**ООО «БАЗАЛЬТ СПО»**

# АЛЬТ ПЛАТФОРМА

## Описание функциональных характеристик

**Ред. 1.0**

**МОСКВА 2023**

*Содержание*

# 1 ОТЕЧЕСТВЕННАЯ ПРОГРАММНАЯ ИНФРАСТРУКТУРА ПОДДЕРЖКИ, РАЗРАБОТКИ И ОБНОВЛЕНИЯ ПО

«Альт Платформа» («ALT Platform») – технологический комплекс для сборки прикладного программного обеспечения и выпуска продуктов ООО «Базальт СПО».

«Альт Платформа» – инфраструктура в основе, которой, лежат технологии отечественного репозитория свободного программного обеспечения Sisyphus (Сизиф). Репозиторий не только является хранилищем пакетов программ, но и сопровождается набором оригинальных технологий, которые поддерживают его целостность, обеспечивают возможность взаимодействия разработчиков (в том числе и сторонних), и позволяют создавать на этой базе решения различного назначения, в том числе и дистрибутивы Linux.

«Альт Платформа» – основана на стабильной ветке репозитория Sisyphus, предназначена для разработки, тестирования, распространения, обновления и поддержки комплексных решений всех уровней – от встроенных устройств до серверов предприятий и датацентров, создается и поддерживается ООО «Базальт СПО».

«Альт Платформа» позволяет обеспечить интеграцию и поддержку программных продуктов различных разработчиков в окружение операционных систем «Альт».

Состав «Альт Платформа»:

- дистрибутив «alt-platform-builder», содержащий следующие компоненты:
  - hasher – средство воспроизводимой сборки пакетов в изолированном окружении;
  - gear – инструмент для хранения исходных текстов в git и извлечения заданной версии;
  - mkimage – набор утилит для создания образов (в основном ISO);
  - mkimage-profiles – метапрофиль со множеством готовых «кирпичиков» и конфигураций образов;
  - alternator-mirror – модуль центра управления системой для зеркалирования репозиториев;
- репозиторий «Альт Платформа».

Инфраструктура поддержки, разработки и обновления ПО включает в себя следующие компоненты:

**1. Банк исходных кодов программ**

В банке хранятся все исходные коды вместе с историей изменений. Обеспечивается ведение нескольких веток одного проекта как одним разработчиком, так и разными, в целях поддержки нескольких версий дистрибутива, а также нескольких дистрибутивов. Банк исходных кодов интегрирован с системой сборки пакетов.

2. **Система контроля зависимостей между пакетами**

При формировании пакетов программ, разработчик указывает разного рода зависимости с другими пакетами (по выполнению, по сборке, и др.). Автоматическая система контроля анализирует как исполняемые бинарные файлы, так и программы на скриптовых языках в целях обнаружения неразрешенных зависимостей или паразитных зависимостей. Такая проверка обеспечивает высокий уровень интеграции всех пакетов в единый репозиторий, на основании, которого можно создавать целостные решения, а также уменьшает объём ручного тестирования, необходимого в технологическом процессе.

3. **Система сборки пакетов в гарантированных сборочных средах**

При сборке пакета из исходных кодов автоматически формируется виртуальная файловая система, гарантирующая воспроизводимость сборки независимо от конфигурации сборочной системы. Сборка пакетов в виртуальной среде позволяет фиксировать среду сборки, а также обеспечивает её безопасность. Система сборки интегрирована с банком исходных кодов программ и позволяет отслеживать связь между собранными пакетами и исходным кодом, что облегчает тестирование и отладку решений.

4. **Система распараллеливания сборки**

Обеспечивает распределенную сборку пакетов на доступных компьютерных мощностях, что позволяет достигать разумного времени сборки пакетов и, при необходимости, пересборки всего репозитория.

5. **Единая модульная система управления**

Управление различными системными объектами обеспечивается средствами системы Alterator, которая предоставляет API для создания и подключения различных модулей управления, специфичных для данного дистрибутива, а также набор готовых модулей для управления типичными Unix-сервисами. Для «Альт Платформа» существует специальный модуль alterator-mirror с помощью которого можно получить локальную копию репозитория.

6. **Единая система управления пакетами программ**

Управление программным обеспечением осуществляется единой системой, представляющей ПО в виде так называемых пакетов и контролирующей зависимости между пакетами, включая версии пакетов. Наличие этой системы позволяет формировать дистрибутивы с контролированной замкнутостью по зависимостям между программными компонентами, а в дальнейшем – организовывать выборочную установку пакетов или их обновление, без нарушения целостности системы и без потери пользовательских настроек.

**7. Единая система установки**

Установка дистрибутива осуществляется единой настраиваемой системой, интегрированной с системой управления системными объектами и с системой управления пакетами.

**8. Система генерации дистрибутивов**

Данная система позволяет создавать инсталляционные образы дистрибутивов (наборы iso-образов CD или DVD, образа сетевой загрузки) по формализованному описанию (набору целевых пакетов) с учётом зависимостей между пакетами.

# 2 БАНК ИСХОДНЫХ КОДОВ ПРОГРАММ

Все исходные тексты находятся в специально созданном для этого банке исходных кодов. Все изменения фиксируются в системе контроля версий. Именно из этого банка в дальнейшем и создаются бинарные пакеты. Таким образом, исходные программный код из банка исходных кодов проходит стадию сборки в гарантированной сборочной среде, в результате чего попадает в банк бинарных пакетов.

Исходный текст является свободно доступным, что позволяет производить его своевременный аудит и адаптацию. Структура хранения исходных кодов может быть произвольной. Для сборки пакетов из произвольно устроенного git-репозитория используется утилита gear.

# 3 СИСТЕМА КОНТРОЛЯ ЗАВИСИМОСТЕЙ МЕЖДУ ПАКЕТАМИ

В современных системах на базе Linux огромное число общих ресурсов, которыми пользуются сразу несколько программ: разделяемых библиотек, содержащих стандартные функции, исполняемых файлов, сценариев и стандартных утилит и т. д. Удаление или изменение версии одного из составляющих систему компонентов может повлечь неработоспособность других, связанных с ним компонентов, или даже вывести из строя всю систему. В контексте системного администрирования проблемы такого рода называют нарушением целостности системы. Задача администратора – обеспечить наличие в системе согласованных версий всех необходимых программных компонентов (обеспечение целостности системы).

Для установки, удаления и обновления программ и поддержания целостности системы в Linux в первую очередь стали использоваться менеджеры пакетов. С точки зрения менеджера пакетов программное обеспечение представляет собой набор компонентов – программных пакетов. Такие компоненты содержат в себе набор исполняемых программ и вспомогательных

файлов, необходимых для корректной работы программного обеспечения. Менеджеры пакетов облегчают установку программ: они позволяют проверить наличие необходимых для работы устанавливаемой программы компонент подходящей версии непосредственно в момент установки, а также производят необходимые процедуры для регистрации программы во всех операционных средах пользователя: сразу после установки программа может быть доступна пользователю из командной строки и – если это предусмотрено – появляется в меню всех графических оболочек.

Часто компоненты, используемые различными программами, выделяют в отдельные пакеты и помечают, что для работы ПО, предоставляемого пакетом A, необходимо установить пакет B. В таком случае говорят, что пакет A зависит от пакета B или что между пакетами A и B существует зависимость.

Отслеживание зависимостей между такими пакетами представляет собой серьёзную задачу для любого дистрибутива – некоторые компоненты могут быть взаимозаменяемыми: может обнаружиться несколько пакетов, предлагающих затребованный ресурс.

Задача контроля целостности и непротиворечивости установленного в системе ПО ещё сложнее. Представим, что некие программы A и B требуют наличия в системе компоненты C версии 1.0. Обновление версии пакета A, требующее обновления компоненты C до новой, использующей новый интерфейс доступа, версии (например, до версии 2.0), влечёт за собой обязательное обновление и программы B.

Однако менеджеры пакетов оказались неспособны предотвратить все возможные коллизии при установке или удалении программ, а тем более эффективно устранить нарушения целостности системы. Особенно сильно этот недостаток сказывается при обновлении систем из централизованного репозитория пакетов, в котором последние могут непрерывно обновляться, дробиться на более мелкие и т. п. Этот недостаток и стимулировал создание систем управления программными пакетами и поддержания целостности системы.

Для автоматизации этого процесса в «Альт Платформа» применяется **Усовершенствованная система управления программными пакетами APT**[1] (от англ. Advanced Packaging Tool). Такая автоматизация достигается созданием одного или нескольких внешних репозиториев, в которых хранятся пакеты программ и относительно которых производится сверка пакетов, установленных в системе. Репозитории могут содержать как официальную версию дистрибутива, обновляемую его разработчиками по мере выхода новых версий программ, так и локальные наработки, например, пакеты, разработанные внутри компании.

Таким образом, в распоряжении APT находятся две базы данных: одна описывает установленные в системе пакеты, вторая – внешний репозиторий. APT отслеживает целостность

[1] В «Альт Платформа» используется собственный форк apt-rpm, отличающийся от аналогичного инструмента проекта debian и работающий с пакетами формата rpm.

установленной системы и, в случае обнаружения противоречий в зависимостях пакетов, руководствуется сведениями о внешнем репозитории для разрешения конфликтов и поиска корректного пути их устранения.

# 4 СИСТЕМА СБОРКИ ПАКЕТОВ В ГАРАНТИРОВАННЫХ СБОРОЧНЫХ СРЕДАХ

## 4.1 Сборочная среда hasher

hasher – инструмент для сборки пакетов в чистой и контролируемой среде. Это достигается с помощью создания в chroot минимальной сборочной среды, установки туда указанных в source-пакете сборочных зависимостей и сборке пакета в созданной «на лету» среде. Для сборки каждого пакета сборочная среда создаётся заново.

Такой принцип сборки имеет несколько следствий:

- все необходимые для сборки зависимости должны быть указаны в пакете. Для облегчения поддержания сборочных зависимостей в актуальном состоянии в «Альт Платформа» имеется инструмент под названием `buildreq`;
- сборка не зависит от конфигурации компьютера пользователя, собирающего пакет, и может быть повторена на другом компьютере;
- изолированность среды сборки позволяет с лёгкостью собирать на одном компьютере пакеты для разных дистрибутивов и веток репозитория – для этого достаточно лишь направить hasher на различные репозитории для каждого сборочного окружения.

Дополнительно к сборке пакетов hasher:

- проверяет их с помощью утилиты `sisyphus_check`;
- создает локальный APT-репозиторий с результатами сборки, позволяя последовательно собирать пакеты, опираясь на уже собранные.

Система сборки пакетов в гарантированных сборочных средах предъявляет следующие требования:

- не снижать уровень безопасности хост-системы;
- обеспечивать собственную безопасность от атак со стороны пакетов;
- обеспечивать безопасность сборки пакетов от атак со стороны других пакетов;
- гарантировать надёжность (воспроизводимость) результатов сборки;
- обеспечивать приемлемый уровень производительности.

## 4.2 Архитектура hasher

В основе архитектуры hasher лежит трёхпользовательская модель: вызывающий непривилегированный пользователь (C) и два непривилегированных вспомогательных псевдопользователя; первый (R) играет роль root в порождаемой сборочной среде, второй (U) – обычного пользователя, собирающего программы.

Переключение между вызывающим и вспомогательными пользователями осуществляется с помощью специальной привилегированной программы (вызываемой посредством `sudo`), написанной с применением мер защиты от непривилегированных пользователей. Кроме того, с помощью этой программы удаляются процессы, запущенные вспомогательными псевдопользователями и не завершившиеся в срок, а также создаются устройства. Наконец, эта программа предоставляет возможность контролировать ресурсы, выделяемые процессам непривилегированных пользователей, для защиты от DOS-атак.

Путь пакета через сборочную систему в общих чертах выглядит следующим образом:

Пользователь **C** порождает среду (aptbox) для работы с apt.

Полностью удаляется сборочная среда, возможно оставшаяся от предыдущей сборки. Удаление происходит последовательно в chroot пользователем **U**, в chroot пользователем **R** и, наконец, пользователем **C**.

Пользователь **C** создаёт каркас новой сборочной среды, состоящий из вспомогательных каталогов и вспомогательных статически слинкованных программ (ash, find и cpio). С помощью вспомогательной привилегированной программы создаётся фиксированный набор устройств, достаточный для нормального функционирования сборочной среды и при этом не несущий угрозы host-системе.

Порождается базовая установочная среда, представляющая собой набор средств, необходимых для штатной установки пакетов в эту среду. Пользователь **C** с помощью apt определяет набор пакетов, необходимых для порождения базовой установочной среды. Пользователь **R** с помощью вспомогательных статически слинкованных программ распаковывает эти пакеты.

Порождается базовая сборочная среда, представляющая собой набор средств, необходимых для сборки любого пакета. Пользователь **C** с помощью apt определяет набор пакетов, пользователь **R** устанавливает их.

Порождается сборочная среда для данного пакета. Пользователь **U** извлекает сборочные зависимости пакета, пользователь **C** с помощью apt определяет набор пакетов для установки, и пользователь **R** устанавливает их.

Пользователь **U** осуществляет сборку пакета.

Такая схема призвана исключить атаки вида **U->R**, **U->C**, **R->C**, а также все виды атак на **root**.

Для повышения производительности, особенно важной при сборке большого числа пакетов, применяется кэширование базовой сборочной среды. С помощью средств apt реализована возможность использования собранных ранее пакетов для сборки последующих пакетов.

# 5 СИСТЕМА РАСПАРАЛЛЕЛИВАНИЯ СБОРКИ

Благодаря свойству воспроизводимости результатов сборки hasher можно использовать для параллельной сборки большого числа пакетов на нескольких серверах. Таким образом удаётся достичь разумного времени сборки при средних вычислительных ресурсах. Открывается возможность организовать регулярное тестирование на пересобираемость большого репозитория пакетов.

# 6 ЕДИНАЯ МОДУЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ

На платформе Alterator построены инсталлятор системы и штатный ее конфигуратор. В качестве языка описаний интерфейсов используется встроенный интерпретатор Scheme. Alterator позволяет, например, построить на своей основе удобный интерфейс для выполнения наиболее востребованных административных задач: добавление и удаление пользователей, настройка сетевых подключений, просмотр информации о состоянии системы, и т.п.

Важной особенностью является возможность сетевого доступа к такого рода интерфейсу, что позволяет осуществлять администрирование, в том числе и удаленно.

# 7 ЕДИНАЯ СИСТЕМА УСТАНОВКИ

Единая система установки состоит из модулей, написанных на платформе Alterator. Установка может проходить сразу после загрузки с установочного диска или образа.

# 8 СИСТЕМА ГЕНЕРАЦИИ ДИСТРИБУТИВОВ

Система генерации дистрибутивов использует все преимущества банка пакетов и позволяет получить установочные образы. Для сборки используется утилита mkimage, которая использует для сборки «профиль», представляющий из себя набор файлов Makefile. В результате из пакетов репозитория создается установочный диск CD/DVD. Целостность репозитория и его непротиворечивость позволяют с легкостью генерировать новые образы при необходимости.

## 8.1 Образы бездисковых станций

Помимо создания установочных образов предусмотрена система для создания образов ОС для бездисковых станций. Для этого используется утилита `mknfsroot`.

Утилита принимает единственный параметр – местоположение профиля. Профиль – это каталог, содержащий следующие файлы:

- packages – список пакетов для установки;
- modules – список модулей ядра для сетевых адаптеров;
- pxelinux.cfg – конфигурационный файл для pxelinux;
- autoinstall.scm – сценарий для инсталлятора.

Последний файл содержит инструкции для настройки системы, например:

- настройка системной локали;
- настройка раскладки клавиатуры;
- настройка часового пояса;
- задание пароля администратору.

Запуск утилиты:

```
# mknfsroot /etc/mknfsroot/profiles/sample
```

В результате появляется файл `/var/lib/mknfsroot/mknfsroot.tar`, содержащий:

- настроенную систему;
- ядро, initrd, образ загрузчика pxelinux и конфигурационный файл для него.

### 8.1.1 Установка и удаление образа

Развёртывание образа осуществляется при помощи утилиты `setupnfsroot`. Данная утилита принимает два параметра: путь к tar-архиву и целевой каталог.

```
# setupnfsroot /var/lib/mknfsroot/nfsroot.tar /var/lib/tftpboot
```

Утилита развёртывает образ и настраивает необходимые точки монтирования. Если к этому моменту в системе уже настроены сервера TFTP и DHCP, то можно уже попробовать загрузить бездисковую станцию.

Обратная операция осуществляется утилитой removenfsroot.

```
# removenfsroot /var/lib/tftpboot
```

В результате каталог очищается, и точки монтирования удаляются.

### 8.1.2 Особенности созданного образа

- имя машины (hostname) выставляется по результатам резолвинга ее IP-адреса;
- поскольку один и тот же образ используется для загрузки большого количества бездисковых узлов, то для каждого узла создаётся персональный каталог `/var`. Это

перестраховка, поэтому для конкретного случая созданный автоматом образ желательно подправить;

- при остановке машины сеть не останавливается, так как корневая файловая система – сетевая.

8.1.3 Нестандартное расширение профиля

Работает `mknfsroot` следующим образом:

1. Утилита переключается на псевдопользователя, настроенного так, чтобы работал `hasher`.

2. Две части профиля - общая для всех образов (`/etc/mknfsroot/template`) и специфичная (`/etc/mknfsroot/profiles/*`) – объединяются в один профиль `mkimage`.

3. Запускается `mkimage`.

Меняя содержимое `/etc/mknfsroot/template` можно неограниченно изменять поведение mknfsroot вплоть до того что утилита начнёт делать iso-образы вместо tar-файлов.