



链滴

常见开源协议介绍

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1607042899164>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

世界上的开源许可证（Open Source License）大概有上百种，今天我们来介绍下集几种我们常见的源协议。大致有GPL、BSD、MIT、Mozilla、Apache和LGPL等。

许可证	版本	包含许可证	包含源代码	链接	状态变化	商业使用	散布	修改	专利授权	私人使用	授权转售	无担保责任	没有商标
Apache许可证	2.0	是			是	是	是	是	是	是	是	是	是
3句版BSD许可证		是				是	是	是		是	是	是	是
2句版BSD许可证		是				是	是	是		是	是	是	
GNU通用公共许可证	2.0	是	是		是	是	是	是	是	是	否	是	
GNU通用公共许可证	3.0	是	是		是	是	是	是	是	是	是	是	
GNU宽通用公共许可证	2.1	是	是	是		是	是	是	是	是	是	是	
GNU宽通用公共许可证	3.0	是	是	是		是	是	是	是	是	是	是	
MIT许可证		是				是	是	是		是	是	是	
Mozilla公共许可证	2.0	是	是			是	是	是	是	是	是	是	是
Eclipse公共许可证	1.0	是	是			是	是	是	是	是	是	是	
Affero通用公共许可证		是	是		是	是	是	是		是	是	是	
一般的著作权 (note 1)		是				是	否	否		是	否		

Apache License

Apache License (Apache 许可证)，是Apache软件基金会发布的一个自由软件许可证。

Apache License是著名的非盈利开源组织Apache采用的协议。该协议和BSD类似，同样鼓励代码共享和最终原作者的著作权，同样允许源代码修改和再发布。但是也需要遵循以下条件：

- 需要给代码的用户一份Apache License。
- 如果修改了代码，需要再被修改的代码中说明。
- 在衍生的代码中（修改和有源代码衍生的代码中）需要带有原来代码中的协议，商标，专利声明和他原来作者规定需要包含的说明。
- 如果再发布的产品中包含一个Notice文件，则在Notice文件中需要带有Apache Licence。你可以在Notice中增加自己的许可，但是不可以表现为对Apache Licence构成更改。
- Apache Licence也是对商业应用友好的许可。使用者也可以再需要的时候修改代码来满足并作为开或商业产品发布/销售。

使用这个协议的好处是：

- 永久权利一旦被授权，永久拥有。
- 全球范围的权利在一个国家获得授权，适用于所有国家。假如你在美国，许可是从印度授权的，也有问题。
- 授权免费无版税，前期、后期均无任何费用。
- 授权无排他性任何人都可以获得授权
- 授权不可撤消一旦获得授权，没有任何人可以取消。比如，你基于该产品代码开发了衍生产品，你用担心会在某一天被禁止使用该代码。

BSD

BSD是"Berkeley Software Distribution"的缩写，意思是"伯克利软件发行版"。

BSD开源协议:是一个给予使用者很大自由的协议。可以自由的使用，修改源代码，也可以将修改后的码作为开源或者专有软件再发布。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，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，在必要的时候可以改或者二次开发。

GPL

GPL (GUN General Public License) : GUN 通用公共许可协议。

Linux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上各种各样的由人，组织，以及商业软件公司开发的免费软件了。

LPGL

LPGL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。LPGL允许商业软件通过类库引用(link)方式使用LPGL类库而不需要开源业软件的代码。这使得采用LPGL协议的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LPGL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都必须采用LPGL协议。因此LPGL协议的源代码很适合作为第三方类库被商业软件引用，但不适合望以LPGL协议代码为基础。

商业软件采用。

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

MIT

MIT是和BSD一样宽范的许可协议,源自麻省理工学院(Massachusetts Institute of Technology,MIT)，又称x11协议。作者只想保留版权，而无任何其他限制。MIT与BSD类似，但是比BSD协议更加松，是目前最少限制的协议。这个协议唯一的条件就是在修改后的代码或者发行包包含原作者的许可息。适用商业软件。使用MIT的软件项目有:jquery、Node.js。

MPL(Mozilla Public License 1.1)

MPL协议允许免费重发布、免费修改，但要求修改后的代码版权归软件的发起者。这种授权维护了商软件的利益，它要求基于这种软件的修改无偿贡献版权给该软件。这样，围绕该软件的所有代码的版都集中在发起开发人的手中。但MPL是允许修改，无偿使用得。MPL软件对链接没有要求。

EPL (Eclipse Public License 1.0)

EPL允许Recipients任意使用、复制、分发、传播、展示、修改以及改后闭源的二次商业发布。

使用EPL协议，需要遵循以下规则：

- 当一个Contributors将源码的整体或部分再次开源发布的时候，必须继续遵循EPL开源协议来发布，不能改用其他协议发布.除非你得到了原"源码" Owner的授权;
- EPL协议下，你可以将源码不做任何修改来商业发布。但如果你要发布修改后的源码，或者当你再布的是Object code的时候，你必须声明它的Source Code是可以获取的，而且要告知获取方法;
- 当你需要将EPL下的源码作为一部分跟其他私有的源码混和着成为一个Project发布的时候，你可以整个Project/Product以私人的协议发布,但要声明哪一部分代码是EPL下的，而且声明那部分代码继续遵循EPL;
- 独立的模块(Separate Module)，不需要开源。

Creative Commons 知识共享协议

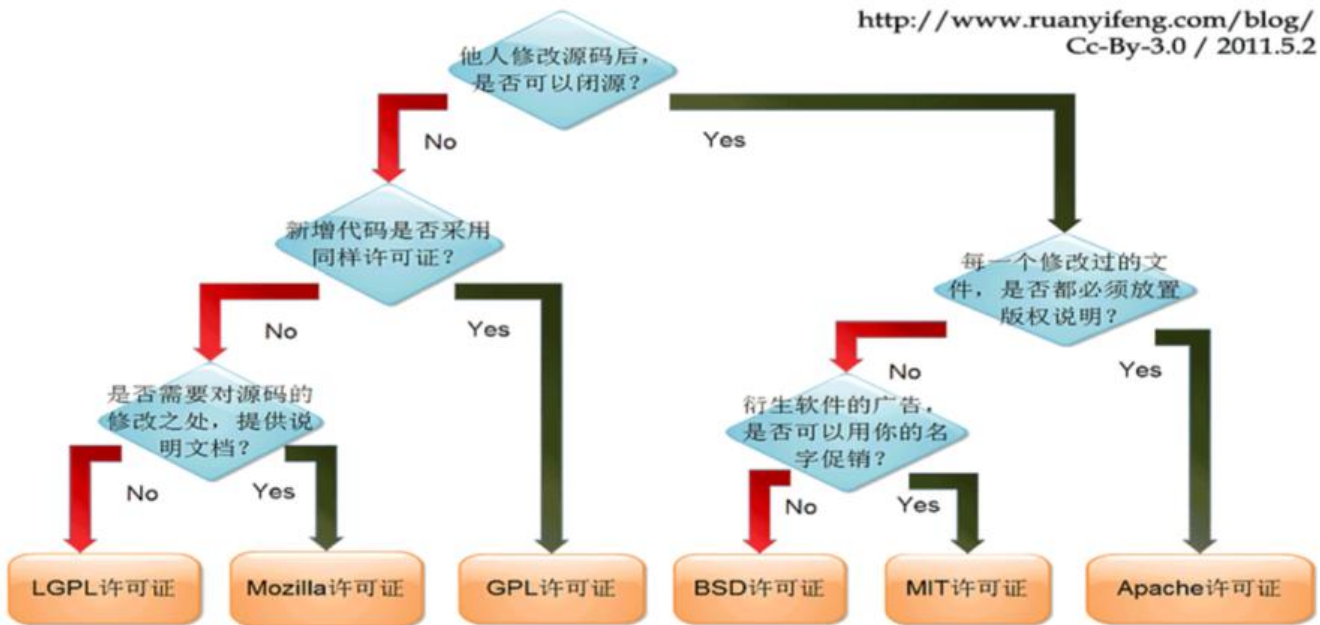
Creative Commons(CC)许可协议并不能说是真正的开源协议，它们大多是被使用于设计类的工程上CC协议种类繁多，每一种都授权特定的权利。一个CC许可协议具有四个基本部分，这几个部分可以独起作用，也可以组合起来。下面是这几部分的简介：

- 署名作品上必须附有作品的归属。如此之后，作品可以被修改，分发，复制和其它用途。
- 相同方式共享作品可以被修改、分发或其它操作，但所有的衍生品都要置于CC许可协议下。
- 非商业用途作品可以被修改、分发等等，但不能用于商业目的。但语言上对什么是"商业"的说明十分含糊不清(没有提供精确的定义)，所以你可以在你的工程里对其进行说明。例如，有些人简单的解释"商业"为不能出售这个作品。而另外一些人认为你甚至不能在有广告的网站上使用它们。还有些人认为"商业"仅仅指你用它获取利益。
- 禁止衍生作品

CC许可协议的这些条款可以自由组合使用。大多数的比较严格的CC协议会声明"署名权，非商业用途禁止衍生"条款，这意味着你可以自由的分享这个作品，但你不能改变它和对其收费，而且必须声明作品的归属。这个许可协议非常的有用，它可以让你的作品传播出去，但又可以对作品的使用保留部分完全的控制。最少限制的CC协议类型当属"署名"协议，这意味着只要人们能维护你的名誉，他们对的作品怎么使用都行。

CC许可协议更多的是在设计类工程中使用，而不是开发类，但没有人或妨碍你将之使用于后者。只是必须要清楚各部分条款能覆盖到的和不能覆盖到的权利。

图解分析



我只想安心的写代码
别人怎么搞就怎么搞吧

MIT协议

- 1、允许别人以任何方式使用。
- 2、署名作者。
- 3、原作者不承担代码使用后的风险。

我想保留专利
但你们可以随便用

Apache协议

- 1、授权使用者免费使用个人专利。
- 2、使用者必须放置协议说明。
- 3、使用者需要对修改部分进行说明。
- 4、禁止用作者的名号进行商业广告。
- 5、原作者不承担代码使用后的风险。

我希望的代码能生根发芽,
遍地开花

GPL协议

- 1、使用者需遵循相同的协议进行开源。
- 2、原作者不承担代码使用后的风险。