项目名称：提取 debian 软件包的控制信息生成 SPEC 文件

导师：王铮

申请人：杨凌云

邮箱：tinaaaaa@sjtu.edu.cn

日期：2023 年 5 月 27 日

目录

1 项目背景

1.1 项目简介

为 openEuler 引入 debian 的软件包, 需要把 deb 形式的软件包, 转成 rpm 形式的软件包。而生成 rpm 软件包, spec 文件是必须项。

1.2 项目产出要求

- 生成 SPEC 文件需要符合规范, 重点关注多子包的情况
- 软件通过 SPEC 文件能够在 openEuler 完成构建

1.3 项目链接

1.3.1 参考资料

- https://www.debian.org/doc/debian-policy/ch-controlfields.html
- https://www.debian.org/doc/debian-policy/ch-source.html#main-building-script-debian-rules
- https://rpm-packaging-guide.github.io/#what-is-a-spec-file

1.3.2 成果仓库

https://gitee.com/openeuler/oec-application

2 技术可行性分析

2.1 Linux 相关

我从半年前配置 CSAPP 环境时开始使用 Ubuntu 系统, 目前使用的主力系统是 Arch Linux, 因此我对 Linux 系统以及各大发行版有较多了解, 对 Linux 的命令和脚本较为熟悉。

2.2 软件包相关

使用 Ubuntu 系统时, 我对 apt 命令下载软件包很有兴趣, 了解过软件包的结构, 知道软件包的安装构建过程。

2.3 python 相关

我在大一开始时自学了 python, 并在自学国外课程时通过 python 写爬虫爬取资料 (比如 MIT 微积分网站上的数百个 pdf 文件)。其中使用到的正则表达式等知识同样可以运用到本项目中。

3 项目基本知识

这部分内容我在阅读了参考资料后自己进行了总结, 对应地整理出了三份文档存放在我的 项目仓库中。

下面我进一步地介绍一下这三份文档的内容。

3.1 deb 包的 control 文件

deb 包的 control 文件是一个文本文件, 包含了软件包的基本信息, 比如软件包的名称, 版本, 作者, 依赖等等。它既会在源包中出现, 也会在二进制包中出现。其基本格式为若干个键值对。其中有如下重要的字段:

- Package 软件包的名称
- Source 源包的名称
- Version 软件包的版本
- Architecture 软件包的架构
- Depends 软件包的依赖
- Description 软件包的描述
- Maintainer 软件包的维护者
- Homepage 软件包的主页
- Section 软件包的分类
- Priority 软件包的优先级

3.2 deb 包的 rule 文件

deb 包的 rule 文件是一个脚本文件, 包含了软件包的构建过程, 比如软件包的编译, 安装, 打包等等。其基本格式为若干个规则, 每个规则包含了若干个目标和依赖。其中典型的有:

- build 构建软件包
- build-arch 构建软件包 (特定架构)
- build-indep 构建软件包 (架构无关)
- clean 清理软件包
- install 安装软件包
- binary 构建二进制包
- binary-arch 构建二进制包 (特定架构)
- binary-indep 构建二进制包 (架构无关)

3.3 rpm 包的 spec 文件

rpm 包的 spec 文件是一个文本文件, 包含了 rpm 软件包的基本信息, 比如软件包的名称, 版本, 作者, 依赖等等。其基本格式为*前言段*和*主体段*, 前者包含包的元数据, 类似3.1中的 control 文件; 后者包含构建等过程的脚本, 类似3.2中的 rule 文件。

*前言段*中有如下重要的段落:

- Name 软件包的名称
- Version 软件包的版本
- Release 软件包的发布号
- Summary 软件包的简介
- License 软件包的许可证
- URL 软件包的主页
- Source0 软件包的源码

- Patch0 软件包的补丁
- BuildArch 软件包的架构
- BuildRequires 软件包的构建依赖
- Requires 软件包的运行依赖

主体段中有如下重要的段落:

- %description 软件包的描述
- %prep 构建前准备
- %build 构建阶段
- %install 安装阶段
- %check 测试阶段
- %files 文件列表
- %changelog 修改日志

4 项目实现

根据3中的知识, 可以发现 deb 包和 rpm 包的构建过程有很多相似之处。比如 rpm 包的 spec 文件中的前言段主要包含元数据, 可以与 deb 包中的 control 文件对应; spec 文件中的主体段主要包含构建脚本等内容, 可以与 deb 包中的 rule 文件对应。更具体的, 我们可以在将 spec 前言段和 control 文件在字段层面进行对应, 如下表所示:

rpm spec 前言段	deb control
Name	Package
Version	Version 或 Standards Version
Release	Version
Summary	Description
License	License
URL	Homepage
Source0	Source
BuildArch	Architecture
BuildRequires	Build-Depends
Requires	Depends

同样的, 我们可以在将 spec 主体段和 rule 文件在规则层面进行对应, 如下表所示:

rpm spec 主体段	deb rule
%description	Description
%prep	Source, Build
%build	build
%install	install
%check	test
%files	source

另外, rpm 包的 spec 文件中的*%changelog* 可以通过 deb 包中的*.change* 文件来提取。

5 时间安排

5.1 研发阶段 (7.1-8.15)

- 完善 deb 包和 rpm 包的对应关系
- 完成 deb 包和 rpm spec 文件的转换工具
- 在 openEuler 上对一些基础包进行测试

5.2 测试阶段 (8.16-9.30)

- 在 openEuler 上对更多包进行测试, 并完善细节处理
- 完成对包转换工具的测试, 并对其进行优化
- 完成文档的书写