Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ЮГОРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Высшая инженерная школа цифровых технологий

09.03.04, Программная инженерия

(код и наименование направления подготовки, специальности)

Программная инженерия

(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 Щербаков Дмитрий Ильич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия имя отчество) (личная подпись)

Научный Карпов Дмитрий Викторович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

руководитель (Фамилия имя отчество) (личная подпись)

Нормоконтролер Самарина Ольга Владимировна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия имя отчество) (личная подпись)

Допустить к защите

Руководитель Пронькина Татьяна Васильевна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

образовательной (Фамилия имя отчество) (личная подпись)

программы

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

г. Ханты-Мансийск

2024 годРеферат

на выпускную квалификационную бакалаврскую работу

Щербакова Д.И.

Бакалаврская работа содержит 75 с., 12 рисунков, 1 таблицу, 5 приложений, 22 источника.

СБОРКА, ДИСТРИБУТИВ, ДОМЕН, LINUX, ОПЕРАЦИОННАЯ СИСТЕМА, КОНФИГУРАЦИЯ, СКРИПТ, WINDOWS

Целью бакалаврской работы является разработка сборка дистрибутива операционной системы с прикладным программным обеспечением на базе ядра Linux.

Объектом исследования является программное обеспечение Югорского государственного университета. Предметом исследования является исследования процесса сборки 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 системы.

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

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

Оглавление

[ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 5](#_Toc1)

[ВВЕДЕНИЕ 8](#_Toc2)

[1 ОБЗОР ОПЕРАЦИОННЫХ СИСТЕМ НА БАЗЕ ЯДРА LINUX И СПОСОБОВ ИХ СБОРКИ 11](#_Toc3)

[1.1 Описание собираемой системы и её функционала 11](#_Toc4)

[1.2 Обзор аналогов 12](#_Toc5)

[1.3 Обзор способов сборки ОС на базе ядра Linux 15](#_Toc6)

[1.4 Описание live-build и принцип работы его утилит 18](#_Toc7)

[2 ОПИСАНИЕ СТРУКТУРЫ КОНФИГУРАЦИИ ДЛЯ СБОРКИ 20](#_Toc8)

[2.1 Структура конфигурации для lb build 20](#_Toc9)

[2.1.1 Директория package-lists 20](#_Toc10)

[2.1.2 Директория archives 22](#_Toc11)

[2.1.3 Директория includes.chroot 23](#_Toc12)

[2.1.4 Директория hooks 25](#_Toc13)

[2.1.5 Директория preseed 25](#_Toc14)

[2.2 Параметры сборки системы 25](#_Toc15)

[2.3 Обзор основных компонентов системы 27](#_Toc16)

[2.3.1 Samba 27](#_Toc17)

[2.3.2 Kerberos 28](#_Toc18)

[2.3.3 PAM 29](#_Toc19)

[2.3.4 Winbind 30](#_Toc20)

[2.4 Обзор прикладного программного обеспечения 30](#_Toc21)

[2.5 Способы запуска Windows приложений 32](#_Toc22)

[3 СОСТАВЛЕНИЕ КОНФИГУРАЦИИ ДЛЯ СБОРКИ СИСТЕМЫ И СОЗДАНИЕ ИНФРАСТРУКТУРЫ 34](#_Toc23)

[3.1 Настройка системы в процессе сборки 34](#_Toc24)

[3.2 Настройка подключения к домену и доменной авторизации 35](#_Toc25)

[3.2.1 Настройка DNS 35](#_Toc26)

[3.2.2 Настройка синхронизации времени 38](#_Toc27)

[3.2.3 Настройка авторизации через Kerberos 39](#_Toc28)

[3.2.4 Настройка Samba и Winbind 41](#_Toc29)

[3.2.5 Настройка PAM 43](#_Toc30)

[3.2.6 Подключение к домену 44](#_Toc31)

[3.3 Настройка системы после установки 48](#_Toc32)

[3.3.1 Запуск systemd сервисов и включение темы оформления 51](#_Toc33)

[3.3.2 Установка Flatpak приложений и VirtualBox 52](#_Toc34)

[3.3.3 Установка драйверов GPU 54](#_Toc35)

[3.3.4 Настройка списка репозиториев 55](#_Toc36)

[3.4 Автоматизация процесса установки системы 56](#_Toc37)

[3.5 Создание Debian репозитория ЮГУ 58](#_Toc38)

[Заключение 62](#_Toc39)

[Список использованных источников 63](#_Toc40)

# ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ЮГУ – Югорский государственный университет

ОС – операционная система

ПО – программное обеспечение

ВКР – выпускная квалификационная работа

Graphics processing unit (GPU) – графический процессор

Central processing unit (CPU) – центральный процессор

Coordinated Universal Time (UTC) – всемирное координированное время

Real Time Clock (RTC) – электронная схема, предназначенная для учёта хронометрических данных

Identifier (ID) – уникальный идентификатор

Security Identifier (SID) – идентификатор безопасности

Active Direcory (AD) – службы каталогов корпорации Microsoft для операционных систем семейства Windows Server

Доменная служба Active Directory (домен) – 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

Debian – свободный дистрибутив GNU/Linux

LiveCD (live) – операционная система, которая загружается со сменного носителя

Критерии Debian по определению свободного ПО (Debian Free Software Guidelines, DFSG) – набор правил, по которым проект Debian определяет, какие лицензии являются свободными

KDE Plasma – технология пользовательского интерфейса, созданная сообществом KDE

Pluggable Authentication Modules (PAM) – набор разделяемых библиотек, которые позволяют интегрировать различные низкоуровневые методы аутентификации в Unix-подобных системах

Flatpak – утилита для развёртывания, управления пакетами и виртуализации для Linux

Flathub – репозиторий Flatpak для загрузки и распространения приложений для всех настольных версий Linux

Kerberos – сетевой протокол аутентификации, который предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними

Samba – пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB/CIFS

Domain Name System (DNS) – компьютерная распределённая система для получения информации о доменах

NetworkManager – программа для управления сетевыми соединениями в Linux

KDC – key distribution center

TGS – ticket-granting ticket

Демон – компьютерная программа в UNIX-подобных системах, запускаемая самой системой и работающая в фоновом режиме

Name Service Switch (NSS) – интерфейс, позволяющий указать различные источники конфигураций и механизмов разрешения имён

Bash – командная оболочка для UNIX-подобных операционных систем

Systemd – подсистема инициализации и управления службами в Linux

apt – программа для установки, обновления и удаления программных пакетов в операционных системах Debian

X Windows System (X) - оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя

Xorg – реализация сервера X Window System с открытым исходным кодом

Wayland – современная графическая платформа, используемая в ОС Linux и пришедшая на замену устаревшей X

GRand Unified Bootloader (GRUB) – загрузчик операционных систем от проекта GNU

Unified Extensible Firmware Interface (UEFI) – интерфейс между операционной системой и микропрограммами, управляющими низкоуровневыми функциями оборудования

Network Time Protocol (NTP) – сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей

Command line interface (CLI) – способ взаимодействия между человеком и компьютером путём отправки компьютеру команд

Secure Shell (SSH) – сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой

# 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. С 30 сентября 2023 года Microsoft 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. Сейчас скачать официальный образ диска Windows 10 или Windows 11 с сайта Microsoft без использования VPN не представляется возможным.

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.

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. 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. 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 программ 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 программ.

Целью данной работы является сборка дистрибутива операционной системы на базе ядра Linux с прикладным программным обеспечением.

Объектом исследования является переход от использования операционных систем семейства Windows и прикладных приложений от Microsoft к операционным системам на базе ядра Linux и приложением с открытым исходным кодом. Предметом исследования является изучение методов и процесса сборки операционных систем на базе ядра Linux.

Выполнение данной работы включает в себя несколько задач:

1. Изучение преимуществ и недостатков российских операционных си на базе ядра Linux;
2. Выбор оптимального инструмента для сборки системы;
3. Формирование конфигурации для сборки системы;
4. Написание скриптов настройки системы в процессе и после сборки;
5. Сборка системы;
6. Тестирование системы в виртуальном окружении;
7. Тестирование системы в инфраструктуре организации;
8. Установка системы на компьютеры организации.

Организацией, для которой собирается конечная операционная система, является Югорский государственный университет. Создание своего собственного Linux дистрибутива позволит ЮГУ использовать на своих компьютерах операционные системы из двух разных семейств, если на то будет необходимость, а, в случае полного ухода компании Microsoft с российского рынка, полностью отказаться от использования операционных систем семейства Windows.

## ОБЗОР ОПЕРАЦИОННЫХ СИСТЕМ НА БАЗЕ ЯДРА LINUX И СПОСОБОВ ИХ СБОРКИ

### Описание собираемой системы и её функционала

Собираемая система будет функционировать на базе ядра Linux и иметь набор прикладного программного обеспечения. Такую систему можно считать дистрибутивом Linux. 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.

За основу собираемой системы была выбрана стабильная версия дистрибутива Debian. Причиной такого выбора является его проверенная сообществом стабильность и надёжность. Дистрибутив, построенный и выпускающий релизы на основе другого дистрибутива, называется деривативом. 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.

Для того, чтобы сотрудники и студенты могли успешно пользоваться системой, она должна удовлетворять ряду требований:

1. Возможность легко ввести компьютер в домен;
2. Авторизация пользователей в систему по доменным учётным записям;
3. Возможность монтирования сетевых директорий доменных пользователей;
4. Работа с принтерами и сканерами;
5. Возможность запуска приложений, написанных под Windows;
6. Отсутствие системных пользователей (за исключением Администратора);
7. Тема оформления, схожая с Windows 10;
8. Наличие прикладного ПО для выполнение стандартных задач (работа с документами, таблицами, графикой и т. д.)

Всем этим требованием система должна удовлетворять сразу после установки, без необходимости дополнительной настройки.

### Обзор аналогов

ALT Linux – отечественное семейство дистрибутивов, являющееся отдельной ветвью дистрибутивов Linux [6-7]. Основан на дистрибутиве 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 – операционная система на базе ядра Linux, внедряемая в России в качестве альтернативы Microsoft Windows в государственных организациях [8-9]. 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 — это российская операционная система на базе ядра Linux, которая разработана компанией «РЕД СОФТ» с использованием открытых исходных кодов и собственных решений [10-11]. 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, Cinnamon и Gnome.

Rosa Linux — линейка дистрибутивов Linux (изначально основанных на Mandriva), разработку которых ведёт российская компания «НТЦ ИТ РОСА» [12-13]. 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 Plasma.

Все редакции системы, кроме ФРЕШ, являются платными.

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

Таблица 1 - Сравнительная таблица аналогов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Astra Linux | ALT Linux | РЕД ОС | РОСА ОС | Ugrach |
| Стоимость лицензии, руб./год | Для физических лиц | Не поставляется физическим лицам | Бесплатно | Бесплатно | Бесплатно | Не поставляется физическим лицам |
| Для юридических лиц | 8950 | 5280 | 4600 | 4400 | Распространяется под лицензией GNU GPL |
| Стоимость поддержки | | 4200 | 2160 | 2500 | 3300 | Силами университета |
| Возможность доменной авторизации «из коробки» | | Нет | Да | Да | Да | Да |
| Программное обеспечение | | Офис, почта, браузер и прочие прикладные программы | Офис, почта, браузер и прочие прикладные программы | Офис, почта, браузер и прочие прикладные программы | Офис, почта, браузер и прочие прикладные программы | Офис, почта, браузер и прочие прикладные программы, а также программы из собственного Debian репозитория ЮГУ |
| 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 прикладных приложений, и имеют сравнительно невысокую стоимость поддержки. Однако возможность включить в Ugrach набор приложения и настроек, специфичных для ЮГУ, и отсутствие необходимости приобретать лицензию являются сильными аргументами в пользу выбора собираемой системы.

### Обзор способов сборки ОС на базе ядра Linux

Существует множество способов собрать свою операционную систему на базе ядра Linux. Выбор способа зависит от цели сборки, дистрибутива, на котором должна основываться сборка, возможностей настройки, сложности реализации задачи и прочих факторов. Рассмотрим несколько методов, выделим их основные преимущества и недостатки с сделаем выбор в пользу одного из них.

Linux From Scratch (LFS) - проект, содержащий набор руководств по сборки минимального Linux дистрибутива «с нуля». Они позволяют «вручную» создать ОС на основе исходных кодов пакетов ядра Linux. Каждый включаемый компонент компилируется из исходных кодов с помощью минимального набора инструментов. Данный способ позволяет углубить свои знания о Linux, и понять, как система работает изнутри. Имеется огромный потенциал для кастомизации, так как в сборку попадает ровно то, что в неё включит администратор. Минусы данного способа очевидны – это сложность и время сборки. Также LFS содержит руководства по созданию только самой базовой системы, что нам не подходит.

Cubic (Custom Ubuntu ISO Creator) – графическое приложение, которое позволяет создать настраиваемый Live ISO образ на основе официального образа Ubuntu. Оно позволяет легко перемещаться по этапам настройки ISO и имеет встроенную виртуальную среду командной строки для настройки файловой системы Linux, внутри которой можно проводить любые манипуляции, которые в результате попадут в итоговый ISO образ. В графическом интерфейсе можно выбрать или исключить основные компоненты будущей системы путем простого нажатия на чекбоксы. Этот вариант походит почти по всем критериям. Во-первых – сборка строится на основе официального образа Ubuntu, то есть имеется уже минимально функционирующая система. Во-вторых – наличие широких возможностей для настройки. В-третьих – Ubuntu является одним из самых популярных дистрибутивов, поэтому пользователям будет проще работать с ней. Однако есть серьёзный недостаток – Ubuntu разрабатывается коммерческой организацией Canonical, и, как следствие, имеется зависимость от принятых ей решений. Также стоит отметить излишнюю простоту данного метода, что может оказаться неприемлемым в рамках выполнения ВКР.

Systemback – приложение для создания и восстановления снимков системы. Оно позволяет делать резервные копии и точки восстановления. Systemback имеет функцию создания Live ISO образа на основе установленной системы. Можно выполнить «чистую» установку любого дистрибутива, произвести необходимые настройки, установить приложения и создать на основе этого ISO образ. Плюсами данного способа является скорость и простота – для получения желаемого результата нужно всего лишь установить программу и привести систему к желаемом состоянию. Минусом данного подхода является то, что для его реализации уже нужна установленная система. Для внесения изменений нужно вносить изменения в систему, на основе которой создаётся образ. Помимо всего вышеперечисленного, этот метод чрезмерно прост для ВКР.

Ещё один способ сборки – использование archiso. Это широко настраивая утилита для сборки live ISO образов на основе Arch Linux. Именно с помощью неё создаётся официальный образ этого дистрибутива. С помощью archiso можно как создать как установочный диск, так и диск восстановления. Основным преимуществом данного метода является маленький размер конфигурации для сборки. Однако имеются трудности с созданием и запуском скриптов настройки системы во время сборки. Помимо этого, Arch Linux является «rolling release» дистрибутивом, что означает то, что он получает обновления сразу же по мере их поступления. Это может негативно сказаться на стабильности работы системы.

Окончательный выбор пал на набор утилиту live-build, разработанную проектом Debian Live. Он позволяет на основе директории с конфигурацией создать live ISO образ на основе дистрибутива Debian. Данный способ идеально подходит для поставленной задачи. Конфигурация для live-build занимает мало дискового пространства и в неё очень легко вносить изменения. Для того, чтобы изменения вступили в силу, нужно заново собрать систему на основе обновлённой конфигурации. Дистрибутив Debian, также, как и Ubuntu, является очень популярным, что облегчит пользователям работу с будущей системой. Нельзя не отметить проверенную сообществом стабильность этого дистрибутива. Более того, разработкой Debian занимается сообщество, а не коммерческая организация, что также добавляет уверенности в использовании данного метода. Ещё одним плюсом стоит указать возможность включения установщика в систему и дальнейшую его настройку. По совокупности скорости реализации поставленной задачи, сложности освоения, надёжности и дальнейшей поддержки данный способ сборки Linux дистрибутива является наиболее подходящим.

### Описание live-build и принцип работы его утилит

Так как на этапе выбора методов сборки системы была выбрана утилита live-build, стоит подробнее рассмотреть, что она из себя представляет и как работает. live-build – это набор утилит, разработанный проектом Debian Live, позволяющий собрать свой образ live системы на основе директории с конфигурацией [1]. Установить его на Debian и основанных на нём системах можно с помощью команды «sudo apt install live-build». live-build состоит из трёх утилит, которые также называются командами:

1. lb config [2];
2. lb build [3];
3. lb clean [4].

Команда lb config инициализирует конфигурационную директорию и сообщает параметры сборки live системы. Каркас конфигурации можно составить и вручную, но эта утилита практически полностью автоматизирует данный процесс.

Команда lb build считывает конфигурацию из директории config/ и запускает процесс сборки системы.

Команда lb clean позволяет легко удалить продукты сборки, чтобы можно было начинать последующие сборки «с нуля». С помощью параметров можно указать утилите на удаление только определённых этапов сборки, речь о которых пойдёт далее.

Процесс самой сборки разделён на стадии, к каждой из которых можно применить свои настройки. Всего их три:

1. Стадия bootstrap;
2. Стадия chroot;
3. Стадия binary.

На стадии bootstrap выполняется заполнение chroot директории (директории, по отношению к которой в дальнейшем будет применена команда «chroot», позволяющая сменить корень файловой системы) пакетами для создания базовой Debian системы.

Стадия chroot завершает построение chroot директории, заполняя её пакетами и прочими файлами, включёнными в конфигурацию сборки. Большинство настроек системы применяется именно здесь.

Стадия binary завершает сборку, создавая загружаемый образ, используя содержимое chroot директории для построения корневой файловой системы live системы. На этом этапе добавляется установщик и применяемые к нему параметры.

Передачу параметров в утилиты live-build можно автоматизировать с помощью скриптов, находящихся в директории auto, которая располагается в корне конфигурации. Названия скриптов должны соответствовать названию команд (config, build, clean). Например, можно перенаправить весь вывод lb build в файл для последующего анализа и поиска ошибок, создав sh-скрипт со следующей строчкой:

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

## ОПИСАНИЕ СТРУКТУРЫ КОНФИГУРАЦИИ ДЛЯ СБОРКИ

### Структура конфигурации для lb build

Для сборки системы сначала необходимо составить конфигурацию, из которой утилита lb build 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. Иерархия конфигурации первого уровня описывается следующим изображением:

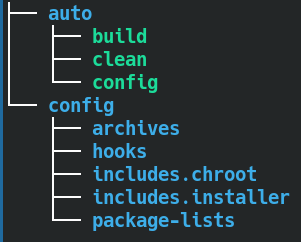


Рисунок 1 - Иерархия первого уровня конфигурации для сборки

#### 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 и не должно содержать синтаксических ошибок. В противном случае он не будет найдет и, либо не будет включен в систему, либо станет причиной преждевременного завершения сборки 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.

Листинг 1 – пример использования условных конструкций в файле устанавливаемых пакетов:

#if ARCHITECTURES i386 amd64

memtest86+

#endif

В данном примере 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:

1. development.list.chroot – программы для разработки ПО и моделирования;
2. graphics.list.chroot – программы для работы с графикой;
3. media.list.chroot – программы для работы с мультимедиа;
4. office.list.chroot – офисный пакет и прочие программы для работы с документами разных форматов;
5. virtualization.list.chroot – программы для виртуализации;
6. system.list.chroot – системные программы и утилиты;
7. 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 [21].

Названия файлов с репозиториями имеют следующий шаблон:

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

Название может быть любым, но лучше, чтобы оно отражало содержимое репозитория во избежание 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:

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

Пример файла 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 оттуда, 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. Также допустимо наличие директории includes.binary, файлы из которой попадают в образ live системы, но поскольку для нашей системы в ней нет необходимости, в конфигурацию она включена не была. Иерархия её структуры представлена на рисунке 2. Её содержимое будет подробнее рассмотрено далее.

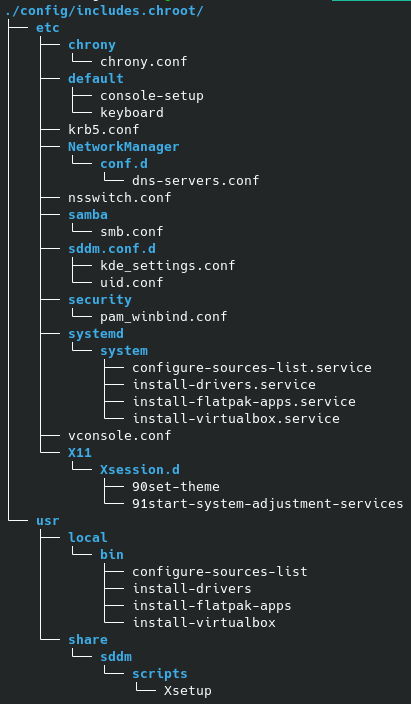


Рисунок 2 - иерархия содержимого директории config/includes.chroot

#### Директория hooks

Данная директория может состоять из поддиректорий hooks/normal и hooks/live, содержащих скрипты, исполняемые на этапе сборки системы. Они позволяют выполнить 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 четырёхзначное 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, который предопределяет параметры установщика системы. Он позволяет практически полностью автоматизировать процесс установки.

### Параметры сборки системы

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

Листинг 2 – код скрипта auto/config:

#!/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:

1. architectures amd64 – архитектура системы;
2. archive-areas "main contrib non-free non-free-firmware" – ветки репозитория, откуда будут скачиваться пакеты. Установлены все ветки, чтобы в процессе сборки были установлены проприетарные программы и драйвера;
3. 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.
4. debian-installer live – тип установщика. Данная опция добавляет установщик в меню live системы;
5. debian-installer-gui true – версия установщика. Добавляется графический вариант. Текстовая версия установщика при этом также присутствует;
6. distribution bookworm – релиз Debian. Используется последний – Debian 12 Bookworm;
7. mirror-bootstrap "http://mirror.yandex.ru/debian/" – репозиторий, используемый на bootstrap стадии сборки. В данном случае используется репозиторий Яндекса;
8. mirror-chroot "http://mirror.yandex.ru/debian/" – репозиторий, используемый на chroot стадии сборки. Также используются репозитории Яндекса;
9. binary-images iso-hybrid – тип собираемого образа. Используется iso-hybrid для возможности записи на USB носитель в будущем.

### Обзор основных 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 [14]. 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 системах, таких как Linux, BSD, Solaris, macOS и др. Является стандартом практически для всех Linux дистрибутивов, в которые 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 [18]. 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 KDC, 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:

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

Поскольку в собираемой системе требуется аутентификация как локальных пользователей (Администратора), 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 Microsoft Windows, являющийся частью пакета Samba. Предоставляет набор сервисов для NSS, присутствующей в большинстве современных библиотек языка C, произвольных приложений PAM, а также для самой Samba.

NSS 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 winbind имеет одноименное название и может быть использован для разрешения информации о группах и пользователях Windows Server. Он предоставляет механизм аутентификации через соответствующий PAM модуль.

### Обзор прикладного программного обеспечения

Для того, чтобы пользователи могли без затруднений выполнять свои задачи, пользуясь системой, она должна поставляться со всеми необходимыми прикладными программами. Среди них офисный пакет, инструменты для работы с документами, графикой, видео, аудио и так далее. Рассмотрим, какие конкретно программы были выбраны для выполнения этих задач.

В качестве ключевого компонента прикладного ПО выступает офисный пакет, так как он призван заменить лицензионное решение от Microsoft в виде Microsoft Office. В его роли выступает LibreOffice. Это мощный офисный пакет, переведенный на множество языков и поддерживающий большинство ОС. LibreOffice бесплатен как для частного, так и для образовательного или коммерческого использования. Он предустановлен на множестве популярных дистрибутивах, что фактически делает его стандартом для ОС на базе ядра Linux. LibreOffice состоит из нескольких программ, среди которых:

1. Writer – аналог Microsoft Word;
2. Cacl – аналог Microsoft Excel;
3. Impress – аналог Microsoft PowerPoint;
4. Draw – приложение для работы с векторной графикой;
5. Math – приложение для работы с математическими и химическими формулами;
6. Base – аналог Microsoft Access.

Интерфейс этих программ несколько отличается от своих аналогов. Однако после привыкания к нему они становятся достойными альтернативами программам пакета Microsoft Office.

Перечень всего прикладного ПО и выполняемых им задач:

1. Редактирование видео – Kdenlive;
2. Офисный пакет – LibreOffice;
3. Работа с документами – Okular;
4. Медиаплеер – VLC;
5. Работа с графикой – GIMP;
6. Файловый менеджер – Dolphin;
7. Запись видео и трансляция – OBS Studio;
8. Редактирование аудио файлов – audacity;
9. Виртуализация – VirtualBox и Docker;
10. Разработка – VisualStudioCode и Pycharm;
11. Браузер – Yandex Browser и Firefox;
12. Эмулятор терминала – Konsole, Kitty;
13. Мессенджер – Telegram;
14. Торрент клиент – Qbittorrent;
15. Работа с архивами – Ark;
16. Запуск изолированных десктопных приложений – Flatpak;
17. Игровой лаунчер – lutris;
18. Запись дисков и ISO образов - Xfburn.

### Способы запуска Windows приложений

Для запуска Windows приложение предлагается использовать программу Bottles. Она упрощает запуск и настройку приложений на базе Proton и Wine, предоставляя интерфейс для управления префиксами Wine-окружений и параметров запуска. В Bottles применяется полноценная система управления зависимостями по типу той, что используется в пакетных менеджерах дистрибутивов. Для конкретного Windows приложения определяется список необходимых зависимостей, которые будут загружены и установлены для его корректной работы. Все они находятся в специальном репозитории. За счёт этого нет необходимости устанавливать разные версии Wine для каждого приложения.

Окружения содержат готовые настройки библиотеки и зависимости для определённых классов приложений. Они называются «бутылками». Имеется три базовых окружения: приложение – для прикладных программ, игры – для игр, и пользовательское – для экспериментальных окружений. При необходимости есть возможность установить несколько разных версий Wine и переключаться между ними «на лету».

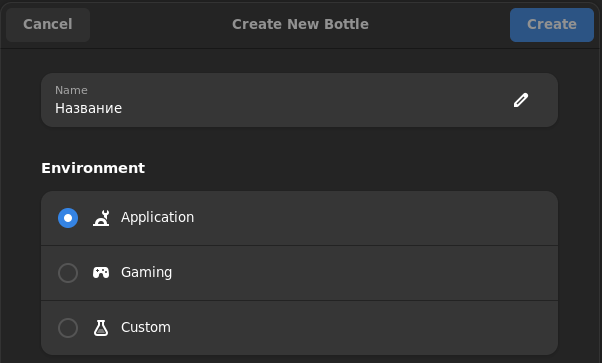


Рисунок 3 – Создание «бутылки» в Bottles

Возможен импорт окружений из других менеджеров, например, Lutris.

Windows приложения можно запустить и без Bottles, используя только Wine и Winetriks. Также возможно сделать это через Lutris, добавив приложение как игру. Однако Bottles предоставляет централизованный и настраиваемый способ сделать это, поэтому рекомендуется использовать именно его.

## СОСТАВЛЕНИЕ КОНФИГУРАЦИИ ДЛЯ СБОРКИ СИСТЕМЫ И СОЗДАНИЕ ИНФРАСТРУКТУРЫ

### Настройка системы в процессе сборки

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

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

Листинг 3 – код скрипта установки темы оформления

#!/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 [5]. Затем переходим в директорию с файлами и запускаем скрипт установки. После успешной установки можем удалить клонированный репозиторий.

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

Листинг 4 – код скрипта настройки PAM модулей:

#!/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 менеджере входа в систему systemd и mkhomedir – для создания домашнего каталога пользователя при входе в систему, если таковой отсутствует.

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

Листинг 5 – код скрипта добавления flathub репозитория:

#!/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 сотрудников. 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. Это можно сделать в файле /etc/resolf.conf. Записи 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 .

Листинг 6 – конфигурация NetworkManager для настройки DNS:

[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 системы получим следующий файл /etc/resolf.conf:

Листинг 7 – содержимое файла /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 возникает одна проблема. Дело в том, что lb build в процессе сборки переопределяет содержимое этого файла, поэтому нельзя 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 скрипте joindomain:

Листинг 8 – код функции configure\_hosts скрипта 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:

Листинг 9 – содержимое файла /etc/hosts:

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 и 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 реализаций протокола NTP и может подключиться к NTP службе контроллера домена [20]. Состоит из демона chronyd, CLI chronyc и конфигурационного файла chrony.conf. Демон автоматически включается после установки пакета, поэтому это не нужно делать вручную. Файл конфигурации Chrony хранится в /etc/chrony/chrony.conf, поэтому добавим соответствующий файл в config/includes.chroot/. Он содержит следующие настройки:

Листинг 10 – конфигурация Chrony:

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

1. keyfile – расположение файла, содержащего пары ID/ключ для NTP аутентификации;
2. driftfile – расположение файла, в котором хранится информация о скорости, с которой системные часы опережают реальное время или же отстают от него;
3. log tracking measurements statistics – данная настройка указывает, какую именно информацию нужно логировать;
4. logdir – директория, в которой сохраняются логи;
5. maxupdateskew – устанавливает порог, при котором отставание или опережение реального времени системным считается недостаточно надёжным для использования;
6. hwclockfile – расположение файла adjtime, который используется программой hwclock в Linux, позволяющей определить, использует ли компьютер местное время или UTC;
7. rtcsync – включает синхронизацию времени между ядром и RTC;
8. 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;
9. bindcmdaddress – указывает, на каких интерфейса принимать пакеты. В нашем случае на всех;
10. 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:

Листинг 11 – конфигурация Kerberos:

[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:

1. default\_realm – адрес KDC. Контроллера домена;
2. dns\_lookup\_realm – данный параметр позволяет включить или отключить поиск kerberos-имени домена через DNS;
3. dns\_lookup\_realm – поиск kerberos-настроек домена через DNS;
4. default\_tgs\_enctypes – поддерживаемые алгоритмы шифрования для TGS;
5. defaukt\_tkt\_enctypes – поддерживаемые алгоритмы шифрования для TGT;
6. permitted\_enctypes – определяет все поддерживаемые алгоритмы шифрования для ключей сессий;
7. 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 4 – Получение TGT с помощью команды «kinit»

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

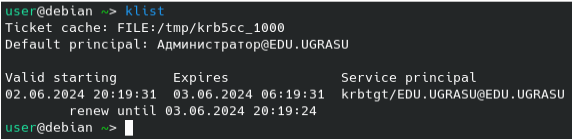


Рисунок 5 – Вывод команды «klist» после получения TGT

#### 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 [15]. Также необходимо настроить проецирование групп и пользователей домена в систему, чтобы можно было совершать операции, завязанные на них. Например, назначение пользователей домена владельцами директорий и файлов. За это отвечает Winbind [18]. Оба этих компонента настраиваются в одном файле – /etc/samba/smb.conf [16]. Добавим его в конфигурацию lb build в config/includes.chroot/etc/samba/smb.conf. Рассмотрим его настройки:

Листинг 12 – конфигурация Samba

[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:

1. workgroup – указывает, в какой группе будут находится пользователи домена, авторизовавшиеся в системе. Этот параметр должен совпадать с рабочей группой контроллера домена;
2. 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;
3. realm – указывает на KDC сервер, который нужно использовать при аутентификации пользователей;
4. kerberos method – указывает на то, как проверяются TGT. При значении параметра «system keytab» используется локальная таблица ключей;
5. winbind use default domain – указывает, нужно ли winbind обрабатывать пользователей, не имеющих названия домена в их имени;
6. 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;
7. winbind enum users - определяет, будет ли winbind перечислять пользователей домена;
8. winbind enum groups - определяет, будет ли winbind перечислять группы домена;
9. winbind cache time – определяет время кеширования информации о пользователях и группах перед повторным запросом;
10. idmap config \* – отвечает за сопоставление между SID пользователей домена и ID пользователей Linux системы. При текущем значении параметра пользователям системы будут присваиваться ID из диапазона 20000-40000;
11. 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;
12. 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;
13. 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, настроив 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:

Листинг 13 – конфигурация NSS

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.

#### 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:

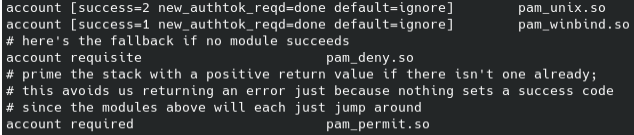


Рисунок 6 – содержание файла common-account

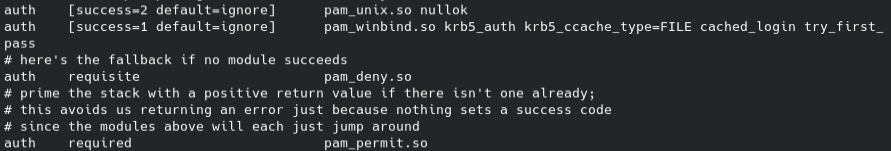


Рисунок 7 – содержание файла common-auth

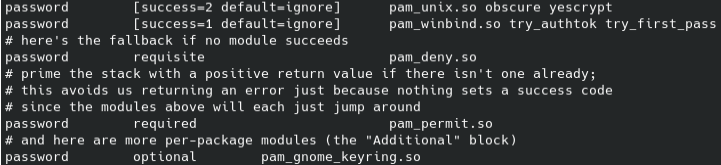


Рисунок 8 – содержание файла common-password

#### Подключение к домену

С версии Samba 4.15.0 появилась мощная утилита samba-tool, позволяющая осуществлять полное администрирование Samba [22]. В том числе с помощью неё можно подключиться к домену. Достаточно вызвать команду «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 необходимо включить systemd сервисы smbd, nmbd и winbind для возможности подключения и работоспособности взаимодействия с доменом после перезагрузки системы.

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 подключении к другому домену. Его полный код приведён в [Приложении Б](#_Приложения). 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.

Листинг 14 – обработка параметров в скрипте joindomain

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 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()»:

Листинг 15 – функции включения сервисов скрипта joindomain

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:

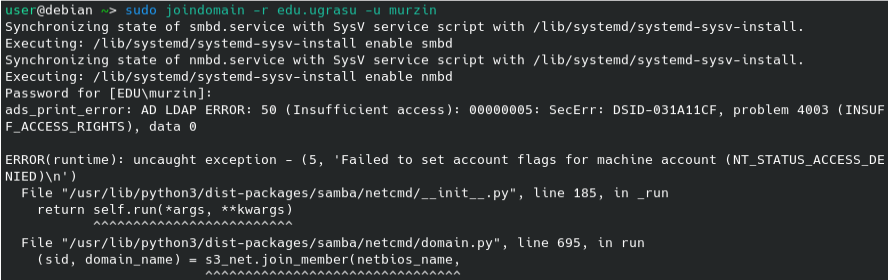


Рисунок 9 – Ошибка подключения к домену по причине отсутствия прав администратора

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

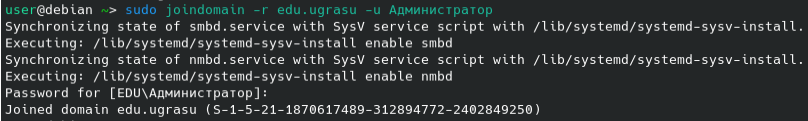


Рисунок 10 – Успешное подключение к домену

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

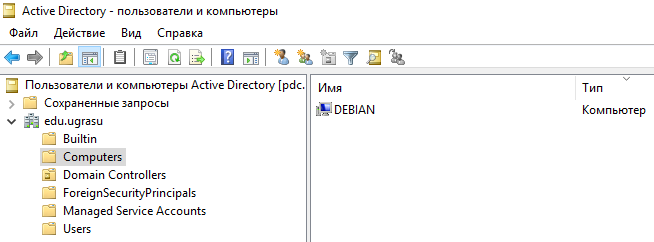


Рисунок 11 – Список компьютеров домена после подключения

### 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:

Листинг 16 – общий вид кода сервиса для запуска скриптов настройки системы

[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 сервисов. 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, выполняющихся 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 избежание этого, был написан дополнительный файл конфигурация для SDDM, который при первой загрузке системы выполняет вход в графическое окружение 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:

Листинг 17 – часть конфигурации SDDM, отвечающая 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:

Листинг 18 – код скрипта 91start-system-adjustment-services

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 нужных операций 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»:

Листинг 19 – код скриптаа включения темы оформления

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 и VirtualBox

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:

Листинг 20 – спислк Flatpak приложения 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:

Листинг 21 – код установки Flatpak приложений

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 приложений в системе, и, при отрицательном результате, также выйдем с кодом «1»:

Листинг 22 – код проверки факта установки Flatpak приложений

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 приложений. По этой причине, после первого запуска системы стоит немного подождать, пока они появятся в системе.

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

Листинг 23 – код скриптаа установки VirtualBox

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, есть ли установленный пакет в системе. Если нет, выходим с кодом «1».

#### Установка драйверов GPU

Базовые драйвера GPU устанавливаются в процессе сборки системы и позволяют спокойно пользоваться 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:

Листинг 24 – код скриптаа настройки драйверов

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 не нативных игр. После с помощью команды «lspcu» и «grep» получаем производителя GPU и, в зависимости от результата, устанавливаем или удаляем необходимые пакеты.

#### Настройка списка репозиториев

Список репозиториев представляет собой текстовый файл, располагающийся в /etc/apt/sources.list. Его структура была подробно рассмотрена в главе 2.1.2. 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 избежание 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:

Листинг 25 – код скрипта настройки репозиториев системы

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 [17]. Установка состоит из множества этапов. Preseed предоставляет способ установки ответов на вопросы всех этапах в процессе установки.

Русская локализация устанавливается с помощью строки:

Листинг 26 – установка русской локализации установщика

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

Для настройки клавиатуры используется:

Листинг 27 – установка параметров клавиатуры установщика

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.

Листинг 28 – настройка пользователей в установщике

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

Строка «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.

Листинг 29 – настройка групп в установщике

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»:

Листинг 30 – установка часового пояса в установщике

d-i time/zone string Asia/Yekaterinburg

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

Листинг 31 – настройка apt в установщике

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

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

После нужно настроить параметры загрузчика:

Листинг 32 – настройка параметров загрузчика установщика

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 на накопитель USB).

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

Листинг 33 – код для моментальной перезагрузки после установки

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.

### Создание Debian репозитория ЮГУ

Чтобы Linux дистрибутив можно было считать полноценным, ему нужна своя инфраструктура. В качестве таковой выступают в том числе репозитории со своими, специфичными для конкретного дистрибутива пакетами. В рамках данной работы был поднят и настроен собственный Debian репозиторий, куда был добавлен пакет со скриптом подключения системы к домену. Рассмотрим процесс его создания.

Для данной задачи нужен сервер, находящийся в сети ЮГУ. Он был выделен научным руководителем и имеет адрес 192.168.254.66. Далее на него необходимо установить операционную систему на базе ядра Linux, в качестве которой выступит Debian. Процесс установки Debian на сервер не является сложным и рассматриваться не будет.

Сам репозиторий представляет из себя директорию с deb пакетами и информации о них. Поэтому на следующем шаге нужно сделать свой deb пакет, который будет содержать скрипт подключения к домену, а также информацию о его размещении в системе и требуемых зависимостях. Сначала создаётся директория, название которой должно соответствовать следующему шаблону:

(имя\_пакета)\_(версия)-(номер\_релиза)\_(архитектура)

Номер релиза используется для используется для отслеживания разных релизов пакетов одной и той же версии. Обычно устанавливается «1». В нашем случае название директории имеет следующий вид:

joindomain\_0.0.1-1\_amd64

Эта директория является корневой для нашего пакета. Чтобы указать путь установки скрипта, просто создадим вложенные директории, названия которых должны соответствовать директориям в целевой системе, и скопируем туда скрипт.

Каждый пакет должен иметь control файл, который содержит основную информацию о нём. Он должен находится в директории DEBIAN. Содержание файла:

Листинг 34 – код control файла пакета

Package: joindomain

Version: 0.0.1

Maintainer: Dmitry Shcherbakov <dmitri.sherbak@gmail.com>

Depends: acl, attr, samba, samba-common-bin, winbind, libpam-winbind, libnss-winbind, krb5-config, krb5-user, python3-setproctitle, chrony, dnsutils

Architecture: amd64

Homepage: http://example.com

Description: Script to join computer to Windows AD domain

В директиве Depends перечисляются все необходимые зависимости пакета. При его установке все недостающие зависимости будут установлены автоматически.

Для того, чтобы собрать пакет из составленной директории, понадобится утилита dpkg:

Листинг 35 – команда для создания deb пакета

dpkg --build ./joindomain\_0.0.1-1\_amd64/

В результате вызова данной команды будет создан пакет joindomain\_0.0.1.deb. Теперь его можно установить в систему с помощью менеджера пакетов apt.

После создания пакета можно заниматься наполнением содержимого директории самого репозитория. Создадим директорию usu-repo, внутри которой создадим директорию pool/main. Именно здесь будут храниться наши пакеты, поэтому скопируем пакет, полученный на предыдущем этапе сюда. Список всех доступных пакетов должен хранится в директории dists/stable/main/binary-amd64 в файле Packages. Его можно сгенерировать с помощью утилиты dpkg-scanpackages. Также необходимо создать Release файл, содержащий информацию о репозитории и информацию о пакетах в виде их хешэй и размеров. Создадим его с помощью специального скрипта и переместим в директорию dists/stable.

На этом этапе директория нашего репозитория готова. Осталось скопировать её на сервер ЮГУ и запустить веб-сервер. В качестве веб-сервера используется nginx. Его конфигурация приведена в [Приложении Б](#_Приложения).

После запуска веб-сервера и подключения к сети ЮГУ можно проверить работоспособность сервера, перейдя по ссылке <http://192.168.254.66.> На странице браузера должно отобразиться содержимое директории репозитория.

Для того, чтобы репозиторием можно было пользоваться в установленной системе, а также во время её сборки, необходимо добавить файл с адресом репозитория в config/archives/ конфигурации live-build. Создадим файлы usu-repo.list.chroot и usu-repo.list.binary и добавим в них следующую строчку:

deb [trusted=yes] http://192.168.254.66/var/usu-repo stable main

Опция «trusted=yes» позволяет не проверять ключ репозитория. Так как сервер находится внутри сети ЮГУ и используется только компьютерами университета, в настройке ключей нет необходимости.

# Заключение

В результате выполнения данной работы был собран live ISO образ операционной системы на базе ядра Linux с прикладным программным обеспечением. ПО, использующиеся в данной сборке, подобрано под спектр задач, которые нужно выполнять студентам и сотрудникам ЮГУ. Установленная операционная система без каких-либо проблем подключается к любому из доменов университета. После подключения к домену пользователи имеют возможность авторизоваться под своими учётными записями, после чего им становится доступно использование сетевых корпоративных ресурсов. Возможность работы в live режиме позволяет загрузиться и посмотреть, как выглядит и функционирует система без установки её на диск. Интерфейс интуитивно понятен и не вызывает у пользователей серьёзный проблем после использования ОС семейства Windows. Процесс установки занимает около 10-ти минут и зависит от общего быстродействия компьютера. Установленная система является менее ресурсоёмкой, чем Windows 10 и Windows 11. По умолчанию у пользователей отсутствует привилегированный режим, что не позволяет им намерено или случайно навредить системе. Созданный собственный Debian репозиторий позволяет размещать и устанавливать только проверенные пакеты.

Очевидно, пользователям понадобится некоторое время, чтобы привыкнуть к работе в новом окружении. Даже в имеющихся ограничениях, сотрудники и студенты не должны испытывать трудностей при выполнения стандартных задач за компьютером. Едва ли получится совершить моментальный переход от ОС семейства Windows к ОС на базе ядра Linux. Но данная работа, определённо, приблизит нас к этому моменту. Придёт время, и мы полностью сможем отказаться от использования зарубежных операционных систем.

# Список использованных источников

1. Debian Live Manual – URL: https://live-team.pages.debian.net/live-manual/html/live-manual/index.en.html (дата обращения: 01.01.2024)
2. live-build(7) — live-build — Debian testing — Debian Manpages – URL: https://manpages.debian.org/testing/live-build/live-build.7.en.html (дата обращения: 12.01.2024)
3. lb\_config(1) — live-build — Debian unstable — Debian Manpages – URL: https://manpages.debian.org/unstable/live-build/lb\_config.1.en.html (дата обращения: 12.01.2024)
4. Ubuntu Manpage: lb clean - Clean build directory – URL: https://manpages.ubuntu.com/manpages/trusty/de/man1/lb\_clean.1.html (дата обращения: 12.01.2024)
5. We10XOS-dark Global Theme - KDE Store – URL: https://store.kde.org/p/1368860 (дата обращения: 27.03.2024)
6. ALT Linux Wiki – URL: https://www.altlinux.org/Главная\_страница (Дата обращения: 02.04.2024)
7. ALT Linux – URL: https://en.wikipedia.org/wiki/ALT\_Linux (Дата обращения: 02.04.2024)
8. Российская операционная система Astra Linux Special Edition – URL: https://astralinux.ru/ (Дата обращения: 08.04.2024)
9. Astra Linux – URL: https://ru.wikipedia.org/wiki/Astra\_Linux (Дата обращения: 02.04.2024)
10. РЕД ОС - российская операционная система – URL: https://redos.red-soft.ru/ (Дата обращения: 02.04.2024)
11. РЕД ОС – URL: https://ru.wikipedia.org/wiki/РЕД\_ОС (Дата обращения: 02.04.2024)
12. НТЦ ИТ РОСА – URL: https://rosa.ru/ (Дата обращения: 03.04.2024)
13. Rosa Linux – URL: https://ru.wikipedia.org/wiki/Rosa\_Linux (Дата обращения: 03.04.2024)
14. Samba - ArchWiki – URL: https://wiki.archlinux.org/title/Samba#Remote\_control\_of\_Windows\_computer (дата обращения: 15.01.2024)
15. Setting up Samba as a Domain Member - SambaWiki – URL: https://wiki.samba.org/index.php/Setting\_up\_Samba\_as\_a\_Domain\_Member (дата обращения: 15.01.2024)
16. smb.conf – URL: https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html (дата обращения: 17.01.2024)
17. Appendix B. Automating the installation using preseeding – URL: https://www.debian.org/releases/stable/amd64/apb.en.html (дата обращения: 20.03.2024)
18. winbindd – URL: https://www.samba.org/samba/docs/current/man-html/winbindd.8.html (дата обращения: 04.03.2024)
19. MIT Kerberos Documentation – URL: https://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf\_files/krb5\_conf.html (дата обращения: 20.02.2024)
20. chrony – chrony.conf(5) – URL: https://chrony-project.org/doc/4.5/chrony.conf.html (дата обращения: 09.04.2024)
21. Настройка репозиториев в Debian – URL: https://serveradmin.ru/nastrojka-repozitoriev-v-debian/ (дата обращения: 12.03.2024)
22. samba-tool – URL: https://www.samba.org/samba/docs/current/man-html/samba-tool.8.html (дата обращения: 15.04.2024)

ПРИЛОЖЕНИЕ А

Техническое задание

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

«СБОРКА ДИСТРИБУТИВА ОПЕРАЦИОННОЙ СИСТЕМЫ С ПРИКЛАДНЫМ ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ НА БАЗЕ ЯДРА LINUX ДЛЯ ПОТРЕБНОСТЕЙ ЮГУ»

Ханты-Мансийск 2024 г.

Перечень принятых сокращений:

ГОСТ – государственный стандарт

ФГБОУ – Федеральное государственное образовательное учреждение

высшего образования

ЮГУ – Югорский государственный университет

ПК – персональный компьютер

ВКР – выпускная квалификационная работа

АС – автоматизированная система

1. Общие сведения
   1. Полное и краткое наименования информационной системы

Полное наименование системы – «Дистрибутив операционной системы с прикладным программным обеспечением на базе ядра Linux Ugrach”;

Условное обозначение системы – «Ugrach», «сборка», «дистрибутив».

Наименование предприятия заказчика системы.

* 1. Заказчик АС

Заказчик – ФГБОУ ВО "ЮГУ", "Югорский государственный университет" 628012, Ханты-Мансийский автономный округ - Югра, г. Ханты-Мансийск, ул. Чехова, 16.

1. Цели и назначение создания дистрибутива
   1. Назначение приложения

Создание дистрибутива операционной системы с прикладным программным обеспечением на базе ядра Linux Ugrach для замены существующей системы на базе Windows.

* 1. Цели создания приложения

Установка и функционирование на ПК студентов и сотрудников ЮГУ.

1. Характеристика объектов автоматизации
   1. Основные сведения об объекте автоматизации

Дистрибутив представляет из себя операционную систему на базе ядра Linux, основывающийся на дистрибутиве Debian. Поставляется в виде ISO образа диска и устанавливается со загрузочного USB носителя. Включает в себя прикладное программное обеспечение и настройки для взаимодействия с домен Windows AD.

1. Требования к автоматизированной системе
   1. Требования к структуре дистрибутива в целом

Загрузочный ISO образ операционной системы с предустановленным программным обеспечением.

* 1. Требования к функциям, выполняемым дистрибутивом

Дистрибутив должен обладать следующим функционалом:

1. Возможность ввода машины в домен с помощью специальной консольной утилиты;
2. Авторизация пользователей в систему по доменным учётным записям;
3. Возможность монтирования сетевых директорий доменных пользователей;
4. Работа с принтерами и сканерами;
5. Возможность запуска приложений, написанных под Windows;
6. Невозможность создания и авторизации локальных пользователей (за исключением Администратора);
7. Наличие программ для виртуализации;
8. Тема оформления, схожая с Windows 10;
9. Наличие программ для работы с документами, таблицами, графикой, мультимедиа, базами данных, носителями данных.
   1. Общие технические требования к дистрибутиву
      1. Требования к эргономике и технической эстетике

Дистрибутив должен иметь графический интерфейс, схожий с графическим интерфейсом Windows 10.

* + 1. Требования к эксплуатации и техническому обслуживанию дистрибутива

Для поддержания корректной и стабильной работы дистрибутива, администратор должен периодически проверять и устанавливать обновления компонентов системы.

* + 1. Требования к защите информации от несанкционированного доступа;

Непривилегированные пользователи не должны иметь доступа к средствам администрирования.

1. Состав и содержание работ по созданию дистрибутива

|  |  |  |
| --- | --- | --- |
| Наименование задачи | Начало | Окончание |
| Разработка технического задания | 01.13.24 | 20.01.24 |
| Выбор инструмента для сборки системы | 22.01.24 | 25.01.24 |
| Составление конфигурации для сборки системы | 28.01.24 | 20.05.24 |
| Написание утилиты подключения к домену | 04.05.24 | 23.05.24 |
| Создание Debian репозитория ЮГУ | 21.05.24 | 28.05.24 |
| Сборка системы | 29.05.24 | 30.05.24 |
| Тестирование дистрибутива | 01.06.24 | 07.06.24 |
| Подготовка промежуточного отчета о  результатах проекта | 27.03.24 | 28.02.24 |
| Подготовка промежуточного отчета о  результатах проекта | 29.04.24 | 30.03.24 |
| Подготовка промежуточного отчета о  результатах проекта | 29.04.24 | 30.04.24 |
| Подготовка отчета о результатах проекта | 26.05.24 | 30.05.24 |
| Завершение разработки дистрибутива «Ugrach» – 07.06.24 | | |

* 1. Выбор инструмента для сборки системы

Выбор инструмента для сборки подразумевает выбор оптимальных для выбранной задачи инструментов и средств сборки дистрибутива

* 1. Составление конфигурации для сборки системы

Составление конфигурации включает в себя составление иерархии директории конфигурации, написание и включение конфигурационных файлов, написание скриптов настройки системы и выбор параметров сборки

* 1. Создание Debian репозитория ЮГУ

Создание Debian репозитория ЮГУ включает в себя установка Debian на один из серверов сети университета, установка веб-сервера, создание самого репозитория, копирование его на сервер и проверку возможности установки пакетов.

* 1. Сборка системы

Сборка система подразумевает запуск утилиты сборки из составленной конфигурации для получения загрузочного ISO образа диска.

1. Порядок контроля и приемки автоматизированной системы
   1. Виды, состав и методы испытаний дистрибутива и его составных частей

Для испытания дистрибутива необходимо произвести его установка на один из компьютеров ЮГУ. После чего должна быть протестирована возможность подключения к домену, авторизация доменных пользователей по своим учётным записям в систему, доступ к сетевым ресурсам, работа с принтерами и общая стабильность работы системы.

1. Требования к документированию
   1. Перечень подлежащих разработке документов

Основными документами, регламентирующими разработку будущей программы, являются документы Единой системы программной документации (ЕСПД):

1. Техническое задание (ГОСТ 19.201-78);
2. Пояснительная записка (ГОСТ 19.404-79).
   1. Вид представления и количество документов

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа. Пояснительная записка должна быть загружена в систему Антиплагиат.ВУЗ. Лист, подтверждающий загрузку пояснительной записки, сдается в учебную часть вместе со всеми материалами не позже, чем за день до защиты выпускной квалификационной работы.

Техническое задание и пояснительная записка, титульные листы других документов должны быть напечатаны, подписаны академическим руководителем образовательной программы 09.03.04 «Программная инженерия», руководителем разработки и исполнителем перед сдачей ВКР в учебную часть не позже одного дня до защиты.

Документация сдается в электронном виде в формате .pdf или .docx. в архиве формата .zip или .rar.

Документация должна содержать информацию инструментах сборки, структуре конфигурации, способе установки, использовании Debian репозитория ЮГУ и способе подключения к домену.

ПРИЛОЖЕНИЕ Б

Листинг 1 – код скрипта joindomain

#!/bin/bash

function help(){

echo "This script disigned to join computer to domain"

echo "Usage $0 [-u] [-p] [-r] [-w] [-h]"

echo "Options:"

echo " -u Specify domain user."

echo " -p Specify domain user's password."

echo " -r Specify domain realm."

echo " -w Specify domain workgroup. Pass these parameter to override current Samba and Kerberos configuration."

echo " -h Display help."

}

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

}

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

}

function change\_domain\_conf(){

sudo sed -i "s/.\*workgroup.\*/ workgroup = $(echo $1 | tr '[:lower:]' '[:upper:]')/" /etc/samba/smb.conf

sudo sed -i "s/.\*realm.\*/ realm = $(echo $2 | tr '[:lower:]' '[:upper:]')/" /etc/samba/smb.conf

sudo sed -i "s/.\*default\_realm.\*/ default\_realm = $(echo $2 | tr '[:lower:]' '[:upper:]')/" /etc/krb5.conf

}

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

if [[ -n "$user" ]] && [[ -n "$realm" ]]; then

configure\_hosts "$realm";

enable\_smbd\_nmbd;

if [ -n "$workgroup" ]; then

if [ ! -n "$password" ]; then

change\_domain\_conf $workgroup $realm;

sudo samba-tool domain join $realm MEMBER -U $user --workgroup $workgroup;

exit 0;

else

change\_domain\_conf $workgroup $realm;

sudo samba-tool domain join $realm MEMBER -U $user --workgroup $workgroup --password $password;

exit 0;

fi

else

if [ ! -n "$password" ]; then

sudo samba-tool domain join $realm MEMBER -U $user;

exit 0;

else

sudo samba-tool domain join $realm MEMBER -U $user --password $password;

exit 0;

fi

fi

enable\_winbind;

else

echo "You must specify user and realm to join to domain";

exit 1;

fi

ПРИЛОЖЕНИЕ В

Листинг 2 – конфигурация nginx веб-сервера репозитория ЮГУ

server {

listen 80;

access\_log /var/log/nginx/apt-repository.access;

error\_log /var/log/nginx/apt-repository.error;

location / {

root /var/www/repository/;

autoindex on;

}

location ~ /(.\*)/conf {

deny all;

}

location ~ /(.\*)/db {

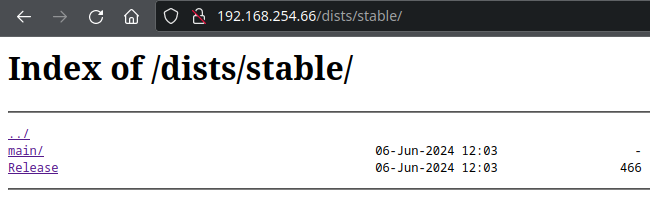
deny all;

}

}

ПРИЛОЖЕНИЕ Г

Страница Debian репозитория ЮГУ



ПРИЛОЖЕНИЕ Д

Скриншот установленной системы

