开源软件开发

曹东刚

caodg@sei.pku.edu.cn

Linux 程序设计环境

http://c.pku.edu.cn/

内容提要

1 开源软件

2 开源许可证

3 开源软件开发实践

软件分类

- 商业软件 (Business Software/Commercial Software)
- 共享软件 (Shareware)
- 公有软件 (Public Domain Software)
- 自由软件 (Free Software, Open-source Software)

自由软件

自由软件是一种版权法意义上的定义

体现在软件知识产权保护层面上. 使用者可以

- 可以自由使用该软件, 无论出于何种目的
- 自由学习该程序怎样工作, 并使之适应被许可人的需求
- 可以自由重新分发、复制以便帮助亲友
- 可以自由改善该程序, 并发布给公众, 让整个社会得利

自由软件 vs 开放源码软件

- 定义角度不同
  - 开源软件: 技术层面
  - 自由软件: 被许可的权利层面
- 许可证中对被许可人权利限制的严格程度不同
  - 开源软件: 稍宽
  - 自由软件: 严格
- 另一种说法为: 自由软件专指遵守 GPL 的软件, 自由软件包含于开源软件范畴

开源软件的认定机构

OSIA: Open Source Initiative Association[1]

- Eric S. Raymond 等人 1998 年成立的非营利性组织
- 将 OSI 申请为证明商标 (OSI Certified)
- 将 OSI 商标许可给经审核认定为开源软件的软件提供者
- 开源软件许可证都可以标明 OSI 商标, 从而得到开源社区认可
- 目前有 50 多个许可证

[1] http://www.opensource.org

开源软件及许可证的认定标准 –1

- Free Redistribution（自由发布）
- Source Code（对源代码的要求）
- Derived Works（演绎作品）
- Integrity of The Author's Source Code（保持源代码的完整性）
- No Discrimination Against Persons or Groups（不得歧视任何个人或团体）

开源软件及许可证的认定标准 –2

- No Discrimination Against Fields of Endeavor（不得歧视任何应用领域）
- Distribution of License（许可证的发布）
- License Must Not Be Specific to a Product（不得限制许可协议专属于某一个软件）
- License Must Not Restrict Other Software（许可证不能影响其他软件）
- License Must Be Technology-Neutral（许可证应保持技术中立性）

开源软件的著作权分析 —1

- 开放源码软件受著作权保护
- 开放源码软件作者并没有放弃任何权利, 只是有条件的将某些权利授予所有愿意接受条件的人
- 被许可人只有在发布了软件的修改版本或演绎作品时才承担相应的义务
- 开放源码软件受著作权保护

开源软件的著作权分析 —2

- 演绎作品问题
  - 演绎权: 改编、翻译、整理、注释、编辑
  - 对开放源码软件进行演绎, 必须获得原始著作权人的许可
  - 各种开源软件许可证之间的最大区别往往体现在对演绎作品的态度

内容提要

1 开源软件

2 开源许可证

3 开源软件开发实践

开源许可证的共同点

- 承认版权
- 发布的义务 —将获得的源代码再发布
- 对发布的源代码的要求 —须保证源代码的完整和可以被获得
- 允许修改 —可以根据获得的源代码产生演绎作品
- 没有担保

开源软件许可证的不同点

- 是否允许同其他非开源软件代码混合
- 是否必须公开修改后的程序
- 是否明确了专利许可授权
- 是否明确了专利侵权诉讼导致许可证协议终止
- 是否明确允许与函数库连接
- 是否只能按本许可证发布源代码
- 是否要求对于获得的源代码可能存在知识产权问题进行明确提示

开源软件许可证的比较

不同点对比	是否允许同非开源软件代码混合	是否可以将对源代码的修改不公开	是否明确了专利许可授权	是否明确了专利侵权诉讼导致许可证协议终止	是否明确禁止与函数库连接	是否只能按本许可证发布源代码	是否要求对源代码可能存在的知识产权问题明确提示
GPL	×	×	×	×	√	√	×
LGPL	√	×	×	×	×	×	×
BSD	√	√	×	×	×	×	×
MPL	√	√	×	×	×	×	×
Apache	√	√	×	×	×	×	×
Artistic	√	√	√	√	×	×	×

MIT 或 X Consortium

最宽松的许可证

- 在所有修改版本中保留版权和许可证条款
- 无限制的复制、使用、修改和再发行

MIT 许可示例

Copyright (c) 1898-2012 Peking University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

MIT 许可示例 (cont.)

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

BSD

比 MIT 稍微严格. 最先用于 BSD 4.4 版本上, 现被 Apache 和 BSD 操作系统等开源软件采纳。

- 在所有修改版本中保留版权和许可证条款, 并在广告和软件包相关文档中包含对所有程序开发者的致谢
- 无限制的复制、使用、修改和再发行

最新的 BSD 许可证删除了对广告的要求, 实际上等价于 MIT 许可证

新 BSD 许可示例

Copyright (c) 2010, Peking University (PKU).
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the PKU nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

新 BSD 版权声明示例 (cont.)

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MPL: The Mozilla Public License

- 允许将别人的源代码用于自己的商业行为
  - 在自己的源代码库上加一个接口，除了接口程序的源代码以 MPL 许可发布，源代码库中的源代码不必用 MPL 许可对外发布
- 可与其他类型的代码混合得到自己的软件
- 源码提供者不能提供受专利保护的源码，也不能在将代码以开源许可证许可后再申请专利
- 对修改的源代码以网络形式发布有时间上的要求：网页保留至少 12 个月

GPL: General Public License

- 自由软件联盟 GNU 的开源软件许可证的一种
- 开源软件领域最富盛名
- 对被许可人权利限制最严
- 最大限度保证软件自由

GPL 的理念: 在承认版权的前提下, 通过软件的版权许可来实现自由软件的自由权利的要求

- 复制权、发行权、修改权、翻译权

GPL 的传染性

- 凡是在逻辑上与 GPL 软件相联系并作为整体发布的程序中，只要有任何一部分代码是以 GPL 发布的，那么全部程序作为整体就必须接受 GPL 的约束
- 适用于 GPL 的程序代码，包括该程序以及由该程序衍生出来的“基于程序的作品”

GPL 被许可人的义务 —1

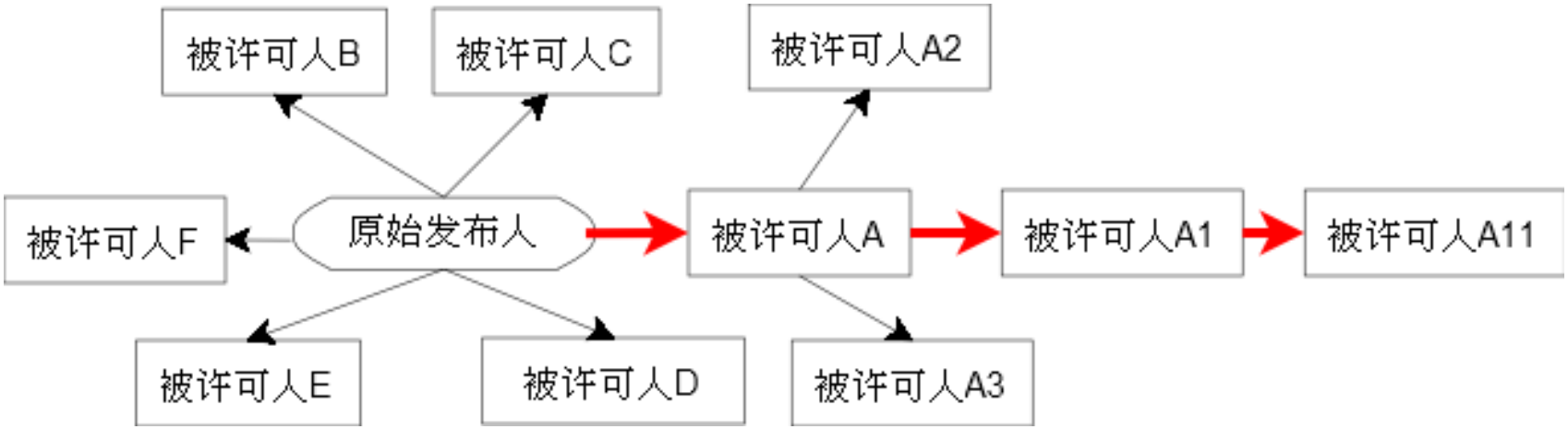
- 声明: 版权声明, 免责声明, 保证声明完整无损
- 修改说明: 在修改的文件中附有明确的修改说明以及具体日期
- 传递 GPL 条款: 当被许可人发布或出版作品时, 必须遵照 GPL 条款作为整体全部免费许可给任何第三方

GPL 被许可人的义务 —2

- 必要的提示：交互式程序运行前打印声明
- 发布时提供源代码
- 被许可人没有再许可的权利与义务：从原始著作人处获得权利
- 无担保和免责条款：非强制性

GPL 当事人及其关系的确定

- GPL 许可证的一方当事人始终是“原始许可证颁布者”, 即首先发布受 GPL 保护的程序的人
- GPL 许可证的另一方当事人是程序的下游被许可人, 他们无差别的享有原始发布者赋予的权利和承担相同的义务



LGPL: Lesser General Public License

主要针对函数库类型软件

- 允许其它应用程序与库进行连接
- 区分了“基于函数库的作品”与“使用函数库的作品”
  - 前者包含来自函数库修改过的源代码
  - 后者必须与函数库结合才能执行
- 对于选用 LGPL 许可证公开的源代码，可以变更为适用 GPL 许可证；但不能反向变更

开源软件的侵权纠纷

- 侵犯 OSIA 及许可人的商标权
- 发布不享有版权的软件产生的版权纠纷
- 发布侵犯他人专利权产生的专利权纠纷

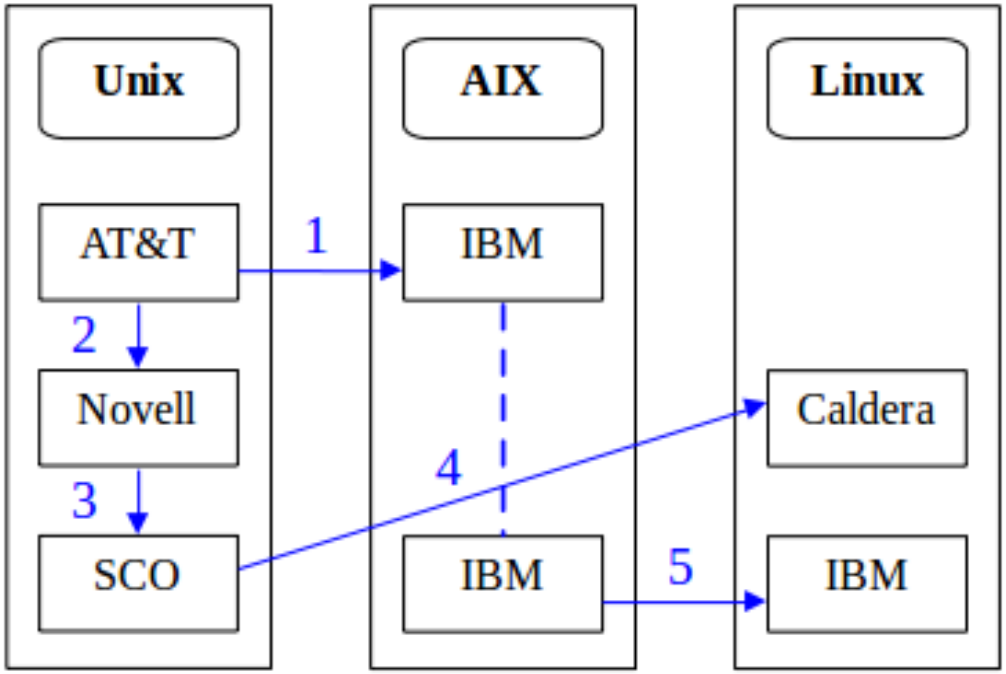
开源软件的违约纠纷

- 发布软件时未标明版权信息及必要的修改信息
- 发布软件时没有附上相应的许可证
- 源代码的提供不符合许可证的要求
- 违反许可证的规定将软件代码与其他代码混合
- 发布软件时违反许可证规定收取不适当的费用
- 违反许可证的规定以开源软件申请专利
- 许可证间的冲突带来的违约行为

开放源码司法案例: SCO vs IBM vs Novell

- 2003 年 3 月美国犹他州盐湖城法院, SCO 诉 IBM 侵权 Unix, 后相继起诉 Novell, AutoZone, DaimlerChrysler
- SCO 警告财富榜 1000, 全球 500 强, 全部 Linux 用户（赔偿 1399$/处理器）
- IBM 和 Red Hat 公司反诉讼
- SCO 将 Unix 代码授权给微软
- SCO 变换指控, 案情越来越复杂
- 2006, 法官驳回 SCO 的 294 条主张中的 182 条
- 2007 和 2010, 法院两次判决 Novell 是 Unix 版权拥有者
- SCO 破产,Nasdaq 摘牌

SCO vs IBM vs Novell: Unix 版权



自由和开源社区的反应

- SCO 甚至并不拥有那些有争议的源码
- Linux 不大可能会有 Unix 的代码
- Linux 采用 SCO UNIX 代码无意义
- 即使 Linux 和 SCO UNIX 某代码相似, 也并不能证明其来源于 SCO
- SCO 之前是 Linux 厂商, 可能从 Linux 复制了代码
- 即使 Linux 从 SCO 复制了代码, 但 SCO UNIX 代码之前已经在没有保密约定的情况广泛传播了, 不存在商业泄密问题
- 即使 Linux 有 UNIX 代码, SCO 也没权利以泄漏商业秘密或侵权为由起诉 IBM

案件背后的问题

- GPL 的法律效力
- 开源软件代码规范性
- 版权陷阱
- 专利威胁
- 商业化
- 自主知识产权

内容提要

1 开源软件

2 开源许可证

3 开源软件开发实践

开源软件开发的规则 —1

- 源码公开
  - 代码和过程公开, 鼓励同行复审
- 尽早发布, 经常发布
  - 确保第一次发布能编译和运行, 实现了承诺的功能
  - 缩短和加速用户与开发者之间的反馈

开源软件开发的规则 —2

- 对贡献者致谢并表扬
  - 即使有物质奖励, 也别忘了精神表扬
  - 核心团队特权最小化
- 遵循 Unix 尽可能自动化的传统

协同开发的最佳实践

开源软件开发的最佳实践实际上就是分布式协同开发实践

- 良好的修补实践
- 良好的项目命名实践
- 良好的开发实践
- 良好的发行制作实践
- 良好的交流实践

良好的修补实践 –1

- 发送补丁而不是完整的档案包或文件
  - 接受补丁的开发者可能已经更改了主干版本
  - diff 文件节省时间
- 如果补丁被拒绝, 不要太在意
- 发送针对当前版本代码的补丁
  - 补丁提交者应当跟踪主干代码库

良好的修补实践 –2

- 不包含可生成文件的补丁
  - 如 Bison 或 Flex 产生的 C 文件, 以及 autoconf 产生的 configure 脚本
- 使用-c或者-u 格式的 diff 文件
  - 缺省的 diff 输出 (-e) 不包含上下文, 非常脆弱
- 在补丁中包含手册页和其它文档文件
  - 尤其是补丁增加了用户接口部分或改变了软件功能特征

良好的修补实践 –3

- 在补丁中包含解释 (告诉维护者)
  - 谦逊而自信地说明为什么补丁是必须和有用的, 例如:

    “This patch solves the immediate problem, but I realize it complicates the memory allocation in an unpleasant way. Works for me, but you should probably test it under heavy load before shipping”

良好的修补实践 –4

- 在代码中包含有用的注释
  - 糟糕的注释:

    ```
    /* norman newbie fixed this 13 Aug 2001 */
    ```

  - 良好的注释:

    ```
    /*
     * This conditional needs to be guarded so that
     * crunch_data() never gets passed a NULL pointer.
     * <norman_newbie@foosite.com>
     */
    ```

良好的命名实践 –1

GNU 风格命名法: 主干 + major.minor.patch 的编号法

1. project prefix (全小写, 只包含字母和数字)
2. dash
3. version number
4. dot
5. “src” or “bin” (optional)
6. dot
7. binary type and options (optional)
8. archiving and compression extensions

GNU 命名示例

一个叫 “foobar” 的项目, 主版本号 1, 次版本号 2, 补丁级别 3, 则

foobar-1.2.3.tar.gz	√
foobar123.tar.gz	×
foobar1.2.3.tar.gz	×
foobar-v1.2.3.tar.gz	×
`foo_bar-1.2.3.tar.gz`	×
FooBar-1.2.3.tar.gz	×
foobar-1.2.3.bin.i386.tar.gz	√

良好的命名实践 –2

- 采用适合计算机程序可以解析和理解的规则模式
- 尊重适当的本地项目和社区的约定
  - 例如, Apache 的模块命名: `mod_foo` 合法
- 选择一个唯一的、容易读写和记忆的项目名称.
  到下述站点作名称搜索

  ibiblio: http://www.ibiblio.org/pub/Linux/

  freshmeat: http://www.freshmeat.net

  sourceforge: http://www.sourceforge.net

良好的开发实践 –1

- 不依赖专有语言、函数库或其他代码
  - 开源开发者不相信他们无法评审的源码
- 使用 GNU 自动工具处理移植性问题

  `configure; make; make test; make install`
  - 使用autoconf, autoheader 或automake
  - 安装软件时尽可能少地向系统询问配置信息

好的开发实践 –2

- 先测试再发布代码
  - 建立强大易用的测试框架, 便于进行回归测试
  - 将测试用例 (test suite) 和代码一起发布
- 代码发布前进行健全检查, 清除易忽略的错误
  - 用gcc编译 C/C++ 程序时打开 -Wall 选项, 清除所有的警告
  - 使用查找内存泄露和运行时刻错误的软件, 如 Electric Fence 和 Valgrind

好的开发实践 –3

- 发布前对文档和 README 进行拼写检查
- C/C++ 移植性实践
  - 对 C 语言应完全使用 ANSI 功能特征
  - 不使用编译器的专门特征
  - 需要移植的代码被隔离在一个单独区域 (如 os 子目录), 创造一个分离的移植层
  - 在移植层外尽量少用`#ifdef`和`#if`
  - 选择并严格坚持一个编码规范

良好的制作发行实践 –1

- 遵从标准的文件命名实践
- 确保打包文件总是解包到一个单一的新目录下

  例: `foo-0.23.tar.gz` 的文件包应该解压到`foo-0.23` 目录
- 为可升级性设计
  - 考虑支持多个软件版本在同一系统中共存, 尤其是库

良好的制作发行实践 –2

- 项目根目录下包含如下文件

README	最先被阅读的重要文件
INSTALL	配置、编译和安装向导
HISTORY	项目历史
COPYING	项目许可证条款 (GNU 惯例)
FAQ	项目常见问题解答的纯文本文档
AUTHORS	项目贡献者列表 (GNU 惯例)
NEWS	最近的项目新闻
CHANGES	修订版本之间重大更改的日志

良好的制作发行实践 –3

REDAME 文件应该短小精悍, 包含:

1. 项目的简短描述
2. 指向项目站点的 WEB 链接, 项目邮件列表地址
3. 开发者编译环境的注意事项以及潜在的移植性问题
4. 重要文件和子目录的描述
5. 项目贡献者的列表
6. 编译/安装指令或指向同样内容的文件 (INSTALL 文件)
7. 项目的最近新闻或指向同样内容的文件 (NEWS 文件)

良好的制作发行实践 –4

- 为 Linux 提供 RPM, deb 等预安装包
  - 由 Makefile 自动生成
- 提供二进制文件的校验和, 允许人们验证文件和正确性
  - 可用命令: sum, cksum, md5sum, gpg
  - 在项目网页上列出发布的二进制文件的校验和及其生成命令

良好的交流实践 –1

- 在 freshmeat 上或主题新闻组 (usenet) 发布通告
  通告应该言简意赅, 说明软件的目的
- 建立项目网站, 至少包含如下内容
  - 项目说明 (项目建立的理由, 面向的群体等)
  - 项目源码/二进制码的下载链接
  - 如何加入项目邮件列表的指导
  - 常见问题回答列表 (FAQ)
  - 项目文档的 HTML 版本
  - 相关或竞争项目的链接

良好的交流实践 –2

- 提供项目邮件列表. 对于项目 foo
  - 私有的开发者列表: foo-dev@mail.domain
  - 公开的通告列表: foo-announce@mail.domain
- 发布到主要的档案站点
  - ibiblio: http://www.ibiblio.org/pub/Linux/
  - freshmeat: http://www.freshmeat.net
  - sourceforge: http://www.sourceforge.net
  - 专门站点

选择合适的许可证

- 发布于公共域, 无许可证
- MIT 或 X Consortium 许可证
- BSD 许可证
- Artistic 许可证
- GNU GPL 和 LGPL 许可证
- Mozilla 公共许可证
- Apache 许可证