Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ЮГОРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Высшая инженерная школа цифровых технологий

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки, специальности)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

(наименование профиля, специализации)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(БАКАЛАВРСКАЯ РОБОТА)

На тему «Сборка дистрибутива операционной системы с прикладным программным обеспечением на базе ядра Linux для потребностей ЮГУ»

Студент (ка) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия имя отчество) (личная подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия имя отчество) (личная подпись)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия имя отчество) (личная подпись)

Допустить к защите

Руководитель образовательной программы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученая степень, звание, фамилия имя отчество) (личная подпись)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

г. Ханты-Мансийск

2024 год

Реферат

на выпускную квалификационную бакалаврскую работу

Щербакова Д.И.

Бакалаврская работа содержит (?) с., (?) рисунков, (?) таблиц, (?) приложений, (?) источников.

(?)(Перечень ключевых слов (с заглавной буквы))

Целью бакалаврской работы является разработка информационной системы «Ugrach».

Объектом исследования является (?). Предметом исследования является сборка операционной системы на базе ядра Linux, способной взаимодействовать с доменом Windows Active Directory.

Выполнение работы включает в себя несколько этапов.

На первом этапе составляются требования к операционной системе, формируется список предустановленных программ и настроек. Пользователь должен иметь возможность с тем же успехом выполнять поставленные задачи, что и на операционной системе под управлением Windows.

На втором этапе формируется конфигурация для утилиты «live-build», которая будет использоваться для дальнейшей сборки.

На третьем этапе с помощью утилиты «lb build» реализуется сборка системы, по завершению которой формируется файл  с расширением «.iso», являющийся образом конечной системы.

На последнем этапе полученная система устанавливается на рабочую машину и тестируется.

Оглавление

[Термины и определения 5](#_Toc1)

[Перечень сокращений и обозначений 7](#_Toc2)

[ВВЕДЕНИЕ 8](#_Toc3)

[Основная часть 11](#_Toc4)

[1 Аналитическая часть 11](#_Toc5)

[1.1 Описание разрабатываемой системы 11](#_Toc6)

[1.2 Обзор аналогов 12](#_Toc7)

[1.2.1 ALT Linux 12](#_Toc8)

[1.2.2 Astra Linux 12](#_Toc9)

[1.2.3 РЕД ОС 13](#_Toc10)

[1.2.4 РОСА ОС 14](#_Toc11)

[1.2.5 Сравнительная таблица аналогов 14](#_Toc12)

[2 Проектная часть 16](#_Toc13)

[2.1 Функционал разрабатываемой системы 16](#_Toc14)

[2.2 Обзор важных компонентов системы 17](#_Toc15)

[2.2.1 Samba 17](#_Toc16)

[2.2.2 Kerberos 18](#_Toc17)

[2.2.3 PAM 20](#_Toc18)

[2.2.4 Winbind 21](#_Toc19)

[3 Практическая часть 22](#_Toc20)

[3.1 Структура конфигурации для live-build 22](#_Toc21)

[3.1.1 Директория package-lists 22](#_Toc22)

[3.1.2 Директория archives 23](#_Toc23)

[3.1.3 Директория includes.chroot 25](#_Toc24)

[3.1.4 Директория hooks 25](#_Toc25)

[3.1.5 Директория preseed 26](#_Toc26)

[3.2 Параметры сборки системы 26](#_Toc27)

[3.3 Настройка системы в процессе сборки 27](#_Toc28)

[3.4 Настройка подключения к домену и доменной авторизации 28](#_Toc29)

[3.4.1 Настройка DNS 29](#_Toc30)

[3.4.2 Настройка синхронизации времени 31](#_Toc31)

[3.4.3 Настройка авторизации через Kerberos 32](#_Toc32)

[3.4.4 Настройка Samba и Winbind 34](#_Toc33)

[3.4.5 Настройка PAM 37](#_Toc34)

[3.4.6 Подключение к домену 37](#_Toc35)

[3.5 Настройка системы после установки 40](#_Toc36)

[3.5.1 Запуск systemd сервисов и включение темы оформления 43](#_Toc37)

[3.5.2 Установка Flatpak приложений 44](#_Toc38)

[3.5.3 Установка драйверов GPU 45](#_Toc39)

[3.5.4 Установка VirtualBox 46](#_Toc40)

[3.5.5 Настройка списка репозиториев 46](#_Toc41)

[3.6 Автоматизация процесса установки системы 47](#_Toc42)

[Заключение 50](#_Toc43)

[Список использованных источников 51](#_Toc44)

[Приложения 52](#_Toc45)

# Термины и определения

Active Direcory – службы каталогов корпорации Microsoft для операционных систем семейства Windows Server;

Доменная служба Active Directory – решение Microsoft для операционных систем Windows Server, позволяющая загружать, группировать и хранить информацию об объектах ИТ-инфраструктуры;

Дистрибутив GNU/Linux – общее определение и название операционных систем, использующих ядро Linux, готовых для конечной установки на пользовательское и серверное оборудование;

Debian – свободный дистрибутив GNU/Linux.

LiveCD (Live) – операционная система, которая загружается со сменного носителя (компакт-диска или USB-накопителя);

Критерии Debian по определению свободного ПО (англ. Debian Free Software Guidelines) – набор правил, по которым проект Debian определяет, какие лицензии являются свободными, и следовательно, приемлемыми для операционной системы Debian.

KDE

PAM

Flatpak

Flathub

Kerberos

Samba

DNS

NetworkManager

KDC

TGS

демон

NSS

Bash

Systemd

Flatpak runtime

apt

Xorg

Wayland

GRUB

UEFI

# Перечень сокращений и обозначений

ЮГУ – Югорский государственный университет

AD – Active Directory

ОС – операционная система

Домен – Доменная служба Active Directory

ВКР – выпускная квалификационная работа

GPU (graphics processing unit) – графический процессор

CPU (central processing unit) – центральный процессор

PAM

DNS

NTP

UTC (Coordinated Universal Time) – всемиирное координиированное время

ID (identifier) – уникальный идентификатор

SID (Security Identifier) – идентификатор безопасности

RTC (Real Time Clock) -

CLI

SSH

nss

# ВВЕДЕНИЕ

После событий 24 февраля 2022 года на нашу страну многими зарубежными корпорациями были наложены санкции. Американская публичная транснациональная корпорация «Microsoft», наиболее известная производством семейства операционных систем «Windows» и офисных приложений «Microsoft Office», не стала исключением. В марте 2022 года было объявлено о приостановке продаж продуктов и услуг россиянам. Лицензии на офисные программы не продлевались даже частным пользователям. В сентябре того же года была заблокирована возможность обновления Windows 10 и 11. Сделать это можно было только с помощью обходных маневров — через VPN или приобретение ключей активации по параллельному импорту. В апреле 2023 года Microsoft объявила, что не будет предоставлять в России гарантийное обслуживание игровых консолей «Xbox». С 30 сентября 2023 года прекратит продлевать лицензии на программное обеспечение для корпоративных пользователей. Об этом сообщило агентство TelecomDaily со ссылкой на письмо, полученное от технической поддержки разработчика. Причиной является то, что американская компания больше не может принимать платежи за продление лицензий на российские счета.

Все эти ограничения создают серьёзные неудобства как для обычных пользователей, так и для организаций, поскольку и те и другие в основной совей массе используют операционные системы на базе «Windows» и офисный пакет «Microsoft Office». В текущих реалиях вполне вероятно, что «Microsoft» полностью уйдет с российского рынка, что сделает невозможным использования лицензионных продуктов компании.

Все эти факторы побуждают искать аналоги вышеупомянутых цифровых продуктов. В первую очередь, в этом заинтересованы организации, поскольку использование ими нелицензионных продуктов недопустимо(?).

Среди офисных решений можно найти немало продуктов с открытым исходным кодом, которые могут стать достойной заменой «Microsoft Office». (Среди них: «LibreOffice», «OnlyOffice», «OpenOffice», «МойОфис» и другие.) Выбрать подходящий аналог не является сложной задачей.

С операционными системами же ситуация обстоит иначе. Существует немало российских систем на базе ядра «Linux», которые могут стать достойной альтернативой Microsoft Windows. (Самые известные из них: «ALT Linux», «РЕД ОС», «Astra Linux», «ЛотОС», «ROSA Linux».) Хотя некоторые из них являются бесплатными для физических лиц, для юридических лиц все из них, также как и «Microsoft Windows», являются платными. Более того, любая из вышеперечисленных систем требует детальной настройки для того, что удовлетворять всем требованиям конкретной организации. Это могут требования к безопасности, производительности, наличию тех или иных программных пакетов и так далее.

Решением может стать сборка своей собственной операционной системы, удовлетворяющей всем необходимым критериям, чему и посвящена данная исследовательская работа. Поскольку ядро «Linux» имеет открытый исходный код и распространяется по лицензии «GNU General Public License», сборка на его основе является наиболее удобной. Создание своей операционной системы «с нуля» является весьма нетривиальной задачей, требующей немало временных и трудовых ресурсов, а также квалифицированных специалистов, что может сделать разработку очень недешёвой. Альтернативой может быть сборка «Linux» системы на основе уже существующих дистрибутивов с помощью предназначенных для этого утилит. Наиболее известными среди них являются «live-build» и «Archiso», позволяющие создать свою кастомизированную(?) операционную систему на базе дистрибутивов «Debian» и «Arch Linux» соответственно.

Организацией, для которой будет собираться конечная операционная система, является Югорский государственный университет. Утилитой для сборки была выбрана «live-build». Причиной данного выбора является то, что «Arch Linux» - «роллинг-релиз» дистрибутив (?)и, как следствие, требует регулярных обновлений, которые могут негативно сказаться на стабильности работы системы. «Debian» же, напротив, получает глобальные обновления раз в два года. При этом, каждый релиз предварительно тщательно тестируется, что повышает отказоустойчивость системы.

Создание своего собственного дистрибутива «Linux» позволит Югорскому государственному университету использовать на своих компьютерах операционные системы из двух разных семейств, если на то будет необходимость, и, случае полного ухода компании «Microsoft» с российского рынка, совершить плавный переход с операционной системы «Windows».

# Основная часть

## Аналитическая часть

### Описание разрабатываемой системы

Дистрибутив Linux – общее определение и название операционных систем, использующих ядро Linux, готовых для конечной установки на пользовательское и серверное оборудование. Такая операционная система состоит из ядра Linux, набора библиотек и утилит, а также, как правило, графической подсистемы и набора приложений, требуемых для работы с документами, таблицами, мультимедиа, графикой, базами данных и т. д.

Дериватив – дистрибутив, построенный и выпускающий релизы на основе другого дистрибутива. Это в первую очередь означает, что пакетная база дериватива полностью наследуется из оригинального дистрибутива, и, при необходимости, расширяется дополнительными пакетами.

Разработка дериватива удобна тем, что изначальный дистрибутив уже готов к установке, способен выполнять базовые задачи и, как правило, тщательно протестирован. С другой стороны, имеется зависимость от решений, принимаемых разработчиками базовой системы. Любые изменения, сделанные в базовой системе, даже если они являются нежелательными, в любом случае попадут в дериватив со следующим релизом.

За основу собираемой системы была выбрана стабильная версия дистрибутива Debian. Таким образом, конечная система будет являться деривативом. Этот подход к разработке был выбран по следующим причинам:

* Проверенная надежность дистрибутива Debian;
* Скорость разработки;
* Значительно меньшая сложность разработки по сравнению со сборкой системы «с нуля»;
* Не требует вложения денежных средств;
* Возможность собрать систему, работающую в Live режиме.

### Обзор аналогов

#### ALT Linux

ALT Linux – отечественное семейство дистрибутивов, являющееся отдельной ветвью дистрибутивов Linux. Основан на дистрибутиву Mandrake. Разработкой занимается компания «Базальт СПО». Является одним из самых старых отечественных дистрибутивов – его разработка началась в 1999 году.

Дистрибутив доступен в нескольких редакциях: «Альт Рабочая станция», «Альт Рабочая станция K», «Альт Сервер», «Альт Сервер Виртаулизации», «Альт Образование» и «Симпли Линукс». Все редакции являются бесплатными для скачивания и использования физическим лицам, но платными для юридических.

Интересной для рассмотрения является редакция «ALT Рабочая Станция». Это операционная система широкого назначения, имеющая широкий набор программ и драйверов для современных устройств. Поставляется с графической оболочкой «MATE». «ALT Рабочая Станция K», в свою очередь, поставляется с графической оболочкой «KDE Plasma». Дистрибутив настроен для работы в корпоративной сети, в том числе с доменной структурой. В качестве контроллера домена может выступать как Microsoft AD, так и «Альт Сервер».

#### Astra Linux

Astra Linux – операционная система на базе ядра Linux, внедряемая в России в качестве альтернативы Microsoft Windows в государственных организациях. Основывается на Debian. Разработкой системы занимается группа компания «Астра» – один из лидеров российского рынка информационных технологий в области разработки ПО и средств защиты информации.

Доступно два варианта операционной системы – «Astra Linux Common Edition», «Astra Linux Special Edition». «Astra Linux Common Edition» является системой общего назначения, предназначения для решения повседневных задач. «Astra Linux Special Edition» в свою очередь подразделяется на четыре редакции: «Astra Linux Server», «Astra Linux Desktop», «Astra Linux Mobile» и «Astra Linux Embedded». Ключевой особенностью данной редакции является её высокая защищенность. Система может без труда противостоять как различным киберугрозам, так и вредоносной активности, вызванной пользователями или системным администратором. Является единственной системой в России, которая полностью соответствует требованиям безопасности информации всех основных регуляторов страны.

В «Astra Linux Common Edition» доступно три режима защищенности: «базовый» - «Орёл», «усиленный» - «Воронеж» и «максимальный» - «Смоленск». Начиная с режима «Усиленный», интерфейсы средств защиты информации «Astra Linux Special Edition» существенно сужают поверхность атаки, и формируют основной рубеж обороны от вредоносного внешнего воздействия.

«Astra Linux Common Edition» хоть и является бесплатной для физических лиц, но на данный момент неактуальна, и не лицензируется для использования юридическими лицами. «Astra Linux Special Edition» является полностью платной и её стоимость варьируется от режима защищенности системы.

#### РЕД ОС

РЕД ОС — это российская операционная система на базе ядра Linux, которая разработана компанией «РЕД СОФТ» с использованием открытых исходных кодов и собственных решений. РЕД ОС имеет ряд преимуществ, которые делают её привлекательной для различных сфер применения, особенно для государственных и корпоративных заказчиков.

Операционная система поставляется в двух конфигурациях – РЕД ОС «Рабочая станция» и РЕД ОС «Сервер», и в двух редакциях – сертифицированная и стандартная. Сертифицированная прошла ряд испытаний в системе сертификации средств защиты информации. Стандартная система предназначена для серверов и рабочих станций и распространяется свободно, бесплатно для некоммерческого использования. Сертифицированная редакция является платной и требует приобретения лицензии.

РЕД ОС позволяет настраивать различные параметры системы, такие как интерфейс, сеть, безопасность, обновления и т.д. Система поддерживает различные графические оболочки, такие как «Mate» и «Cinnamon» и «Gnome».

#### РОСА ОС

Rosa Linux — линейка дистрибутивов Linux (изначально основанных на Mandriva), разработку которых ведёт российская компания «НТЦ ИТ РОСА». Так же как и РЕД ОС, представлена в двух версиях – для серверов и рабочих станций. Имеется несколько редакций: РОСА «Кобальт», РОСА «ХРОМ» и РОСА «КОБАЛЬТ». РОСА «ФРЕШ» – полностью бесплатная версия ОС для дома с обширным набором предустановленного ПО. РОСА «КОБАЛЬТ» и «ХРОМ» – универсальные операционные системы, предназначенные для оснащения рабочих мест пользователей в корпоративной среде. Редакция «КОБАЛЬТ» оснащена дополнительными средствами для защиты от внешних и внутренних угроз. В качестве основной графической среды рабочего стола используется KDE.

Все редакции системы, кроме «ФРЕШ», являются платными.

#### Сравнительная таблица аналогов

Для того, чтобы подытожить всю информацию об аналогах, составим таблицу, в которой наглядно будут видны преимущества и недостатки каждого варианта:

Таблица 1: Сравнение аналогов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Astra Linux | ALT Linux | РЕД ОС | РОСА ОС | Ugrach |
| Стоимость лицензии, руб./год | Для физических лиц | - | Бесплатно | Бесплатно | Бесплатно | Бесплатно для ЮГУ |
| Для юридических лиц | 8950 | 5280 | 4600 | 4400 | Бесплатно для ЮГУ |
| Стоимость поддержки | | 4200 | 2160 | 2500 | 3300 | Бесплатно для ЮГУ |
| Возможность доменной авторизации «из коробки» | | Нет | Да | Да | Да | Да |
| Программное обеспечение | | Офис, почта, веб-сервер, средства работы с изображениями | Офис, почта, веб-сервер, средства работы с изображениями | Офис, почта, веб-сервер, средства работы с изображениями | Офис, почта, веб-сервер, средства работы с изображениями | Заточено строго под потребности университета |
| Семейство ОС | | Debian/Linux | Mandrake/Linux | Red Hat | Mandriva/Linux | Debian/Linux |

Как видим, все аналоги, кроме Astra Linux, предоставляют доменную авторизацию «из коробки», обладают набором жизненно необходимых приложений, и имеют сравнительно невысокую стоимость поддержки. Однако возможность включить в Ugrach набор приложения и настроек, специфичных только для ЮГУ, и отсутствие необходимости приобретать лицензию являются сильными аргументами в пользу выбора разрабатываемой системы.

## Проектная часть

### Функционал разрабатываемой системы

Для того, чтобы сотрудники и студенты могли успешно пользоваться системой, она должна удовлетворять ряду требований. Требования были получены от научного руководителя данной работы. Приведем их ниже:

* Возможность ввода машины в домен с помощью специальной консольной утилиты;
* Авторизация пользователей по домену;
* Возможность монтирования сетевых директорий доменных пользователей;
* Работа с принтерами и сканерами;
* Возможность запуска приложений, написанных под Windows;
* Невозможность авторизации локальными пользователями (за исключением Администратора);
* Наличие программ для виртуализации;
* Тема оформления схожая с Windows 10;
* Наличие программ для работы с документами, таблицами, графикой, мультимедиа, базами данных, носителями данных.

### Обзор важных компонентов системы

#### Samba

Ключевым компонентом системы, обеспечивающим подключение к домену «Windows Active Directory», возможность монтирования сетевых папок и взаимодействие с принтерами и сканерами, является Samba. Samba – пакет программ, являющийся свободной реализацией сетевого протокола SMB, изначально разработанный Триджеллом Эндрю. Samba предоставляет файловые сервисы и сервисы печати, а также может интегрироваться с доменом «Microsoft Windows Server», и как контроллер, и как член домена.

Samba работает на большинстве Unix-like подобных системах, таких как Linux, BSD, Solaris, macOS и др. Является стандартом практически для всех Linux дистрибутивов, в которых обычно включается как базовый компонент.

Название Samba происходит от SMB (Server Message Block), названия проприетарного протокола, использующегося в сетевой файловой системе Microsoft Windows.

Для аутентификации пользователей Samba может использовать либо Kerberos, либо NTLM протоколы. Поскольку начиная с версии Windows 2000 Kerberos используется как стандартный протокол аутентификации, рассмотрим его подробнее.

#### Kerberos

Kerberos – сетевой протокол аутентификации, который предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними. Kerberos выполняет аутентификацию в качестве службы аутентификации доверенной третьей стороны, используя криптографический ключ. Построен на криптографии симметричных ключей и требует наличия центра распределения ключей. Расширения Kerberos могут обеспечить использование криптографии с открытым ключом на определённых этапах аутентификации.

Центр распределения ключей (KDС) — это служба, работающая на физически защищённом сервере. Центр хранит базу данных с информацией об учётных записях всех клиентов сети. Вместе с информацией о каждом абоненте в базе центра распределения ключей хранится криптографический ключ, известный только этому абоненту и службе центра. Этот ключ служит для связи клиента с центром.

Процесс аутентификации пользователя по протоколу Kerberos можно описать следующим образом:

1. Запрос к серверу аутентификации. Клиент отправляет незашифрованный запрос  аутентификации KDC серверу;
2. Ответ сервера аутентификации. Если пользователь, отправивший запрос не найден в базе данных, то он не может быть авторизован, и процесс аутентификации прекращается. Если же он был найден, то сервер аутентификации выдает клиенту TGT (Ticket Grant Ticket) – билет на получение билетов и ключ сессии;
3. Как только клиент прошел процедуру аутентификации сервером KDS, он обращается к TGS (Ticket Granting Server) – серверу выдачи мандатов и разрешений. Запрос должен сопровождаться TGT, полученном на предыдущем этапе;
4. Если TGS может осуществить аутентификацию клиента, он отправляет реквизиты для входа и билет обратно клиенту. Эта операция шифруется ключом сессии;
5. Теперь клиент авторизован и может совершать запросы к сервисам.

#### PAM

Linux Pluggable Authentication Modules (PAM, подключаемые модули аутентификации) — набор библиотек, позволяющих настроить методы аутентификации пользователей в UNIX-системах. Он предоставляет возможность гибко и централизованно переключать методы аутентификации для защищенных приложений, используя конфигурационные файлы вместо изменения кода самих приложений. Является одной из частей стандартного механизма обеспечения безопасности UNIX-систем.

Linux-PAM разделяет задачи аутентификации на четыре независимые группы:

* модуль «account» проверят, что указанная учетная запись является допустимой целью аутентификации;
* модуль «authentication» проверят личность пользователя путем запроса пароля или другого секрета;
* модуль «password» отвечает за обновление паролей и, как правило, связаны с модулями, используемыми на этапе аутентификации. Также используются для проверки надежности паролей;
* модуль «session» определяет действия, которые исполняются в начале и конце каждой сессии пользователя. Сессия начинается, когда пользователь успешно прошел аутентификацию.

Поскольку в разрабатываемой системе требуется аутентификация как локальных пользователей, так и пользователей домена Windows Active Directory, необходимо так настроить каждый PAM модуль, чтобы аутентификация одних не мешала аутентификации других.

#### Winbind

Winbind — Name Service Switch «демон» для распознания имен Micfrosoft Windows, являющийся частью пакета Samba. Предоставляет набор сервисов для Name Service Switch, присутствующей в большинстве современных библиотек C, произвольных приложений PAM, а также для самой Samba.

The Name Service Switch позволяет позволяет получать информацию о пользователях и системную информацию из различных баз данных, таких как NIS и DNS. Точное поведение может быть настроено в файле /etc/nsswitch.conf. Пользователи и группы выделяются по мере их разрешения в диапазоне идентификаторов, указанных администратором системы Samba.

Сервис, предоставляемый winbindd называется «winbind» и может быть использован для разрешения информации о группах и пользователях Windows Server. Сервис также предоставляет механизм аутентификации через соответствующий PAM модуль.

Существует множество способов собрать свою собственную операционную систему на базе ядра «Linux». Среди них можно выделить следующие: «Linux From Scratch (LFS)», «Buildroot», «SUSE Studio Express», «Cubic» и т. д. Разберем возможности каждой из них.

.....

## Практическая часть

### Структура конфигурации для live-build

Для сборки системы сначала необходимо составить конфигурацию, из которой утилита lb build уже будет собирать систему. Она представляет из себя набор вложенный директорий, содержащих файлы, из которых читается конфигурация, а также файлы, которые будут включены в конечную файловую систему.

В корне конфигурации находятся директории auto и config. Директория auto содержит sh-скрипты, включающие в себя параметры, которые применяются при вызове утилит пакета live-build. Директория config содержит основную конфигурацию и состоит из нескольких поддиректорий, каждая из которых отвечает за определенные компоненты собираемой системы. Рассмотрим её структуру подробно.

#### Директория package-lists

Директория package-lists содержит списки пакетов, которые будут включены в конечную систему. Может состоять из одного или нескольких файлов, имеющих расширение .list.chroot и .list.binary. Пакеты из файлов с расширением .list.chroot попадают как в устанавливаемую, так и в live систему. Пакеты из файлов с расширением .list.binary включаются только на binary стадии и, как следствие, попадают только в live систему.

Название каждого пакета в файле должно быть с новой строки, а также не должно содержать синтаксических ошибок. В противном случае он не будет найдет и, либо не будет включен в систему, либо прервет сборку с ошибкой.

В файлах допускается использование комментариев и условных конструкций. Например можно включить те или иные пакеты в сборку в зависимости от указанной в конфигурации архитектуры:

#if ARCHITECTURES i386 amd64

memtest86+

#endif

В данном пример в систему будет включена программа для проверки целостности оперативной памяти memtest86+, если в параметрах сборки указаны архитектуры i386 либо amd64.

В текущей конфигурации используется следующий набор файлов со списками пакетов:

* development.list.chroot – программы для разработки ПО и моделирования;
* graphics.list.chroot – программы для работы с графикой;
* media.list.chroot – программы для работы с мультимедиа;
* office.list.chroot – офисный пакет и прочие программы для работы с документами разных форматов;
* virtualization.list.chroot – программы для виртуализации;
* system.list.chroot – системные программы и утилиты;
* games.list.chroot – драйвера для GPU и игровые лаунчеры (также могут использоваться для запуска Windows приложений).

#### Директория archives

Директория archives содержит список дополнительных репозиториев, с которых live-build может скачивать пакеты. При добавлении соответствующего репозитория нужный пакет можно прописать в package-lists также, как и все остальные пакеты. После сборки прописанные репозитории попадают в директорию /etc/apt/sources.list.d/, где хранятся все дополнительные репозитории в Debian системе.

Названия файлов с репозиториями имеют следующий шаблон:

(название\_репозитория).list.(стадия\_сборки)

Название может быть любым, но лучше, чтобы оно отражало содержимое репозитория во избежании путаницы. В качестве стадии сборки могут быть указаны либо chroot, либо binary. От неё зависит, на каком этапе сборки пакеты будут установлены в систему, и, как следствие, наличие или отсутствие соответствующих программ в live системе.

Само содержимое файлов должно иметь следующую структуру:

(тип\_архива) (url репозитория) (псевдоним\_релиза) [ветка...]

Тип архива может быть либо deb, либо deb-src. deb указывает на бинарные пакеты, а deb-src – на пакеты с исходным кодом. В качестве псевдонима релиза указывается одно из названий стабильных релизов Debian. Ветки репозиториев могут быть следующими:

* main – пакеты, считающиеся частью дистрибутива Debian. Являются полностью свободными для любого использования;
* contrib – свободные пакеты, имеющие зависимости, которые могут быть не свободными;
* non-free – не свободные пакеты;
* nom-free-firmware – не свободные пакеты драйверов устройств.

Пример файлы репозитория Visual Studio Code:

deb https://packages.microsoft.com/repos/vscode stable main

Помимо самих списков репозиториев необходимо также добавить соответствующие ключи. Это текстовые файлы, содержащие ключи в формате PGP. Шаблон их именования имеет почти такую же структуру:

(название\_репозитория).key.(стадия\_сборки)

Ключи, также как и списки репозиториев, попадают в конечную систему. За счёт этого пакеты, скачанные в процессе сборки от туда, будут обновляться вместе со всеми остальными после установки системы.

#### Директория includes.chroot

Эта директория содержит файлы, которые буду включены в корневую директорию конечной системы на этапе chroot. Здесь в основном содержатся конфигурации для работы некоторых программ системы, в том числе всех программ, отвечающих за взаимодействие с доменом. Её содержимое будет подробнее рассмотрено далее.

Также допустимо наличие директории includes.binary, файлы из которой попадают в образ live системы, но поскольку для нашей системы в ней нет особой необходимости, в конфигурацию она включена не была.

#### Директория hooks

Данная директория может состоять из поддиректорий hooks/normal и hooks/live, содержащих скрипты, исполняемые на этапе сборки системы. Они позволяют произвести более точную настройку. Директория hooks/normal содержит скрипты, исполняемые при сборки регулярной системы, а hooks/live – при сборке live системы.

Шаблон именования файлов имеет следующий вид:

(номер)(название\_скрипта).(этап сборки)

Номер – это четырёх-значное число, указывающие на очередность исполнения скрипта. Чем оно больше, тем позднее скрипт будет выполнен. Этапами сборки могут быть либо chroot, либо binary, указывающие на соответствующие этапы сборки системы.

Содержание данных скриптов будет подробнее рассмотрено далее.

#### Директория preseed

Директория preseed содержит файл preseed.cfg, которые позволяет предопределить параметры установщика системы. Это позволяет практически полностью автоматизировать процесс установки.

### Параметры сборки системы

Параметры для сборки системы хранятся в файле auto/config. Содержимое файла config представлено ниже:

#!/bin/sh

set -e

lb config noauto \

—apt-recommends true \

—apt-indices true \

—architectures amd64 \

—archive-areas "main contrib non-free non-free-firmware" \

—binary-images iso-hybrid \

—binary-filesystem fat32 \

—bootappend-live "boot=live components locales=ru\_RU.UTF-8 keyboard-layouts=ru,us keyboard-options=grp:alt\_shift\_toggle timezone=Asia/Yekaterinburg" \

—clean \

—color \

—debian-installer live \

—debian-installer-distribution bookworm \

—debian-installer-gui true \

—debootstrap-options "--include=apt-transport-https,ca-certificates,openssl" \

—distribution bookworm \

—firmware-binary true \

—firmware-chroot true \

—security true \

—iso-volume ugrach \

—image-name ugrach.beta \

—linux-packages "linux-image linux-headers" \

—mirror-bootstrap "http://mirror.yandex.ru/debian/" \

—mirror-chroot "http://mirror.yandex.ru/debian/" \

"${@}"

Рассмотрим основные параметры:

* architectures amd64 – архитектура системы
* archive-areas "main contrib non-free non-free-firmware" – ветки репозитория, откуда будут скачиваться пакеты. Установлены все ветки, чтобы в процессе сборки были установленны проприетарные программы и драйвера.
* bootappend-live "boot=live components locales=ru\_RU.UTF-8 keyboard-layouts=ru,us keyboard-options=grp:alt\_shift\_toggle timezone=Asia/Yekaterinburg" – установка дополнительных параметров для live системы: русская локализация, раскладки клавиатуры, комбинация для переключения раскладок и регион.
* debian-installer live – тип установщика. Данная опция добавляет установщик в меню live системы.
* debian-installer-gui true – версия установщика. Добавляется графический вариант. Текстовая версия установщика при этом также присутствует.
* distribution bookworm – релиз Debian. Используется последний – Debian 12 «Bookworm»
* mirror-bootstrap "http://mirror.yandex.ru/debian/" – репозиторий, используемый на bootstrap стадии сборки. В данном случае используется репозиторий Яндекса.
* mirror-chroot "http://mirror.yandex.ru/debian/" – репозиторий, используемый на chroot стадии сборки. Также используются репозитории Яндекса.
* binary-images iso-hybrid – тип собираемого образа. Используется iso-hybrid для возможности записи на USB носитель в будущем.

### Настройка системы в процессе сборки

Для настройки системы в процессе сборки используются скрипты из директории config/hooks/normal. Рассмотрим, что делает каждый из них:

* install-we10xos-theme – скрипт для установки темы оформления KDE.

#!/bin/sh

set -e

git clone https://github.com/yeyushengfan258/We10XOS-kde.git

cd We10XOS-kde

./install.sh

cd ../

rm -rf We10XOS-kde

Сначала клонируем файлы темы из репозитория. Затем переходим в директорию с файлами и запускаем скрипт установки. После успешной установки можем удалить клонированный репозиторий.

* pam-auth-update – скрипт для настройки модулей PAM. Необходим для корректной работы авторизации с систему как доменных пользователей, так и локальных (Администратора).

#!/bin/bash

set -e

sudo pam-auth-update --disable krb5 --enable unix winbind systemd mkhomedir --force

С помощью специальной утилиты для настройки PAM включаемый все нужные нам модули: unix – для авторизации локальных пользователей, winbind – для авторизации доменных пользователей, systemd – для регистрации сессий пользователя в менеджер входа в систему systemd и mkhomedir – для создания домашнего каталога пользователя при входе в систему, если таковой отсутствует.

* add-flathub-repo – скрипт для добавления flathub репозитория в систему для установки flatpak приложений.

#!/bin/sh

set -e

flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo

### Настройка подключения к домену и доменной авторизации

Для реализации доменной авторизации в систему и взаимодействия с сетевыми папками используется пакет программ Samba, система сетевой аутентификации Kerberos и служба Winbind пакета Samba.

В ЮГУ имеется два домена: edu.ugrasu – студенческий домен, и ugrasu.ru – домен для преподавателей. Необходимо настроить систему так, чтобы она могла взаимодействовать с каждым из них. Машина не может одновременно находится в двух доменах, поэтому необходимо реализовать простой способ выхода из одного домена и присоединения к другому. Настройку по умолчанию произведем для преподавательского домена.

Для упрощения этой задачи был написан скрипт «joindomain», отдельные части которого будут рассмотрены в главах далее.

#### Настройка DNS

Для того, чтобы система смогла найти контроллеры домена, необходимо произвести настройку DNS. На установленной системе это можно сделать в файле /etc/resolf.conf. Записи DNS серверов в нём имеют следующий вид:

nameserver (адрес\_сервера)

Директива nameserver указывает на адрес DNS сервера, используемого при разрешении доменных имен. Если от указанного DNS сервера не был получен IP адрес доменного имени, проверяется следующий по списку.

В ЮГУ имеется два контроллера домена: dc.edu.ugrasu – с адресом 192.168.0.200, и pdc.ugrasu.ru – с адресом 192.168.0.128. Так как настройку по умолчанию мы делаем для ugrasu.ru, адрес его контроллера домена должен находится в списке выше.

Однако нельзя просто добавить файл с необходимым содержимым в config/includes/chroot/etc/resolf.conf, поскольку NetworkManager будет перезаписывать его при каждом запуске системы. В этом случае необходимо произвести настройку самого NetworkManager’a, чтобы при каждом запуске он записывал в /etc/resolf.conf DNS адреса контроллеров домена. Сделать это можно в файле /etc/NetworkManager/conf.d/dns-servers.conf. Добавим файл config/includes.chroot/etc/NetworkManager/conf.d/dns-servers.conf со следующим содержимым:

[global-dns-domain-\*]

servers=192.168.0.200,192.168.6.128

Секция global-dns-domain позволяет определить глобальную конфигурацию DNS для конкретного домена. \* в конце указывает на то, что применить это нужно ко всем доменам. Таким образом, после установки сисетмы получим следующий файл /etc/resolf.conf:

# Generated by NetworkManager

nameserver 192.168.0.200

nameserver 192.168.6.128

После чего доменные имена ugrasu.ru и edu.ugrasu будут успешно разрешаться. За разрешения всех остальных доменных имен отвечают DNS службы контроллеров домена.

Помимо файла /etc/resolf.conf необходимо также настроить файл /etc/hosts так, чтобы в нём была запись с полным доменным именем компьютера и коротким именем хоста. Здесь имеются определенные подводные камни. Дело в том, что lb build в процессе сборки переопределяет содержимое этого файла, поэтому мы не сможем просто добавить файл с нужным содержимым в конфигурацию. В отличие от /etc/resolf.conf, /etc/hosts не генерируется NetworkManagerom’ом, так что можно внести необходимые изменения прямо перед подключением к домену. Сделаем это в joindomain:

function configure\_hosts(){

sudo sed -i "s/.\*127.0.0.1.\*/127.0.0.1 $(hostname).$1 localhost/" /etc/hosts

sudo sed -i "s/.\*127.0.1.1.\*/127.0.1.1 $(hostname).$1 $(hostname)/" /etc/hosts

sudo sed -i "0,/::1/ s/.\*::1.\*/::1 $(hostname).$1 localhost ip6-localhost ip6-loopback/" /etc/hosts

}

С помощью утилиты редактирования текста sed подменяем в файле все исходные записи на нужные. Команда «$(hostname)» вернет нам имя хоста, указанное пользователем во время установки. Параметр «$1» передаётся в функцию и содержит название домена. После подключения компьютера с именем хоста «ugrach» к домену «ugrasu.ru» файл /etc/hosts будет иметь следующий вид:

127.0.0.1 debian.ugrasu.ru localhost

127.0.1.1 debian.ugrasu.ru debian

# The following lines are desirable for IPv6 capable hosts

::1 debian.ugrasu.ru localhost ip6-localhost ip6-loopback

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

#### Настройка синхронизации времени

Далее необходимо произвести настройку синхронизации времени между компьютером с контроллером домена. В случае, если разница во времени между ними будет более пяти минут, Kerberos не сможет установить подключение.

Для этой задачи будем использовать программу Chrony, которая является одной из реализация протокола NTP и может подключиться к NTP службе контроллера домена. Состоит из демона chronyd, CLI chronyc и конфигурационного файла chrony.conf. Демон автоматически включается после установки пакеты, поэтому это не нужно делать вручную. Файл хранится в /etc/chrony/chrony.conf, поэтому добавим соответствующий файл в config/includes.chroot/. Он содержит следующие настройки:

keyfile /etc/chrony/chrony.keys

driftfile /var/lib/chrony/chrony.drift

log tracking measurements statistics

logdir /var/log/chrony

maxupdateskew 100.0

hwclockfile /etc/adjtime

rtcsync

makestep 1 3

bindcmdaddress 0.0.0.0

bindcmdaddress ::

server dc.ugrasu.ru iburst

server pdc.edu.udgrasu iburst

* keyfile – расположение файла, содержащего пары ID/ключ для NTP аутентификации;
* driftfile – расположение файла, в котором хранится информация о скорости, с которой системные часы опережают реальное время или же отстают от него;
* log tracking measurements statistics – данная настройка указывает, какую именно информацию нужно логировать;
* logdir – директория, в которой сохраняются логи;
* maxupdateskew – устанавливает порог, при котором отставание или опережение реального времени системным считается недостаточно надёжным для использования;
* hwclockfile – расположение файла adjtime, который используется программой hwclock в Linux, позволяющей определить, использует ли компьютер местное время или UTC;
* rtcsync – включает синхронизацию времени между ядром и RTC;
* makestep – позволяет установить системные часы на необходимое значение вместо постепенной настройки сразу после старта системы, когда разница во времени с контроллером домена может быть ощутимой;
* bindcmdaddress – указывает, на каких интерфейса принимать пакеты. В нашем случае на всех;
* server – доменное имя источника, с которым нужно синхронизировать время. В нашем случае адреса контроллеров домена.

При правильной настройке синхронизации времени присоединение к домену и вход доменных пользователей в систему должен происходить успешно и без задержек.

#### Настройка авторизации через Kerberos

Поскольку для авторизации пользователей домена используется протокол Kerberos, необходимо осуществить его настройку. Файл конфигурации находится в /etc/krb5.conf. Также включим его в конфигурацию для lb build, добавив в config/includes.chroot/etc/krb5.conf. Файл имеет следующее содержание:

[libdefaults]

default\_realm = UGRASU.RU

dns\_lookup\_realm = false

dns\_lookup\_kdc = true

default\_tgs\_enctypes = rc4-hmac des-cbc-crc des-cbc-md5

defaukt\_tkt\_enctypes = rc4-hmac des-cbc-crc des-cbc-md5

permitted\_enctypes = rc4-hmac des-cbc-crc des-cbc-md5

allow\_weak\_crypto = true

Секция [libdefaults] содержит основные настройки для Kerberos. Рассмотрим значение параметров этой секции:

* default\_realm – адрес KDC. Контроллера домена;
* dns\_lookup\_realm – данный параметр позволяет включить или отключить поиск kerberos-имени домена через DNS;
* dns\_lookup\_realm – поиск kerberos-настроек домена через DNS;
* default\_tgs\_enctypes – поддерживаемые алгоритмы шифрования для TGS;
* defaukt\_tkt\_enctypes – поддерживаемые алгоритмы шифрования для TGT;
* permitted\_enctypes – определяет все поддерживаемые алгоритмы шифрования для ключей сессий;
* allow\_weak\_crypto – определяет, можно ли использовать слабые алгоритмы шифрования.

При правильной настройке Kerberos, пользователь должен иметь возможность получить TGT, выполнив команду «kinit [(имя\_пользователя)@(имя\_домена](http://username@DOMAIN.COM))». Если после её выполнения ничего не было выведено, настройка выполнена правильно.



Рисунок 1 – Получение TGT с помощью команды «kinit»

Выполнив команду klist можно вывести список всех TGT, и убедиться в том, что билет был действительно получен.

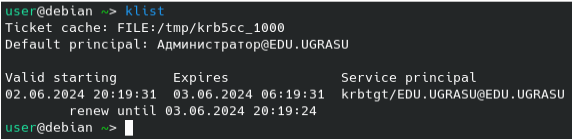


Рисунок 2 – Вывод команды «klist» после получения TGT

#### Настройка Samba и Winbind

Самым важным этапом в настройке подключения к домену является конфигурация Samba, так именно она отвечает за присоединение к домену и взаимодействие с ним. Также необходимо настроить проецирование групп и пользователей домена в систему, чтобы можно было совершать операции, завязанные на них. Например, назначения пользователей домена владельцами директорий и файлов. За это отвечает Winbind. Оба этих компонента настраиваются в одном файле – /etc/samba/smb.conf. Добавим его в конфигурацию lb build в config/includes.chroot/etc/samba/smb.conf. Рассмотрим его настройки:

[global]

workgroup = UGRASU

security = ADS

realm = UGRASU.RU

kerberos method = system keytab

winbind use default domain = Yes

winbind expand groups = 1

winbind enum users = yes

winbind enum groups = yes

winbind cache time = 36

idmap config \* : range = 20000-40000

template homedir = /home/%D/%U

template shell = /usr/bin/false

obey pam restrictions = yes

В секции [global] указываются параметры, применяемые к клиенту в целом и являющиеся значениями по умолчанию для всех остальных разделов, где явно не определенны соответствующие элементы.

Значения параметров:

* workgroup – указывает, в какой группе будут находится пользователи домена, авторизовавшиеся в системе. Этот параметр долежн совпадать с рабочей группой контроллера домена;
* security – указывает на то, как клиенты взаимодействуют с Samba. Если указано значение ADS, то Samba будет взаимодействовать как член домена. Для работы в этом режиме на клиенте должен быть установлен и настроен Kerberos;
* realm – указывает на KDC сервер, который нужно использовать при аутентификации пользователей;
* kerberos method – указывает на то, как проверяются TGT. При значении параметра «system keytab» используется локальная таблица ключей;
* winbind use default domain – указывает, нужно ли winbindd обрабатывать пользователей, не имеющих названия домена в их имени;
* winbind expand groups – управляет максимальной глубиной, которую будет проходить winbindd при выравнивании членства во вложенных группах пользователей домена, так как Linux не имеет вложенных групп. При значении параметра «0» группы пользователей ну будут запрашиваться вовсе;
* winbind enum users - определяет, будет ли winbind перечислять пользователей домена;
* winbind enum groups - определяет, будет ли winbind перечислять группы домена;
* winbind cache time – определяет время кеширования информации о пользователях и группах перед повторным запросом;
* idmap config \* – отвечает за сопоставление между SID’ами пользователей домена и ID’ами Linux системы. При текущем значении параметра пользователям системы будут присваиваться ID из диапазона 20000-40000;
* template homedir – расположения и шаблон создания домашних директория доменных пользователей. Параметр «%D» подставляет имя домена и используется по причине того, что системе предстоит взаимодействовать с двумя доменами. Параметр «%U» подставляет имя пользователя домена;
* template shell – оболочка входа в систему, которая будет присваиваться пользователям домена. Поскольку им нет необходимости входить в систему через терминал или подключаться по SSH, оболочка для них отключена;
* obey pam restrictions – указывает, нужно ли подчиняться настройкам PAM по управлению сессиями и учётными записями.

Для того, чтобы система могла полноценно взаимодействовать с пользователями и группами, необходимо сделать Winbind дополнительным источником информации о них. Сделать это можно, настроев файл диспетчера службы имён /etc/nsswitch.conf. Он представляет из себя набор колонок, в первой из которых указывается база данных, а в остальных – порядок запроса к источникам. Включим его в config/includes.chroot/etc/nsswitch.conf и укажем следующие настройки:

passwd: files systemd winbind

group: files systemd winbind

shadow: files systemd

gshadow: files systemd

hosts: files mdns4\_minimal [NOTFOUND=return] dns mymachines

networks: files

protocols: db files

services: db files

ethers: db files

rpc: db files

netgroup: nis

Все строчки, кроме первых двух, являются стандартными для Debian. В первых же двух был дописан Winbind, как дополнительный источник информации о пользователях и группах.

Для проверки корректности работы NSS можно выполнить следующие команды:

getent passwd

getent group

Первая из них должна отобразить содержимое файла /etc/passwd с информацией о пользователях системы и домена, вторая – содержимое файла /etc/group.

#### Настройка PAM

После включения настроек, описанных выше, группы и пользователи домена вместе со своими домашними директориями появятся в системе. Однако доменные пользователи всё ещё не могут авторизоваться, так как необходимо добавить в PAM модуль Samba, который будет при попытке авторизации подключаться к домену и проверять введённые данные. Сделать это можно вручную, отредактировав соответствующие файлы в /etc/pam.d, после чего добавив их в конфигурацию. Но есть более простой способ, который был описан в главе 3.3, а именно использование утилиты pam-auth-update. Так как это происходит на этапе сборки системы, вносить никакие изменения не требуется. Содержание файлов PAM приведено ниже на изображениях:

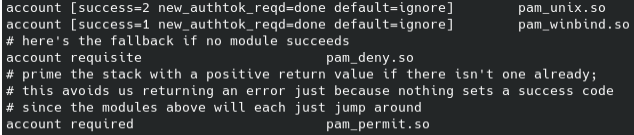


Рисунок 3 – содержание файла common-account

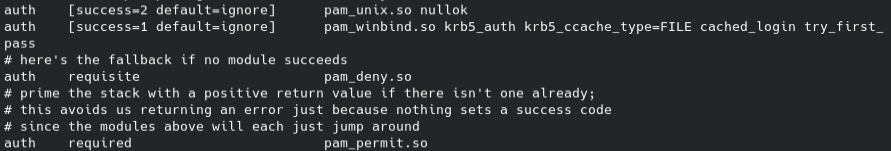


Рисунок 4 – содержание файла common-auth

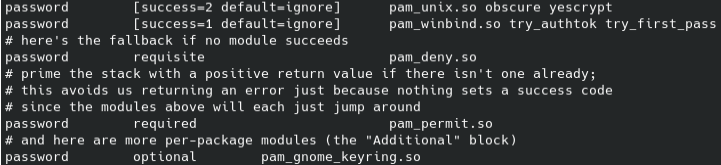


Рисунок 5 – содержание файла common-password

#### Подключение к домену

С версии Samba 4.15.0 появилась мощная утилита samba-tool, позволяющая осуществлять полное администрирование Samba. В том числе с помощью неё можно подключиться к домену. Достаточно вызвать samba-tool join domain и указать адрес домена, тип членства и тип пользователя в параметрах. Однако утилита не позволяет подключиться к домену, отличному от указанного в /etc/samba/smb.conf, при указании его в параметрах. Во время подключения к указанному домену будут запрошены его адрес и рабочая группа. Они не будут совпадать с прописанным в конфигурации Samba, что приведет к ошибке, поскольку samba-tool не может переопределить эти параметры.

Также нужно включить необходимо включить systemd сервисы smbd, nmbd и winbind для возможности подключения и работоспособности взаимодействия с доменом после перезагрузки системе.

Для решения этой проблемы был написан bash-скрипт «joindomain», который включает все необходимые сервисы в процессе подключения системы к домену, выполняет сам вход в домен и позволяет переопределить конфигурацию Samba при подключению к другому домену. Рассмотрим подробнее принцип его работы.

Для реализации требуемого поведения необходимо, чтобы скрипт мог принимать информацию о пользователе домена, от имени которого осуществляется присоединение, и информацию о самом домене (KDC и адрес) в качестве аргументов. Сделано это с помощью функции «getopts», которая позволяет оперировать позиционными параметрами.

getpots обрабатывает аргументы в цикле с оператором «case», и, в случае совпадения, присваивает их значения в соответствующие переменные. Рассмотрим, как это реализовано.

unset -v user

unset -v password

unset -v realm

unset -v workgroup

OPTSTRING=":hu:p:c:r:w:"

while getopts ${OPTSTRING} opt; do

case ${opt} in

h ) help

exit 0

;;

u ) user=$OPTARG;;

p ) password=$OPTARG;;

r ) realm=$OPTARG;;

w ) workgroup=$OPTARG;;

:)

echo "Option -${OPTARG} requires an argument."

exit 1

;;

\?)

echo "Invalid option: -${OPTARG}."

exit 1

;;

esac

done

if [ $OPTIND -eq 1 ]; then

help

exit 0

fi

Переменная «OPTSTRING» содержит список допустимых аргументов, с которыми getopts будет работать. Двоеточие после буквы аргумента означает, что он обязательно должен принимать какое-то входное значение. Переменная «OPTARG» получает значение параметра аргумента от getopts. При передаче аргумента «h» или вызове скрипта без аргументов будет выведена справка по использованию, а сам скрипт завершится с кодом «0». Для проверки последнего варианта необходимо убедиться, равняется ли переменная «OPTIND», хранящая индекс следующего аргумента, который должен быть обработан, единице. Случай «:)» вызывается, если если аргументам, требующих параметр, не было ничего передано. В этом случае скрипт завершится с кодом «0». Случай «\?» вызовется, если среди переданный аргумент отсутствует в OPTSTRING. В этом случае скрипт также завершится с кодом «1».

После того, как все переданные параметры были обработаны, необходимо настроить файл /etc/hosts для корректной работы DNS. Функция, осуществляющая это, была рассмотрена в главе 3.4.1.

Сразу после настройки DNS можно включать нужные сервисы и присоединяться к домену. За включение сервисов отвечают функции «enable\_smbd\_nmbd()» и «enable\_winbind()»:

function enable\_smbd\_nmbd(){

sudo systemctl enable smbd.service --now 2>&1 >/dev/null

sudo systemctl enable nmbd.service --now 2>&1 >/dev/null

}

function enable\_winbind(){

sudo systemctl enable winbind.service --now 2>&1 >/dev/null

}

Включение winbind сервиса вынесено в отдельную функцию, которая вызывается в самом конце, так как для его работы сначала нужно подключиться к домену.

Стоит уточнить, что запускать этот скрипт имеет смысл только администратору или пользователю с правами администратора домена, поскольку рядовой пользователь не может ввести машину в домен. В противном случае joindomain выдаст ошибку, изображённую ниже.

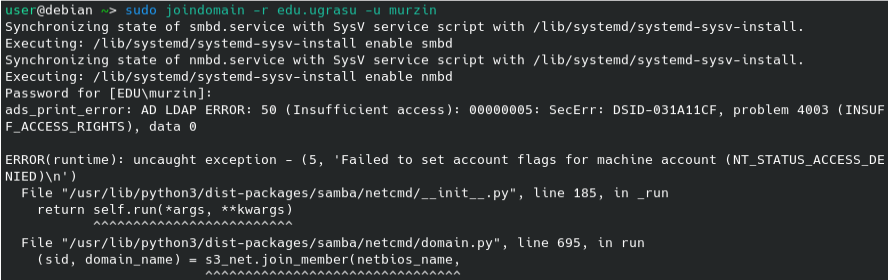


Рисунок 6 – ошибка подключения к домену по причине отсутствия прав администратора

??? (картинка ошибка подключения от рядового пользователя)

При успешном подключении к домену будет выведено сообщение «Joined to domain...», как на рисунке ниже.

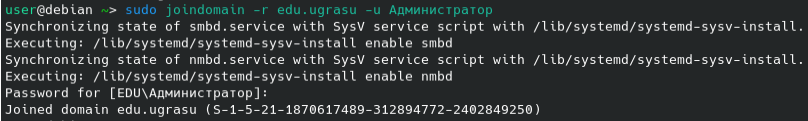


Рисунок 7 – Успешное подключение к домену

В панели администрирования Active Direcotry диспетчера серверов контроллера домена машина должна появиться в списке компьютеров.

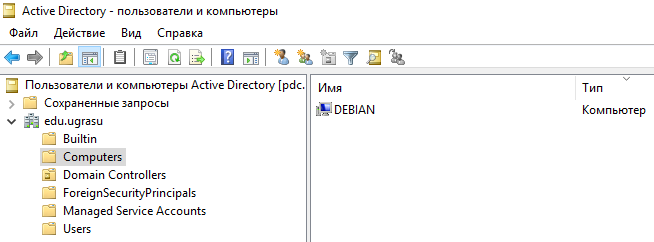


Рисунок 8 – Список компьютеров домена после подключения

### Настройка системы после установки

К сожалению, не все приложения и настройки можно включить в систему на этапе создания конфигурации или сборки. Например, не представляется возможным установить ряд Flatpak приложений с помощью «хуков» live build. Для решения подобных проблем были написаны скрипты, которые начинают выполняться в фоновом режиме при первом запуске системы, и делают это до тех пор, пока не будут успешно выполнены.

Реализовать выполнение задач в фоновом режиме можно с помощью сервисов systemd, которые были добавлены в конфигурацию в config/includes.chroot/etc/systemd/system. Вид сервисов для запуска скриптов настройки:

[Unit]

Description=Install flatpak apps

After=multi-user.target

[Service]

User=root

Group=root

ExecStart=/usr/bin/bash (путь\_к\_скрипту)

Type=simple

[Install]

WantedBy=multi-user.target

Сервис состоит из трёх секций. Секция «[Unit]» содержит основную информацию о нём. Директива «After» устанавливает зависимости от других сервисов. Секция «[Service]» содержит информацию о исполнении и прекращении работы сервиса. «User» и «Group» определяют, от имени какого пользователи и группы должен запускаться скрипт, определённый в «ExecStart».

Секция «[Install]» определяет, куда будет устанавливаться сервис. В данном случае в директорию /etc/systemd/system/multi-user.target.wants.

Все сервисы и соответствующие им скрипты нужны только для разовой задачи, поэтому после успешного выполнения их можно удалить. Делается это автоматически. Если же какой-то из скриптов завершился с кодом «1», этого не произойдёт, и он снова запустится при следующем старте системы. Таким образом, как только все необходимые условия будут выполнены (например, наличие соединение с интернетом), все они будут завершены с кодом «0» и удалены.

Debian имеет одну особенность, касающиеся systemd – он автоматически не запускает сервисы, размещенные в директориях загрузки. Их нужно запускать вручную. Написание отдельного сервиса, запускающего все остальные, не избавит от проблемы, так его тоже нужно будет запустить.

В качестве решения было принято использование скриптов, запускающихся при старте графической подсистемы Xorg. Они находятся в /etc/X11/Xsession.d/ и имеют похожий шаблон именования, что и «хуки» lb build. Перед названием скрипта ставится двузначное число, указывающее на очередность его запуска. Чем меньше его значение, тем раньше он исполняется. Последним запускается скрипт «99x11-common\_start», стартующий Xorg. Таким образом, общее их количество, включая те, что находятся здесь по умолчанию, не может превышать девяносто девяти. Можно было бы отказаться от systemd сервисов вовсе, и осуществить всю дополнительную настройку системы через них, но имеется одна проблема – Xorg не запустится, пока не будут выполнены все предыдущие скрипты. Это приводит к появлению черного экрана после авторизации пользователя через экран входа в систему, который остается висеть до тех пор, пока не запустится графическая подсистема. Поэтому было принято решение использовать данный функционал только для автоматического запуска systemd сервисов и включения темы оформления.

Здесь имеется ещё одна проблема. Дело в том, что KDE, выбранное в качестве графического окружения, поставляется с двумя графическими подсистемами – Xorg и Wayland. Выбрать, какое именно будет использоваться в конкретном сеансе пользователя, можно в менеджере сессий SDDM на экране входа в систему. Он запомнит выбор и будет использовать его при последующих входах в систему. Однако сразу после установки системы SDDM устанавливает Wayland как значение по умолчанию, и, соответственно, Xorg нужно выбирать вручную. Во избежании этого, был написан дополнительный файл конфигурация для SDDM, который при первой загрузке системы выполняет вход в графическое окружение с Xorg администратором без запроса пароля. Он был добавлен в конфигурацию в config/includes.chroot/etc/sddm.conf.d/kde\_settings.conf. Рассмотрим строчки, отвечающие за автоматический вход:

[Autologin]

Relogin=False

Session=plasma

User=adminuser

В секции «[Autologin]» указываются директивы, отвечающие за автоматический вход. Параметр «Relogin» отвечает за то, будет ли работать автоматический вход после выхода из сеанса. В «Session» указывается нужное нам графическая подсистема. В качестве пользователя указан администратор.

После входа в графическое окружение дополнительная конфигурация SDDM удаляется в скрипте «91start-system-adjustment-services», за счёт чего при последующих запусках системы экран входа будет отображаться, и пользователи смогут вводить свои данные.

#### Запуск systemd сервисов и включение темы оформления

Добавленные в конфигурацию systemd сервисы запускаются из скрипта «91start-system-adjustment-services» при старте Xorg. В отличие от самих сервисов, скрипт запускается не от имени суперпользователя (root). Поэтому для использования команд, требующих прав суперпользователя, необходимо использовать команду «sudo», куда в качестве параметра передаётся пароль администратора. Рассмотрим код скрипта включения сервисов:

echo password | sudo -S systemctl start configure-sources-list.service

echo password | sudo -S systemctl enable configure-sources-list.service

echo password | sudo -S systemctl start install-flatpak-apps.service

echo password | sudo -S systemctl enable install-flatpak-apps.service

echo password | sudo -S systemctl start install-virtualbox.service

echo password | sudo -S systemctl enable install-virtualbox.service

echo password | sudo -S systemctl start install-drivers.service

echo password | sudo -S systemctl enable install-drivers.service

echo password | sudo -S rm -f /etc/X11/Xsession/90set-theme

echo password | sudo -S rm -f /etc/X11/Xsession/91start-system-adjustment-services

Утилита «systemctl» осуществляет взаимодействие и управление в systemd и менеджером сервисов. Параметр «start» включает сервис прямо сейчас, а «enable» говорит о том, что ему нужно запускаться при каждом старте системы. После выполнения всех нужным операция можно удалить созданные нами Xorg скрипты.

Включение темы оформления посредством Xorg скриптов позволяет сделать это незаметным для пользователя. Сделать это можно в одну команду, с помощью утилиты «lookandfeeltool»:

lookandfeeltool -a com.github.yeyushengfan258.We10XOS-light

echo password | sudo -S rm -f /etc/sddm.conf.d/kde\_settings.conf

Аргумент «a» принимает название темы оформления. Получить его можно, предварительно выполнив команду «lookandfeeltool --list» и найдя нужную тему в выведенном списке.

После включения темы оформления нужно удалить дополнительную конфигурацию для SDDM.

#### Установка Flatpak приложений

Проблема с Flatpak заключается в том, что он не может работать в chroot окружении, которое создаётся на этапе сборки. Соответственно, нужно написать скрипт, который будет устанавливать каждое отдельное приложение, после чего проверять факт их наличия в системе и удаляться.

Сначала обозначим список приложений для установки:

apps=(

com.usebottles.bottles

com.bitwarden.desktop

com.getpostman.Postman

)

Далее в цикле установим каждое приложение из списка, предварительно проверив подключение к интернету и разрешение доменных имён:

if ping -q -c 1 -W 1 google.com >/dev/null; then

flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo

for app in "${apps[@]}"

do

flatpak install -y flathub "$app"

done

else

exit 1

fi

Если подключение к интернету отсутствует, скрипт завершится с кодом «1».

После проверим наличие приложение в системе, и, при отрицательном результате, также выйдем с кодом «1»:

for app in "${apps[@]}"

do

if [ $(flatpak list --app --columns=application | tail -n +1 | grep -w -c "$app") -eq 0 ]; then

exit 1

fi

done

Если проверка пройдена, можем спокойно удалять скрипт и сервисы.

Установка первых Flatpak приложений занимает немало времени, так как происходит установка runtime, где содержатся базовые библиотеки Flatpak приложение. По этой причине после первого запуска системы стоит немного подождать, пока они появятся в системе.

#### Установка драйверов GPU

Базовые драйвера GPU устанавливаются в процессе сборки системы и позволяют спокойно её функционалом. Однако для лучшей работы игр и других приложений, утилизирующих использования графического процессора, стоит произвести дополнительную настройку. За это отвечает скрипт install-drivers. Разберём принцип его работы:

if ping -q -c 1 -W 1 google.com >/dev/null; then

sudo dpkg --add-architecture i386 && sudo apt update

if lscpu | grep Architecture | grep x86\_64; then

apt install -y linux-headers-amd64

else

apt install -y linux-headers-686

fi

if lspci -nn | grep '\[03[02]' | grep 'NVIDIA'; then

apt install -y nvidia-driver firmware-misc-nonfree

fi

if lspci -nn | grep '\[03[02]' | grep 'AMD'; then

apt install -y libglx-mesa0:i386 mesa-vulkan-drivers:i386 libgl1-mesa-dri:i386

fi

if lspci -nn | grep '\[03[02]' | grep 'Intel'; then

apt remove xserver-xorg-video-intel

fi

else

exit 1

fi

Сначала проверяется подключение к интернету и разрешение доменных имён, при отрицательном результате скрипт завершается с ошибкой. Далее устанавливаются 32-битные библиотеки для возможности запуска ненативных игр. После с помощью команды «lspcu» и «grep» получаем производителя GPU и в зависимости от результата устанавливаем или удаляем необходимые пакеты.

#### Установка VirtualBox

VirtualBox также как и Flatpak приложения невозможно поставить во время сборки, поэтому также установим его через скрипт. Установка происходит через apt, так что необходимо добавить репозитория и соответствующий ключ в config/archives. Рассмотрим содержание скрипта:

if ping -q -c 1 -W 1 google.com >/dev/null; then

apt install linux-headers-$(uname -r) dkms -y

apt install virtualbox-7.0 -y

else

exit 1

fi

if ! dpkg -s virtualbox-7.0; then

exit 1

fi

Так же, как и раньше, проверяем доступность интернет соединения. После этого с помощью apt устанавливаем необходимые зависимости, а именно заголовки и поддержку динамических модулей ядра. Команда «uname –r» вернет и подставит значение текущей версии ядра. Далее можем установить сам VirtualBox. С помощью dpkg проверяем, если ли установленный пакет в системе. Если нет, выходим с кодом «1».

#### Настройка списка репозиториев

Список репозиториев представляет собой текстовый файл, располагающийся в /etc/apt/sources.list. Его структура была подробно рассмотрена в главе 3.1.3. Можно было бы просто добавить файл с нужными репозиториями в config/includes.chroot/etc/apt/sources.list, однако после его попадания в chroot окружение содержимое переопределяется в процессе. Для устанавливаемой системы в этот файл попадают репозитории, используемые для загрузки дополнительных компонентов в процессе установки. Но установка нашей системы происходит исключительно в оффлайн режиме, чтобы это можно было сделать без возможности подключения к интернету и во избежании потенциальных ошибок. В результате чего список репозиториев оказывается пустым, и его нужно заполнять вручную. Чтобы это исправить, был написан bash-скрипт и запускающий его сервис, которые заполняют содержимое /etc/apt/sources.list сразу после установки. Содержание скрипта:

echo "deb http://ftp.ru.debian.org/debian/ bookworm main contrib non-free non-free-firmware" > /etc/apt/sources.list

echo "deb-src http://ftp.ru.debian.org/debian/ bookworm main contrib non-free non-free-firmware" >> /etc/apt/sources.list

echo "deb http://security.debian.org/debian-security bookworm-security main contrib non-free non-free-firmware" >> /etc/apt/sources.list

echo "deb-src http://security.debian.org/debian-security bookworm-security main contrib non-free non-free-firmware" >> /etc/apt/sources.list

echo "deb http://ftp.ru.debian.org/debian/ bookworm-updates main contrib non-free-firmware" >> /etc/apt/sources.list

echo "deb-src http://ftp.ru.debian.org/debian/ bookworm-updates main contrib non-free-firmware" >> /etc/apt/sources.list

apt update

Команда «echo» просто отображает полученную на вход строку, а оператор «>» перенаправляет результат предыдущей команды в файл. С помощью «apt update» обновляем списки пакетов из указанных репозиториев.

### Автоматизация процесса установки системы

Процесс установки можно автоматизировать, с помощью файла preseed.cfg, находящийся в директории config/preseed конфигурации. Установка состоит из множества этапов. Preseed предоставляет способ установки ответов на вопросы всех этапах в процессе установки.

Русская локализация устанавливается с помощью строки:

d-i debian-installer/locale string ru\_RU

Для настройки клавиатуры используется:

d-i keyboard-configuration/xkb-keymap select ru

d-i keyboard-configuration/toggle select Alt+Shift

Первая строчка добавляет русскую раскладку клавиатуры. Вторая устанавливает комбинацию переключения раскладок.

Далее следует автоматизировать создание суперпользователя и администратора.

d-i passwd/root-login boolean true

d-i passwd/root-password password password

d-i passwd/root-password-again password password

d-i passwd/user-fullname string Admin user

d-i passwd/username string adminuser

d-i passwd/user-password password password

d-i passwd/user-password-again password password

Значение параметра «root-login» true позволяет пропустить этап создания суперпользователя и назначить ему пароль автоматически. Для создания администратора помимо пароля также нужно прописать полное имя и имя пользователя.

Для упрощения работы с docker и VirtualBox, администратора нужно добавить в группы «docker» и «vboxusers» соответственно.

d-i passwd/user-default-groups string audio cdrom video sudo docker vboxusers

Допишем эти группы к тем, что используются по умолчанию.

Часовой пояс указывается с помощью параметра «time/zone»:

d-i time/zone string Asia/Yekaterinburg

Далее нужно произвести настройку apt. Отключим вхождение cdrom и зеркал в /etc/sources.list:

d-i apt-setup/disable-cdrom-entries boolean true

d-i apt-setup/use\_mirror boolean false

После нужно настроить параметры загрузчика:

d-i grub-installer/only\_debian boolean true

d-i grub-installer/with\_other\_os boolean true

d-i grub-installer/bootdev string /dev/sda

d-i grub-installer/bootdev string default

Первые две строчки позволяют установить загрузчик GRUB на UEFI раздел, даже если на диске присутствует другая ОС. Последние две строчки указывают на то, что установку нужно произвести на основной диск (не USB).

Для того, чтобы сразу перезагрузиться в установленную систему добавим строку:

d-i finish-install/reboot\_in\_progress note

В результате применения настроек из preseed.cfg будут автоматизированы все этапы установщика, кроме этапов указания имени хоста и разметки диска. Имя хоста стоит прописать вручную, чтобы не было путаницы в структуре домена. Разметку диска стоит осуществлять вручную, так как на компьютере может присутствовать более одного диска. В этом случае администратор сам выбирает, куда установить систему. Также, в зависимости от размера диска, администратор может выбрать ту или иную структуру разделов.

# Заключение

# Список использованных источников

# Приложения

sh-скрипты build, clean и config, вызывающиеся при запуске утилит lb build, lb clean и lb config соответственно. Они содержат список параметров, которые будут применены при вызове каждой конкретной утилиты, что позволяет нам не прописывать их при всяком вызове.

В файле build мы перенаправляем вывод lb build в файл build.log для дальнейшего изучения логов сборки и отслеживания возможных ошибок.

lb build noauto "${@}" 2>&1 | tee build.log

В файле clean помимо параметров для утилиты lb clean содержатся команды для удаления оставшихся продуктов сборки. Это позволяет запускать последующие сборки «с нуля».

Наибольший интерес представляет файл config. В нём прописаны все параметры, применяемые во время сборки системы.

Для реализации поставленной задачи в первую очередь понадобится утилита «live-build». Скачать её можно из официальных репозиториев дистрибутива «Debian» с помощью команды «sudo apt install live-build» на компьютере с уже установленной системой «Debian» или его производными, однако для корректной работы лучше использовать ту же систему, на основе которой будет осуществляться сборка. Для локального (?) тестирования потребуется программа для виртуализации, с помощью которой можно будет запустить виртуальную машину с собранной системой.

«live-build» позволяет собрать систему, которую можно запустить в «live» режиме (не устанавливая на диск) и далее установить её с того же носителя.