

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики и  
информационных технологий

АВТОМАТИЗАЦИЯ РАЗВЁРТЫВАНИЯ СЕРВЕРОВ С  
ИСПОЛЬЗОВАНИЕМ KICKSTART

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы  
направления 09.03.01 — Информатика и вычислительная техника  
факультета КНиИТ  
Разумкова Юрия Станиславовича

Научный руководитель

к.ф.-м.н., доцент

\_\_\_\_\_

В.А. Поздняков

Заведующий кафедрой

к. ф.-м. н.

\_\_\_\_\_

Л. Б. Тяпаев

Саратов 2017

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Linux .....	4
1.1 Fedora .....	4
1.2 Anaconda .....	5
1.3 Kickstart .....	5
2 Используемые языки и технологии .....	6
2.1 Go .....	6
2.2 HTML .....	6
2.3 CSS .....	7
2.4 JavaScript .....	7
2.5 Materialize.css .....	7
3 Описание приложения .....	8
3.1 База данных .....	8
3.2 Структура приложения .....	9
3.2.1 Пакет server .....	9
3.2.2 Kickstart template .....	9
3.2.3 Пакет generator .....	9
3.2.4 Статическое содержимое .....	10
3.2.5 Шаблоны страниц сайта .....	10
3.2.6 Пакет main .....	11
3.3 Конфигурация TFTP сервера .....	12
3.4 Конфигурация PXE .....	13
ЗАКЛЮЧЕНИЕ .....	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	15

## ВВЕДЕНИЕ

У большинства крупных компаний довольно часто встаёт вопрос о развёртывании большого количества серверов, несмотря на переход части из них в облачные хостинги. Для автоматизации развёртывания серверов в облаке есть много различных приложений, но для развёртывания физических серверов количество решений не так велико.

Одно из таких решений – использование файлов конфигурации kickstart. Такие файлы конфигурации используют установщики разных дистрибутивов linux. В частности, в этом проекте будет использоваться дистрибутив Fedora версии 25. Приложение делает возможным определение мак-адреса сервера и генерирование по нему нужного файла, который можно настроить в этом же приложении.

Целью бакалаврской работы является автоматизация развёртывания серверов, основанных на RPM пакетах, так как развёртывание занимает много времени, так как для этого необходимо вручную написать kickstart файл.

Для достижения этой цели были сформулированы следующие задачи:

- разработка приложения для автоматического генерирования файлов kickstart
- настройка TFTP сервера
- настройка PXE

## 1 Linux

Linux - это семейство Unix-подобных операционных систем на базе ядра Linux, которые включают тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе как правило создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения.

Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов, то есть в форме, которая готова для установки и удобна для сопровождения и обновления, а также имеющая свой набор системных и прикладных компонентов, как свободных, так, возможно, и собственных.

В наше время открытое свободное программное обеспечение всё больше и больше набирает популярность как по причине возможной слежки за пользователями в закрытом программном обеспечении, а также из-за удобства его использования и настройки.

В этой работе используется дистрибутив Fedora версии 25, так как она rpm based и, что важно для задачи, поставленной в данной работе, довольно часто обновляется, из-за чего приходится часто обновлять или переставлять серверы.

### 1.1 Fedora

Fedora – это дистрибутив операционной системы GNU/Linux. Этот дистрибутив спонсируется фирмой Red Hat и поддерживается сообществом.

Проект служит для тестирования новых технологий, которые в дальнейшем включаются в продукты Red Hat и других производителей. Компания Red Hat не предоставляет поддержку пользователям Fedora, поддержка осуществляется открытым сообществом.

Целью проекта Fedora является построение целостной операционной системы из свободного программного обеспечения. Версии выходят каждые 6-8 месяцев.

Разработчики дистрибутива fedora поддерживают только 3 последних версии, то есть максимальный срок использования одной версии составляет всего лишь полтора года. С приложением, разработанным в рамках этой работы переустановка системы на серверы и обычные компьютеры становится

намного проще и делается буквально в пару кликов.

## 1.2 Anaconda

Anaconda – это специальная программа, которая используется для установки операционной системы, используемая как в fedora, так и в следующих дистрибутивах:

- Fedora
- Red Hat Enterprise Linux
- FrameOS Linux
- Scientific Linux

Anaconda написана на языке программирования python и находится в свободном доступе на сервисе github. Именно файлы конфигурации для этого установщика называются kickstart

## 1.3 Kickstart

Используя kickstart установку возможно установить операционную систему с помощью всего лишь одного файла. Эта система была разработана Red Hat в первую очередь для установки RHEL и Fedora. Для использования kickstart необходимо:

- Создать kickstart файл
- Сделать этот файл доступным, например, по http и указать путь до него программе установки
- Начать установку операционной системы, используя установщик

## 2 Используемые языки и технологии

### 2.1 Go

Go – это компилируемый многопоточный язык программирования, разработанный компанией Google. Первоначальная разработка Go началась в сентябре 2007 года, а его непосредственным проектированием занимались Роберт Гризмер, Роб Пайк и Кен Томпсон, занимавшиеся до этого проектом разработки операционной системы Inferno. Официально язык был представлен в ноябре 2009 года. На данный момент его поддержка осуществляется для операционных систем FreeBSD, OpenBSD, Linux, Mac OS X, Windows, начиная с версии 1.3 в язык Go включена экспериментальная поддержка DragonFly BSD, Plan 9 и Solaris, начиная с версии 1.4 была реализована поддержка платформы Android. Сейчас Go очень активно развивается и разрабатывается множество библиотек для него. [1]

Go используется, в основном, для написания микросервисов и серверных приложений. Он сочетает в себе высокую скорость и кроссплатформенность.

Этот язык был выбран по причине его кроссплатформенности, то есть приложение можно запустить как на операционной системе linux, так и на windows и macOS.

Язык продолжает развиваться, и разработчики рассматривают возможность включения в язык средств обобщённого программирования. В «Часто задаваемых вопросах» по языку приводятся аргументы против использования утверждений, а наследование без указания типа, наоборот, отстаивается.

### 2.2 HTML

HTML - стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. В данном проекте используется HTML5, целью разработки которого было улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости, удобочитаемости кода для человека и простоты анализа для парсеров. [2]

## 2.3 CSS

CSS (каскадные таблицы стилей) – это формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML. [3]

## 2.4 JavaScript

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript.

Этот язык программирования обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. [4]

JavaScript используется в клиентской части веб-приложений: клиент-серверных программ, в котором клиентом является браузер, а сервером — веб-сервер, имеющих распределённую между сервером и клиентом логику. Обмен информацией в веб-приложениях происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами.

## 2.5 Materialize.css

В этом проекта также используется css библиотека materialize.css, которая позволяет уделять меньше внимания разметке и больше – функциональности приложения. Также использование данной библиотеки позволяет сделать красивый и удобный интерфейс с точки зрения пользователя приложения.

### 3 Описание приложения

В процессе разработки приложения были использованы следующие языки и фреймворки: Go 1.8.1 [1] [6], MariaDB 10.1.21, HTML 5, CSS 3, materialize.css. Сначала была разработана база данных для хранения информации о серверах, пакетах, пользователях и скриптах, затем был написан backend на языке go. Для отображения интерфейса был использован встроенный web-сервер go и go templates для встраивания данных из базы данных в страницы.

#### 3.1 База данных

База данных построена на основе СУБД MariaDB 10.1.21, запущенной на сервере с fedora 25. Она состоит из 7 таблиц.

В таблице ksHosts указаны основные характеристики сервера, которые нужны для правильного создания файла конфигурации, такие как: MAC-адрес устройства, количество дисков, версия операционной системы fedora и имя хоста. Также у каждой строки этой таблицы есть уникальный идентификатор, который используется для однозначного определения хоста, что позволяет получать доступ к данным по идентификатору быстрее и позволяет делать связи между таблицами, используя внешние ключи.

В таблице ksPackages хранятся имена пакетов с уникальным идентификатором. Таблица ksHostPackages нужна для связи хоста с пакетами, которые необходимо установить на него. Таблица ksUsers служит для хранения пользователей серверов. В ней хранится идентификатор, логин, хэш пароля и публичный ssh ключ для доступа на сервер. Пароль хранится в хэшированном виде для того, чтобы его не мог украсть злоумышленник. В таблице ksHostUsers хранятся связи пользователя и сервера.

В таблице ksPostScripts хранятся скрипты, которые выполняются после установки операционной системы на сервер. В ней находится уникальный идентификатор, название скрипта (для упрощения выбора скриптов на странице сервера) и сам скрипт. Скрипты должны быть написаны на bash. Связи между скриптом и сервером установлены с помощью таблицы ksHostsPostScripts.

Используя данные из этих таблиц происходит создание файла Kickstart, который используется для установки операционной системы на сервер.



## 3.2 Структура приложения

Основное приложение написано на языке программирования Go и содержит пакеты `server`, `generator` и `main`, файлы интерфейса, файлы `kickstart`, статический контент, также была проведена настройка `tftp` сервера и конфигурирование `pxe`.

### 3.2.1 Пакет `server`

Данный пакет нужен для хранения соединения с базой данных, чтобы не переподключаться к ней с каждым запросом, то есть на всё приложение будет только одно соединение с базой, что ускоряет загрузку, а также уменьшает нагрузку как на приложение, так и на базу данных

### 3.2.2 Kickstart template

В данной директории хранится шаблон для создания `kickstart` файла.

### 3.2.3 Пакет `generator`

Пакет `generator` используется для непосредственного генерирования `Kickstart` файла.

В пакете `generator` используется несколько структур:

- `User` - структура повторяет таблицу `ksUsers` с дополнительным полем `IsSelected`, в котором хранится значение для возможности удобного редактирования серверов
- `ServerPackage` - структура повторяет таблицу `ksPackages` с дополнительным полем `IsSelected`, в котором хранится значение для возможности удобного редактирования серверов
- `PostScript` - структура повторяет таблицу `ksPostScripts` с дополнительным полем `IsSelected`, в котором хранится значение для возможности удобного редактирования серверов
- `MainServerData` - структура повторяет таблицу `ksHosts`
- `SystemConfig` - структура, которая используется вместе с шаблоном файла `kickstart`. Она содержит данные о версии операционной системы, имя сервера, данные о разделах, список пакетов, скрипт, который выполняется после установки и список пользователей.

Данные о разделах генерируются функцией `generatePartitions`. Для разделов используется файловая система `btrfs`.

В процессе работы этой программы описываются разделы `/boot/`, `swap` и `/`. Раздел `/boot/` нужен для хранения загрузчика и ядра. `Swap` нужен для хранения информации из оперативной памяти при её нехватке. Раздел `/` – для хранения операционной системы, приложений и всех данных, включая пользовательские.

Функция `generateKickstartFile` служит для создания строки, в которой содержится конфигурационный файл для установки операционной системы на сервер, используя переменную структуры `SystemConfig` и шаблон файла конфигурации `kickstart`.

Функция `GenerateKickstartFileFromServerData` запрашивает информацию из базы данных с помощью вспомогательных функций, которые выполняют запросы из базы данных, которые указаны в разделе «база данных».

Также в этом пакете есть функция `IsConfigExist`, которая принимает мак-адрес и возвращает 2 значения – первое `bool` – существует ли конфигурация для данного мак-адреса и информация о сервере, которая будет пустой, если конфигурация не найдена.

### 3.2.4 Статическое содержимое

Для статического содержимого в приложении предусмотрена директория `/static/`. В ней хранятся файлы, которые не должны каждый раз создаваться динамически, такие как:

- файлы стилей (`css`)
- файлы скриптов (`js`)
- шрифты

Все данные из этой папки доступны любому человеку по адресу `/static/`

### 3.2.5 Шаблоны страниц сайта

В директории `/newInterface/templates/` находятся шаблоны основных страниц веб интерфейса приложения:

- Главная страница приложения
- Страница пакетов
- Страница скриптов
- Страница серверов
- Страница пользователей

В файле `index.tmpl.html` находится страница авторизации, и по совместительству, главная страница приложения.

В файлах `packages.tmpl.html`, `postScripts.tmpl.html`, `servers.tmpl.html` и `users.tmpl.html` соответственно находятся шаблоны для страниц пакетов, скриптов, серверов и пользователей серверов. Это шаблоны впоследствии используются приложением для отображения данных.

### 3.2.6 Пакет main

В пакете `main` находятся все функции для отображения интерфейса, вывода данных в него, редактирования полей и их удаления, а также для определения физического адреса сервера.

В данном пакете используются следующие библиотеки:

- `html/template` - для работы с шаблонами [7]
- `strconv` - для конвертации строк в числа и обратно [8]
- `strings` - для работы со строками [9]
- `crypto/sha256` - для шифрования паролей алгоритмом `sha256` для обеспечения безопасности паролей [10]
- `encoding/hex` - для перевода хэшей в шестнадцатичную систему для удобства хранения [11]
- `encoding/json` - для парсинга и закидывания в переменную файла конфигурации, а также для чтения запроса и ответа на него [12]
- `errors` - для создания сообщений об ошибке [13]
- `flag` - для считывания флагов командной строки [14]
- `fmt` - для вывода информации в поток вывода `http` сервера [15]
- `io/ioutil` - для работы с файлом конфигурации [16]
- `log` - для логирования информации в консоль приложения [17]
- `net/http` - `http` сервер [18]

Также используются библиотеки с открытым исходным кодом для более удобной работы с базой данных:

- `github.com/gorilla/sessions` - для работы с сессиями [19]
- `gopkg.in/ldap.v2` - для работы с доменом и авторизации [20]
- `github.com/go-sql-driver/mysql` - как драйвер для MySQL [21]
- `github.com/jmoiron/sqlx` для упрощения работы БД с сложными структурами данных. [22]

Функция `serversHandler` отвечает за отображение страницы серверов.

В ней запрашиваются все пакеты, пользователи, скрипты и серверы. Первые три нужны для добавления данных к серверу, а переменная с серверами нужна для отображения списка серверов в интерфейсе приложения. При редактировании пакеты, пользователи и скрипты запрашиваются специально написанными нестандартными функциями, для того, чтобы выбранные элементы были выбраны при редактировании.

Функция `packagesHandler` нужна для отображения страницы пакетов. В ней запрашиваются только пакеты, для отображения в общем списке и отдельный пакет при редактировании.

Функция `usersHandler` нужна для отображения страницы пользователей. В ней запрашивается список пользователей из базы данных и отдельный пользователь при редактировании

Функция `postScriptsHandler` отвечает за отображение страницы скриптов. В ней запрашиваются список скриптов и отдельный скрипт при редактировании.

На этом закончено описание функций для отображения страниц для авторизованных пользователей. Сама же авторизация происходит на главной странице, используя `ldap`. За это отвечают функции:

Слующие функции нужны для авторизации пользователя:

- `indexHandler` – для отображения главной страницы и перенаправления с неё на страницу серверов для уже авторизованных пользователей
- `auth` – функция, которая производит авторизацию по `ldap` и возвращает имя пользователя, если авторизация прошла успешно
- `loginHandler` – вызывает функцию `auth`, и, если авторизация прошла успешно, то записывает в сессию эти данные и перенаправляет на страницу сервера
- `logoutHandler` – удаляет сессию

В функции `main` происходит чтение флагов флагов, загрузка конфигурации в приложение, соединение с базой данных, привязка функций к страницам и запуск сервера на определённом порту.

### 3.3 Конфигурация TFTP сервера

Для работы `tftp` сервера на операционную систему `fedora` нужно выполнить команду: `sudo dnf install tftpd`. Тогда в папке `/tftpboot` будут храниться данные, который будут доступны по `tftp`

### 3.4 Конфигурация PXE

Pxelinux конфигурация выглядит следующий образом:

```
label fedora25-x86_64_ksgen
kernel http://path/to/kernel
append initrd=http://path/to/inird.img inst.repo=http://link/to/repo ip=dhcp
lang=ru_RU.UTF-8 keymap=us highres=off resolution=1024x768 biosdevname=0
ipv6=auto net.ifnames=0 ks=http://link.to.ksgen.server/ks/ kssendmac
```

Важные параметры для работы нашего приложения – это `ks`, который представляет собой ссылку на наш сервер с запущенным сервисом и `kssendmac`, который заставляет сервер передавать мак-адрес в заголовках `http`.

Листинг приложения находится на диске, приложенном к работе.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы было разработано приложение для автоматического создания файлов kickstart и автоматизирована установка операционной системы linux дистрибутива fedora на серверы. Также была проведена установка и настройка TFTP сервера и PXE.

Таким образом, все поставленные в бакалаврской работе задачи выполнены.

В результате было достигнуто значительное уменьшение затраченного времени на развёртывания серверов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Mark Summerfield Programming in go / Mark Summerfield // Addison-Wesley / 2015
- 2 HTML MDN [Электронный ресурс] / 2017 URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (Дата обращения: 20.05.2017).
- 3 CSS docs [Электронный ресурс] / 2017 URL: <https://www.w3schools.com/css/> (Дата обращения: 20.05.2017).
- 4 JS docs [Электронный ресурс] / 2017 URL: <https://www.w3schools.com/js/> (Дата обращения: 20.05.2017).
- 5 Bootstrap [Электронный ресурс] / 2017 URL: <http://getbootstrap.com/> (Дата обращения: 20.05.2017).
- 6 Documentation – The go programming language [Электронный ресурс] / 2017 URL: <https://golang.org/doc/> (Дата обращения: 21.05.2017).
- 7 Package html/template [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/html/template> (Дата обращения: 21.05.2017).
- 8 Package strconv [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/strconv> (Дата обращения: 21.05.2017).
- 9 Package strings [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/strings> (Дата обращения: 21.05.2017).
- 10 Package crypto/sha256 [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/crypto/sha256> (Дата обращения: 21.05.2017).
- 11 Package encoding/hex [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/encoding/hex> (Дата обращения: 21.05.2017).
- 12 Package encoding/json [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/encoding/json> (Дата обращения: 21.05.2017).
- 13 Package errors [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/errors/> (Дата обращения: 21.05.2017).
- 14 Package flag [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/flag/> (Дата обращения: 21.05.2017).

- 15 Package fmt [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/fmt/> (Дата обращения: 21.05.2017).
- 16 io/ioutil [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/io/ioutil> (Дата обращения: 21.05.2017).
- 17 Package log [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/log/> (Дата обращения: 21.05.2017).
- 18 Package http [Электронный ресурс] / 2017 URL: <https://golang.org/pkg/net/http/>
- 19 Sessions in golang [Электронный ресурс] / 2017 URL: <https://github.com/gorilla/sessions> (Дата обращения: 21.05.2017).
- 20 LDAP [Электронный ресурс] / 2017 URL: <http://gorpkg.in/ldap.v2> (Дата обращения: 21.05.2017).
- 21 Mysql [Электронный ресурс] / 2017 URL: <https://github.com/go-sql-driver/mysql> (Дата обращения: 21.05.2017).
- 22 Sqlx [Электронный ресурс] / 2017 URL: <https://github.com/jmoiron/sqlx> (Дата обращения: 21.05.2017).