

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

УТВЕРЖДАЮ

Зав.кафедрой,

доцент, к. ф.-м. н.

\_\_\_\_\_ М. В. Огнева

**ОТЧЕТ О ПРАКТИКЕ**

студента 3 курса 341 группы факультета КНиИТ  
Шарова Кирилла Владимировича

вид практики: производственная

кафедра: информатики и программирования

курс: 3

семестр: 6

продолжительность: 4 нед., с 22.06.2024 г. по 19.07.2024 г.

Руководитель практики от университета,

доцент, к. ф.-м. н.

\_\_\_\_\_

? ? ?

Руководитель практики от организации (учреждения, предприятия),

Заместитель генерального директора \_\_\_\_\_

Е. А. Синельников

Тема практики: «Разработка древовидной иерархии модулей управления компонентами ALT Linux»

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Программы и их хранение в ALT Linux .....	5
1.1 RPM-пакеты и средства пакетизации ALT Linux .....	5
1.2 Проект "Сизиф" и процедура Join .....	8
1.3 Системы межпроцессного взаимодействия .....	8
2 Разработка системы конфигурации операционной системы ALT Linux...	9
2.1 Инструментарий для разработки и пакетизации .....	9
2.2 Реализация иерархии компонентов alterator-application-components.	9
ЗАКЛЮЧЕНИЕ .....	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	11
Приложение А Какое-то приложение .....	12

## ВВЕДЕНИЕ

ALT Linux — это семейство российских дистрибутивов Linux, разрабатываемых компанией «Базальт СПО». Они предназначены для использования в государственных учреждениях, образовательных организациях и других структурах, где требуется надёжное и безопасное программное обеспечение.

Одной из особенностей ALT Linux является его ориентация на безопасность и защиту информации. В дистрибутивы включены инструменты для шифрования данных, защиты от вирусов и несанкционированного доступа. Также ALT Linux поддерживает работу с отечественными криптографическими алгоритмами и средствами аутентификации.

Компания «Базальт СПО» предоставляет техническую поддержку и обновления для ALT Linux, что обеспечивает его надёжность и стабильность. Разработчики также проводят обучение и консультации по использованию системы, что способствует её распространению и внедрению.

В целом, ALT Linux представляет собой надёжную и безопасную операционную систему, которая может быть использована в различных областях деятельности. Она позволяет снизить зависимость от иностранных поставщиков программного обеспечения и обеспечить безопасность информационных систем [1], что соответствует текущей государственной политике импортозамещения.

Целью работы является разработка древовидной иерархии модулей управления компонентами операционной системы ALT Linux.

Поставленная цель определила следующие задачи:

- Научиться собирать RPM-пакеты инструментами ALT Linux.
- Начать прохождение процедуры Join.
- Изучить систему межпроцессного взаимодействия D-Bus.
- Научиться разрабатывать и пакетировать приложения на C++ и Qt5.
- Реализовать древовидную иерархию компонентов в alterator-application-components.

## 1 Программы и их хранение в ALT Linux

### 1.1 RPM-пакеты и средства пакетизации ALT Linux

#### *RPM-пакеты*

Неотъемлемой частью дистрибутивов Linux являются хранилища программного обеспечения (далее ПО), которые зачастую являются собственными и индивидуальными для конкретного дистрибутива. Такое хранилище ПО называется репозиторием пакетов Linux. В общем случае пакеты содержат директории с бинарными файлами, метаданными и информацией о зависимостях для их установки. Репозитории пакетов Linux предназначены для стандартизации процесса установки ПО из удалённого хранилища, что предоставляет удобство как разработчикам, так и пользователям.

Самыми популярными форматами таких пакетов являются DEB (свойственны Debian-подобным дистрибутивам) и RPM (Red Hat Package Manager). Репозитории пакетов семейства дистрибутивов ALT Linux основаны на пакетах RPM.

RPM-пакетизация состоит из следующих этапов: у

- Нахождение исходного текста программы (опционально).
- Написание инструкции сборки пакета.
- Непосредственная сборка пакета.

Исходный текст программы часто можно получить на официальном сайте или странице программы. Исходный текст может быть в виде архива tar, репозитория git, cvs, меркурий и т.п. Также альтернативным источником исходного текста может быть пакет формата src.rpm или deb-src (у Debian-подобных дистрибутивов).

В качестве сценария для сборки выступает файл формата spec (далее spec-файл). Структура spec-файла следующая:

- Шапка с информацией о пакете.
- Описание пакета.
- Секция предварительной обработки исходных данных.
- Секция сборки исходного текста.
- Секция установки результата сборки.
- Секция файлов.
- Секция метаданных о журнале изменений версий пакета.

В общем случае шапка spec-файла содержит следующую информацию:

- Название пакета (Name).
- Версия ПО, включенного в пакет (Version).
- Версия пакета (Release).
- Резюме ПО (Summary).
- Лицензия распространения ПО (License).
- Категория, к которой относится ПО (Group).
- Электронный ресурс ПО (URL).
- Имена архивов с исходными текстами (Source).
- Имена файлов исправлений (патчей), применяемых к исходным текстам (Patch).
- Архитектуры процессоров, на которых собирается пакет (BuildArch).
- Требуемые пакеты для сборки (BuildRequires).
- Требуемые пакеты для запуска (Requires).

Секция предварительной обработки исходных данных (%prep) включает в себя распаковку архива с исходниками в директорию сборки с установкой соответствующих пользовательских прав доступа. Также при необходимости накладываются патчи, перечисленные в шапке под соответствующим ключом (например, макрос %patch0 разворачивается в установку первого патча из перечисления).

Секция сборки исходного текста (%build) включает в себя инструкции для непосредственной сборки предварительно обработанных исходных текстов в директории сборки.

Результат работы сборки проходит следующий этап в секции (%install) с установкой собранного ПО в локальный корневой каталог пакета с настройкой пользовательских прав доступа.

Секция файлов (%files) содержит перечисление файлов, полученных в результате сборки и установки в локальный корневой каталог, которые устанавливаются в пользовательскую систему при установке пакета.

Секция журнала изменений (%changelog) включает в себя историю релизов пакета.

Сборка RPM-пакета происходит в директории, содержащей, в общем случае, архив с исходными текстами программы, а также спрес-файл. Классическая сборка RPM-пакета из исходных текстов происходит посредством вызова следующей команды:

```
\$ rpmbuild -ba имя_спес-файла.спес
```

### *Средства пакетизации ALT Linux*

Дистрибутивы ALT Linux предоставляют инструменты собственной разработки для более надёжной и удобной в большинстве случаев сборки RPM-пакетов.

Например, классическая сборка RPM-пакета опирается на уже предустановленные в системе требуемые зависимости. В процессе написания спес-файла можно забыть про некоторые требуемые зависимости для сборки, которые уже содержатся в системе. Также требуемые зависимости для сборки могут быть специфичными для сборки конкретных исходных файлов и быть нужными только непосредственно при сборке. Для того, чтобы решить проблему в точности требуемых пакетов и остаточных пакетов после процесса сборки необходимо производить сборку в "чистой" и временной системе.

Такую систему эмулирует Hasher – собственная разработка Базальт СПО. Hasher создает "чистую" и контролирующую среду внутри операционной системы, в которой производится сборка RPM-пакета. Изолированность среды сборки позволяет вне зависимости от конфигурации системы пользователя повторить результат сборки RPM-пакета на другом компьютере и для любой из веток репозитория (подробнее в 1.2).

Сборка при помощи Hasher происходит от обычного пользователя, добавленного с помощью `hasher-useradd` (подробнее про настройку Hasher в 2.1):

```
\$ hsh ~/hasher имя_архива.src.rpm
```

Где `/hasher` – директория, в которой строится сборочная среда (`chroot`). Рекомендуется это делать внутри домашней директории.

Также Базальт СПО был разработан инструмент Gear, который делает процесс сборки пакетов из исходных файлов более удобным. Например, очень часто исходные тексты ПО содержатся в Git-репозиториях. Таким образом, Gear позволяет собирать RPM-пакеты напрямую из клонированного Git-репозитория, являясь, грубо говоря, более высокоуровневой надстройкой над `rpmbuild` с использованием `Git`.

Gear-репозиторий – это Git-репозиторий, содержащий спес-файл и инструкцию архивации в `.gear/rules`, которая, чаще всего, содержит:

tar: .

Что означает упаковку в tar-архив данных, содержащихся в текущей директории. Причем архивация происходит не столько из исходных файлов репозитория, сколько из истории Git-репозитория. Поэтому все изменения в Gear-репозитории необходимо сохранять в истории (команда `git commit` содержимого `git`-индекса).

Сборка при помощи Gear происходит посредством вызова следующей команды (находясь в директории Gear-репозитория):

```
\$ gear-rpm -ba
```

Спец-файл в таком случае не требуется передавать явно, так как он будет найден в истории Git.

Однако и Gear, и Hasher могут быть использованы вместе, что даёт удобную, надёжную и "чистую" сборку RPM-пакета. И производится это при помощи следующей команды (находясь в директории Gear-репозитория):

```
\$ gear-hsh
```

## **1.2 Проект "Сизиф" и процедура Join**

## **1.3 Системы межпроцессного взаимодействия**



- 2 Разработка системы конфигурации операционной системы ALT Linux**
  - 2.1 Инструментарий для разработки и пакетизации**
  - 2.2 Реализация иерархии компонентов alterator-application-components**

## **ЗАКЛЮЧЕНИЕ**

Заключение.

Показано, как можно оформить документ в соответствии:

- с правилами оформления курсовых и выпускных квалификационных работ, принятых в Саратовском государственном университете в 2012 году;
- с правилами оформления титульного листа отчета о прохождении практики в соответствии со стандартом.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- 1 Создание приложения Windows Forms с помощью .NET Framework (C++) [Электронный ресурс]. — URL: [http://msdn.microsoft.com/ru-ru/library/vstudio/ms235634\(v=vs.100\).aspx](http://msdn.microsoft.com/ru-ru/library/vstudio/ms235634(v=vs.100).aspx) (Дата обращения 12.07.2013). Загл. с экр. Яз. рус.

**ПРИЛОЖЕНИЕ А**  
**Какое-то приложение**