码，在必要的时候可以修改或者二次开发。

4. GPL 许可协议

我们很熟悉的 Linux 就是采用了 GPL。GPL 协议和 BSD，Apache Licence 等鼓励代码重用的许可很不一样。GPL 的出发点是代码的开源/免费使用和引用/修改/衍生代码的开源/免费使用，但不允许修改后和衍生的代码做为闭源的商业软件发布和销售。

这也就是为什么我们能免费的各种 linux，包括商业公司的 linux 和 linux 上各种各样的由个人，组织，以及商业软件公司开发的免费软件了。

GPL 协议的主要内容是只要在一个软件中使用（“使用”指类库引用，修改后的代码或者衍生代码）GPL 协议的产品，则该软件产品必须也采用 GPL 协议，既必须也是开源和免费。这就是所谓的“传染性”。GPL 协议的产品作为一个单独的产品使用没有任何问题，还可以享受免费的优势。

由于 GPL 严格要求使用了 GPL 类库的软件产品必须使用 GPL 协议，对于使用 GPL 协议的开源代码，商业软件或者对代码有保密要求的部门就不适合集成/采用作为类库和二次开发的基础。

其它细节如再发布的时候需要伴随 GPL 协议等和 BSD/Apache 等类似。

5. LGPL 许可协议

LGPL 是 GPL 的一个主要为类库使用设计的开源协议。和 GPL 要求任何使用/修改/衍生之 GPL 类库的软件必须采用 GPL 协议不同。LGPL 允许商业软件通过类库引用 (link) 方式使用 LGPL 类库而不需要开源商业软件的代码。这使得采用 LGPL 协议的开源代码可以被商业软件作为类库引用并 发布和销售。

但是如果修改 LGPL 协议的代码或者衍生，则所有修改的代码，涉及修改部分的额外代码和衍生的代码都必须采用 LGPL 协议。因此 LGPL 协议的开源 代码很适合作为第三方类库被商业软件引用，但不适合希望以 LGPL 协议代码为基础，通过修改和衍生的方式做二次开发的商业软件采用。

GPL/LGPL 都保障原作者的知识产权，避免有人利用开源代码复制并开发类似的产品。

6. MPL 许可协议

MPL 是 The Mozilla Public License 的简写，是 1998 年初 Netscape 的 Mozilla 小组为其开源软件项目设计的软件许可证。

MPL 许可证出现的最重要原因就是，Netscape 公司认为 GPL 许可证没有很好地平衡开发者对源代码的需求和他们利用源代码获得的利益。同著名的 GPL 许可证和 BSD 许可证相比，MPL 在许多权利与义务的约定方面与它们相同（因为都是符合 OSI 认定的开源软件许可证）。

但是，相比而言 MPL 还有以下几个显著的不同之处：

- MPL 虽然要求对于经 MPL 许可证发布的源代码的修改也要以 MPL 许可证的方式再许可出来，以保证其他人可以在 MPL 的条款下共享源代码。但是，在 MPL 许可证中对“发布”的定义是“以源代码方式发布的文件”，这就意味着 MPL 允许一个企业在自己已有的源代码库上加一个接口，除了接口程序的源代码以 MPL 许可证的形式对外许可外，源代码库中的源代码就可以不用 MPL 许可证的方式强制对外许可。这些，就为借鉴别人的源代码用做自己商业软件开发的行为留了一个豁口。

- MPL 许可证第三条第 7 款中允许被许可人将经过 MPL 许可证获得的源代码同自己其他类型的代码混合得到自己的软件程序。

对软件专利的态度，MPL 许可证不像 GPL 许可证那样明确表示反对软件专利，但是却明确要求源代码的提供者不能提供已经受专利保护的源代码（除非他本人是专利权人，并书面向公众免费许可这些源代码），也不能在将这些源代码以开放源代码许可证形式许可后再去申请与这些源代码有关的专利。

对源代码的定义

- 而在 MPL（1.1 版本）许可证中，对源代码的定义是：“源代码指的是对作品进行修改最优先择取的形式，它包括：所有模块的所有源程序，加上有关的接口的

定义，加上控制可执行作品的安装和编译的‘原本’（原文为‘Script’），或者不是与初始源代码显著不同的源代码就是被源代码贡献者选择的从公共领域可以得到的程序代码。”

- MPL 许可证第 3 条有专门的一款是关于对源代码修改进行描述的规定，就是要求所有再发布者都得有一个专门的文件就对源代码程序修改的时间和修改的方式有描述。

小结

GPL 协议、LGPL 协议与 BSD 协议的法律区别。

简而言之，GPL 协议就是一个开放源代码协议，软件的初始开发者使用了 GPL 协议并公开软件的源程序后，后续使用该软件源程序开发软件者亦应当根据 GPL 协议把自己编写的源程序进行公开。GPL 协议要求的关键在于开放源程序，但并不排斥软件作者向用户收费。

虽然如此，很多大公司对 GPL 协议还是又爱又恨，爱的是这个协议项下的软件历经众多程序员千锤百炼的修改，已经非常成熟完善，恨的是必须开放自己后续的源程序，导致竞争对手也可以根据自己修改的源程序开发竞争产品。

正因大公司对 GPL 协议在商业上存在顾虑，因此，另两种协议被采用的更多，第一种是 LGPL（亦称 GPL V2）协议，可以翻译为更宽松的 GPL 协议。与 GPL 协议的区别为，后者如果只是对 LGPL 软件的程序库的程序进行调用而不是包含其源代码时，相关的源程序无需开源。

调用和包含的区别类似在互联网网页上对他人网页内容的引用：如果把他人的内容全部或部分复制到自己的网页上，就类似包含，如果只是贴一个他人网页的网址链接而不引用内容，就类似调用。有了这个协议，很多大公司就可以把很多自己后续开发内容的源程序隐藏起来。

第二种是 BSD 协议（类似的还有 MIT 协议）。BSD 协议鼓励软件的作者公开自己后续开发的源代码，但不强求。在 BSD 协议项下开发的软件，原始的源程序是开放源代码的，但使用者修改以后，可以自行选择发布源程序或者二进制程序（即目标程序），当然，使用者有义务把自己原来使用的源程序与 BSD 协议在软件对外发布时一并发布。因为比较灵活，所以 BSD 深受大公司的欢迎。

参考文章

常用开源软件许可协议简介

<https://www.cnblogs.com/yefengmeander/p/4173673.html> 开源协议在再发布过程中的授权总结 <http://blog.csdn.net/bqlyj/article/details/60322655> 如何为你的代码选择一个开源协议 - 刘哇勇的部落格

http://www.cnblogs.com/Wayou/p/how_to_choose_a_license.html 重要开源协议的比较(BSD,Apache,GPL,LGPL,MIT) – TechBirds 在路上

http://blog.csdn.net/techbirds_bao/article/details/8785413 游云庭律师的回答 - 主流开源协议之间有何异同? - 知乎

<https://www.zhihu.com/question/19568896/answer/12284793> Choose a license

<https://choosealicense.com/licenses/> Understanding Copyright And License

<https://www.smashingmagazine.com/2011/06/understanding-copyright-and-lic...>

开源协议 - 百度百科

<http://baike.baidu.com/item/%E5%BC%80%E6%BA%90%E5%8D%8F%E8%AE%AE> 如何选择开源许可证?

http://www.ruanyifeng.com/blog/2011/05/how_to_choose_free_software_licenses

更多文章请参考 <http://www.cnblogs.com/pengdai/>