

大家应该都听到过开源许可协议，但是这玩意儿是干啥的，使用哪一种，有些人就犯嘀咕了。

下面我们来看下几种常用的开源许可。

GNU 通用公共许可(General Public License, **GPL**): 目前广泛使用的开源软件许可协议之一，用户享有运行、学习、共享和修改软件的自由。GPL最初是自由软件基金会创始人 Richard Stallman 起草的，其版本目前已经发展到了第3版。GPL的目的是保证程序员在开源社区中所做的工作对整个世界是有益的，所开发的软件也是自由的，并极力避免开源软件被私有化以及被无良软件公司所剥削。

现在，只要软件中包含了遵循GPL许可证的产品或代码，该软件就必须开源、免费，因此这个许可证并不适合商业收费软件。遵循该许可证的开源软件数量极其庞大，包括Linux内核在内的大多数的开源软件都是基于GPL许可证的。GPL赋予了用户著名的五大自由。

使用自由：允许用户根据需要自由使用这个软件。

复制自由：允许把软件复制到任何人的计算机中，并且不限制复制的数量。

修改自由：允许开发人员增加或删除软件的功能，但软件修改后必须依然基于GPL许可证。

衍生自由：允许用户深度定制化软件后，为软件注册自己的新商标，再发行衍生品的自由。

收费自由：允许在各种媒介上出售该软件，但必须提前让买家知道这个软件是可以免费获得的。因此，一般来讲，开源软件都是通过为用户提供有偿服务的形式来营利的。

较宽松通用公共许可证 (Lesser GPL, **LGPL**)：一个主要为保护类库权益而设计的GPL开源协议。与标准GPL许可证相比，LGPL允许商业软件以类库引用的方式使用开源代码，而不用将其产品整体开源，因此普遍被商业软件用来引用类库代码。简单来说，就是针对使用了基于LGPL许可证的开源代码，在涉及这部分代码，以及修改过或者衍生出来的代码时，都必须继续采用LGPL协议，除此以外的其他代码则不强制要求。

如果您觉得LGPL许可证更多地是关注对类库文件的保护，而不是软件整体，那就对了。因为该许可证最早的名字是Library GPL，即GPL类库开源许可证，保护的對象有glibc、GTK widget toolkit等类库文件。

伯克利软件发布版许可证 (Berkeley Software Distribution, **BSD**) : 另一款被广泛使用的开源软件许可协议。相较于GPL许可证, BSD更加宽松, 适合于商业用途。用户可以使用、修改和重新发布遵循该许可证的软件, 并且可以将软件作为商业软件发布和销售, 前提是满足下面3个条件。

如果再发布的软件中包含开源代码, 则源代码必须继续遵循BSD许可证。

如果再发布的软件中只有二进制程序, 则需要在相关文档或版权文件中声明原始代码遵循了BSD许可证。

不允许用原始软件的名字、作者名字或机构名称进行市场推广。

Apache许可证 (Apache License) : 顾名思义, 是由Apache软件基金会负责发布和维护的开源许可协议。作为当今世界上最大的开源基金会, Apache不仅因此协议而出名, 还因市场占有率第一的Web服务器软件而享誉行业。目前使用最广泛的Apache许可证是2004年发行的2.0版本, 它在为开发人员提供版权及专利许可的同时, 还允许用户拥有修改代码及再发布的自由。该许可证非常适合用于商业软件, 现在热门的Hadoop、Apache HTTP Server、MongoDB等项目都是基于该许可证研发的。程序开发人员在开发遵循该许可证的软件时, 要严格遵守下面4个条件。

该软件及其衍生品必须继续使用Apache许可证。

如果修改了程序源代码, 需要在文档中进行声明。

若软件是基于他人的源代码编写而成的, 则需要保留原始代码的许可证、商标、专利声明及原作者声明的其他内容信息。

如果再发布的软件中有声明文件, 则需在此文件中注明基于了Apache许可证及其他许可证。

MIT许可证 (Massachusetts Institute of Technology License: 源于麻省理工学院, 又称为X11协议。MIT许可证是目前限制最少的开源许可证之一, 用户可以使用、复制、修改、再发布软件, 而且只要在修改后的软件源代码中保留原作者的许可信息即可, 因此普遍被商业软件 (例如jQuery与Node.js) 所使用。也就是说, MIT许可证宽松到一个新境界, 即用户只要在代码中声明了MIT许可证和版权信息, 就可以去做任何事情, 而无须承担任何责任。

Mozilla公共许可证 (Mozilla Public License, **MPL**) : 于1998年初由Netscape公司的Mozilla小组设计, 原因是它们认为GPL和BSD许可证不能很好地解决开发人员对源代码的需求和收益之间的平衡关系, 因此便将这两个协议进行融合, 形成了MPL。2012年年初, Mozilla基金会发布了MPL 2.0版本 (目前为止也是最新的版本), 后续被用在Firefox、Thunderbird等诸多产品上。最新版的MPL公共许可证有以下特点。

在使用基于MPL许可证的源代码时, 后续只需要继续开源这部分特定代码即可, 新研发的软件不用完全被该许可证控制。

开发人员可以将基于MPL、GPL、BSD等多种许可证的代码一起混合使用。

开发人员在发布新软件时，必须附带一个专门用于说明该程序的文件，内容要有原始代码的修改时间和修改方式

大家看到这里，会新生疑虑：“说了半天，这不都差不多吗，还是不知道用哪个啊。。。”

说到这里一个外国程序员 整理了一个流程图，对前面我们说的几种许可证做了归纳，效果图如下：

