ОПЕРАЦИОННАЯ СИСТЕМА АЛЬТ СЕРВЕР 8.0

Описание жизненного цикла

СОДЕРЖАНИЕ

1 Отечественная программная инфраструктура поддержки, разработки и обновления ПО 2

2 Банк исходных кодов программ ... 3

3 Система контроля зависимостей между пакетами ... 4

4 Система сборки пакетов в гарантированных сборочных средах ... 5

4.1 Сборочная среда hasher ... 5

4.2 Архитектура hasher ... 6

5 Единая система управления пакетами программ ... 7

5.1 Репозитории ... 7

5.2 Поиск пакетов ... 9

5.3 Установка или обновление пакета ... 10

5.4 Удаление установленного пакета ... 11

5.5 Обновление всех установленных пакетов ... 12

6 Система распараллеливания сборки ... 13

7 Единая модульная система управления ... 13

8 Единая система установки ... 13

9 Система генерации дистрибутивов ... 13

9.1 Образы бездисковых станций ... 13

9.1.1 Установка и удаление образа ... 14

9.1.2 Особенности созданного образа ... 14

9.1.3 Нестандартное расширение профиля ... 15

1 ОТЕЧЕСТВЕННАЯ ПРОГРАММНАЯ ИНФРАСТРУКТУРА ПОДДЕРЖКИ, РАЗРАБОТКИ И ОБНОВЛЕНИЯ ПО

В качестве технологического комплекса для выпуска Альт Сервер используется инфраструктура в основе, которой лежат технологии отечественного репозитория свободного программного обеспечения Sisyphus. Репозиторий не только является хранилищем пакетов программ, но и сопровождается набором оригинальных технологий, которые поддерживают его целостность, обеспечивают возможность взаимодействия для разработчиков (в том числе и сторонних), и позволяют создавать на этой базе дистрибутивы Linux и другие решения различного назначения.

Кроме того, данная инфраструктура позволяет обеспечить интеграцию в окружение операционной системы и поддержку программных продуктов различных разработчиков.

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 репозитарий, 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 репозитария.

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.

6.**Единая система управления пакетами программ**

Управление программным обеспечением осуществляется единой системой, представляющей ПО в виде т.н. 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, и проч.) по формализованному описанию (набору целевых пакетов) с учётом зависимостей между пакетами.

2 БАНК ИСХОДНЫХ КОДОВ ПРОГРАММ

Все исходные текста входящих в дистрибутив программ 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 репозитария 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 (скажем, 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 и применяется **Усовершенствованная система управления программными пакетами APT** (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 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 свежесозданной 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 Sisyphus придуман 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;
- создаёт 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 ЕДИНАЯ СИСТЕМА УПРАВЛЕНИЯ ПАКЕТАМИ ПРОГРАММ

Используемая усовершенствованная система управления программными пакетами APT, используемая является системой управления пакетами с простым пользовательским интерфейсом, позволяющую производить установку, обновление и повседневные «хозяйственные» работы с установленными на машине программами без необходимости изучения тонкостей используемого в дистрибутиве менеджера программных пакетов.

Система APT состоит из нескольких утилит. Чаще всего используется утилита управления пакетами apt-get: она автоматически определяет зависимости между пакетами и строго следит за их соблюдением при выполнении любой из следующих операций: установка, удаление или обновление пакетов.

5.1 Репозитории

Репозитории, с которыми работает APT, отличаются от обычного набора пакетов наличием мета информации – индексов пакетов, содержащихся в репозитории, и сведений о них. Поэтому, чтобы получить всю информацию о репозитории, APT достаточно получить его индексы.

APT может работать с любым количеством репозиториев одновременно, формируя единую информационную базу обо всех содержащихся в них пакетах. При установке пакетов APT обращает внимание только на название пакета, его версию и зависимости, а расположение в том или ином репозитории не имеет значения. Если потребуется, APT в рамках одной операции установки группы пакетов может пользоваться несколькими репозиториями.

Важно: Подключая одновременно несколько репозиториев, нужно следить за тем, чтобы они были совместимы друг с другом по пакетной базе, т. е. отражали один определённый этап разработки. Например, совместимыми являются основной репозиторий дистрибутива и репозиторий обновлений по безопасности к данному дистрибутиву. В то же время смешение среди источников APT репозиториев, относящихся к разным дистрибутивам, или смешение стабильного репозитория с нестабильной веткой разработки (Sisyphus) чревато различными неожиданными трудностями при обновлении паке-тов.

APT позволяет взаимодействовать с репозиторием с помощью различных протоколов доступа. Наиболее популярные – HTTP и FTP, однако существуют и некоторые дополнительные методы.

Для того, чтобы APT мог использовать тот или иной репозиторий, информацию о нем необходимо поместить в файл /etc/apt/sources.list. Описания репозиториев заносятся в этот файл в следующем виде:

rpm [подпись] метод: путь база название

rpm-src [подпись] метод: путь база название

rpm или rpm-src

Тип репозитория (скомпилированные программы или исходные тексты).

[подпись]

Необязательная строка-указатель на электронную подпись разработчиков. Наличие этого поля подразумевает, что каждый пакет из данного репозитория должен быть подписан соответствующей электронной подписью. Подписи описываются в файле /etc/apt/vendor.list.

Метод

Способ доступа к репозиторию: ftp, http, file, cdrom, copy.

Путь

Путь к репозиторию в терминах выбранного метода.

База

Относительный путь к базе данных репозитория.

Название

Название репозитория.

Для добавления в sources.list репозитория на компакт-диске в APT даже предусмотрена специальная утилита – apt-cdrom. Чтобы добавить запись о репозитории на компакт-диске, достаточно вставить диск в привод и выполнить команду apt-cdrom add. Если при установке ОС был выбран профиль «Рабочая станция», то записи для cdrom в файле /etc/fstab не будет. Потребуется примонтировать cdrom вручную (командой mount /dev/cdrom /media/cdrom) и использовать коман-

ду добавления диска с дополнительным ключом: apt-cdrom add -m. После этого в sources.list появится запись о подключённом диске примерно такого вида:

```
rpm cdrom:[Server Disk 1]/ AltServer main
rpm-src cdrom:[Server Disk 1]/ AltServer main
```

После того как отредактирован список репозиториев в sources.list, необходимо обновить локальную базу данных APT о доступных пакетах. Это делается командой apt-get update.

Если в sources.list присутствует репозиторий, содержимое которого может изменяться (как происходит с любым постоянно разрабатываемым репозиторием, в частности, обновлений по безопасности - updates), то прежде чем работать с APT, необходимо синхронизировать локальную базу данных с удалённым сервером командой apt-get update. Локальная база данных создаётся заново каждый раз, когда в репозитории происходит изменение: добавление, удаление или переименование пакета. Для репозиториев, находящихся на компакт-дисках и подключённых командой apt-cdrom add, синхронизация производится единожды в момент подключения.

При выборе пакетов для установки, APT руководствуется всеми доступными репозиториями вне зависимости от способа доступа к ним. Так, если в репозитории, доступном по сети Интернет, обнаружена более новая версия программы, чем на компакт-диске, то APT начнёт загружать данный пакет из Интернет. Поэтому если подключение к Интернет отсутствует или ограничено низкой пропускной способностью канала или высокой стоимостью, то следует закомментировать те строчки в /etc/apt/sources.list, в которых говорится о ресурсах, доступных по Интернет.

5.2 Поиск пакетов

Если вы не знаете точного названия пакета, для его поиска можно воспользоваться утилитой apt-cache, которая позволяет искать не только по имени пакета, но и по его описанию.

Команда apt-cache search подстрока позволяет найти все пакеты, в именах или описании которых присутствует указанная подстрока. Например:

```
$ apt-cache search ^gimp
gimp - The GNU Image Manipulation Program
libgimp - GIMP libraries
libgimp-devel - GIMP plugin and extension development kit
gimp-help-en - English help files for the GIMP
gimp-help-ru - Russian help files for the GIMP
gimp-plugin-separateplus - Improved version of the CMYK Separation plug-in [...]
gimp-script-ISONoiseReduction - Gimp script for reducing sensor noise [...]
gimp-plugin-gutenprint - GIMP plug-in for gutenprint
gimp-plugin-ufraw - GIMP plugin for opening and converting RAW files [...]
```

Обратите внимание, что в данном примере в поисковом выражении используется символ ^, указывающий на то, что необходимо найти совпадения только в начале строки (в данном случае – в начале имени пакета).

Для того, чтобы подробнее узнать о каждом из найденных пакетов и прочитать его описание, можно воспользоваться командой apt-cache show, которая покажет информацию о пакете из репозитория:

```
$ apt-cache show gimp-help-ru
Package: gimp-help-ru
Section: Graphics
Installed Size: 37095561
Maintainer: Alexey Tourbin <at@altlinux.ru>
Version: 2.6.1-alt2
Pre-Depends: rpmlib(PayloadIsLzma)
Provides: gimp-help-ru (= 2.6.1-alt2)
Obsoletes: gimp-help-common (< 2.6.1-alt2)
Architecture: noarch
Size: 28561160
MD5Sum: 0802d8f5ec1f78af6a4a19005af4e37d
Filename: gimp-help-ru-2.6.1-alt2.noarch.rpm
Description: Russian help files for the GIMP
Russian help files for the GIMP.
```

apt-cache позволяет осуществлять поиск и по русскому слову, однако в этом случае будут найдены только те пакеты, у которых есть описание на русском языке. К сожалению, русское описание на настоящий момент есть не у всех пакетов, хотя описания наиболее актуальных для пользователя пакетов переведены.

5.3 Установка или обновление пакета

Установка пакета с помощью APT выполняется командой:

```
# apt-get install имя_пакета
```

apt-get позволяет устанавливать в систему пакеты, требующие для работы другие, пока ещё не установленные. В этом случае он определяет, какие пакеты необходимо установить, и устанавливает их, пользуясь всеми доступными репозиториями.

Установка пакета gimp командой apt-get install gimp приведёт к следующему диалогу с APT:

```
# apt-get install gimp
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие дополнительные пакеты будут установлены:
icc-profiles libbabl libgegl libgimp libjavascriptcoregtk2 libopenraw libspiro lib-
webkitgtk2 libwmf
Следующие НОВЫЕ пакеты будут установлены:
gimp icc-profiles libbabl libgegl libgimp libjavascriptcoregtk2 libopenraw libspiro
libweb-kitgtk2 libwmf
```

```
0 будет обновлено, 10 новых установлено, 0 пакетов будет удалено и 0 не будет обновлено.
Необходимо получить 0B/24,6MB архивов.
После распаковки потребуется дополнительно 105MB дискового пространства.
Продолжить? [Y/n] y
. . .
Получено 24,6MB за 0s (44,1MB/s).
Совершаем изменения...
Preparing... ######################################## [100%]
1: libbabl ######################################## [ 10%]
2: libwmf ######################################## [ 20%]
3: libjavascriptcoregtk2 ######################################## [ 30%]
4: libwebkitgtk2 ######################################## [ 40%]
5: icc-profiles ######################################## [ 50%]
6: libspiro ######################################## [ 60%]
7: libopenraw ######################################## [ 70%]
8: libgegl ######################################## [ 80%]
9: libgimp ######################################## [ 90%]
10: gimp ######################################## [100%]
Running /usr/lib/rpm/posttrans-filetriggers
Завершено.
```

Команда apt-get install имя_пакета используется и для обновления уже установленного пакета или группы пакетов. В этом случае apt-get дополнительно проверяет, не обновилась ли версия пакета в репозитории по сравнению с установленным в системе.

При помощи APT можно установить и отдельный бинарный rpm-пакет, не входящий ни в один из репозиториев (например, полученный из Интернет). Для этого достаточно выполнить команду apt-get install путь_к_файлу.rpm. При этом APT проведёт стандартную процедуру проверки зависимостей и конфликтов с уже установленными пакетами.

Иногда, в результате операций с пакетами без использования APT, целостность системы нарушается, и apt-get отказывается выполнять операции установки, удаления или обновления. В этом случае необходимо повторить операцию, задав опцию -f, заставляющую apt-get исправить нарушенные зависимости, удалить или заменить конфликтующие пакеты. В этом случае необходимо внимательно следить за сообщениями, выдаваемыми apt-get. Любые действия в этом режиме обязательно требуют подтверждения со стороны пользователя.

5.4 Удаление установленного пакета

Для удаления пакета используется команда apt-get remove имя_пакета. Для того чтобы не нарушать целостность системы, будут удалены и все пакеты, зависящие от удаляемого: если отсутствует необходимый для работы приложения компонент (например, библиотека), то само приложение становится бесполезным. В случае удаления пакета, который относится к базовым ком-

понентам системы, apt-get потребует дополнительного подтверждения производимой операции с целью предотвратить возможную случайную ошибку.

Если вы попробуете при помощи apt-get удалить базовый компонент системы, вы увидите такой запрос на подтверждение операции:

```
# apt-get remove filesystem
Обработка файловых зависимостей... Завершено
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие пакеты будут УДАЛЕНЫ:
basesystem filesystem ppp sudo
Внимание: следующие базовые пакеты будут удалены:
В обычных условиях этого не должно было произойти, надеемся, вы точно
представляете, чего требуете!
basesystem filesystem (по причине basesystem)
0 пакетов будет обновлено, 0 будет добавлено новых, 4 будет
удалено(заменено) и 0 не будет обновлено.
Необходимо получить 0B архивов. После распаковки 588kБ будет
освобождено.
Вы делаете нечто потенциально опасное!
Введите фразу 'Yes, do as I say!' чтобы продолжить.
```

Каждую ситуацию, в которой APT выдаёт такое сообщение, необходимо рассматривать отдельно. Однако, вероятность того, что после выполнения этой команды система окажется неработоспособной, очень велика.

5.5 Обновление всех установленных пакетов

Для обновления всех установленных пакетов используется команда apt-get upgrade. Она позволяет обновить те, и только те установленные пакеты, для которых в репозиториях, перечисленных в /etc/apt/sources.list, имеются новые версии; при этом из системы не будут удалены никакие другие пакеты. Этот способ полезен при работе со стабильными пакетами приложений, относительно которых известно, что они при смене версии изменяются несущественно.

Иногда, однако, происходит изменение в именовании пакетов или изменение их зависимостей. Такие ситуации не обрабатываются командой apt-get upgrade, в результате чего происходит нарушение целостности системы: появляются неудовлетворённые зависимости. Например, переименование пакета MySQL-shared, содержащего динамически загружаемые библиотеки для работы с системой управления базами данных MySQL, в libmysqlclient (отражающая общую тенденцию к наименованию библиотек в дистрибутиве) не приводит к тому, что установка обновлённой версии libmysqlclient требует удаления старой версии MySQL-shared. Для разрешения этой проблемы существует режим обновления в масштабе дистрибутива – apt-get dist-upgrade.

В случае обновления всего дистрибутива APT проведёт сравнение системы с репозиторием и удалит устаревшие пакеты, установит новые версии присутствующих в системе пакетов, а также отследит ситуации с переименованиями пакетов или изменения зависимостей между старыми и новыми версиями программ. Всё, что потребуется поставить (или удалить) дополнительно к уже имеющемуся в системе, будет указано в отчёте apt-get, которым APT предварит само обновление.

Для обновления всей системы рекомендуется использовать команду apt-get dist-upgrade.

6 СИСТЕМА РАСПАРАЛЛЕЛИВАНИЯ СБОРКИ

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

7 ЕДИНАЯ МОДУЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ

На платформе Alterator построены инсталлятор системы и штатный её 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.

8 ЕДИНАЯ СИСТЕМА УСТАНОВКИ

Единая система установки состоит из модулей, написанных на платформе Alterator. Установка может проходить как из режима Live CD, так и сразу после загрузки с установочного диска или образа.

9 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.

9.1 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.

9.1.1 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 tftp и dhcp сервера, то можно уже попробовать загрузить бездисковую станцию.

Обратная операция осуществляется утилитой removenfsroot:

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

В результате каталог очищается, и точки монтирования удаляются.

9.1.2 Особенности созданного образа

- имя машины (hostname) выставляется по результатам резолвинга её 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.

9.1.3 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.

Стало быть меняя содержимое /etc/mknfsroot/template вы можете неограниченно изменять поведение mkfsroot вплоть до того что утилита начнёт делать iso образы вместо tar-файлов.