

Debian & Packaging

LI Daobing (Debian Developer)

lidaobing@gmail.com

Who Am I?

LI Daobing <lidaobing@gmail.com>



- Debian Developer
- 盛大创新院云计算分院后台开发
- 参与过的其他开源项目
 - 中文维基百科(前管理员)
 - translationproject.org
 - [iso-codes](http://iso-codes.org)
 - [chmsee](http://chmsee.org)(前维护人员)
 - [i18n-zh](http://i18n-zh.org)
 - 写过一些没人用的小库, 小程序 (github, google code)

Outline

- 我们为什么需要一个发行版？
- 软件如何进入 Debian ？
- 如何成为 Debian Developer？
- 近期 Debian 的打包方法的改进情况

为什么我们需要一个发行版

- 发行版能帮你安装一个系统
 - 一张光盘，20分钟，就可以帮你安装好一个系统
 - 如果玩 LFS(Linux From Scratch), 大约需要一个星期



发行版能帮你安装软件

- 如果你自己安装软件
 - 先要搜索并下载软件
 - 开始编译，发现缺乏依赖，编译不过
 - 搜索依赖
 - 下载，编译，安装依赖
 - 开始编译，编译失败，作者说 `gcc` 版本不对
 - 修改代码，或者安装一份新的 `gcc`
 - 编译成功了（或者失败了）

如果你使用发行版

- Debian/Ubuntu:
 - `sudo apt-get install gromacs`
- Fedora/CentOS:
 - `sudo yum install gromacs`

实际上自己安装软件会面临更多的问题

- 上游已经不再维护软件了，甚至找不到官方下载
- 库版本冲突
- 编译时间过长：比如 boost, libreoffice, GNOME, KDE, ...

发行版提供快速低风险的安全升级

- 不用紧盯安全通知(CVE)
- 不用考虑如何将安全补丁移植到老版本
- 更方便地在大集群上部署补丁

提供清晰的授权声明

- 很多软件没有提供清晰的授权声明
 - 作者不注意版权，使用了有版权问题的文件
 - 文件来源复杂，授权不一，但没有整合说明
 - 没有使用常见的授权协议，而是单独的协议
 - 究竟能不能用？
- 对于 Debian/Ubuntu
 - **main**区的可以无限制地使用，修改，再分发
 - 直接查看 `/usr/share/doc/<pkgname>/copyright`

发行版的其他优点

- 社区, 有问题可以去社区的 IRC, 论坛询问, 或者直接报 bug
- 一致的软件布局
 - `/usr/bin`, `/var/lib`, `/var/log`, ...
- 辅助文件
 - 启动文件: `init.d/`
 - 日志滚动: `logrotate`
 - `menu`
 - `.desktop`
 - `autostart`
- 集中力量, 推进 Linux 的改进

Debian 发行版的特点

- Debian 是纯社区支持的发行版，没有一个公司主导
- Debian 是同时发布源码包和二进制包，所以与 gentoo 不同，他受到更多的限制，某些 gentoo 能采用的软件不能放到 debian 里，比如很多游戏，gentoo 其实只发布了安装脚本。
- Debian 要求main 区软件满足 DFSG，其中第8款说授权必须是针对所有人，不能只针对 Debian，所以 Debian 也不能去跟中易谈字体授权。带来的好处是由于没有法律风险，Debian 的衍生版本特别多。

软件进入 Debian 的流程

- 评估软件进入 Debian 是否合适
- 对软件进行打包
- 交给他人 review, 确认打包没有问题
- 由 DD/DM 进行上传
- **ftp-master** 审察, 确认是否有违反授权(仅首次)
- 进入 unstable
- 没有问题的话, 10天后进入 testing
- 下次发布稳定版本时进入 stable

什么样的软件可以进入 Debian

- 首先是授权协议，所有进入 Debian 的软件都必须满足 DFSG，值得注意的是，DFSG 与自由软件的定义有不一致的地方，比如 cc-by-nc 协议就不是 DFSG 兼容的，部分 GFDL 文档(含有不可变文本)也不满足 DFSG，如果你对软件授权是否满足 DFSG 拿不准，那么可以到 [debian-legal](#) 咨询
- 上游是否仍然在维护这个软件，这个尽管不是强制标准，但除非这个软件不可替代(比如 mutt?)，否则一般不会收纳这种软件.
- 与现在仓库中同类软件相比，这个软件的特点是什么？这条就比较容易达到，毕竟哪个软件没有自己的特点呢？功能，基础库(比如 GTK vs QT)，速度，大小，等等都可以是自己软件的特点。

什么样的软件可以进入 Debian

- 软件质量如何？是否经常崩溃？
- 软件的用户数多少？几乎没人使用的软件很难保证质量。
- 如果没有 Debian 打包流程，用户直接使用是否方便，比如不依赖于本地库的 firefox 插件就倾向于不要打包，因为安装起来实在是太方便了
- 如果你在打包一个库的话，这个库是否有程序使用？如果没有程序使用，那么这是否是一个重要的开发工具包？如果都不是话，那么一般不会收纳这种库。
- 打包人员自己是否在使用这个软件

软件如何进入 Debian

- 首先你应该提交一份 ITP bug, ITP 的含义是“Intent To Packaging”, 最简单的方式是运行 “reportbug wnpp”, 接着选择 “ITP”, 填完他要求的内容, 即可提交, 这个 bug 报告最主要的作用是为了防止重复劳动, 可以追踪谁在负责这个包, 也方便直接联系负责人询问进度, 所以也没有必要太详细, 范例可以参见 ibus 的 ITP bug: <http://bugs.debian.org/501106>
- 稍后你可以收到一封邮件, 里面应该有这个 bug 对应的号码, 比如 ibus 的 ITP bug 号就是 501106, 这个号码在第3步打包时会用到(放在 debian/changelog 文件中)

软件如何进入 Debian

- 接着就是具体的打包工作了, 这个部分内容比较多, 不在这儿说了, 详细情况你可以查看Debian 新维护人员手册(英文版, 简体中文版)。打好的包首先要保证能用, 同时最好用 pbuilder 编译一遍, 确定能编译成功, 用 lintian 检查一遍, 确认没有警告, 然后再复查一下 debian/copyright 里面的版权声明, 是否有遗漏和错误, 这些都检查完毕后, 就可以申请上传了。
- 上传到 mentors.debian.net

软件如何进入 Debian

- 接着呢你要发封信到 `debian-mentors@lists.debian.org`, 找一个Debian Developer 帮你上传, 这个人一般被称之为 sponsor, 这封信的标题一般含有“RFS”, 含义为“Request For Sponsor”。`mentors.debian.net` 一般已经为你准备了一个模板, 就在你的包详细页面下面会有一个链接, 你把模板拷贝一份, 补充一下里面缺少的内容, 就可以发到 `debian-mentors@lists.debian.org`, 值得注意的是, 如果你没有订阅这个邮件列表, 记得在邮件里面加上一句“please cc me, thanks.”。

软件如何进入 Debian

- 接着呢就需要等待几天时间，当然如果运气不好，没人对你的包没有兴趣的话，可能需要更长的时间，一般两周后你可以重新发一遍你的邮件，补充说明一下为什么你的包值得进入 Debian. 如果你想加快这个进程，一个办法是到 IRC 问问，IRC 服务器是 `irc.debian.org` (不是 `freenode`)，频道是 `#debian-mentors`。另外，如果你有认识的 Debian Developer 的话，可以直接给他发信，问他是否愿意帮你上传这个包。
- 包上传后就进入 Debian 的 `unstable`了(即`sid`)，然后一般会在 10 天后进入 `testing` (需要满足如下的要求：a. 包在所有的平台上都成功编译，b. 这10天内没有发现重大 `bug`，c. 你所依赖的包也已经都在 `testing` 存在了)，下一个 `stable` 版发布时就会包含你的包了。

如何成为 Debian Developer

1. 开始使用 Debian

2. 开始参与 Debian 的开发

- 给 Debian 反馈 bug
- 协助 Debian 的翻译工作
- 参与包的维护工作

1. 申请 Debian Maintainer

- 有 DD 帮你做 gpg 签名
- 有一定的包维护经验
- 有一个 DD 愿意推荐你
- 能跟社区友好地交流

如何成为 Debian Developer

- 申请 Debian Developer
 - 成为 DM 后的工作能被大家认可
 - 有一位 DD 愿意推荐你
 - 测试你对Debian社区的理念和方法的认识程度
 - 测试你的打包技能
 - 其他事务审察
 - 成为 DD

Debian 打包方法的演进

原始版本: 逐个写规则

```
#!/usr/bin/make -f
```

```
build: ...
```

```
clean: ...
```

```
install: ...
```

```
binary: ...
```

```
binary-arch: ...
```

```
binary-indep: ...
```

问题: 过于复杂,

Debian 打包方法的演进

- CDDBS: 简化打包方法

```
#!/usr/bin/make -f
```

```
include /usr/share/cdbb/1/rules/debhelper.mk
```

```
include /usr/share/cdbb/1/class/cmake.mk
```

- 通过定义一些变量可以做一些定制，但不够灵活

Debian 打包方法的演进

- debhelper 7: 再次简化

```
#!/usr/bin/make -f
```

```
%:
```

```
dh $@
```

```
override_dh_strip:
```

```
dh_strip -Xfoo
```

其他方面的改进

- **Debian format 3.0**

- 新的补丁体系，取代 dpatch, .diff.gz
- 去除上游提供的 debian 目录（带来的麻烦很多）
- 支持 .gz 之外的上游格式
- 支持多个上游文件

- 二进制包支持 .gz 之外的压缩格式，比如 .bz2, .lzma, .xz

- 进一步的标准化

- 较早的：UTF-8, IPv6, i18n
- 引入 binutils-gold
- ...

Summary

- 玩 Linux 的人需要一个发行版
- **Debian** 是一个社区发行版，需要来自社区的支持
- 给 Debian 打包很简单
- 成为 Debian Developer 也不难
- **Debian** 也在持续改进

多余的话

- 加入一个社区，为社区贡献的时间是因为自己认同社区的理念，而不是因为想给社区灌输自己的理念。wikipedia 里边这种反面例子很多，Debian 最近的“蝶变”事件也是类似。
- 我愿意为 wikipedia, Debian, iso-codes 这类的项目贡献时间，是因为他们在为人类创造价值，并且通过一个非常宽松的授权来放大这个价值。细分下来，这些项目都是大家都用得上，以前没人做过，有了这个项目，其他人不必重复造轮子。
- 在做贡献的时候也不要忘了自己，不要让你的贡献过多妨碍你的职业规划（不过现在支持开源的公司越来越多，所以把这种贡献当成职业规划的一部分也未尝不可 :)

Thanks for your attention