

УДК 004.451

Т.И. КОМАРОВ¹, И.Ю. ЖУКОВ^{1,2},
Н.А. ЧЕПИК^{1,3}, Ю.А. ПОЛОВНЕВА¹

¹Национальный исследовательский ядерный университет «МИФИ», Москва

²ООО «Группа компаний «Инфотактика», Москва

³АНО «Институт инженерной физики», Серпухов

СОСТОЯНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ПАКЕТНЫХ МЕНЕДЖЕРОВ

Анализируются основные сложности и тенденции в области управления пакетами и конфигурациями ОС: проблемы традиционных пакетных менеджеров, распространение «неизменяемых» дистрибутивов и контейнеров, развитие функциональных пакетных менеджеров – в частности, Nix и Guix. Предлагаются подходы к развитию отечественных решений в данной области с учётом мирового опыта.

Пакетный менеджер – это программный комплекс, который обеспечивает автоматизацию процесса установки, обновления, конфигурации и удаления ПО на компьютере. Наиболее распространённые дистрибутивы ОС на базе ядра Linux используют классические пакетные менеджеры apt на основе dpkg (apt) или RPM (yum, dnf, zypper) [1].

Основные недостатки классических пакетных менеджеров:

- возможность некорректного/неполного указания зависимостей;
- «ад зависимостей»;
- возможность взаимного влияния пакетов друг на друга;
- отсутствие атомарных обновлений;
- отсутствие полной воспроизводимости сборки.

Кроме того, широкое распространение получила практика использования пакетных менеджеров, специфичных для конкретных технологий или языков программирования (ЯП), например: pip для ЯП Python. Одновременное использование нескольких пакетных менеджеров – потенциальный источник конфликтов и причина проблем с безопасностью.

Данные проблемы не остаются незамеченными. Большое распространение получают «неизменяемые» (immutable) дистрибутивы ОС на базе ядра Linux (например: Fedora Silverblue, Ubuntu Core и др.),

которые воплощают идею, схожую с той, что применяется в мобильных ОС (например, Android) уже достаточно давно: приложения отделены от основной ОС, которая является неизменяемой и атомарно обновляемой.

Однако переход к «неизменяемым» дистрибутивам видится полумерой, т.к. распространение приложений в контейнерах препятствует обеспечению должного уровня безопасности разработчиками ОС.

Но существует альтернативный подход, который был описан в PhD-диссертации E. Dolstra в 2006 г. [2]. Предложенная автором модель функционального пакетного менеджера Nix заключается в том, что пакеты обрабатываются аналогично значениям в функциональных ЯП – они собираются функциями, не содержащими побочных эффектов.

Согласно принципам Nix, пакеты должны находиться в хранилище, размещённом в каталоге /nix/store. При этом, каждый пакет должен храниться в отдельном каталоге с именем вида: /nix/store/<HASH>-<NAME>-<VER>/, где hash – это хеш-сумма, полученная в результате обработки всех входных данных пакета, name – имя пакета, ver – версия пакета.

Основные преимущества пакетного менеджера Nix:

- отсутствие «ада зависимостей»;
- отсутствие влияния пакетов друг на друга;
- атомарность и возможность лёгкого отката обновлений;
- повторяемость сборки отдельных пакетов и всей системы.

В качестве недостатков Nix и NixOS можно отметить сложность использования «обычных» исполняемых файлов и не самое эффективное использование дискового пространства.

Следует обратить внимание, что пакетный менеджер Nix использует собственный функциональный ЯП Nix. Это послужило основной причиной создания построенного на схожих принципах пакетного менеджера Guix [3], который предполагает использование ЯП Guile.

Несмотря на указанные недостатки, функциональные пакетные менеджеры являются востребованной технологией для отечественных разработчиков безопасного программного обеспечения и для органов сертификации средств защиты информации.

Список литературы

1. Usage statistics of Linux for websites [Электронный ресурс] // URL: <https://w3techs.com/technologies/details/os-linux> (дата обращения: 15.09.2024).
2. Dolstra E. The Purely Functional Software Deployment Model [Электронный ресурс] // URL: <https://edolstra.github.io/pubs/phd-thesis.pdf> (дата обращения: 15.09.2024).
3. Courtès L. Functional Package Management with Guix [Электронный ресурс] // URL: <https://arxiv.org/pdf/1305.4584> (дата обращения: 15.09.2024).