关于GNU GPL

这里简单介绍关于自由软件的一个重要协议:GNU GPL。具体信息需要参照官方网站(http://www.gnu.org/licenses/)。

主要内容：

1，基本概念

2，与其它协议的比较

3,发展历史

4，授权

5，其它

1，基本概念

==================

GNU 通用公共许可协议（英语：GNU General Public License，简称GNU GPL或GPL），是一个广泛被使用的自由软件许可协议，最初由理查德·斯托曼为GNU计划而撰写。此许可协议最新版本为“第3版”（v3），2007年6月29日发布。GNU宽通用公共许可协议（英语：GNU Lesser General Public License，一般简称LGPL）是改自GPL的另一个版本，其目的是为了应用于一些软件库。GPL给予了电脑程序自由软件的定义，并且使用“Copyleft”来确保程序的自由被完善的保留。

2，与其它协议的比较

==================

这里通过和其它的软件许可协议进行比较，大致了解GPL的特点。

GPL授予程序接受人以下权利，或称“自由”：

\* 以任何目的运行此程序的自由；

\* 再发行复制件的自由；

\* 改进此程序，并公开发布改进的自由（前提是能得到源代码）。

相反地，随版权所软件的最终用户许可协议几乎从不授予用户任何权利（除了使用的权利），甚至可能限制法律允许的行为，比如逆向工程。

GPL与其他一些更“许可的”自由软件许可协议（比如BSD许可协议）相比，主要区别就在于GPL寻求确保上述自由能在复制件及演绎作品中得到保障。它通过一种由斯托曼发明的叫Copyleft的法律机制实现，即要求GPL程序的演绎作品也要在GPL之下。相反，BSD式的许可协议并不禁止演绎作品变成专有软件。

采用LGPL之计划本身虽然仍有"著作权脱离"("Copyleft")之限制条件，但这些限制不感染仅仅只联结到本计划的软件。不过此等软件仍会受到其他限制。LGPL主要使用之标的为软件函式库("Software Libraries"),虽然其亦可使用于独立存在的应用程式。比较有名的例子为Mozilla跟OpenOffice.Org.LGPL有一特点是LGPL软件可以被转换成GPL(参见Section 3)。这种特性对于在GPL函式库或应用程式中直接使用LGPL程式有一定程度之帮助。

由于某些原因，GPL成为了自由软件和开源软件的最流行许可协议。到2004年4月，GPL已占Freshmeat上所列的自由软件的约75%，SourceForge的约68%。类似的，2001年一项关于Red Hat Linux 7.1的调查显示一般的代码都以GPL发布。著名的GPL自由软件包括Linux核心和GCC。

3,发展历史

==================

GPL由斯托曼撰写，用于GNU计划。它以GNU Emacs、GDB、GCC的许可协议的早期版本为蓝本。这些许可协议都包含有一些GPL的版权思想，但仅只针对特定程序。斯托曼的目标就是创造出一种四海之内皆可使用的许可协议，这样就能为许多源代码共享计划带来福音。GPL版本1就这样，在1989年1月诞生。到1990年时，因为一些共享库而出现了对比GPL更宽松的许可协议的需求。所以当GPL版本2在1991年6月发布时，另一许可协议——库通用许可协议（Library General Public License，简称LGPL）也随之发布，并记作“版本2”以示对GPL的补充。版本号在LGPL版本2.1发布时不再相同，而LGPL也被重命名为GNU宽通用公共许可协议（Lesser General Public License）以体现GNU哲学观。

个版本分别叙述如下：

GPLv1即GPL版本1，即最初的版本，发布于1989年一月，其目的是防止那些阻碍自由软件的行为，而这些阻碍软件开源的行为主要有两种（一种是软件发布者只发布可执行的二进制代码而不发布具体源代码，一种是软件发布者在软件许可加入限制性条款）。因此按照GPLv1，如果发布了可执行的二进制代码，就必须同时发布可读的源代码，并且在发布任何基于GPL许可的软件时，不能添加任何限制性的条款。

GPLv2即GPL版本2，到了1990年，人们普遍认为一个限制性弱的许可协议对于自由软件的发展是有战略意义上的好处的；因此，当GPL的第二个版本（GPLv2）在1991年6月发布时，与此同时第二个许可协议程序库GNU通用公共许可协议（LGPL,the Library General Public License）也被发布出来并且一开始就将其版本定为第2版本以表示其和GPLv2的互补性。这个版本一直延续到1999年，并分支出一个派生的LGPL版本号为2.1，并将其重命名为轻量级通用公共许可协议（又称宽通用公共许可协议）（Lesser General Public License）以反映其在整个GNU哲学中的位置。

GPLv3即GPL版本3，到2005年，GPL版本3正由斯托曼起草。这里不详细介绍了，在所有的改动中，最重要的四个是：

\* 解决软件专利问题；

\* 与其他许可协议的兼容性；

\* 源代码分区和组成的定义；

\* 解决数位版权管理（DRM） 问题。

4，授权

==================

1)GPL(GNU General Public License)的授权：

此GPL的条款和条件适用于任何收到GPL下的作品的人（即“许可协议接受人”）。任何接受这些条款和条件的许可协议接受人都有修改、复制、再发行作品或作品的演绎版本的授权。许可协议接受人可以对此项服务收取费用 ，反之亦然。这一点是GPL与其他禁止商业用途的自由软件许可协议最大的不同。Stallman认为自由软件不应限制其商业用途，同时GPL清楚地说明了这一点。但GPL又规定发行者不能限制GPL授予的权利。例如，这禁止对软件在单纯沉默（消极默示）式协议或合同下的发行。GPL下的发行者同时也同意在软件中使用的专利可以在其它GPL软件中使用。

GPL不会授予许可协议接受人无限的权利。再发行权的授予需要许可协议接受人开放软件的源代码，及所有修改。且复制件、修改版本，都必须以GPL为许可协议。这些要求就是copyleft，它的基础就是作品在法律上版权所有。由于它版权所有，许可协议接受人就无权进行修改和再发行（除合理使用），除非它有一个copyleft条款。如果某人想行使通常被法律所禁止的权利，只需同意GPL的条款。相反地，如果某人发行软件违反了GPL（比如不开放源代码），他就有可能被原作者起诉。

copyleft只在程序再发行时发生效力。对软件的修改可以不公开或开放源代码，只要不发行。注意copyleft只对软件有效力，而对软件的输出并无效力（除非输出的是软件本身）。不过这在GPL版本3中可能会有改动。

2)LGPL（GNU Lesser General Public License）的授权：

LGPL是GPL的一个为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

3)一些问题

一个关于GPL重要的争议是，非GPL软件是否可以动态链接到GPL库。GPL对GPL作品的演绎作品在 GPL下发布规定很明确。但是对于动态链接到GPL库的作品是否是演绎作品就规定得不清楚了。自由和开放源代码社区为此分成两派，自由软件基金会认为这种作品就是演绎作品，但其他专家并不同意。这个问题根本的并不关乎GPL本身，而是一个版权法如何定义演绎作品。美国联邦上诉法院第九巡回审判庭在 Galoob v. Nintendo案对演绎作品尝试定义，但最终没有明确的结果。不幸的是，许多开发者觉得这是个技术问题。但实际上这完全是法律问题。不过由于迄今为止没有案例表明有人以动态链接的方式来绕过GPL的条款或者并被起诉，动态链接的限制已经是事实上地（de facto）有效，不论它是否是法律上地（de jure）有效。

大多数自由软件许可协议，比如MIT/X许可协议、BSD许可协议、LGPL，都是“GPL兼容的”，即它们的代码与GPL代码混用无冲突（但新代码则是GPL下的）。但是有某些开源软件许可协议不是GPL兼容的。通常意见是开发者仅只使用GPL兼容的许可协议，以免法律问题。

GPL的批评者常常认为GPL是有“传染性”的“病毒”，因为GPL条款规定演绎作品也必须是GPL的。由于“演绎作品”通常被解释为包含GPL代码或动态链接到GPL库（如上）的软件，“病毒说”来源于GPL对于许可协议的强制继承的要求。这正是GPL与BSD式许可协议的哲学思想上的差异。 GPL的支持者确信自由软件世界应具有自我保护能力和可持续发展性——确保自由软件的演绎作品同样“自由”，但其他人认为自由软件应给予“所有人”最大的自由。

不同版本之间的 GPL 并不兼容。例如，当原始的作品以 GPLv2 发布，而补丁以 GPLv3 发布时，因为这样的原因，其编译之后产生的二进制版本不可以再行传播。因此，FSF通常会推荐以 “GPLv3 or later” 这样的形式来标示授权许可协议，或 GPLv2 + GPLv3 双许可协议以规避这一问题。

5，其它

==================

精确的信息，需要参考官方网站（参考资料有列出），这里做一个简单的总结：

1）GNU GPL

GNU General Public Licence (GPL) 有可能是开源界最常用的许可模式。GPL 保证了所有开发者的权利，同时为使用者提供了足够的复制，分发，修改的权利，包括：

\*可自由复制

你可以将软件复制到你的电脑，你客户的电脑，或者任何地方。复制份数没任何限制。

\*可自由分发

在你的网站提供下载，拷贝到U盘送人，或者将源代码打印出来从窗户扔出去（环保起见，请别这样做）。

\*可以用来盈利

你可以在分发软件的时候收费，但你必须在收费前向你的客户提供该软件的 GNU GPL 许可协议，以便让他们知道，他们可以从别的渠道免费得到这份软件，以及你收费的理由。

\*可自由修改

如果你想添加或删除某个功能，没问题，如果你想在别的项目中使用部分代码，也没问题，唯一的要求是，使用了这段代码的项目也必须使用 GPL 协议。

需要注意的是，分发的时候，需要明确提供源代码和二进制文件，另外，用于某些程序的某些协议有一些问题和限制，你可以看一下 @PierreJoye 写的 Practical Guide to GPL Compliance 一文。使用 GPL 协议，你必须在源代码代码中包含相应信息，以及协议本身。

2)GNU LGPL

GNU 还有另外一种协议，叫做 LGPL （Lesser General Public Licence），它对产品所保留的权利比 GPL 少，总的来说，LGPL 适合那些用于非 GPL 或非开源产品的开源类库或框架。因为 GPL 要求，使用了 GPL 代码的产品必须也使用 GPL 协议，开发者不允许将 GPL 代码用于商业产品。LGPL 绕过了这一限制。

3)关于GNU LGPL和GNU GPL

GNU计划在使用库时有两个首要的许可证.一个是GNU LGPL(库GPL);另一个是普通的GNU GPL.选择不同的许可有很大的不同:选择LGPL允许在私有程序中使用该库;选择普通的GPL则只允许在自由软件中使用它.

私软件开发者有金钱上的优势;自由软件开发者需要相互之间利用各自的优势.对一个库采用普通的GPL对自由软件开发者的优势要大于对私有软件开发者: 他们可以使用的库对于私有软件开发者是不可利用的.使用普通的GPL并不是对于所有的库都有好处.在某些情况下更有理由来使用LGPL.最常见的情况就是当一个自由库的特性可以很容易地被私有软件以其他可替代库来实现.在这种情况下,库不能给与自由软件任何特别的优势,因而最好还是为LGPL发布该库。这也就是为什么GNU C 库选择LGPL：有很多的其他C库，如果使用GPL发布该库,将迫使私有软件开发者不得不使用其它的库--对他们来说这不成问题,而对于GNU则有了麻烦.

参考资料：

http://zh.wikipedia.org/wiki/GPL

http://zh.wikipedia.org/wiki/GNU宽通用公共许可证

http://www.gnu.org/licenses/gpl-3.0.html

http://www.cnblogs.com/baizx/archive/2010/07/14/1777210.html

http://www.cnblogs.com/baizx/archive/2010/07/14/1777210.html

http://baike.baidu.com/view/606545.htm

作者：QuietHeart

Email:quiet\_heart000@126.com

日期：2011年7月25日