Доклад

Система управления пакетами dpkg

Верниковская Екатерина Андреевна

Содержание

# 1 Вводная часть

**Актуальность темы и проблема:** тема “система управления пакетами dpkg” остается актуальной, поскольку это основной инструмент управления программным обеспечением в Debian и его производных дистрибутивах Linux. Понимание работы dpkg важно для администраторов систем и разработчиков, чтобы обеспечивать стабильность, безопасность и эффективность в установке, удалении и управлении пакетами программного обеспечения

**Объект и предмет исследования:** система управления пакетами dpkg

**Цель:** цель данного доклада - рассмотреть основные принципы работы системы управления пакетами dpkg, изучить её функциональность, основные команды и возможности

**задачи исследования:** узнать что такое менеджеры пакетов и какие они бывают, что такое пакеты и подробнее узнать про систему управления пакетами dpkg

**Материалы и методы и инструменты исследования:** интернет-ресурсы, аналитика и практические навыки работы на своей операционной системе Linux (Ubuntu)

# 2 Введение

В далекие времена установка программного обеспечения под операционные системы семейства Linux могла серьезно напугать начинающих пользователей этих ОС. Загрузка исходных кодов, управление зависимостями, часто представляющее собой сложную задачу, ручное редактирование конфигурационных файлов и другие аспекты установки приложений в прошлом, сегодня кажутся устаревшими и неактуальными. [1]

Сейчас любой уважающий себя дистрибутив Linux имеет в своем составе возможность установки программного обеспечения с помощью менеджеров пакетов.

## 2.1 Что такое менеджер пакетов

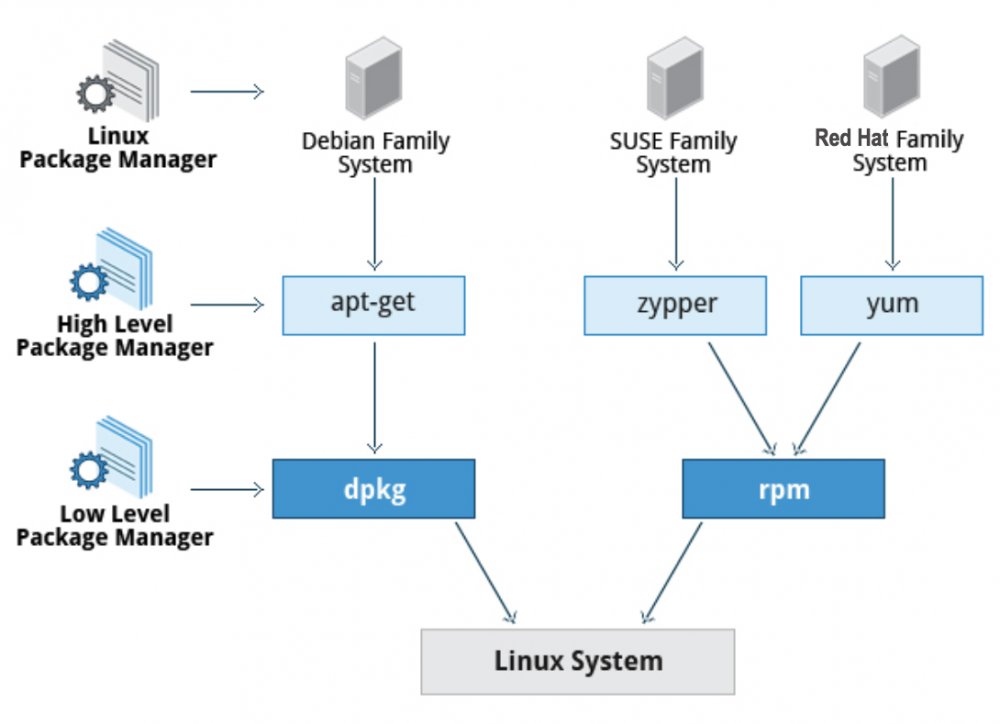
Системы управления пакетами (которые также иногда называются «менеджер пакетов» или «пакетный менеджер») — это набор программного обеспечения, позволяющего управлять процессом установки, удаления, настройки и обновления различных компонентов ПО. [2]

Менеджер пакетов следит за тем, какие программы установлены на компьютере, и позволяет легко устанавливать новые программы, обновлять программы до более новых версий или удалять те программы, которые ранее были установлены.

Использование менеджера пакетов в Linux может сэкономить много времени и усилий по сравнению с установкой программного обеспечения и его зависимостей вручную. Когда мы устанавливаем пакет, менеджер пакетов автоматически проверяет, требуется ли для его корректной работы какое-либо другое программное обеспечение, и устанавливает для нас эти зависимости. Это избавляет нас от необходимости выяснять, какое еще программное обеспечение необходимо, и устанавливать его вручную. Менеджер пакетов также может автоматически проверять наличие обновлений для установленных пакетов и устанавливать их для нас. Это помогает нам поддерживать актуальность и безопасность нашей системы.

Категории пакетных менеджеров:

* **Высокоуровневые менеджеры.** Применяются для поиска и скачивания пакетов из репозиториев. В процессе работы могут задействовать низкоуровневые менеджеры для инсталляции загруженных программ.
* **Низкоуровневые менеджеры.** Используются для установки локальных пакетов, загруженных вручную пользователем, или высокоуровневым пакетным менеджером.



Категории пакетных менеджеров

Популярные пакетные менеджеры: [3]

* **DPKG (Debian Package)** – система управления пакетами в Debian и дистрибутивах на его основе, например Ubuntu.
* **APT (Advanced Packaging Tool)** – консольная утилита, выполняющая роль «поисковика» и загрузчика пакетов из репозиториев. Установка скачанных пакетов производится утилитой DPKG.
* **RPM (Red Hat Package Manager)** – формат пакетов и низкоуровневый пакетный менеджер систем RED HAT (RHEL, CentOS, Fedora и др.)
* **YUM (Yellowdog Updater, Modified)** – высокоуровневый пакетный менеджер, написанный на языке Python для систем RED HAT (RHEL, CentOS, Fedora). Программа представляет собой своеобразную оболочку для утилиты RPM.
* **Pacman** – высокоуровневый пакетный менеджер системы Arch Linux и его родственных дистрибутивов (Manjaro, EndeavourOS и др.). Программа написана на языке C# и совмещает высокую функциональность, легкость и производительность. В качестве пакетов используются архивы pkg.tar.xz.
* **Portage** – система управления пакетами Gentoo или Calculate Linux. Установка программ для данного дистрибутива несколько отличается от остальных систем Linux. В Gentoo пакетный менеджер использует исключительно исходный код, а не готовые пакеты для установки программ.

Как предполагает само название, менеджеры пакетов имеют дело с пакетами — наборами файлов, которые упакованы вместе и могут быть установлены или удалены группой.

## 2.2 Что такое пакет и чем он отличается от обычной программы

В мире Linux термин “пакет” обычно относится к файлу, содержащему программное обеспечение, его зависимости и метаданные, необходимые для установки и управления программным обеспечением в операционной системе. Пакеты представляют собой удобный способ организации и распространения программного обеспечения в Linux. [2]

Обычные программы-установщики обычно устанавливают программное обеспечение из исполняемого файла, который загружается из Интернета или другого источника. Они могут требовать взаимодействия с пользователем для выбора параметров установки и расположения файлов.

Пакеты, с другой стороны, используют пакетный менеджер (например, dpkg или yum) для управления процессом установки. Пакетный менеджер проверяет зависимости, устанавливает необходимые библиотеки и файлы конфигурации и гарантирует, что программное обеспечение будет правильно интегрировано в систему.

Отличие между пакетом в Linux и обычной программой-установщиком заключается в том, что пакет представляет собой архив с программным обеспечением, зависимостями и метаданными, который устанавливается с помощью специальных инструментов управления пакетами. В то время как программы-установщики обычно представляют собой отдельные исполняемые файлы или скрипты, которые запускаются для установки программы и не всегда управляют зависимостями или метаданными так же эффективно, как это делают пакеты.

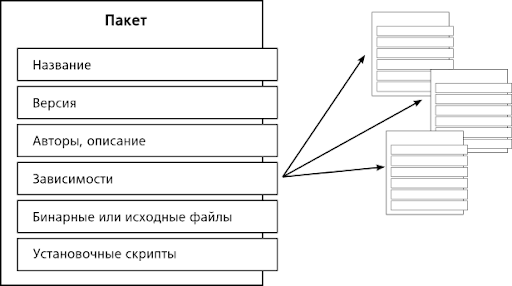
Установщик может сам выполнить все необходимые команды и установить программу, а пакет — нет. Для установки и управления пакетами как раз и нужен менеджер пакетов.

## 2.3 Из чего состоит пакет

Пакет, как правило, содержит само приложение, в откомпилированном  виде, то есть, по сути в виде бинарного файла. Также в пакете указывается метаинформация, то есть информация о другой информации, используемая инструментами управления пакетами.

Метаинформация представляет собой составленное по определённым правилам описание, которое содержит имя пакета, номер версии и сборки, сведения о разработчике и его мастер-сайте, список файлов, их положение в файловой иерархии, список зависимостей. Также, здесь могу присутствовать установочные и настроечные сценарии, необходимые для развертывания приложения.

Помимо установочных сценариев, пакет также содержит набор действий, выполняемых после установки (например, пост-инсталляционную конфигурацию). Также пакет содержит сценарий, выполняемый в случае удаления пакета.

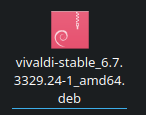


Основные состовляющие пакета

Различные дистрибутивы ОС Linux имеют свои форматы пакетов. Вот основные форматы: [3]

* .deb – Debian и производные (Ubuntu, Mint и т.д.)
* .rpm – Red Hat и производные (CentOS, Fedora и т.д.), OpenSUSE
* .apk – Android
* .ebuild – Gentoo

Структура имени пакетов такова: имя-дополнение-версия\_архитектура.deb Например: vivaldi-stable\_6.7.3329.24-1\_amd64.deb



Пример пакета

# 3 Пакетный менеджер Dpkg

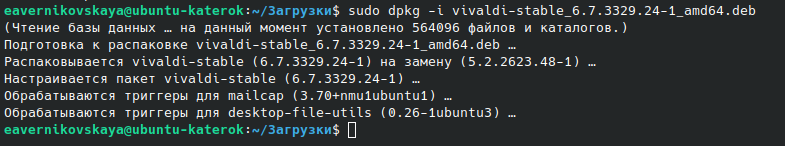
Dpkg - это пакетный менеджер для Debian систем. DPKG используется с пакетами, созданными для Linux на базе Debian, которые заканчиваются расширением .deb. Он может устанавливать, удалять и создавать пакеты, но, в отличие от других систем управления пакетами, он не может автоматически загружать и устанавливать пакеты или их зависимости.

С dpkg можно устанавливать только локальные файлы, которые мы уже загрузили самостоятельно. Он не может искать удаленные репозитории или извлекать из них пакеты.

Debian — это операционная система с открытым исходным кодом, которая состоит из свободного программного обеспечения. На данный момент это самая популярная ОС среди Linux-продуктов. Она стала основой для многих других дистрибутивов, например, Ubuntu и Kali Linux.

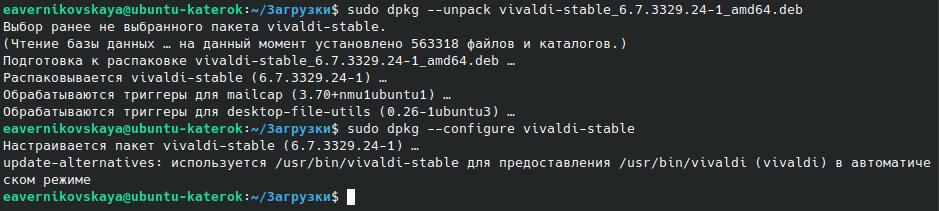
## 3.1 Установка пакета

Dpkg — это, прежде всего, инструмент для установки уже доступных пакетов Debian (поскольку он ничего не загружает). Чтобы установить пакет, используется опция **-i** или **–install**. Мы можем видеть каждый этап, выполняемый dpkg, поэтому мы знаем, в каком месте могла произойти какая-либо ошибка. [4]



Установка пакета с использованием опции -i

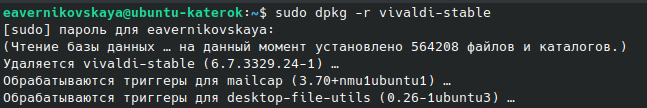
Установку также можно выполнить в два этапа: сперва распаковка (**–unpack**), затем конфигурация (**–configure**). Благодаря этому apt-get делает меньше обращений к dpkg (каждый такой запрос является дорогостоящей операцией из-за необходимости загрузки в память базы данных, включая весь список уже установленных файлов).



Установка пакета в два этапа с использованием опций –unpack и -configure

## 3.2 Удаление пакета

Запуск dpkg с опцией **-r** или **–remove**, за которой следует имя пакета, приведет к удалению этого пакета.



Удаление пакета с использованием опции -r

Это удаление, однако, не полное: все конфигурационные файлы, сценарии сопровождающего, файлы журналов (системные журналы) и другие пользовательские данные, используемые этим пакетом, останутся. Таким путём легко избавиться от программы, деинсталлировав её, но при этом сохраняется возможность установить её ещё раз с той же конфигурацией. Для полного удаления всего, связанного с пакетом, используется опция **-P** или **–purge**, сопровождающаяся именем пакета.

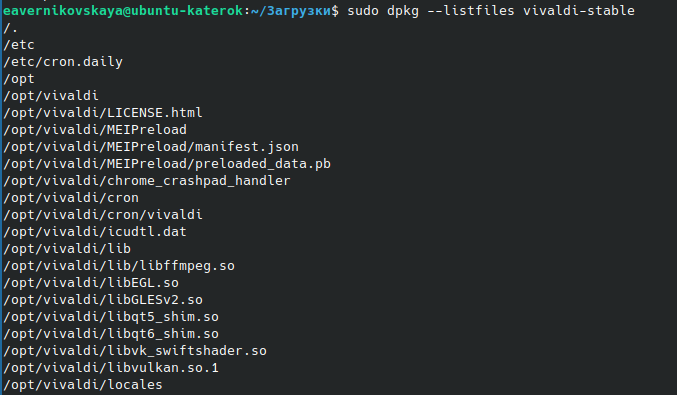
Полное удаление пакета с использованием опции -P

Полное удаление пакета с использованием опции -P

## 3.3 Запросы к базе данных dpkg и анализ файлов .deb

Dpkg также обладает возможностью проведения запросов к своей внутренней базе данных для получения информации. При помощи различных опций можно получить подробную информацию о пакетах, их файловой структуре, а также произвести поиск пакетов, содержащих определенные файлы.

Так, **–listfiles** пакет (или **-L**) выводит список файлов, установленых пакетом;



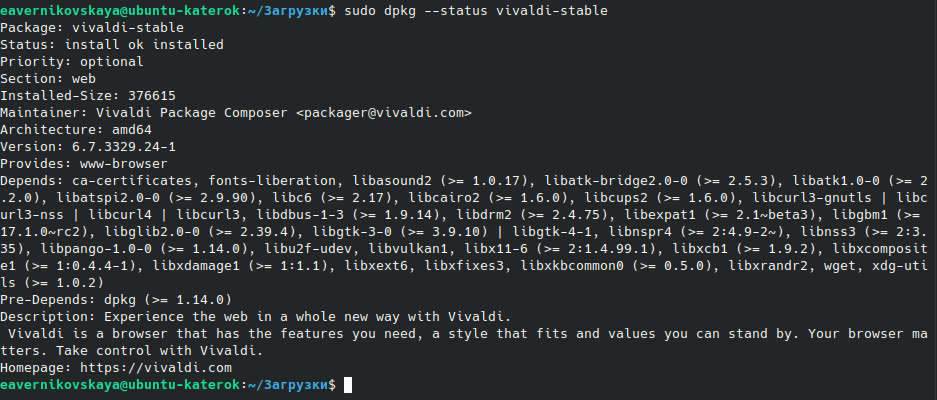
Вывод списка файлов, установленных пакетом с использованием опции –listfiles

**–search** файл (или **-S**) ищет пакет, к которому относится этот файл;

Поиск пакета, к которому относится файл, с использованием опции -S

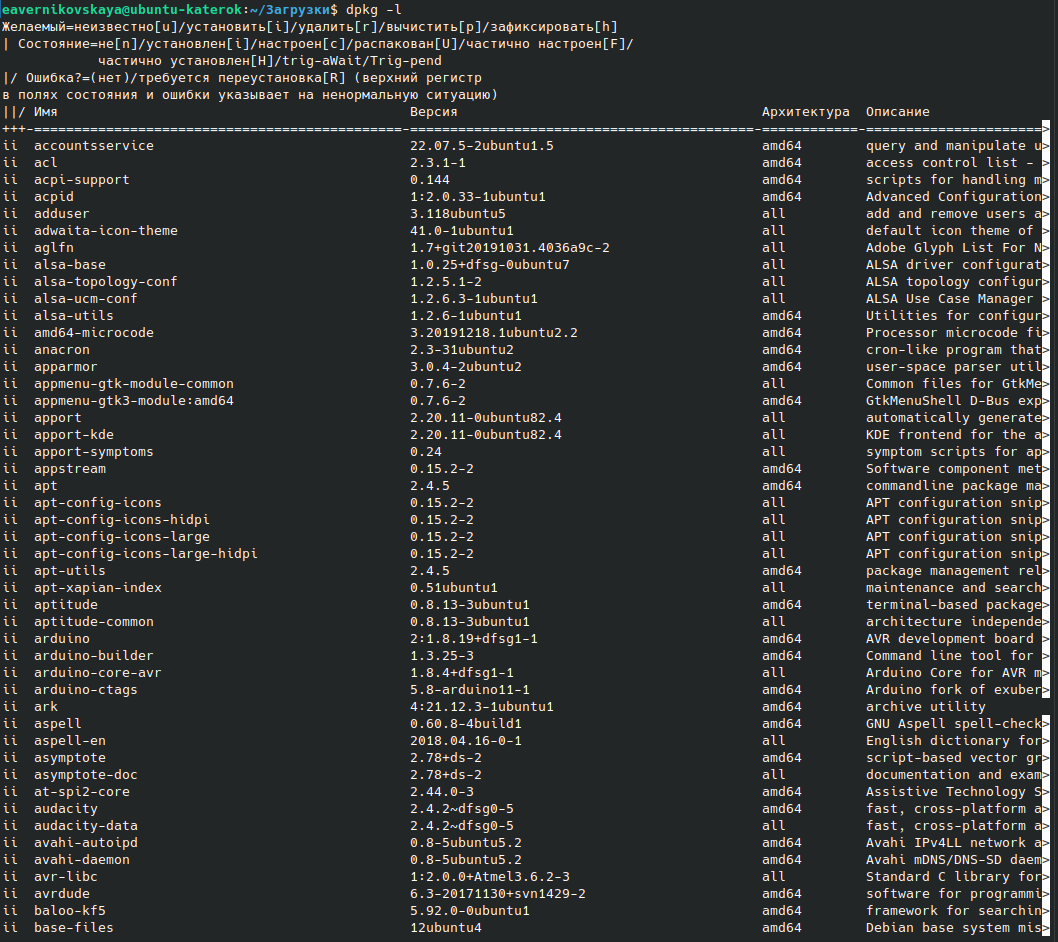
Поиск пакета, к которому относится файл, с использованием опции -S

**–status** пакет (or **-s**) выводит информацию о том или ином установленном пакете;



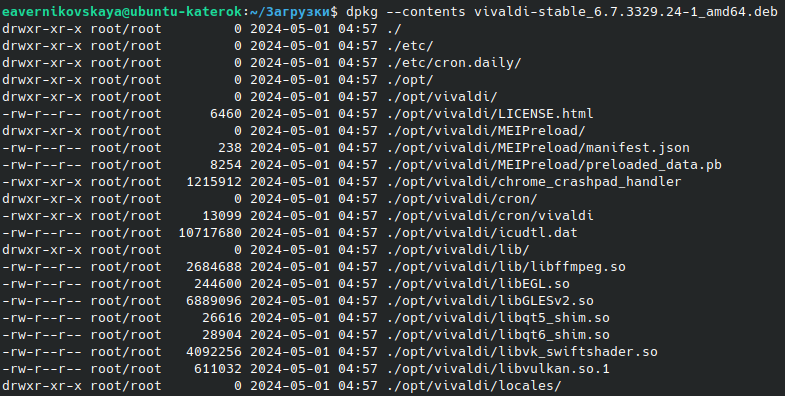
Вывод информации об установленном пакете, с использованием опции –status

**–list** (или **-l**) показывает список пакетов, известных системе, и их статус;



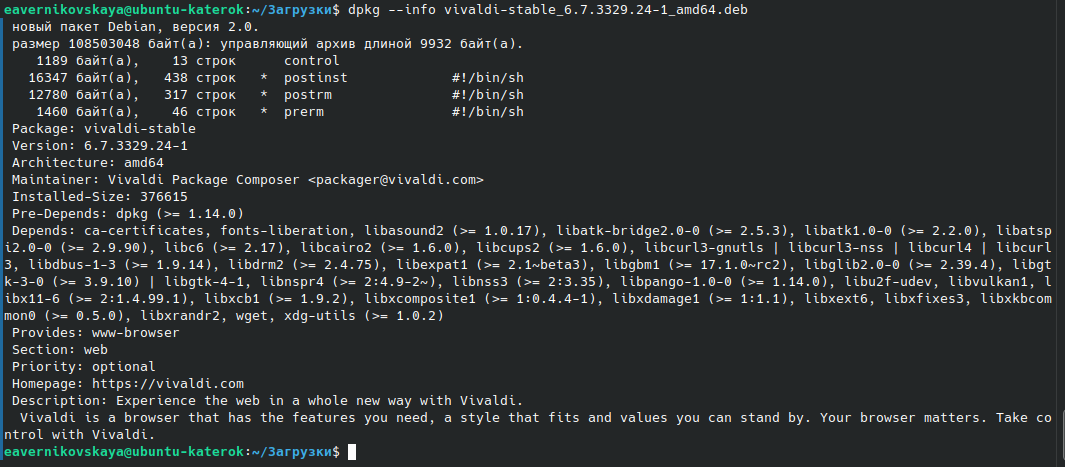
Список всех пакетов известных системе, с использованием опции -l

**–contents** file.deb (или **-c**) показывает список файлов в этом пакете;



Список файлов в пакете, с использованием опции –contents

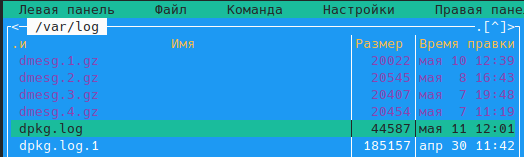
**–info** file.deb  (или **-I**) показывает информацию о пакете Debian.



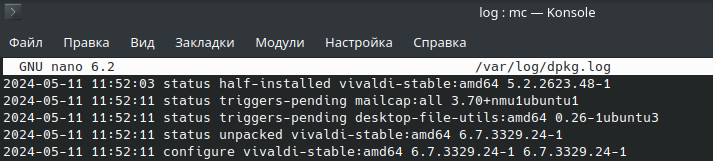
Выод информации о пакете, с использованием опции –info

## 3.4 Файл журнала dpkg

Dpkg сохраняет журнал всех своих действий в **/var/log/dpkg.log**. Этот журнал чрезвычайно подробный: в нём задокументированы все этапы обработки пакетов dpkg. Этот журнал помогает не только отследить поведение dpkg, но и сохранить историю изменений в системе: можно найти точный момент, когда каждый пакет был установлен или обновлён, и эта информация может быть чрезвычайно полезной при выяснении причин изменения поведения системы в целом.



Каталог /var/log/dpkg.log



Файл журнала dpkg (dpkg.log)

## 3.5 Поддержка мультиархитектуры

Все пакеты Debian имеют поле Architecture в своих метаданных. Это поле может содержать либо значение «all» (для пакетов, которые не зависят от архитектуры), либо название конкретной архитектуры, для которой пакет предназначен (например «amd64», «armhf», …). В последнем случае dpkg по умолчанию допустит установку пакета только в том случае, если его архитектура соответствует архитектуре системы, возвращаемой dpkg **–print-architecture**.

Архитектура системы

Архитектура системы

## 3.6 Включение мультиархитектуры

Поддержка мультиархитектуры dpkg позволяет определять «чужеродные архитектуры», которые могут быть установлены в данной системе. Это легко сделать с помощью dpkg **–add-architecture**. Существует и соответствующая команда dpkg **–remove-architecture** для отключения поддержки чужеродной архитектуры, но её можно использовать только в том случае, когда в системе не осталось ни одного пакета этой архитектуры.

# 4 Выводы

Таким образом, dpkg является базовой программой для управления пакетами в системе Debian. Он позволяет устанавливать, удалять и управлять пакетами, обеспечивая фундаментальные операции с пакетами Debian.

Отличительной особенностью dpkg является его простота и прозрачность работы, что делает его привлекательным выбором для опытных пользователей, желающих более тесно контролировать процесс управления пакетами.

Однако, важно отметить, что dpkg является низкоуровневым пакетным менеджером и отличается от более полноценных систем управления пакетами, таких как, например, APT, тем, что не предполагает автоматического решения зависимостей и загрузки пакетов из сети.

В целом, dpkg является незаменимым инструментом для управления пакетами в экосистеме Debian/Ubuntu, обеспечивающим надежную и удобную платформу для установки, обновления и удаления программного обеспечения.

# Список литературы

1. Andrey\_Biryukov. [Управление пакетами в ОС Linuх](https://habr.com/ru/companies/otus/articles/743614/). Habr, 2023.

2. Yadav V. [Package Management in Linux](https://www.scaler.com/topics/cyber-security/package-management-in-linux/). Scaler Topics, 2024.

3. [Популярные пакетные менеджеры Linux](https://eternalhost.net/blog/sistemnoe-administrirovanie/paketnye-menedzhery-linux). Eternalhost, 2020.

4. [Работа с пакетами при помощи dpkg](https://l.github.io/debian-handbook/html/ru-RU/sect.manipulating-packages-with-dpkg.html).