Распространение программного обеспечения на языке Python с использованием системы управления пакетами Debian

Рыжиков Дмитрий Андреевич – магистрант Белорусского государственного университета информатики и радиоэлектроники.

Бирич Сергей Сергеевич – магистрант Белорусского государственного университета информатики и радиоэлектроники.

Аннотация: Целью данной работы является рассмотрение инструмента создания независимых распространяемых программных пакетов для программного обеспечения.

Ключевые слова: Python, управление пакетами, debian, pip.

Python – это современный высокоуровневый язык программирования общего назначения [1]. Основным ориентиром разработки данного языка является повышение производительности программиста и читаемости кода. Синтаксис Python минималистичен, но в то же время стандартная библиотека включает большой объём полезных функций. Кроме того, существует большое количество сторонних программных библиотек для облегчения выполнения задач в разных доменных областях, например, при разработке веб-приложений или решении задач машинного обучения. Эти факторы обеспечили широкое применение данного языка для выполнения разнообразных прикладных задач.

Однако одним из недостатков использования Python при разработке программного обеспечения является отсутствие стандартных качественных инструментов сборки легко распространяемых программных пакетов, которые можно было бы установить, обновить и удалить без особых трудностей, избегая процесса установки зависимостей пакетов, который часто является достаточно времязатратным, а также подвержен трудноразрешимым ошибкам.

Наиболее распространенным способом установки программных пакетов, написанных на языке Python, является получение исходного кода и файла со списком зависимостей и их версий с последующим развертыванием виртуальной среды выполнения. Развертывание виртуальной среды включает в себя установку всех необходимых зависимостей. Как правило установка зависимостей выполняется при помощи стандартного инструмента pip – установщика программных пакетов для Python. Удаление программных пакетов с помощью pip иногда работает с ошибками и пользователям приходится заниматься поиском и удалением файлов той или иной библиотеки вручную. Кроме того, pip не предоставляет возможности произвести откат неудачной установки или обновления библиотек. При возникновении исключений виртуальная среда программного пакета может остаться в неисправном состоянии, починить которое можно только полной переустановкой [2].

Надежность и скорость процесса установки и обновления программного обеспечения особенно важна в высоконагруженных системах, на корректную работу которых могут полагаться сотни и тысячи клиентов одновременно. Каждая секунда простоя таких систем из-за возникших в ходе процесса инсталляции ПО неполадок может стоить огромных убытков владельцам, поэтому очень важно иметь способ проведения безопасной установки и обновления программных пакетов.

Обеспечить надежность, простоту и быстроту установки программных пакетов может помочь использование Debian-пакетов – “бинарных” пакетов для распространения и установки программного обеспечения в операционной системе проекта Debian, и других, использующих систему управления пакетами dpkg.

Начиная с Debian версии 0.93, Debian-пакет (распространяемый обычно как файл с расширением deb) представляет собой архив формата ar. Обычно архив содержит 3 файла в нижеприведенной последовательности:

- debian-binary – текстовый файл, содержащий версию формата deb-пакета (современный формат – версия 2.0);
- tar – tar-архив, содержащий информацию и скрипты установки пакета, может быть сжат с помощью gzip или xz, тип архива отображается в имени файла (к примеру control.tar.gz);
- tar – tar-архив, содержащий дерево устанавливаемых файлов пакета, может быть сжат с помощью gzip, bzip2, lzma или xz, тип архива отображается в имени файла (к примеру data.tar.gz).

Архив control.tar содержит информацию о поставляемом в данном пакете программном обеспечении:

- control – содержит краткую информацию о пакете программного обеспечения: наименование, версия, описание, целевая архитектура, зависимости от других пакетов и так далее;
- md5sums – содержит MD5-суммы всех устанавливаемых файлов;
- conffiles – список файлов пакета, являющихся конфигурационными, при обновлении файлы из этого списка не перезаписываются новыми, если это не указано отдельно;

- preinst, postinst, prerm, postrm – необязательные сценарии оболочки, выполняемые соответственно до и после установки или удаления пакета;
- config – сценарий для debconf – механизма конфигурации;
- shlibs – список разделяемых библиотек пакета.

Архив data.tar содержит устанавливаемые файлы пакета и при установке разворачивается в систему относительно ее корня [3].

Очевидно, deb-пакеты имеют специфическую структуру. Процесс организации правильной структуры проекта и сборки может быть достаточно трудоемким. Однако существуют инструменты, которые позволяют значительно упростить данную задачу. Одним из таких инструментов является dh-virtualenv.

Dh-virtualenv – это инструмент для создания самодостаточных программных пакетов на языке Python, поставляемых с настроенной виртуальной средой выполнения. Поставляемая с пакетами виртуальная среда выполнения содержит все необходимые зависимости программного средства в предустановленном и готовом к работе виде.

Для использования dh-virtualenv программный пакет должен содержать файл с конфигурацией пакета setup.py, а также файл со списком зависимостей и их версиями requirements.txt. Помимо файлов, описывающих python-пакет необходимо создать файл debian/compat, содержащий число, указывающее на версию стандарта debian пакетов (на момент написания статьи актуальной версией является “9”) [4].

Также необходимо создать файл с метаданными о разрабатываемом debian-пакете –

debian/control. Данные должны включать в себя информацию о способе получения исходного кода, контактные данные автора пакета, список целевых архитектур, краткое описание программного пакета и список зависимостей. Ниже приведен пример такого файла:

Source: my-awesome-python-software

Section: python

Priority: extra

Maintainer: Matt Maintainer <matt@example.com>

Build-Depends: debhelper (>= 9), python, dh-virtualenv (>= 0.8)

Standards-Version: 3.9.5

Package: my-awesome-python-software

Architecture: any

Pre-Depends: dpkg (>= 1.16.1), python2.7 | python2.6, ${misc:Pre-Depends}

Depends: ${misc:Depends}

Description: really neat **package**!

second line can contain extra information about it.

В файле debian/rules необходимо указать, что создание пакета должно производиться с использование dh-virtualenv:

#!/usr/bin/make -f

%:

dh $@ --with python-virtualenv

После создания всех необходимых файлов создать готовый к распространению пакет можно одной командой:

$ dpkg-buildpackage -us -uc

Установка полученного пакета на целевой системе производится следующей командой:

$ dpkg -i my-package.deb

В результате использования системы управления пакетами dpkg, поставляемой в комплекте с операционной системой Debian, и инструмента создания deb-пакетов из python-пакетов dh-virtualenv удалось получить способ создавать независимые пакеты, занимающие относительно небольшой объем памяти, которые отличаются простотой в

распространении и эксплуатации.

Список литературы

1. Python / Wikipedia - Режим доступа: https://ru.wikipedia.org/wiki/Python - Дата доступа: 11.06.2020.
2. How We Deploy Python Code / nylas.com - Режим доступа: https://www.nylas.com/blog/packaging-deploying-python/ - Дата доступа: 11.06.2020.
3. dev (формат файлов) / Wikipedia - Режим доступа: https://ru.wikipedia.org/wiki/Deb_(формат_файлов) - Дата доступа: 11.06.2020.
4. Getting Started / dh-virtualenv.readthedocs.io - Режим доступа: https://dh-virtualenv.readthedocs.io/en/1.1/tutorial.html - Дата доступа: 11.06.2020.

{social}