

Debian 及自由软件社区简介

杨博远
(Boyuan Yang)
byang@debian.org

Debian Project

2020 年 8 月 7 日

个人简介

- Boyuan Yang <byang@debian.org>
- Debian 开发者 (“Debian Developer”, 2018 年至今)
- 主要关注方向:
 - 输入法相关软件 (fcitx、ibus)
 - 软件的国际化和本地化 (i18n/l10n)
 - Deepin 相关软件
 - Python 软件库与 Python 所编写软件的打包
 - 整体质保

本次演示文稿可以在 <https://github.com/hosiet/talk-debian-2020> 找到。

目录

- 1 Debian 简介与社区文化
 - Debian 简介
 - 社区文化与特点
- 2 Debian 是如何开发的？
 - 重要的 Debian 站点和服务
 - 正确理解 Git 打包仓库 (Salsa GitLab)
 - 有关 DD/DM 身份
- 3 与开源社区协作的方法、技巧、注意事项
 - 常用缩写和“黑话”
 - 社区协作技巧
- 4 Linux 桌面中文本地化情况简介
- 5 Q&A 问答

Debian 简介

- 最古老经典的 Linux¹发行版之一：始于 1993 年（Linux v0.01 于 1991 年发布）；
- 成为了许多衍生发行版的基石：Ubuntu、Deepin、Linux Mint 等；
- 承诺所包含软件全自由：严格遵循《Debian 自由软件指导方针》²，所有源代码可下载可在线浏览³；
- 开发了强大的软件包管理工具：apt 和 dpkg；
- 完全由社区运营维护，发展方向由 Debian 正式开发者（Debian Developer）经讨论与投票决定，不受商业公司控制；⁴
- 大约每两年发布一个稳定版，每个版本支持时间约为五年；下一个大版本会是 2021 年 5 月的 Debian 11（Bullseye）。

¹Debian 同时有将 Linux 内核替换为 FreeBSD 内核的实验性版本。

²某些硬件的闭源驱动和固件、微码等使用非官方的 non-free 部分提供，默认安装不启用

³可以在 sources.debian.org 浏览 Debian 全部 main 部分的软件包源代码。

⁴法律支持和款项管理由非营利性质的 SPI 基金会提供支撑。

Debian 的社区文化和特点

- 对法律上的问题十分严肃，这尤其体现在对软件包的许可证审查上（通常会精细到源代码文件为单位进行审查）；
- 开发流程上比较严谨⁵；
- 软件包维护传统上有较强的“维护者强所有关系（strong package ownership）”的倾向，但近些年有将软件包由多个维护者共同维护或团队维护的趋势；这个特点对下游发行版向 Debian 反哺贡献的效率有部分影响。
- 以志愿者模式参与开发和维护：Debian 不是公司，大部分成员都是兼职维护 Debian 软件包/维护基础设施，少部分成员由其他公司雇佣并将以志愿者身份参与 Debian 开发作为工作一部分（如 Debian 中存在 Canonical 公司⁶员工等）。

⁵换句话说，流程稍显繁琐。

⁶Canonical 开发并维护 Ubuntu，Ubuntu 是 Debian 的衍生发行版。

Debian 的社区文化和特点（二）

- 主要以协商和讨论的方式决定开发方向；技术问题争论无果时会由数位资深开发者组成的技术委员会（“CTTE”，Technical Committee）作出决断，重大技术/非技术问题会由全体投票的方式决定方案⁷；
- 主要使用邮件列表、电子邮件和 IRC 进行交流；
- 在可能的情况下尽量给用户多种选择，软件功能在合理的前提下尽可能多地启用。例如：
 - 提供多种初始化（init）系统的选择；
 - 同时提供并维护多种桌面环境的软件套件；
 - 软件库的构建 flag 能启用的会尽量启用（如果发现有缺失的功能可以向维护者建议修复）；
 - 然而某些部分严格遵循标准，如 CPU Baseline（AMD64 不会启用 SSE3/SSE4/AVX）⁸；

⁷<https://www.debian.org/vote/>

⁸但 Debian 为特殊的软件包提供了变通方式，详见 `isa-support` 源码包内容。

Debian 常用缩写

Debian 社区语境下有很多不太通用但时常出现的英文缩写，初看可能无法理解其含义，但在社区内工作时需要有一定了解。

DD 即 “Debian Developer”，Debian 社区正式成员，其 PGP Key 登记在案，有内部投票权、被选举权、可以无限制向 Debian 软件仓库上传已有软件的新版本、有内网机器 SSH 和 LDAP 帐号等；

DM 即 “Debian Maintainer”，其 PGP Key 登记在案，只能对某些特定的软件无限制地上传新版本（特定软件上传权限由其他 Debian Developer 授予），除此之外无任何权利；

Salsa 即 salsa.debian.org 站点，是 Debian 自建的 GitLab CE 实例；

DDPO 即 “Debian Developer’s Package Overview”，可以在一个网页上显示某个维护者名下或者某个团队名下所有的软件包列表、健康程度和版本信息。页面位于 qa.debian.org/developer.php。⁹

⁹<https://wiki.debian.org/DDPO>

Debian 常用缩写 (二)

- WNPP** 即 “Work-Needing and Prospective Packages”，有一个信息聚合页面来显示无人维护、需要帮助和需要打包的软件包。站点是 wnpp.debian.net。¹⁰
- RFA** 即 “Request For Adoption”，通常是某位维护者对某个软件包的维护已力不从心，通过 RFA 请求其他人接手维护软件包。
- ITP** 即 “Intent To Package”，是打算打包某个软件的声明；
- RFS** 即 “Request For Sponsorship”，已经完成某个软件的打包需要某位 Debian 开发者协助审查并上传至官方仓库（Debian Archive，不是 Debian 所使用的 GitLab 仓库）；

值得一提的是 ITP/RFS 的流程仅适用于非 Debian 开发者；Debian 开发者（DD）可以直接向 Debian 软件仓库上传软件。

¹⁰<https://www.debian.org/devel/wnpp/>

Debian 是如何开发的？

作为一款完整功能的 Linux 操作系统，Debian 和衍生发行版不同，并没有所谓“上游发行版”。了解 Debian 的开发 workflow 有益于了解一个操作系统应当如何独立存在并维持。

最最核心的内容无非如下三点：

- 以软件包（deb 包）为核心：从系统内核到软件库再到各种软件，所有系统文件均由 .deb 软件包提供；
- Debian 开发者和软件包维护者（即打包人员）从软件原作者处获取源代码，结合自己编写的打包指令便可使用打包工具链生成新的 .deb 包；
- 打包人员在本地测试通过后将软件源代码和打包指令数字签名，以源码包的形式上传¹¹至 Debian FTP Master¹²服务器，服务器端会自动验证数字签名、权限并为不同硬件架构构建 .deb 包、将构建成果放入软件源并随镜像网络分发至全球各地。

¹¹请注意，默认情况下是不会上传 .deb 包的。

¹²请将其理解为“文件服务器”，FTP 的名字是历史遗留下来的。

Debian 是如何开发的？(二)

初次引入软件包，以及后续维护

初次引入新软件包需要由 Debian FTP Masters 团队进行审查，确保软件许可证符合要求（是自由的）且软件质量达标；初次审查通过后，后续更新软件包不再需要 FTP Masters 审查，可以由 DD 自行上传更新。

日常工作：监视上游新版本并进行打包，修复软件包存在的 Bug。

监视上游新版本 只要在打包时正确编写了 `debian/watch` 文件，稍后在 Debian DDPO 和 Debian Maintainer Dashboard 站点上会自动提示新版本需要打包；

修复 Bug 软件包列明的维护者会以电子邮件形式收到用户提交的 Bug 报告，维护者需要相应进行处理。所有 Bug 报告也可在 tracker.debian.org 或 bugs.debian.org 网页上查询到。Bug 提交者、软件包维护者和所有其他人士均可以回复电子邮件的形式在 Bug 报告下留言。

Debian 是如何开发的？(三)

软件包的团队维护

某一类软件包可能有相似的特性，它们通常由一个专门存在的团队进行维护。这里稍举几例：

- Debian GCC Maintainers
- Debian Multimedia Maintainers
- Debian X Strike Force
- Debian Wine Party
- Debian Fonts Task Force
- Debian Python Modules Team
- Debian Rust Maintainers
- Debian Java Maintainers
- Debian Deepin Packaging Team
- Debian Desktop Themes Team
- Debian Chinese Team
- Debian Cloud Team
- Debian Qt and KDE Team
- Debian+Ubuntu MATE Packaging Team

如需修改团队旗下的软件包，与团队取得联系是很重要的；一般这些团队都有属于自己的邮件列表和 GitLab 团队，可以通过查看邮件列表存档的活跃人员或者 GitLab 团队成员名单了解应当与谁联系。

重要的 Debian 站点和服务

完整列表请见<https://wiki.debian.org/Services>

www.debian.org 网站主页

wiki.debian.org Debian 维基，提供网站主页文档的补充内容

salsa.debian.org Debian 自建的 GitLab CE 实例

cdimage.debian.org/images 所有安装镜像

ftp.debian.org Debian 软件仓库最上游

bugs.debian.org Debian 缺陷追踪系统 (BTS)

tracker.debian.org 软件开发跟踪站点，跟踪软件包的历史版本

deb.debian.org Debian 软件仓库的全球 CDN

buildd.debian.org Debian 软件包构建网络的监控页面

manpages.debian.org 在线浏览所有 Debian 内出现的手册页

nm.debian.org 记录 Debian 所有成员的信息

security-tracker.debian.org 追踪安全漏洞的修复情况 (CVE 等)

sources.debian.org 在线浏览/搜索全部 Debian 软件源代码

lists.debian.org Debian 所有邮件列表的网页界面

正确认识 Git 打包仓库 (Debian Salsa GitLab)

Debian 的版本控制系统 (VCS) 在 2018 年前后迁移到了自建 GitLab 平台, 主要用于托管 Debian 打包仓库和其它 Debian 相关项目的代码。

- 在 Debian 中维护软件包时配套 git 打包仓库是推荐的, 但是是完全可选的。
- 没有 git 打包仓库仍然可以正常维护软件包¹³。
- 如果使用 git 打包仓库维护打包用脚本和打包历史, 请遵循 DEP-14 规范¹⁴或者 git-buildpackage 工具可以识别的规范设置 Git 分支名称和布局。
- 如果软件包由个人维护, 一般推荐将 Git 打包仓库设置在 <https://salsa.debian.org/debian/> 团队下; 团队维护的软件包则位于 <https://salsa.debian.org/<teamname-team>> 的命名空间下。
- 没有强制性要求打包用 Git 仓库必须设置在 Debian Salsa GitLab 平台上。

¹³如需了解 Git 打包仓库的采纳比例, 请浏览<https://trends.debian.net>。

¹⁴<https://dep-team.pages.debian.net/deps/dep14>

正确认识 Git 打包仓库 (Debian Salsa GitLab) (二)

如何维护设置了 Git 打包仓库的软件？

- 在 Git 仓库内更新软件包版本和软件源代码，更新软件打包脚本，准备 changelog；
- 使用诸如 `git-buildpackage` 之类的工具从 Git 仓库直接尝试对软件的新版本进行打包；
- 本地测试通过后将打包构建得到的源码包进行数字签名，上传至 Debian FTP Master 服务器。

目前 Debian 的 Git 打包仓库主要用于多人合作和历史记录，正确配置后还可以利用 GitLab Runner 实现自动 CI/CD 集成探测 regression¹⁵。因存在反对的声音，没有实现从 Git 仓库直接上传至 FTP Master 服务器的自动化¹⁶。

¹⁵<https://salsa.debian.org/salsa-ci-team/pipeline>

¹⁶另一个发行版 Chakra Linux 已经这么做了。

Debian 贡献者和 DD/DM 的关系

任何人都可以成为 Debian 贡献者 (“Debian Contributor”), 但成为 Debian 开发者 (DD) 和 Debian 维护者 (DM) 存在一定的门槛:

- 要申请 DM 需要在 Debian 中已有一定贡献, 由至少一个现存 DD 线下验证身份并签署 PGP Key, 同时由至少一个 DD 推荐 (两处的 DD 可以不同);
- 要申请 DD 需要在 Debian 中已有一定贡献, 由至少两个现存 DD 线下验证身份并签署 PGP Key, 同时由至少两个 DD 推荐, 在此之后需要通过知识和技能的测验¹⁷。

根据工作需要, 除非经常性地在 Debian 内进行打包, 否则没有一定要申请 DD/DM 的必要。这时每次更新软件包都需要由某位 DD 对所做的修改进行检查并协助上传。

所有已知贡献者的名单可以在 <https://nm.debian.org/public/people/> 找到。

¹⁷所有测试题目可以在<https://salsa.debian.org/nm-team/nm-templates>找到。

开源社区常见黑话

纯科普性质

中英文的开源社区都有一些常见的单词缩写或者代称；它们是网络用语或者开源文化的一部分，了解这些说法有助于看懂邮件列表、GitHub Issues 以及某些群聊中的一些讨论。这里仅举数例，并不具有代表性：

BTW 即 “By The Way”，意为 “顺便说一下”；

RTFM 即 “Read The Fucking Manual”，意为 “滚去读文档”；

OTOH 即 “On The Other Hand”，意为 “从另一方面来看”；

AFAIK 即 “As Far As I Know”，意为 “就我所知”；

GSoC 即 “Google Summer of Code”（Google 编程之夏）是由 Google 公司所主办的年度开源程序设计项目，项目常年涉及各大开源软件项目并向其贡献代码故有一定的知名度；

xkcd 指 xkcd.com 网站上的连环漫画，其中一部分与开源文化和网络文化有关。

开源社区常见黑话（中文部分）

纯科普性质

中文社区常见的黑话一般较为随意，这里姑且列出博君一笑：

“底裤” 有时候指代 `systemd`；

“菊苣” 类似“大佬”，指代软件核心开发者、主要贡献者或技术大牛；

“(” 常见于非正式场合，可理解为缓和语气用的标点符号；

uuu 指代 Ubuntu，因其名称中有三个字母 U；

滚动/滚挂/回滚 “滚动”指某些滚动 Linux 发行版日常进行的软件升级操作，“滚挂”指升级系统出错失败，“回滚”即指代英文的“rollback”；

开源社区协作技巧

- 除非是完全国人开发维护的项目，否则请全程使用英文。
 - 社区通用的协作语言是英语，使用中文意味着别人只能用谷歌翻译理解你的内容，某些上游可能初看中文看不懂就直接无视你的贡献。
 - 请尽量使用正确英文语法，避免使用缩写（除非缩写已约定俗成）。
 - 请只在全中文项目中使用中文：判断方法很简单，如果大多数其它讨论都是中文，那么可以使用中文。
- 提 Pull Request 或提供补丁时尽量原子化、简单化。
 - 一个补丁精准解决一个问题，避免无关修改（如不必要的空格/制表符/换行）
 - 提交说明中准确描述补丁的作用
 - 使用 Git 协作时善用 `git rebase`，即使开发时有多个提交，在提出 PR 时要基于上游最新代码变基，并将自己的提交合并至单一提交或少数几个意义清晰的提交
- 讨论时永远不要吵起来
 - 任何情况下都对你没有任何好处

开源社区协作技巧（二）

- “固定” 你的身份
 - 与他人交流时选择一个全名（或昵称），选择一个（半永久）电子邮箱地址，不要常常改动
 - 有助于其他人认识你，方便建立身份和声望（“Reputation”）
- 讨论时给出充足的引用信息
 - 如需引用某段文档，应同时给出对应网页超链接
 - 考虑类似科技文献的引用方式（[1][2]...）
 - 在支持 Markdown 等标记语言的场景下（如 GitHub Issue）更加方便（Markdown 等标记语言原生支持超链接等语法）
- 基于电子邮件的协作
 - 如果可能，请用纯文本电子邮件（非 HTML 电子邮件）
 - 回复讨论时适当引用原文
 - 遵循英文信件书写格式（或者看看其他人的写法，模仿来进行书写）
 - 如果常常用电子邮件打交道，建议选择一个靠谱的电子邮件客户端（如 Evolution、Thunderbird 等）
 - 保持耐心和礼貌（邮件不一定会得到及时回复），可以一周一封提醒邮件，做好持续数周/数月的准备

Linux 桌面中文本地化的参与入口

由于时间有限，这里只给出一些参考链接¹⁸：

GNOME 基于 PO 文件翻译，使用 [GNOME Damned Lies \(l10n.gnome.org\)](http://l10n.gnome.org) 自建网站平台，有社区中文团队。

KDE 基于 PO 文件翻译，中文翻译非正式地托管在了 [CrowdIn](https://community.kde.org/KDE_Localization/zh-cn) 平台上，详情请见 https://community.kde.org/KDE_Localization/zh-cn，有社区中文团队。

TP 很多传统命令行工具的翻译托管在 [TranslationProject\(TP\)](https://translationproject.org/team/zh_CN.html) 网站上，使用电子邮件提交 PO 文件翻译（成为提交人员需要申请）。请见 https://translationproject.org/team/zh_CN.html。

其它 其它各大软件项目或是直接在 [Git/SVN](#) 仓库中管理翻译文件（PO 文件或 .ts 文件），或是使用如 [Weblate](#) 或 [CrowdIn](#) 等平台托管翻译等等。

GNOME/KDE 中文社区都有微信群和邮件列表进行组织协调。

¹⁸ 感兴趣的人士欢迎谷歌搜索“大陆简中自由软件本地化工作指南”

- 官方网站: <https://www.debian.org>
- Debian GitLab (Salsa): <https://salsa.debian.org>
- Debian 政策手册: <https://www.debian.org/doc/debian-policy/>
- Debian 开发者参考:
<https://www.debian.org/doc/manuals/developers-reference/>
- Debian 维基上对下游发行版的指导内容:
<https://wiki.debian.org/Derivatives/Guidelines>
- Debian 维基上对上游软件作者的指导内容:
<https://wiki.debian.org/UpstreamGuide>

问答环节

任何与 Debian 相关的问题

本次演示文稿可以在 <https://github.com/hosiet/talk-debian-2020> 找到。

版权声明

Copyright Notice

本档以知识共享 署名-相同方式共享 4.0 协议（国际）发布。

This document is published under Creative Commons Attribution-ShareAlike 4.0 International License.