

# 关于 **GPL** 的几点断言

<http://baiy.cn>

白杨  
2010-2012

# 前言

首先，我要声明一下，我很赞同以及敬佩软件开源这项伟大而崇高的运动。我也很欣赏 RMS 的高尚情操。

同时，在多年的开发工作中，我也为很多开源产品提交了不少 bug 和 feature patch。其中最少 wxWidgets、crypto++、lzo、lz4、mongoose、sqlite 等项目明确地接受了我的 patch。

因此，我并不是个反开源人士，而且我自认为自己对开源的贡献虽然不大，但比起那些不做任何贡献，口头嚷嚷着支持开源，骨子里却是纯粹拿来主义的人强的多。

不过 GPL 霸气的 CopyLeft 病毒式传染策略有时确实让人遭不住，因此俺才在几年前有了以下的胡思乱想。

最后要声明的是，俺不是法律专家。迷信此文的一切后果由信徒自负，谢谢。

## 1. GPL 与 LGPL 兼容（已证实）

即：一个 LGPL 的项目可以依赖于一个 GPL 项目。例子：

★ linux kernel 是 GPL 的，但 glibc（链接到 kernel）是 LGPL 的。

之所以称 glibc “链接到” kernel 是因为我觉得通过 sysenter 之类的机器指令调用 kernel 代码与通过 call 指令调用其它 dll/so 中的代码并无本质区别。如果你还是想不通的话，可以试想一下在 ARM7 之类没有保护模式和 MMU 的处理器中，glibc 调用 kernel 的代码几乎是与普通函数调用相同的。此时 ARM 版的 linux kernel 仍然遵循 GPL 协议、而 glibc 不还是遵循 LGPL 协议吗？

已证实包括LGPL在内的很多协议均与GPL兼容，详情请参考：<http://www.gnu.org/philosophy/license-list.html>。因此，可实现一个遵循LGPL或其它兼容协议的代理库（stub lib）来解决所有GPL产品的依赖问题。但是这里需要注意GPL部分与非GPL部分要分别打包和分发，请参考：<http://www.gnu.org/licenses/gpl-faq.html#GPLInProprietarySystem>和<http://www.gnu.org/licenses/gpl-faq.html#GPLWrapper>。

## 2. GPL 不能间接生效

哪怕有代码上的依赖关系（例如：link 和功能调用），只要不是直接依赖一个 GPL 产品均无需遵循 GPL 协议。例如：

- ★ Linux 版的 ORACLE DB 依赖 glibc (LGPL), glibc 又依赖 linux kernel (GPL), 但显而易见, ORACLE DB 本身是闭源商业软件。
- ★ 事实上, 只要是个 User mode 的 Linux 应用, 基本都要依赖 glibc。但 linux 上还是有很多闭源商业应用的。

但是与第一条相同, 请注意应用的打包和分发方式。特别地, 如果一个应用通过 ODBC DM 之类的通用接口使用一个 GPL 的 ODBC Driver 这样的通用接插件, 那么它不需要遵循 GPL (请参考: <http://mariadb.woqutech.com/en/licensing-faq/index.html>)。

### 3. GPL 不能通过 IPC 生效

就算是同一个应用, 只要不是直接 link 或代码派生都不需要遵循 GPL 协议。例如:

- ★ 通过 Linux Kernel Web Server (kHTTPd, GPL) 访问网页的 IE 浏览器不需要遵循 GPL 协议, 即使它们共同组成了一个 B/S 架构的应用; 并且它们之间发生了 HTTP 调用 (HTTP 请求/响应)。
- ★ 市面上有不少基于 GPL 的 asp.net 应用。为此, 运行它的 IIS、.NET CLR 以及用来访问它的 IE 都不需要遵循 GPL 协议, 即使他们共同组成了一个 B/S 架构的应用; 并且它们之间发生了 HTTP 调用 (HTTP 请求/响应)。
- ★ 通过 Windows 访问位于 linux samba 服务器上的共享目录和打印机时, Windows 不需要遵循 GPL 协议, 即使它与 samba 一起组成了一个完整的文件共享应用; 并且他们之间发生了 RPC 调用。
- ★ 在 Linux KVM 或者 Xen 上运行 Windows 时, Windows 不需要遵循 GPL 许可协议, 即使它 (作为 Guest OS) 与 Linux KVM (作为 Hypervisor) 一起组成了一个完整的云计算应用。

因此, 可以将应用拆分为多个部分 (相互间使用 IPC 通信) 来避免核心技术泄露。最简单的例子就是在 C/S 架构或 B/S 架构的应用中, 客户端 (或浏览器端) 依赖 GPL 协议的组件并遵循 GPL 协议, 而 Server 端仍然可以保持私有 license。更多信息可以参考: <http://www.gnu.org/licenses/gpl-faq.html#NFUseGPLPlugins>; <http://www.gnu.org/licenses/gpl-faq.html#UnreleasedMods> 和 <http://www.gnu.org/licenses/gpl-faq.html#AGPLv3ServerAsUser> 等 (请注意 AGPL 是比 GPL 更为严厉的变体)。

### 4. GPL 仅从机器视角保证 CopyLeft

即: GPL 仅要求开放的源代码能够成功编译, 并没有指定代码必须遵守何种编

码规范，以何种风格写就。也没有强制要求 GPL 产品一定要有满足某种标准的软件说明书。

也就是说 GPL 允许你将所有代码写在一行里、代码注释率为 0%、代码中的变量名和函数名长的都像“idFF9A85B”之类、产品没有总体设计和详细设计文档或者这些文档里都只包含 1 个字等等。GPL 唯一保证的就是你提供的源码必须可以编译通过，而且编译产生的结果与事先提供的二进制版本一致。

因此，我们才说 GPL 仅从机器（编译器）视角保证开源。换句话说，你完全可以发布遵循 GPL 许可协议的，但对人类不可读的源代码。比如经过混淆和 minimize 的 C/C++/JavaScript 代码。

实际上，这一条结合第三点使用，就可以规避掉 GPL 带来的绝大部分问题。

## 5. 只有依赖 GPL 组件的部分需要遵循 GPL

模块 A 为 GPL；模块 B 为其它协议；模块 C 依赖模块 A 和模块 B 实现。此时只有模块 C 需要遵循模块 A 的 GPL 协议，模块 B 则不用。意即：GPL 协议只能依照依赖的方向进行传递，反向则不成立。例子不胜枚举，所有依赖着 Windows kernel32.dll、user32.dll 等模块的 GPL 组件均在此列。但与前文所述相同，请注意打包和分发的方式。