МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владивостокский государственный университет экономики и сервиса»
Колледж сервиса и дизайна

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНО-ПРАКТИЧЕСКИХ РАБОТ

ПМ.02 Организация сетевого администрирования

профессионального цикла образовательной программы среднего профессионального образования подготовки специалистов среднего звена по специальности 09.02.06 Сетевое и системное администрирование

Очная форма обучения

Владивосток 2022

Методические указания для выполнения лабораторно-практических работ ПМ.02 Организация сетевого администрирования разработаны на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.06 Сетевое и системное администрирование, утвержденного приказом Минобрнауки РФ от 09.12.2016, №1548.

Разработана:

Могулева А.В., преподаватель Колледжа сервиса и дизайна ВГУЭС

Рассмотрена на заседании цикловой методической комиссии
Протокол № _9__ от «_4__» __мая____2022 г.

Председатель ЦМК _  Е.А Стефанович

1 ОБЩИЕ СВЕДЕНИЯ

Практические работы по 09.02.06 Сетевое и системное администрирование**,** проводятся в рамках реализации программы Информационные технологии в профессиональной деятельности.

Результатом освоения ПМ.02 Организация сетевого администрирования является овладение обучающимися профессиональными (ПК), указанными в ФГОС по специальности/профессии по 09.02.06 Сетевое и системное администрирование.

Код	Наименование результата обучения
ПК 2.1	Администрировать локальные вычислительные сети и принимать меры по устранению возможных сбоев.
ПК 2.2	Администрировать сетевые ресурсы в информационных системах.
ПК 2.3	Обеспечивать сбор данных для анализа использования и функционирования программно-технических средств компьютерных сетей.
ПК 2.4	Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности.

С целью овладения профессиональными компетенциями обучающийся в ходе освоения профессионального модуля должен освоить основной вид деятельности организация сетевого администрирования и соответствующие ему общие компетенции и профессиональные компетенции.

Студент должен иметь практический опыт в установке, настройке и сопровождении, контроле использования сервера и рабочих станций для безопасной передачи информации и уметь администрировать локальные вычислительные сети, принимать меры по устранению возможных сбоев и обеспечивать защиту при подключении к информационно-телекоммуникационной сети «Интернет».

В конечном итоге обучающему в ходе освоения ПМ требуется знать:

– основные направления администрирования компьютерных сетей;

– утилиты, функции, удаленное управление сервером;

– технологию безопасности, протоколов авторизации, конфиденциальности и безопасности при работе с сетевыми ресурсами.

В процессе освоения ПМ у студенты должны овладеть общими компетенциями (ОК):

Код	Наименование результата обучения
ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес
ОК 2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество
ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК 4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК 5	Владеть информационной культурой, анализировать и оценивать информацию с использованием информационно-коммуникационных технологий
ОК 6	Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.
ОК 7	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации
ОК 8	Быть готовым к смене технологий в профессиональной деятельности
ОК 9	Уважительно и бережно относиться к историческому наследию и культурным традициям, толерантно воспринимать социальные и культурные традиции
ОК 10	Соблюдать правила техники безопасности, нести ответственность за организацию мероприятий по обеспечению безопасности труда

Перечень лабораторно-практических работ по ПМ.02 Организация сетевого администрирования:

	ПМ.02 Организация сетевого администрирования	Кол-во часов	Общ. компетенции	Проф. компетенции
1.	Первоначальная установка ОС Linux – дистрибутива Debian 11	4	ОК 1-10	ПК 2.1-2.4
2.	Установка пакетов в Debian	2	ОК 1-10	ПК 2.1-2.4
3.	Установка ОС Linux – дистрибутива Arch Linux	6	ОК 1-10	ПК 2.1-2.4
4.	Установка оконного менеджер i3 в дистрибутив Arch Linux	4	ОК 1-10	ПК 2.1-2.4
5.	Установка пакетов в Arch Linux	2	ОК 1-10	ПК 2.1-2.4
6.	Основы работы с текстовым редактором vim и файловым менеджером ranger	10	ОК 1-10	ПК 2.1-2.4
7.	Основы работы с Wine	4	ОК 1-10	ПК 2.1-2.4
8.	Помощники в установке программ и библиотек в Wine	2	ОК 1-10	ПК 2.1-2.4

9.	Обзор схемы стенда сети предприятия на основе GNU/Linux	2	ОК 1-10	ПК 2.1-2.4
10.	Настройка шлюза сети предприятия	10	ОК 1-10	ПК 2.1-2.4
11.	Настройка виртуальной машины как программного коммутатора	2	ОК 1-10	ПК 2.1-2.4
12.	Развертывание сервера DHCP	6	ОК 1-10	ПК 2.1-2.4
13.	Развертывание сервера DHCP на основе isc-dhcp-server	10	ОК 1-10	ПК 2.1-2.4
14.	Развертывание сервера DNS	6	ОК 1-10	ПК 2.1-2.4
15.	Развертывание сервера DNS на основе bind9	10	ОК 1-10	ПК 2.1-2.4
16.	Развертывание прокси-сервера HTTP – Squid	6	ОК 1-10	ПК 2.1-2.4
17.	Развертывание веб-сервера – Apache	2	ОК 1-10	ПК 2.1-2.4
18.	Развертывание NTP-сервера	2	ОК 1-10	ПК 2.1-2.4
19.	Развертывание FTP-сервера	6	ОК 1-10	ПК 2.1-2.4
20.	Развертывание Samba в режиме файлового сервера	6	ОК 1-10	ПК 2.1-2.4
21.	Развертывание почтового сервера с использованием Postfix и Dovecot	6	ОК 1-10	ПК 2.1-2.4
22.	Развертывание SSH-сервера		ОК 1-10	ПК 2.1-2.4
23.	Настройка фильтрации пакетов с помощью UFW		ОК 1-10	ПК 2.1-2.4
24.	Фильтрации пакетов и настройка NAT с использованием nftables		ОК 1-10	ПК 2.1-2.4
25.	Настройка fail2ban		ОК 1-10	ПК 2.1-2.4
26.	Настройка Port Knocking (knockd)		ОК 1-10	ПК 2.1-2.4
27.	Развертывание RADIUS-сервера (FreeRADIUS)		ОК 1-10	ПК 2.1-2.4

2 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНО-ПРАКТИЧЕСКИХ РАБОТ

На выполнение каждой лабораторно-практической работы отведено от двух до десяти часов. Лабораторно-практические работы выполняются в электронном виде.

3 ЗАДАНИЯ-ИНСТРУКЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛАБОРАТОРНО-ПРАКТИЧЕСКИХ РАБОТ

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: Первоначальная установка ОС Linux – дистрибутива Debian 11

Цель работы: Выполнить инсталляцию дистрибутива Debian 11

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

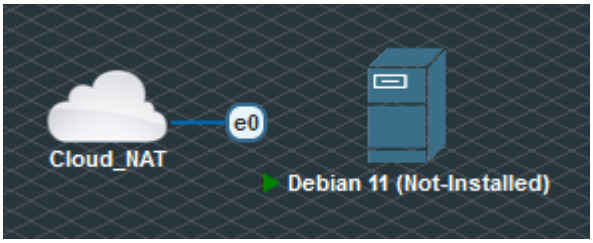
Debian – это операционная система, состоящая из свободного ПО с открытым исходным кодом. В настоящее время Debian GNU/Linux – одн из самых популярных и важных дистрибутивов GNU/Linux, в первичной форме оказавший значительное влияние на развитие этого типа ОС в целом. Также существует проект на основе другого ядра: Debian GNU/Hurd. Debian может использоваться в качестве операционной системы как для серверов, так и для рабочих станций.

Debian имеет наибольшее среди всех дистрибутивов хранилище пакетов – готовых к использованию программ и библиотек, – и если даже не по их числу, то по числу поддерживаемых архитектур: начиная с ARM, используемой во встраиваемых устройствах, наиболее популярных x86-64 и PowerPC, и заканчивая IBM S/390, используемой в мейнфреймах. Для работы с хранилищем разработаны разные средства, самое популярное из которых – Advanced Packaging Tool (APT).

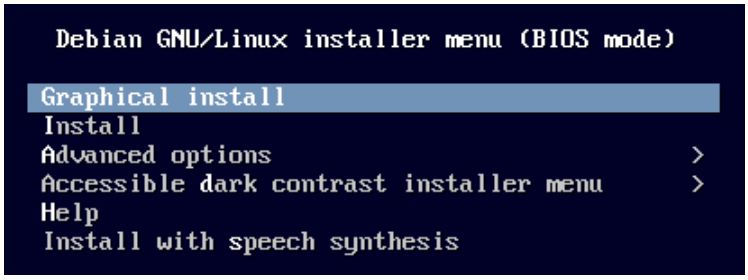
Debian стал основой целого ряда дистрибутивов. Самые известные из них – Kali Linux, Linux Mint, TAILS, Ubuntu.

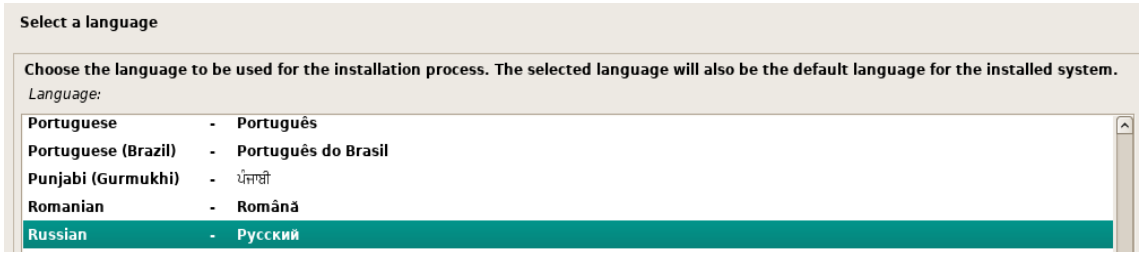
Название «Debian» составлено из имен основателя проекта Иэна Мёрдока (Ian Murdock) и его жены Дебры Линн (Debra Lynn).

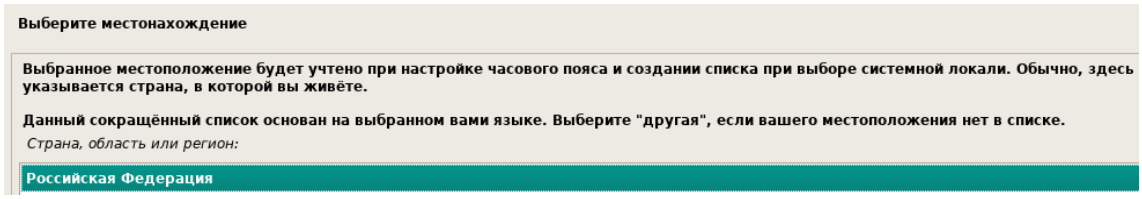
Перед началом установки необходимо подключить виртуальную машину таким образом, как указано на схеме.

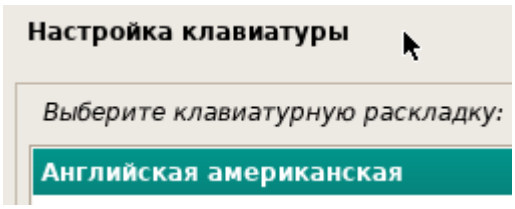


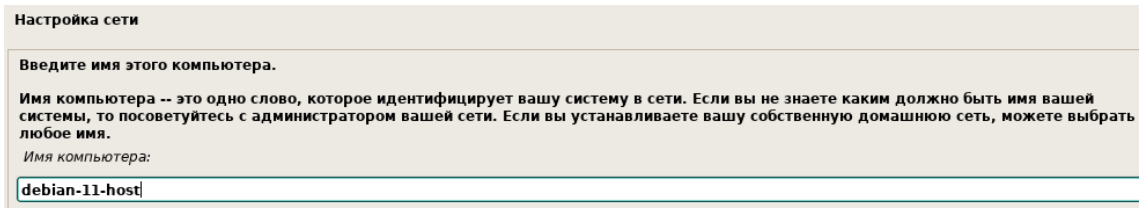
Чтобы установить дистрибутив Debian 11 на виртуальной машине, следуйте графической инструкции, представленной ниже.

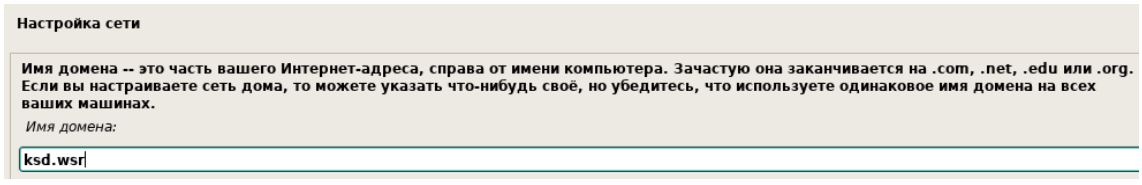












Настройка учётных записей пользователей и паролей

Необходимо ввести пароль учётной записи суперпользователя (root), используемой для администрирования системы. Доступ к компьютеру с использованием этой учётной записи злонамеренных или низкоквалифицированных пользователей может привести к катастрофическим последствиям. Поэтому пароль суперпользователя не должен легко угадываться, подбираться по словарю, и он не должен быть связан с вашей личностью.

Хороший пароль представляет из себя смесь букв, цифр и знаков препинания, и должен периодически меняться.

Пароль учётной записи суперпользователя не должен быть пустым, иначе она будет заблокирована, а настроенной в программе установки пользовательской учётной записи будет разрешено работать с правами суперпользователя через команду "sudo".

Во время ввода пароля вводимые символы не будут отображаться на экране.

Пароль суперпользователя:

guest

☑ Показывать вводимый пароль

Введите тот же самый пароль ещё раз, чтобы убедиться в правильности ввода.

Введите пароль ещё раз:

guest

☑ Показывать вводимый пароль

Настройка учётных записей пользователей и паролей

Будет создана учётная запись пользователя, которая будет использоваться вместо учётной записи суперпользователя (root) для выполнения всех действий, не связанных с администрированием.

Введите реальное имя этого пользователя. Эта информация будет использована в письмах в поле "От кого", посылаемых этим пользователем, а также всеми программами, которые показывают или используют реальное имя пользователя в своей работе. Ваше имя и фамилия вполне подходят.

Введите полное имя нового пользователя:

guest

Настройка учётных записей пользователей и паролей

Выберите имя пользователя (учётную запись), под которым вы будете известны в системе. В качестве учётной записи может быть использовано ваше реальное имя. Учётная запись должна начинаться со строчной латинской буквы, за которой может следовать любое количество строчных латинских букв или цифр.

Имя вашей учётной записи:

guest

Настройка учётных записей пользователей и паролей

Хороший пароль представляет из себя смесь букв, цифр и знаков препинания, и должен периодически меняться.

Введите пароль для нового пользователя:

guest

☑ Показывать вводимый пароль

Проверка правильности ввода осуществляется путём повторного ввода пароля и сравнения результатов.

Введите пароль ещё раз:

guest

☑ Показывать вводимый пароль

Настройка времени

Если нужного часового пояса нет в списке, то вернитесь к шагу "Выбор языка" и выберите страну, в которой используется требуемый часовой пояс (страну, в которой вы живёте или сейчас находитесь).

Выберите часовой пояс:

Москва-01 - Калининград
Москва+00 - Москва
Москва+01 - Самара
Москва+02 - Екатеринбург
Москва+03 - Омск
Москва+04 - Красноярск
Москва+05 - Иркутск
Москва+06 - Якутск
Москва+07 - Владивосток
Москва+08 - Магадан
Москва+09 - Камчатка

Разметка дисков

Если выбрать использование инструмента управления разметкой всего диска, то далее вас попросят указать нужный диск.

Метод разметки:

Авто - использовать весь диск
Авто - использовать весь диск и настроить LVM
Авто - использовать весь диск с шифрованным LVM
Вручную

Разметка дисков

Заметим, что все данные на выбранном диске будут стёрты, но не ранее чем вы подтвердите, что действительно хотите сделать изменения.

Выберите диск для разметки:

SCSI1 (0,0,0) (sda) - 32.2 GB ATA QEMU HARDDISK

Разметка дисков

Выбрано для разметки:

SCSI1 (0,0,0) (sda) - ATA QEMU HARDDISK: 32.2 GB

Диск может быть размечен по одной из следующих схем. Если вы не знаете, что выбрать -- выбирайте первую схему.

Схема разметки:

Все файлы в одном разделе (рекомендуется новичкам)
Отдельный раздел для /home
Отдельные разделы для /home, /var и /tmp

Разметка дисков

Перед вами список настроенных разделов и их точек монтирования. Выберите раздел, чтобы изменить его настройки (тип файловой системы, точку монтирования и так далее), свободное место, чтобы создать новый раздел, или устройство, чтобы создать на нём новую таблицу разделов.

Автоматическая разметка
Настройка программного RAID
Настройка менеджера логических томов (LVM)
Настроить шифрование для томов
Настроить тома iSCSI

▽ SCSI1 (0,0,0) (sda) - 32.2 GB ATA QEMU HARDDISK

>	#1	первичн.	31.2 GB	f	ext4	/
>	#5	логичес.	1.0 GB	f	подк	подк

Отменить изменения разделов
Закончить разметку и записать изменения на диск

Разметка дисков

Если вы продолжите, то изменения, перечисленные ниже, будут записаны на диски. Или же вы можете сделать все изменения вручную.

На этих устройствах изменены таблицы разделов:
SCSI1 (0,0,0) (sda)

Следующие разделы будут отформатированы:
раздел #1 на устройстве SCSI1 (0,0,0) (sda) как ext4
раздел #5 на устройстве SCSI1 (0,0,0) (sda) как подк

Записать изменения на диск?

○ **Нет**
◉ **Да**

Настройка менеджера пакетов

При сканировании установочного носителя найдена метка:

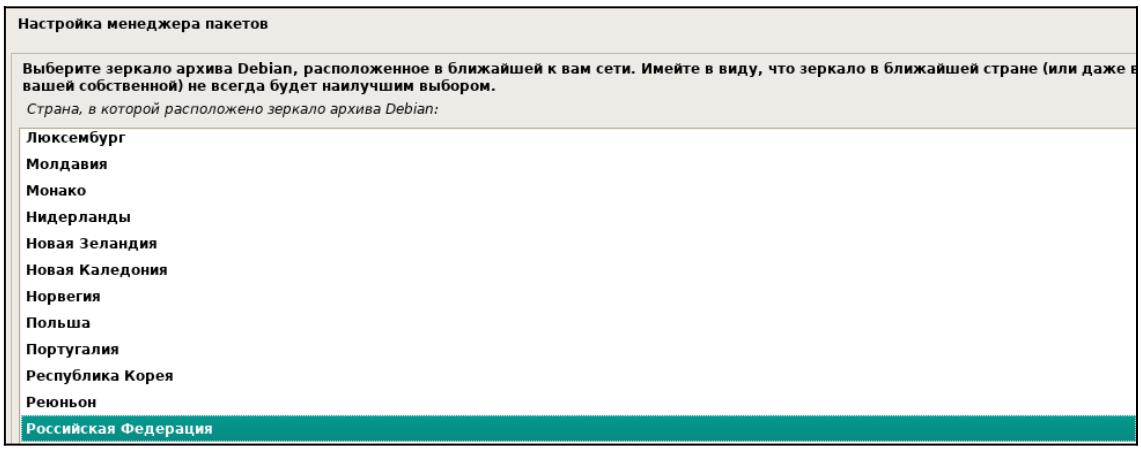
Debian GNU/Linux 11.2.0 _Bullseye_ - Official amd64 NETINST 20211218-11:12

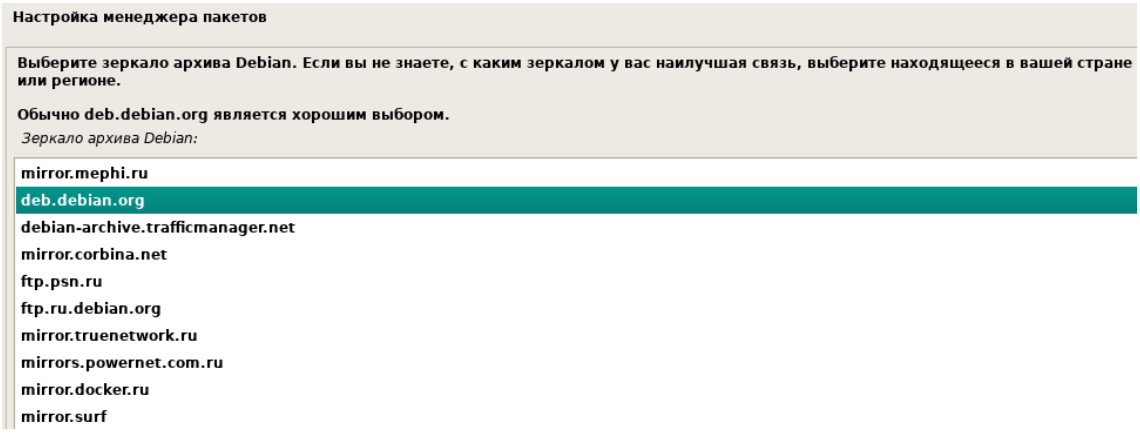
Сейчас вы можете просканировать дополнительные носители, чтобы их можно было использовать из менеджера пакетов (apt). Обычно они должны быть от того же самого набора, что и установочный носитель. Если носителей больше нет, то можно просто пропустить этот шаг.

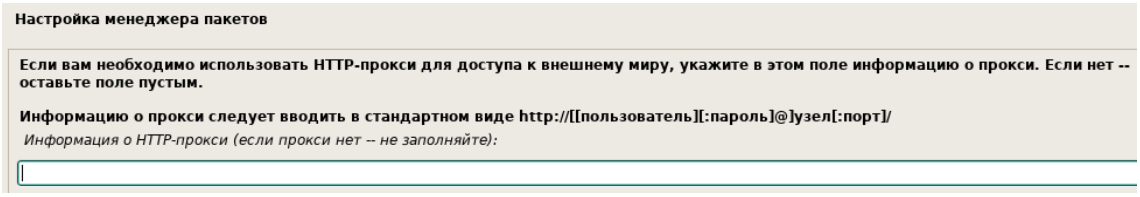
Если вы хотите просканировать другой носитель, вставьте его сейчас.

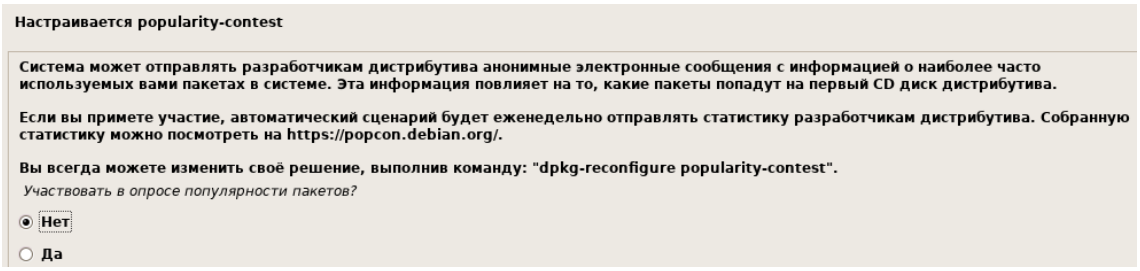
Просканировать дополнительный установочный носитель?

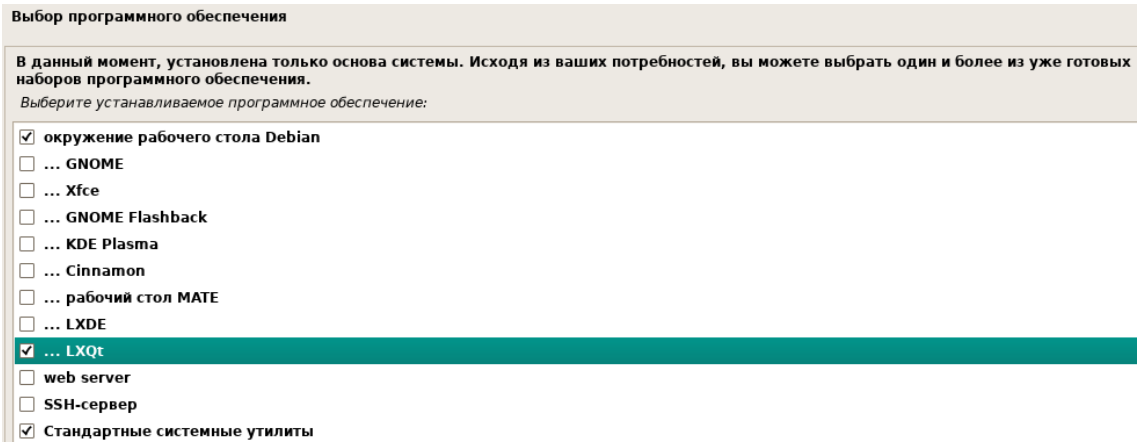
◉ **Нет**
○ **Да**

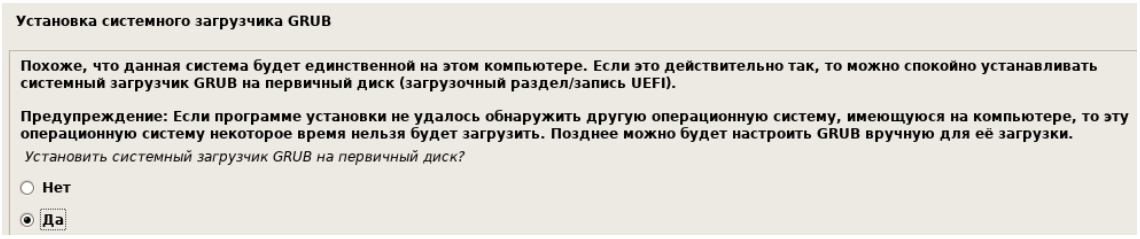

Настройка менеджера пакетов
Выберите зеркало архива Debian, расположенное в ближайшей к вам сети. Имейте в виду, что зеркало в ближайшей стране (или даже в вашей собственной) не всегда будет наилучшим выбором.
Страна, в которой расположено зеркало архива Debian:
Люксембург
Молдавия
Монако
Нидерланды
Новая Зеландия
Новая Каледония
Норвегия
Польша
Португалия
Республика Корея
Реюньон
Российская Федерация

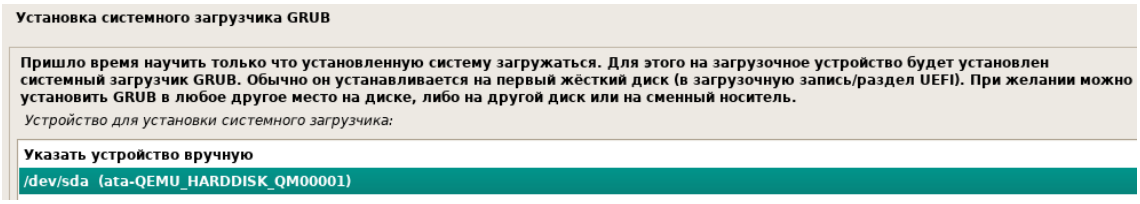

Настройка менеджера пакетов
Выберите зеркало архива Debian. Если вы не знаете, с каким зеркалом у вас наилучшая связь, выберите находящееся в вашей стране или регионе.
Обычно deb.debian.org является хорошим выбором.
Зеркало архива Debian:
mirror.mephi.ru
deb.debian.org
debian-archive.trafficmanager.net
mirror.corbina.net
ftp.psn.ru
ftp.ru.debian.org
mirror.truenetwork.ru
mirrors.powernet.com.ru
mirror.docker.ru
mirror.surf

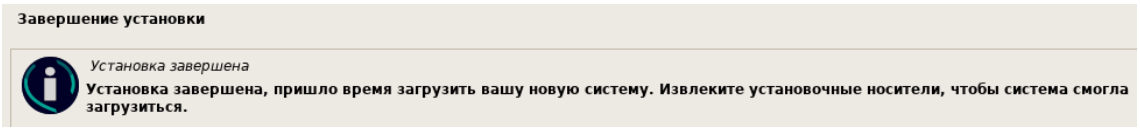

Настройка менеджера пакетов
Если вам необходимо использовать HTTP-прокси для доступа к внешнему миру, укажите в этом поле информацию о прокси. Если нет -- оставьте поле пустым.
Информацию о прокси следует вводить в стандартном виде http://[[пользователь][:пароль]@]узел[:порт]/
Информация о HTTP-прокси (если прокси нет -- не заполняйте):


Настраивается popularity-contest
Система может отправлять разработчикам дистрибутива анонимные электронные сообщения с информацией о наиболее часто используемых вами пакетах в системе. Эта информация повлияет на то, какие пакеты попадут на первый CD диск дистрибутива.
Если вы примете участие, автоматический сценарий будет еженедельно отправлять статистику разработчикам дистрибутива. Собранную статистику можно посмотреть на https://popcon.debian.org/.
Вы всегда можете изменить своё решение, выполнив команду: "dpkg-reconfigure popularity-contest".
Участвовать в опросе популярности пакетов?
Нет
Да


Выбор программного обеспечения
В данный момент, установлена только основа системы. Исходя из ваших потребностей, вы можете выбрать один и более из уже готовых наборов программного обеспечения.
Выберите устанавливаемое программное обеспечение:
окружение рабочего стола Debian
... GNOME
... Xfce
... GNOME Flashback
... KDE Plasma
... Cinnamon
... рабочий стол MATE
... LXDE
... LXQt
web server
SSH-сервер
Стандартные системные утилиты







ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: Установка пакетов в Debian

Цель работы: Установить пакеты с программным обеспечением в Debian 11:

– Синтаксис и опции apt;

– Использование утилиты apt;

– Графический интерфейс apt.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Пакетный менеджер APT или Advanced Package Tool используется во множестве дистрибутивов, основанных на Debian или Ubuntu, а таких дистрибутивов сейчас очень много. Этот пакетный менеджер поддерживает все необходимые функции, вы можете устанавливать и удалять пакеты, обновлять то, что было уже установлено, искать пакеты, устанавливать их из файла или загружать без установки. При этом все зависимости будут разрешаться автоматически.

Утилита apt ничем не уступает пакетному менеджерам dnf и yum, используемых в RPM дистрибутивах. Да, здесь все ещё не поддерживается частичное обновление пакетов, как в Zypper, но в целом всё очень неплохо. В этом разделе мы подробно рассмотрим как пользоваться apt в Linux для решения задач работы с программами.

Синтаксис и опции apt

Синтаксис команды apt очень простой и похож на другие команды Linux:

apt [опция] [команда] <параметры_команды>

Опции указывают общее поведение утилиты, команда – действие, которое надо выполнить, а в параметрах команды обычно передается имя пакета, с которым следует работать, например, установить или удалить. Вот основные опции утилиты:

– -v, --version – выводит версию утилиты;

– -h, --help – выводит справку по использованию утилиты;

– -y, --yes – автоматически отвечать "да" на все возникающие вопросы;

– --assume-no – автоматически отвечать "нет" на все возникающие вопросы;

– -d, --download-only – только скачать пакеты и больше ничего не делать;

– -f, --fix-broken – исправить недостающие зависимости;

– --no-download – ничего не загружать, использовать только пакеты из кэша;

– -s, --simulate – режим симуляции, никакие операции не выполняются, только выводится информация на экран;

– --allow-unauthenticated – позволяет установить пакеты, из репозиториев, для которых нет GPG подписи;

– --no-install-recommends – не устанавливать рекомендованные пакеты, по умолчанию будут установлены;

– -m, --ignore-missing – игнорировать пакеты, которые существует;

– -q, --quiet – выводить минимум информации, не показывать прогресс бар;

– -V, --verbose-versions – показывать полные версии обновленных пакетов;

– --only-upgrade – не устанавливать новые пакеты, только обновлять;

– --allow-downgrades – разрешить откатывать версию пакетов;

– --reinstall – переустановить пакет если он уже установлен.

А теперь давайте пройдёмся по командам apt, которые вы можете использовать:

– install – установить пакет;

– remove – удалить пакет, конфигурационные файлы, которые были изменены в вашей системе удалены не будут;

– purge – полностью удалить пакет, вместе со всеми его конфигурационными файлами;

– autoremove – очистить ненужные пакеты;

– autoclean – очистить кэш пакетов;

– update – обновить списки пакетов из репозиториев;

– upgrade – обновить версию пакета до последней, если пакет не указан будут обновлены все пакеты;

– full-upgrade – полное обновление системы, включая удаление несовместимых или больше ненужных пакетов;

– list – список установленных пакетов;

– search – поиск пакетов;

– show – посмотреть информацию о пакете;

– download – скачать пакет в текущую папку;

– edit-sources – открыть с настройками репозиториев в текстовом редакторе.

– source – скачать исходный код пакета в текущую папку;

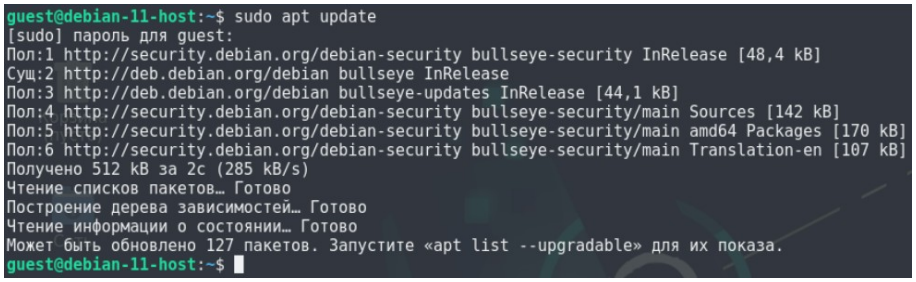
– build-dep – установить зависимости необходимые для сборки выбранного пакета.

Дальше рассмотрим несколько примеров работы с apt, которые пригодятся как начинающим, так и более опытным пользователям.

Использование утилиты apt

Обновление пакетов. Сначала давайте обновим список пакетов apt из репозиториев. Репозитории находятся на удалённых серверах и когда утилита apt ищет пакет для установки, естественно, что она не обращается ко всем репозиториям подряд чтобы узнать где он находится. В системе уже есть сохранённых кэш информации о том, какие пакеты вообще есть и где их можно скачать. Для обновления этого кэша используйте команду update:

guest@localhost:~$ sudo apt update



Во время загрузки URL-репозиториев обозначаются специальными префиксами, вот что они означают:

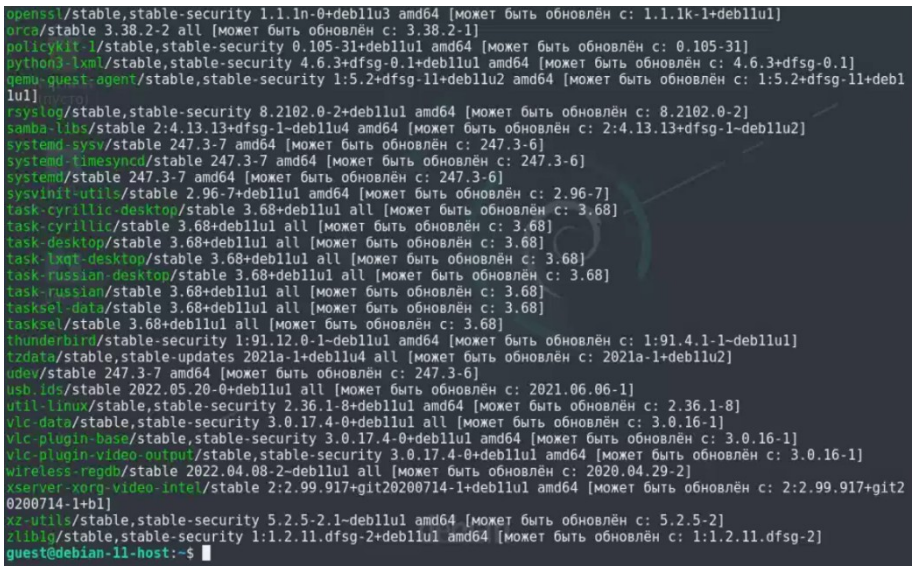
– Hit (Сущ) – список не изменился с момента предыдущей загрузки;

– Ign (Игн) – репозиторий игнорируется, либо он слишком новый, либо произошла незначительная ошибка во время загрузки;

– Get (Пол) – доступна новая версия и она будет загружена.

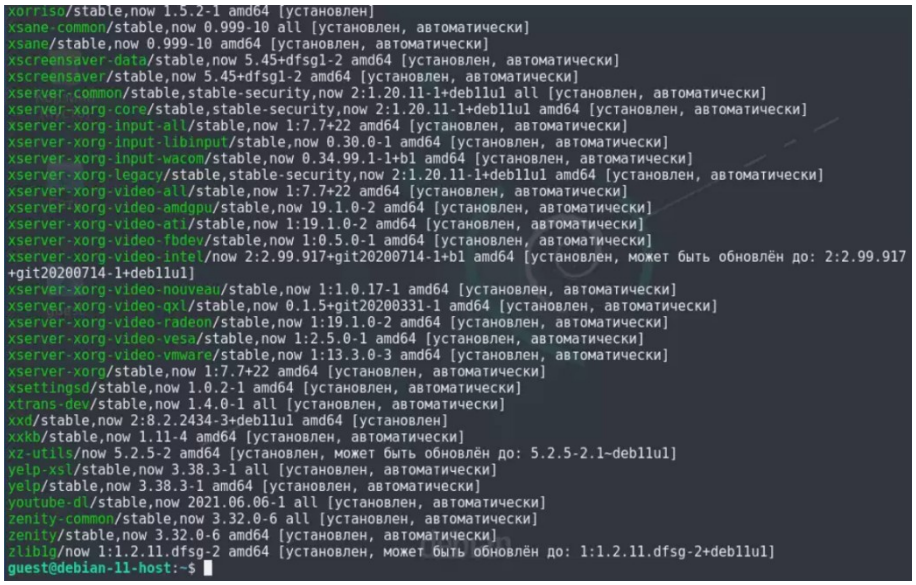
Когда кэш обновлен вы можете посмотреть для каких пакетов доступны обновления:

guest@localhost:~$ sudo apt list --upgradable



Аналогично можно посмотреть установленные пакеты apt:

guest@localhost:~$ sudo apt list --installed

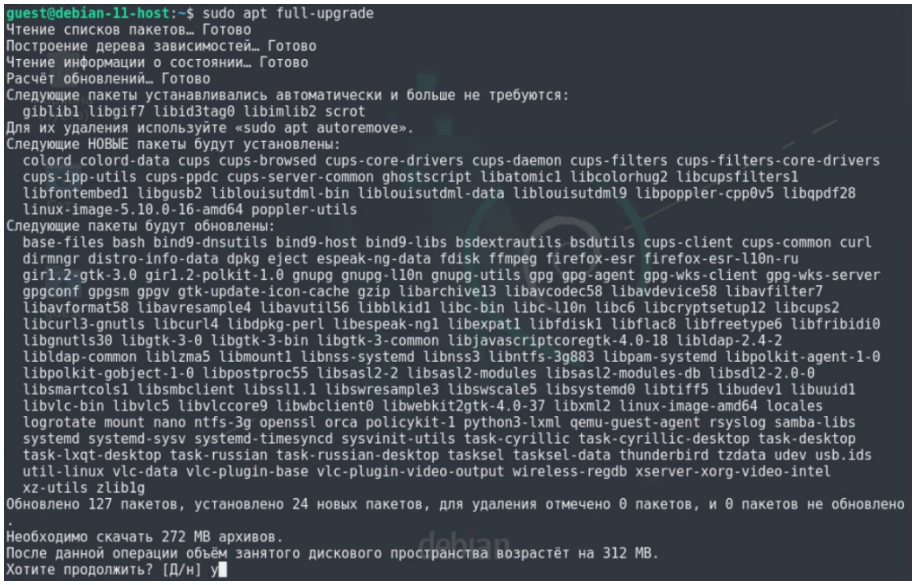


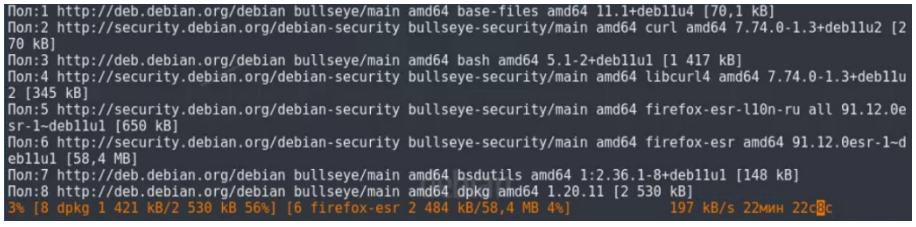
Или всех доступных:

guest@localhost:~$ sudo apt list –all-versions

Затем можно обновить все пакеты в системе:

guest@localhost:~$ sudo apt full-upgrade





Установка пакетов. Чтобы установить пакет apt используйте команду install, например, для установки программы Gimp используйте следующую команду:

guest@localhost:~$ sudo apt install gimp

```
guest@debian-11-host:~$ sudo apt install gimp
[sudo] пароль для guest:
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  giblib1 libgif7 libid3tag0 libimlib2 scrot
Для их удаления используйте «sudo apt autoremove».
Будут установлены следующие дополнительные пакеты:
  fonts-liberation graphviz libann0 libcdt5 libcgraph6 libgts-0.7-5 libgts-bin libgvc6 libgvpr2 libheif1
  liblab-gamut1 libmypaint-1.5-1 libmypaint-common libpathplan4 libwmf0.2-7
Предлагаемые пакеты:
  gimp-help-en | gimp-help gimp-data-extras graphviz-doc libwmf0.2-7-gtk
Следующие НОВЫЕ пакеты будут установлены:
  fonts-liberation gimp graphviz libann0 libcdt5 libcgraph6 libgts-0.7-5 libgts-bin libgvc6 libgvpr2
  libheif1 liblab-gamut1 libmypaint-1.5-1 libmypaint-common libpathplan4 libwmf0.2-7
Обновлено 0 пакетов, установлено 16 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 10,8 MB архивов.
После данной операции объём занятого дискового пространства возрастёт на 36,7 MB.
Хотите продолжить? [Д/н] y
Пол:1 http://deb.debian.org/debian bullseye/main amd64 fonts-liberation all 1:1.07.4-11 [828 kB]
Пол:2 http://deb.debian.org/debian bullseye/main amd64 libann0 amd64 1.1.2+doc-7 [25,3 kB]
Пол:3 http://deb.debian.org/debian bullseye/main amd64 libcdt5 amd64 2.42.2-5 [62,2 kB]
Пол:4 http://deb.debian.org/debian bullseye/main amd64 libcgraph6 amd64 2.42.2-5 [85,5 kB]
Пол:5 http://deb.debian.org/debian bullseye/main amd64 libgts-0.7-5 amd64 0.7.6+darcs121130-4+b1 [158 kB]
Пол:6 http://deb.debian.org/debian bullseye/main amd64 libpathplan4 amd64 2.42.2-5 [64,3 kB]
Пол:7 http://deb.debian.org/debian bullseye/main amd64 libgvc6 amd64 2.42.2-5 [695 kB]
Пол:8 http://deb.debian.org/debian bullseye/main amd64 libgvpr2 amd64 2.42.2-5 [212 kB]
Пол:9 http://deb.debian.org/debian bullseye/main amd64 liblab-gamut1 amd64 2.42.2-5 [221 kB]
Пол:10 http://deb.debian.org/debian bullseye/main amd64 graphviz amd64 2.42.2-5 [632 kB]
30% [10 graphviz 121 kB/632 kB 19%]                                          149 kB/s 55c
```

Если пакет установился неверно и вы хотите его переустановить, можно использовать опцию --reinstall:

guest@localhost:~$ sudo apt install gimp --reinstall

```
guest@debian-11-host:~$ sudo apt install gimp --reinstall
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  giblib1 libgif7 libid3tag0 libimlib2 scrot
Для их удаления используйте «sudo apt autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, переустановлено 1 пакетов, для удаления отмечено 0 пакетов,
и 0 пакетов не обновлено.
Необходимо скачать 7 177 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 0 B.
Пол:1 http://deb.debian.org/debian bullseye/main amd64 gimp amd64 2.10.22-4 [7 177 kB]
25% [1 gimp 2 201 kB/7 177 kB 31%]                                           215 kB/s 23c
```

Как видите, опции можно указывать не только перед командой, но и после неё, аналогично можно установить несколько пакетов сразу, например:

guest@localhost:~$ sudo apt install gimp inkscape -y

```
guest@debian-11-host:~$ sudo apt install gimp inkscape -y
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Уже установлен пакет gimp самой новой версии (2.10.22-4).
Следующие пакеты устанавливались автоматически и больше не требуются:
  giblib1 libgif7 libid3tag0 libimlib2 scrot
Для их удаления используйте «sudo apt autoremove».
Будут установлены следующие дополнительные пакеты:
  fig2dev gawk imagemagick imagemagick-6-common imagemagick-6.q16 libatkmm-1.6-1v5 libcairomm-1.0-1v5 libgc1
  libgdl-3-5 libgdl-3-common libglibmm-2.4-1v5 libgsl25 libgslcblas0 libgtkmm-3.0-1v5 libgtkspell3-3-0
  libimage-magick-perl libimage-magick-q16-perl libjxr-tools libjxr0 liblqr-1-0 libmagick++-6.q16-8
  libmagickcore-6.q16-6 libmagickcore-6.q16-6-extra libmagickwand-6.q16-6 libnetpbm10 libpangomm-1.4-1v5
  libpotrace0 libsigc++-2.0-0v5 libwmf-bin netpbm python3-numpy python3-scour
Предлагаемые пакеты:
  xfig gawk-doc imagemagick-doc autotrace cups-bsd | lpr | lprng enscript gnuplot grads hp2xx html2ps
  mplayer povray radiance texlive-base-bin ufraw-batch dia inkscape-tutorials libsvg-perl libxml-xql-perl
  pstoedit python3-uniconvertor ruby gsl-ref-psdoc | gsl-doc-pdf | gsl-doc-info | gsl-ref-html gcc gfortran
  python-numpy-doc python3-dev python3-numpy-dbg python3-pytest
Следующие НОВЫЕ пакеты будут установлены:
  fig2dev gawk imagemagick imagemagick-6-common imagemagick-6.q16 inkscape libatkmm-1.6-1v5
  libcairomm-1.0-1v5 libgc1 libgdl-3-5 libgdl-3-common libglibmm-2.4-1v5 libgsl25 libgslcblas0
  libgtkmm-3.0-1v5 libgtkspell3-3-0 libimage-magick-perl libimage-magick-q16-perl libjxr-tools libjxr0
  liblqr-1-0 libmagick++-6.q16-8 libmagickcore-6.q16-6 libmagickcore-6.q16-6-extra libmagickwand-6.q16-6
  libnetpbm10 libpangomm-1.4-1v5 libpotrace0 libsigc++-2.0-0v5 libwmf-bin netpbm python3-numpy python3-scour
Обновлено 0 пакетов, установлено 33 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 28,7 MB архивов.
После данной операции объём занятого дискового пространства возрастёт на 139 MB.
Пол:1 http://deb.debian.org/debian bullseye/main amd64 gawk amd64 1:5.1.0-1 [605 kB]
Пол:2 http://deb.debian.org/debian bullseye/main amd64 imagemagick-6-common all 8:6.9.11.60+dfsg-1.3 [211 kB]
3% [2 imagemagick-6-common 201 kB/211 kB 95%]
```

Чтобы не подтверждать установку вручную используем опцию -y. Допустим вы установили пакет с помощью dpkg и теперь хотите установить для него зависимости, запустите команду install без параметров с опцией -f или --fix-broken:

guest@localhost:~$ sudo apt install --fix-broken

Можно скачать deb-пакет в текущую папку без установки:

guest@localhost:~$ apt download gimp

```
guest@debian-11-host:~$ apt download gimp
guest@debian-11-host:~$ ls | grep gimp
gimp_2.10.22-4_amd64.deb
guest@debian-11-host:~$
```

Скачивать пакеты надо от имени обычного пользователя, иначе тогда они не будут доступны для работы с ними. Если вам нужно установить пакет из файла, то используйте утилиту dpkg:

guest@localhost:~$ sudo dpkg -i gimp_2.10.22-4_amd64.deb (Название скачанного файла может отличаться)

```
guest@debian-11-host:~$ sudo dpkg -i gimp_2.10.22-4_amd64.deb
[sudo] пароль для guest:
(Чтение базы данных … на данный момент установлено 216768 файлов и каталогов.)
Подготовка к распаковке gimp_2.10.22-4_amd64.deb …
Распаковывается gimp (2.10.22-4) на замену (2.10.22-4) …
Настраивается пакет gimp (2.10.22-4) …
Обрабатываются триггеры для desktop-file-utils (0.26-1) …
Обрабатываются триггеры для mailcap (3.69) …
Обрабатываются триггеры для man-db (2.9.4-2) …
guest@debian-11-host:~$
```

Чтобы установить определенную версию пакета просто укажите нужную версию после имени пакета через знак =, например:

guest@localhost:~$ sudo apt policy chromium (Чтобы узнать доступные версии пакетов)
guest@localhost:~$ sudo apt install chromium=103.0.5060.53-1~deb11u1 (Номер версии программы может отличаться)

```
guest@debian-11-host:~$ sudo apt policy chromium
chromium:
  Установлен: (отсутствует)
  Кандидат:   103.0.5060.134-1~deb11u1
  Таблица версий:
     103.0.5060.134-1~deb11u1 500
        500 http://security.debian.org/debian-security bullseye-security/main amd64 Packages
     103.0.5060.53-1~deb11u1 500
        500 http://deb.debian.org/debian bullseye/main amd64 Packages
guest@debian-11-host:~$ sudo apt install chromium=103.0.5060.134-1~deb11u1
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  giblib1 libgif7 libid3tag0 libimlib2 libmypaint-1.5-1 libmypaint-common scrot
Для их удаления используйте «sudo apt autoremove».
Будут установлены следующие дополнительные пакеты:
  chromium-common chromium-sandbox libjsoncpp24 libu2f-udev sse3-support
Предлагаемые пакеты:
  chromium-l10n chromium-shell chromium-driver
Следующие НОВЫЕ пакеты будут установлены:
  chromium chromium-common chromium-sandbox libjsoncpp24 libu2f-udev sse3-support
Обновлено 0 пакетов, установлено 6 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
Необходимо скачать 69,2 MB архивов.
После данной операции объём занятого дискового пространства возрастёт на 228 MB.
Хотите продолжить? [Д/н] у
```

Если вы не знаете как точно называется пакет, можно выполнить поиск пакетов apt с помощью команды search:

guest@localhost:~$ sudo apt search gimp

```
mate-hud/stable 19.10.1-3 all
  Run menubar commands, much like the Unity 7 HUD

nip2/stable 8.7.1-2 amd64
  spreadsheet-like graphical image manipulation tool

potrace/stable 1.16-2 amd64
  utility to transform bitmaps into vector graphics

printer-driver-gutenprint/stable 5.3.3-5 amd64
  драйверы принтеров для CUPS

python3-odf/stable 1.4.1-1 all
  Python3 API to manipulate OpenDocument files

rabbit/stable 3.0.0-4 all
  presentation tool using RD, a simple text format

sane/stable 1.0.14-16 amd64
  программы для работы со сканером

vim-syntax-gtk/stable 20110314-1.1 all
  Syntax files to highlight GTK+ keywords in vim

xsane/stable,now 0.999-10 amd64 [установлен, автоматически]
  программа с графическим интерфейсом для работы со сканером

xsane-common/stable,now 0.999-10 all [установлен, автоматически]
  архитектурно-независимые файлы xsane

xzgv/stable 0.9.2-2 amd64
  Picture viewer for X with a thumbnail-based selector
```

Удаление пакетов. Чтобы удалить ненужный пакет используйте команду remove:

guest@localhost:~$ sudo apt remove gimp

```
guest@debian-11-host:~$ sudo apt remove gimp
[sudo] пароль для guest:
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  giblib1 libgif7 libid3tag0 libimlib2 libmypaint-1.5-1 libmypaint-common scrot
Для их удаления используйте «sudo apt autoremove».
Следующие пакеты будут УДАЛЕНЫ:
  gimp
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 1 пакетов, и 0 пакетов не обновлено.
После данной операции объём занятого дискового пространства уменьшится на 22,8 MB.
Хотите продолжить? [Д/н] y
(Чтение базы данных … на данный момент установлено 217055 файлов и каталогов.)
Удаляется gimp (2.10.22-4) …
Обрабатываются триггеры для man-db (2.9.4-2) …
Обрабатываются триггеры для mailcap (3.69) …
Обрабатываются триггеры для desktop-file-utils (0.26-1) …
guest@debian-11-host:~$
```

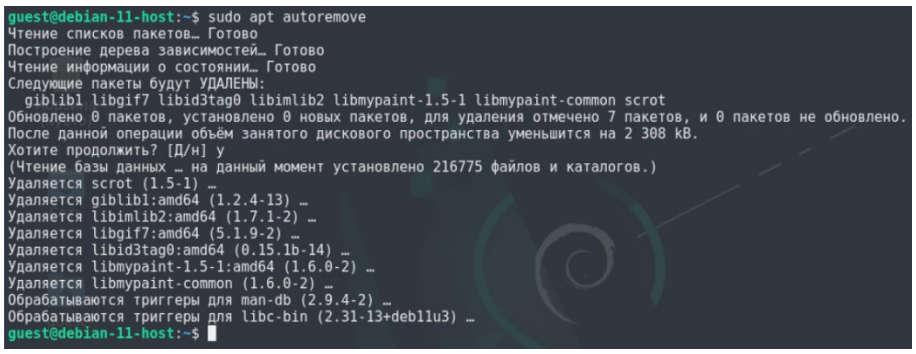
Однако если вы изменяли какие-либо конфигурационные файлы из состава пакета, то при таком способе удаления они останутся в системе, чтобы удалить всё полностью используйте команду purge:

guest@localhost:~$ sudo apt purge gimp

```
guest@debian-11-host:~$ sudo apt purge gimp
Чтение списков пакетов… Готово
Построение дерева зависимостей… Готово
Чтение информации о состоянии… Готово
Пакет «gimp» не установлен, поэтому не может быть удалён
Следующие пакеты устанавливались автоматически и больше не требуются:
  giblib1 libgif7 libid3tag0 libimlib2 libmypaint-1.5-1 libmypaint-common scrot
Для их удаления используйте «sudo apt autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
```

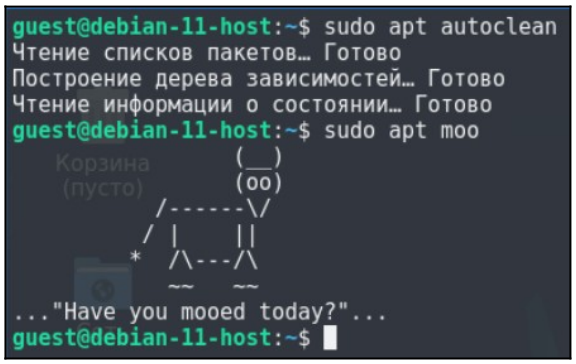
Чтобы удалить лишние пакеты, которые в системе больше не нужны выполните команду autoremove:

guest@localhost:~$ sudo apt autoremove



А для очистки кэша пакетов – autoclean:

guest@localhost:~$ sudo apt autoclean



Графический интерфейс apt

Утилита aptitude – это пакетный менеджер для Ubuntu и Debian. Утилита позволяет делать всё то же самое, что и apt, но только кроме командной строки у неё есть псевдографический интерфейс на основе ncurses[1]. С помощью этого инструмента вы можете устанавливать и удалять программы, обновлять систему, смотреть информацию о пакетах и их зависимости, выполнять поиск и многое другое.

Этот пакетный менеджер в последнее время стал не очень популярным, потому что делает практически всё то же, что и apt и тоже из командной строки, но иногда им пользуются. У этой программе есть псевдографический интерфейс, который делает манипуляции с пакетами гораздо удобнее. Его мы и рассмотрим в этом пункте.

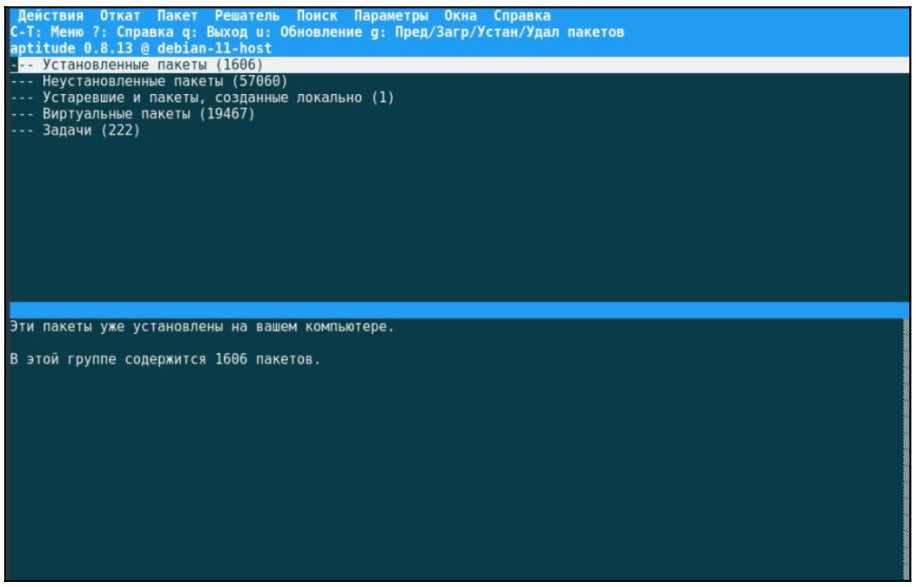
Если программа ещё не установлена в вашей системе, вы можете её установить выполнив такую команду:

guest@localhost:~$ sudo apt install aptitude

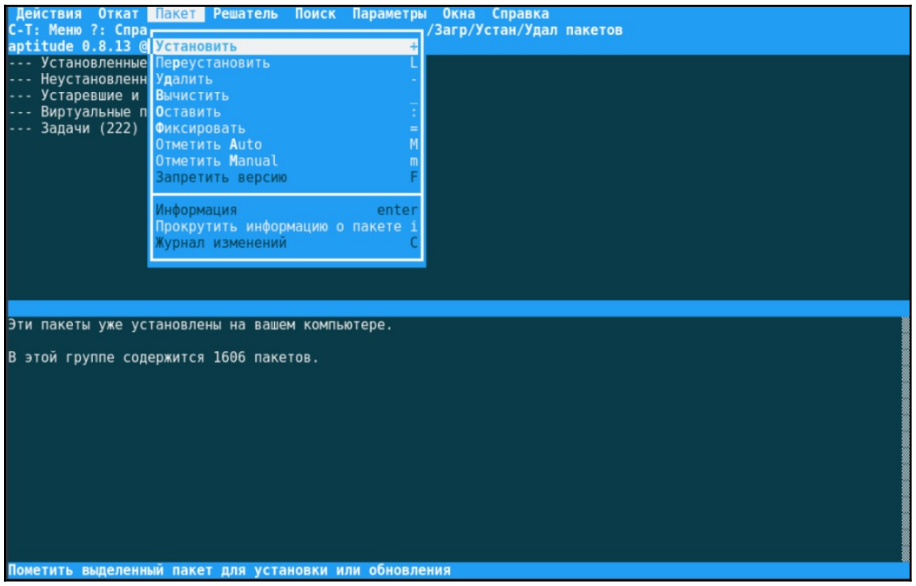
Запуск и интерфейс. Для запуска программы с псевдографическим интерфейсом надо запустить её без параметров:

guest@localhost:~$ sudo aptitude

[1] Библиотека ncurses предназначена для управления вводом выводом на терминал. Она позволяет программисту не беспокоиться об аппаратных различиях терминалов и писать достаточно переносимый код.

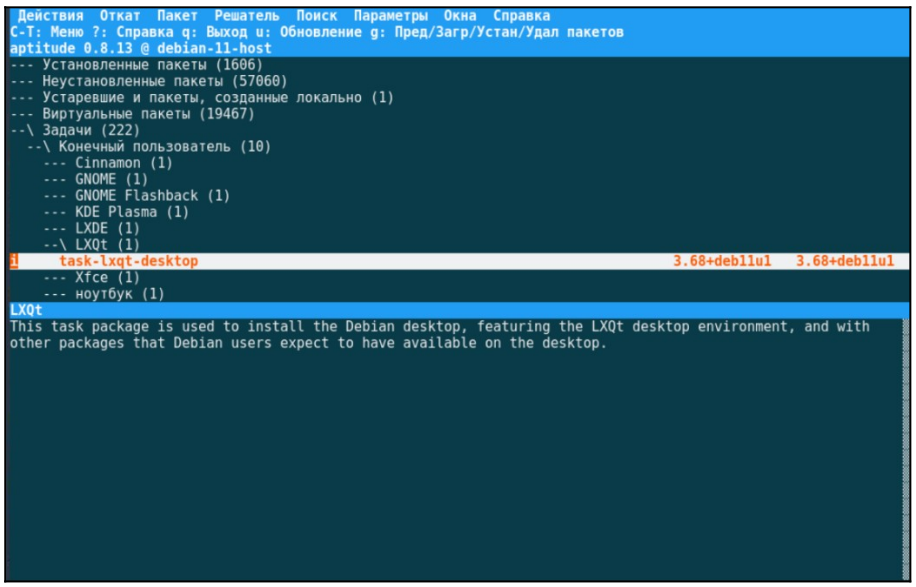


Окно программы разделено на несколько частей: меню, панель вкладок, основная рабочая область и область уведомлений. Вы можете перемещаться по рабочей области с помощью клавиш стрелок вверх/вниз и вправо/влево. Для того чтобы открыть меню нажмите <Ctrl>+<T> и используйте те же стрелки для перемещения по вкладках и пунктах:

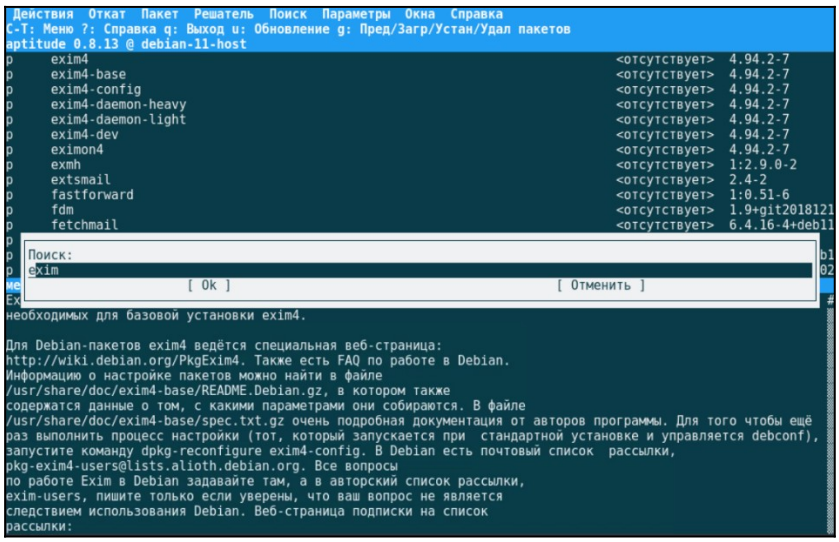


А теперь давайте поговорим про поиск и установку пакетов.

Поиск пакетов. По умолчанию в рабочей области расположены категории программ. Вы можете открывать их и искать пакеты там просто листая их с помощью стрелок. Например, можно открыть раздел Задачи и найти там метапакет task-lqxt-desktop, который включает в себя пакет задач, используемый для установки рабочего стола Debian с окружением рабочего стола LXQt и другими пакетами, которые пользователи Debian ожидают иметь на рабочем столе:

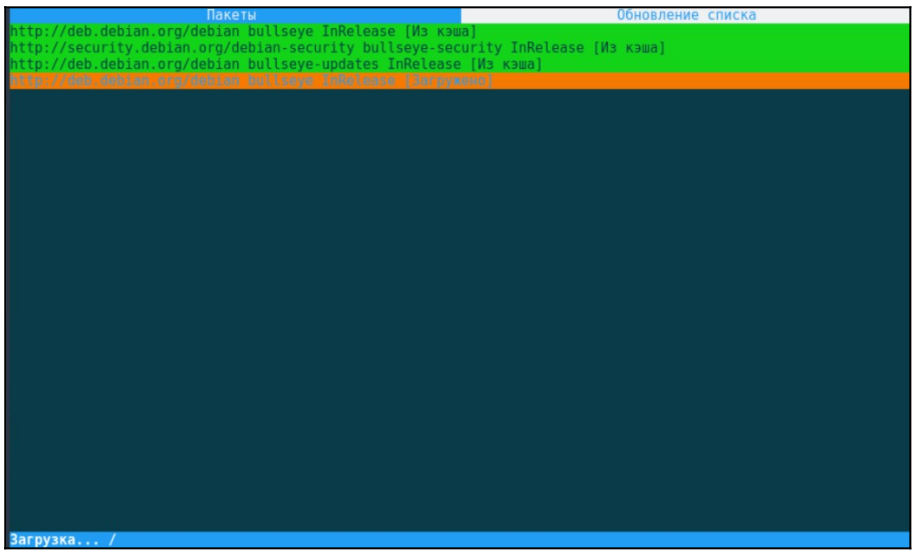


Аналогично можно искать пакеты в других разделах, если вы знаете где они находятся. Или можно выполнять поиск пакетов с помощью горячей клавиши /. Например, давайте найдём пакет exim:

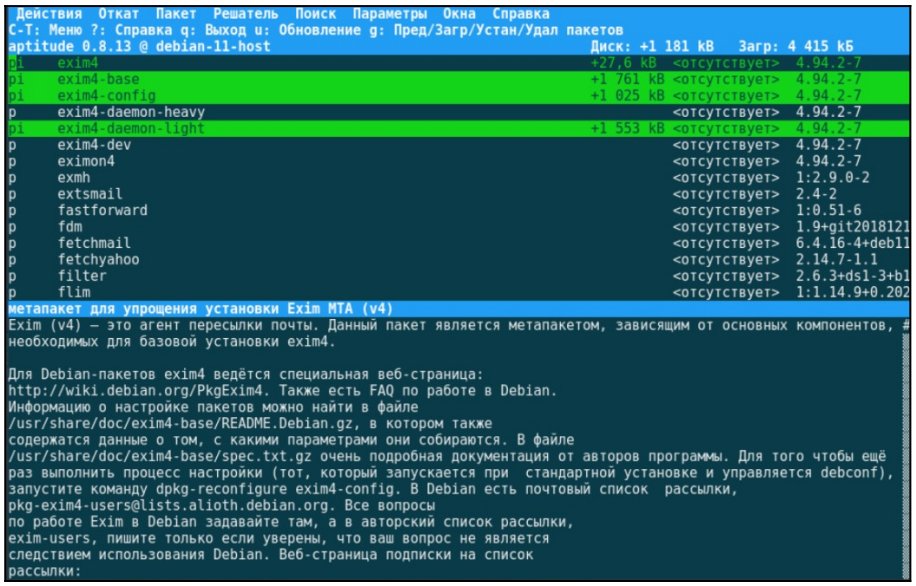


Для того чтобы найти следующее вхождение пакета можно использовать клавишу n.

Установка пакетов. Перед установкой пакета надо обновить списки пакетов из репозиториев. Для этого нажмите кнопку u:

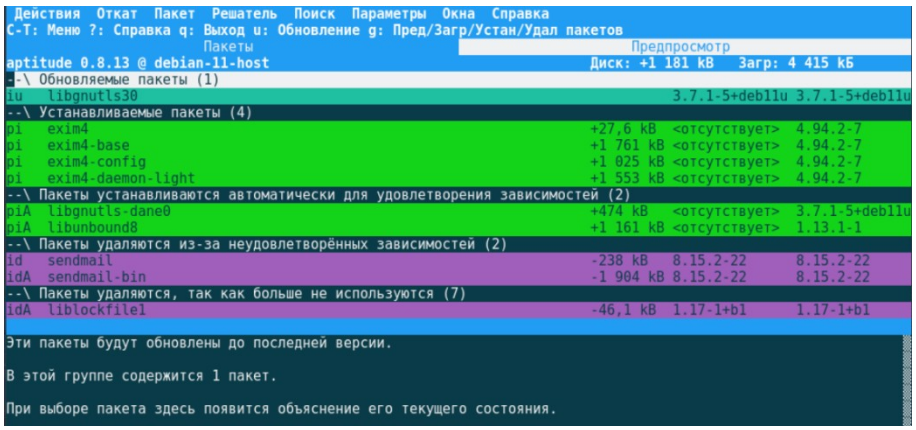


Затем найдите пакет одним из выше предложенных способов и поставьте на него курсор. Например, давайте установим почтовый сервер Exim. Для того чтобы отметить пакет для установки когда курсор находится на нём наберите на клавиатуре символ «+» для этого надо нажать <Shift> + <=>. После этого пакет будет выделен зелёным:



Если этот пакет конфликтует с каким-либо из уже установленных пакетов внизу окна появится красная строка с описанием проблемы и вариантами решения. Вы можете переключаться между ними с помощью клавиш <,> и <.>. А для применения решения нажмите <!>.

После того как все пакеты, которые надо установить отмечены, необходимо применить изменения. Для этого нажмите кнопку <g>. Откроется окно с предпросмотром изменения:

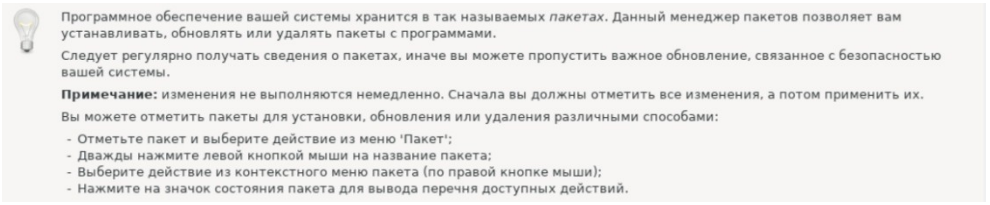


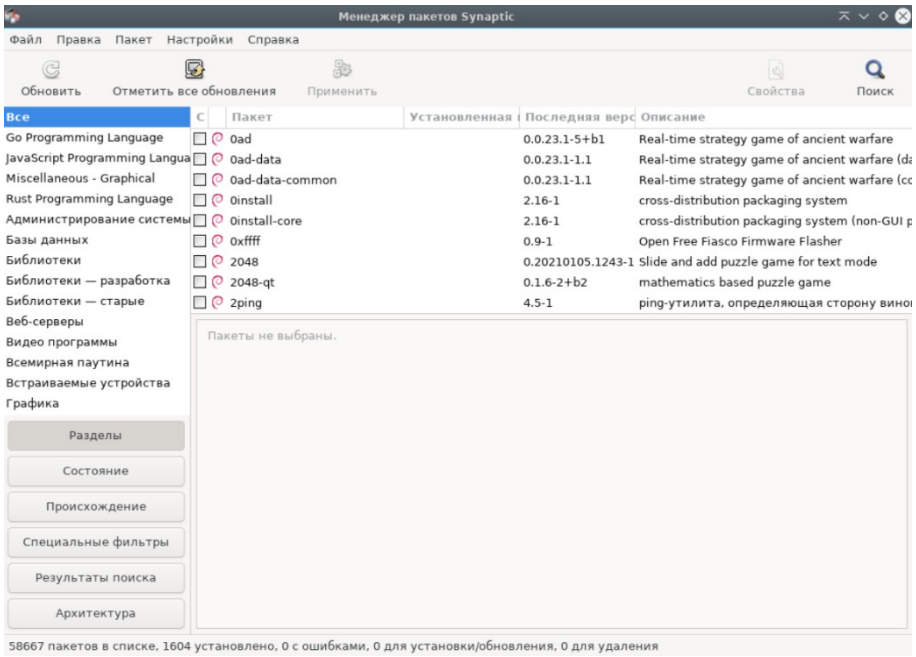
Если вы передумали устанавливать один из пакетов, нажмите кнопку <->. Аналогичным образом пакеты отмечаются для удаления. Когда всё будет готово, нажмите ещё раз <g> чтобы выполнить установку. Удаление пакетов выполняется аналогично.

Мы рассмотрели что делает утилита aptitude, а также как ею пользоваться для установки пакетов. Как видите, всё довольно удобно, а её псевдографический интерфейс достаточно похож на Synaptic по процессу установки пакетов.

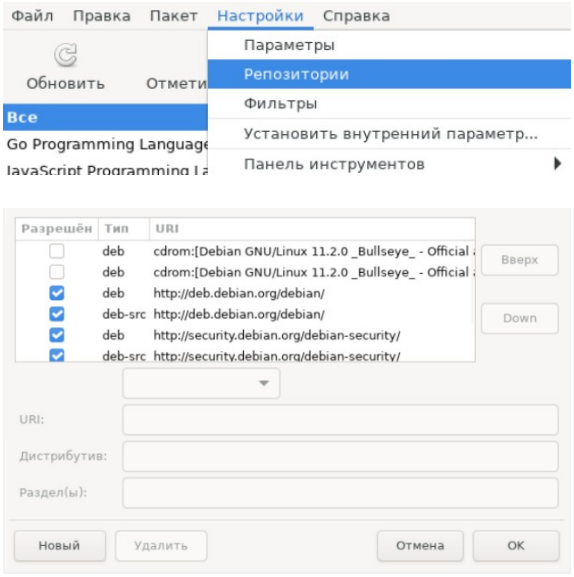
Synaptic – это графическая программа, позволяющая управлять пакетами в Ubuntu и Debian. Предоставляет большую функциональность чем утилита aptitude и ориентирован на простоту использования. Первым делом его нужно установить.

guest@localhost:~# sudo apt install synaptic





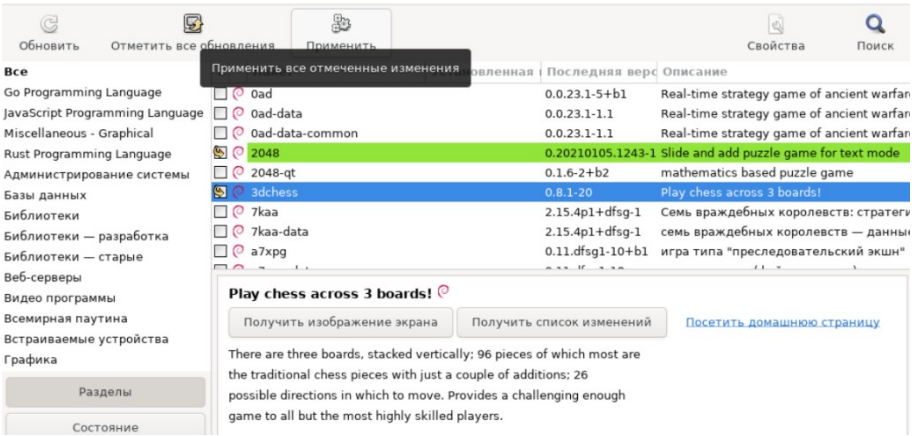
Настройка репозитория выполняется следующим образом:



По умолчанию, Debian использует только раздел main для каждого репозитория. При необходимости, можно добавить разделы contrib и non-free. Обратите внимания, что обновления безопасности предоставляются только для раздела main.

Просмотр, установка и удаление выполняется следующим образом:

Synaptic показывает все доступные пакеты – и отмечает каждый как установленный или доступный для инсталляции. Также можно находить и просматривать пакеты, отмечать которые вы хотите установить (или удалить), нажав на окошко (или щёлкнув правой кнопкой мыши на пакете), затем нажмите «Применить» для внесения изменений.



ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: Установка ОС Linux – дистрибутива Arch Linux

Цель работы: Выполнить инсталляцию дистрибутива Arch Linux:

– Проверка подключения к Интернету;

– Установка раскладки клавиатуры;

– Синхронизация системных часов;

– Разметка дисков;

– Форматирование разделов;

– Монтирование разделов;

– Выбор зеркал ;

– Установка базовой системы Arch Linux;

– Создание файла fstab;

– Конфигурация системы Arch Linux;

– Установка языка системы;

– Установка часового пояса;

– Настройка имени хоста;

– Установка пароля root;

– Установка загрузчика GRUB;

– Перезагрузка и проверка авторизации в установленную ОС Arch Linux;

– Смена шрифта по умолчанию;

– Настройка NetworkManager;

– Создание нового пользователя и добавление пользователя в sudo.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

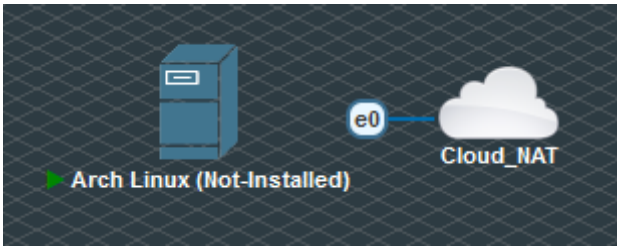
Arch Linux – это независимый дистрибутив GNU/Linux для опытных пользователей, оптимизированный для архитектуры x86-64, который стремится предоставить последние «новейшие» версии программ, следуя модели «Rolling Release»[2].

[2] Rolling Release – это понятие в разработке ПО, характеризующее метод обновления последнего. Наиболее часто употребляется относительно дистрибутивов Linux, и противопоставляется классической системе периодически выходящих версий, содержащих, как правило, уже немного устаревшие версии программ, которые поддерживаются определенное время после выхода выпуска (в большинстве дистрибутивов срок поддержки примерно равен году кроме Long Term Support (LTS) выпусков, имеющих больший срок поддержки). В отличие от нее, система плавающих выпусков позволяет пользователю всегда иметь последние версии устанавливаемых программ, избавляя его от необходимости периодической переустановки системы.

По умолчанию пользователю предоставляется минималистичная базовая система, в которую пользователь может добавить то, что ему требуется. Для установки, удаления и обновления пакетов используется пакетный менеджер Pacman.

В официальных репозиториях Arch Linux содержится преимущественно свободное ПО, однако дистрибутив не одобрен Фондом свободного программного обеспечения, так как Arch придерживается лояльной политики в отношении несвободного ПО. Проект GNU рекомендует к установке основанный на Arch Linux дистрибутив Parabola. В настоящее время в официальных репозиториях насчитывается более 12 000 пакетов, в AUR более 83 000.

Перед началом установки необходимо подключить виртуальную машину таким образом, как указано на схеме.



Чтобы установить дистрибутив Arch Linux на виртуальной машине, следуйте инструкции, представленной ниже.

> *Если Arch Linux отказывается запускаться, то попробуйте выставить ему оперативной памяти больше 1024 Гбайт и затем стереть конфигурацию виртуальной машины полностью (кнопка «Wipe») и перезапустить его.*

Проверка подключения к Интернету

Перед началом установки Arch Linux, убедитесь, что вы получили IP-адрес по DHCP. Для этого введите следующую команду:

root@archiso ~ # ip a

```
root@archiso ~ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 50:78:2b:00:50:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.137.41/24 metric 100 brd 10.0.137.255 scope global dynamic ens3
       valid_lft 2865sec preferred_lft 2865sec
    inet6 fe80::5278:2bff:fe00:5000/64 scope link
       valid_lft forever preferred_lft forever
```

root@archiso ~ # ip link (Проверка обнаружения сетевого интерфейса)

```
root@archiso ~ # ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 50:78:2b:00:50:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
```

root@archiso ~ # ping ya.ru
[ctrl+c]

Установка раскладки клавиатуры

По умолчанию используется *раскладка консоли US*. Чтобы посмотреть список доступных раскладок, запустите:

root@archiso ~ # ls /usr/share/kbd/keymaps/**/*.map.gz

Чтобы выбрать раскладку, передайте имя соответствующего файла команде loadkeys, не указывая полного пути и расширения. Например, чтобы выбрать русскую раскладку, запустите:

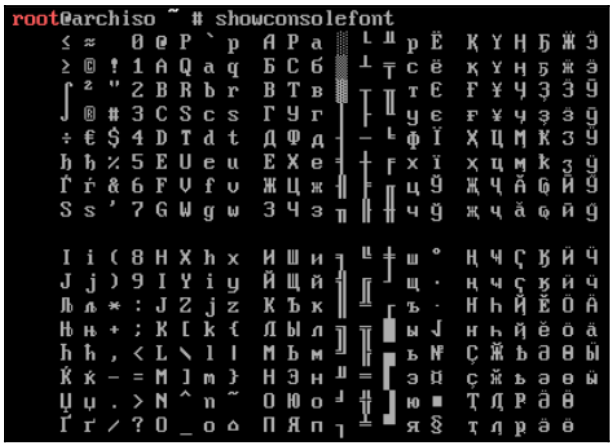
root@archiso ~ # loadkeys ru

Чтобы посмотреть список доступных шрифтов, введите:

root@archiso ~ # ls /usr/share/kbd/consolefonts/

По умолчанию, в Arch Linux стоит шрифт, который не отображает кириллицу. Чтобы это исправить необходимо выбрать необходимый шрифт, поддерживающий кириллические знаки, например, UniCyrExt_8x16:

root@archiso ~ # setfont UniCyrExt_8x16 (Временно меняет шрифт, если команде передано имя шрифта)
root@archiso ~ # showconsolefont (Показывает таблицу символов шрифта)



Проверьте, что вы можете сменить раскладку клавиатуры и увидеть набранный текст:

root@archiso ~ # [ctrl+shift]
root@archiso ~ # Привет, мир!
root@archiso ~ # [ctrl+shift]
root@archiso ~ # Hello, world!

Синхронизация системных часов

Чтобы удостовериться, что время задано правильно, используйте timedatectl:

root@archiso ~ # timedatectl set-ntp true

Для проверки статуса службы используйте timedatectl status:

root@archiso ~ # timedatectl status

```
root@archiso ~ # timedatectl status
               Local time: Wed 2022-05-11 04:26:01 UTC
           Universal time: Wed 2022-05-11 04:26:01 UTC
                 RTC time: Wed 2022-05-11 04:26:01
                Time zone: UTC (UTC, +0000)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
root@archiso ~ #
```

Разметка дисков

Когда запущенная система распознает накопители, они становятся доступны как блочные устройства, например, /dev/sda, /dev/vda, /dev/nvme0n1 или /dev/mmcblk0. Чтобы посмотреть их список, используйте lsblk или fdisk:

root@archiso ~ # fdisk -l

```
root@archiso ~ # fdisk -l
[ 3688.417036] I/O error, dev fd0, sector 0 op 0x0:(READ) flags 0x0 phys_seg 1 prio class 0
[ 3688.446828] I/O error, dev fd0, sector 0 op 0x0:(READ) flags 0x0 phys_seg 1 prio class 0


Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop0: 657.39 MiB, 689319936 bytes, 1346328 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Результаты, оканчивающиеся на rom, loop и airoot, можно игнорировать.

На выбранном накопителе должны присутствовать следующие разделы:

– Раздел для корневого каталога «/»;

– Для загрузки в режиме UEFI также необходим *системный раздел EFI*.

Для изменения таблицы разделов используйте fdisk или parted:

root@archiso ~ # fdisk /dev/vda

Если эта команда не срабатывает и в ответ консоль пишет, что такой диск не найден, то попробуйте ввести команду fdisk /dev/sda.

Вы должны создать раздел boot в начале вашего диска. На этом разделе размещаются файлы настройки и модули загрузчика, которые считываются при старте GRUB и загружают операционную систему. Когда вы введете n, он попросит вас выбрать номер диска, введите 1. Оставьте размер блока по умолчанию, когда он запросит размер раздела, введите +512M. Порядок выполняемых действий изображен ниже.

```
Welcome to fdisk (util-linux 2.37.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x0591878f.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-62914559, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-62914559, default 62914559): +512M

Created a new partition 1 of type 'Linux' and of size 512 MiB.
```

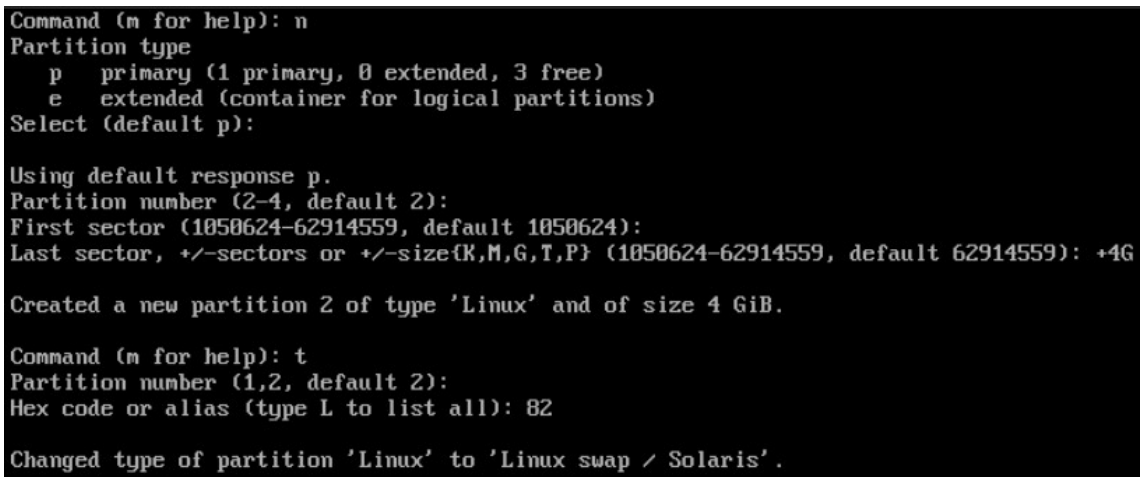
При установке Linux обычной практикой является разбиение дисков на разделы root, swap и home по отдельности. В нашем случае у нас будет корневой раздел и раздел swap, без отдельного home.

Создадите раздел swap в начале вашего диска. Введите n, затем введите 2. Оставьте размер блока по умолчанию, когда он запросит размер раздела, введите +4G.

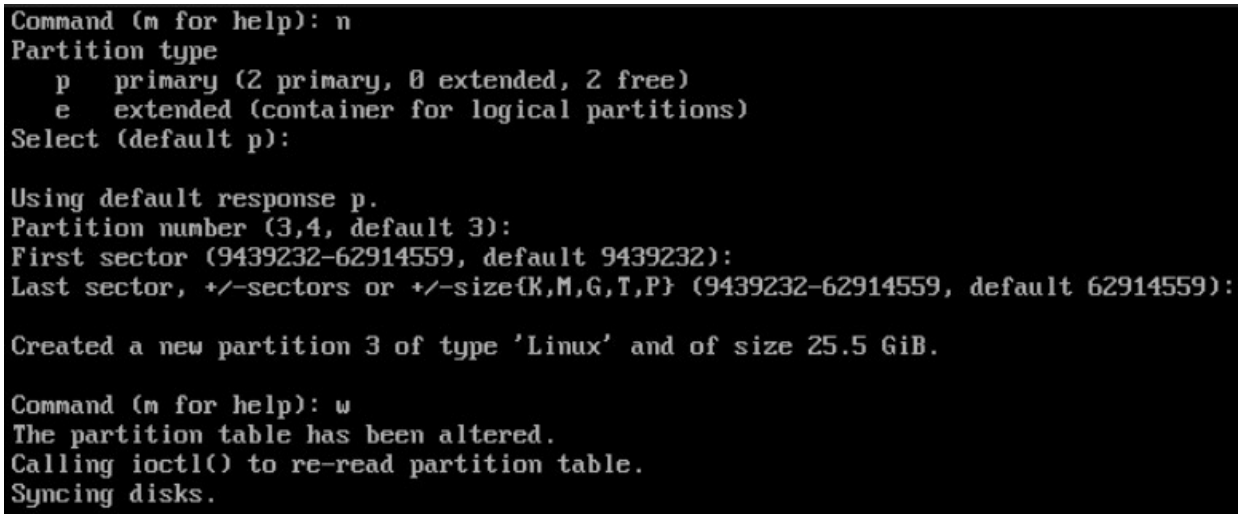
Обратите внимание, что одним из важных шагов является изменение типа разделов. Все типы разделов и их HEX-коды изображен ниже.

```
Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): L

00 Empty            24 NEC DOS          81 Minix / old Lin  bf Solaris
01 FAT12            27 Hidden NTFS Win  82 Linux swap / So  c1 DRDOS/sec (FAT-
02 XENIX root       39 Plan 9           83 Linux            c4 DRDOS/sec (FAT-
03 XENIX usr        3c PartitionMagic   84 OS/2 hidden or   c6 DRDOS/sec (FAT-
04 FAT16 <32M       40 Venix 80286      85 Linux extended   c7 Syrinx
05 Extended         41 PPC PReP Boot    86 NTFS volume set  da Non-FS data
06 FAT16            42 SFS              87 NTFS volume set  db CP/M / CTOS / .
07 HPFS/NTFS/exFAT  4d QNX4.x           88 Linux plaintext  de Dell Utility
08 AIX              4e QNX4.x 2nd part  8e Linux LVM        df BootIt
09 AIX bootable     4f QNX4.x 3rd part  93 Amoeba           e1 DOS access
0a OS/2 Boot Manag  50 OnTrack DM       94 Amoeba BBT       e3 DOS R/O
0b W95 FAT32        51 OnTrack DM6 Aux  9f BSD/OS           e4 SpeedStor
0c W95 FAT32 (LBA)  52 CP/M             a0 IBM Thinkpad hi  ea Linux extended
0e W95 FAT16 (LBA)  53 OnTrack DM6 Aux  a5 FreeBSD          eb BeOS fs
0f W95 Ext'd (LBA)  54 OnTrackDM6       a6 OpenBSD          ee GPT
10 OPUS             55 EZ-Drive         a7 NeXTSTEP         ef EFI (FAT-12/16/
11 Hidden FAT12     56 Golden Bow       a8 Darwin UFS       f0 Linux/PA-RISC b
12 Compaq diagnost  5c Priam Edisk      a9 NetBSD           f1 SpeedStor
14 Hidden FAT16 <3  61 SpeedStor        ab Darwin boot      f4 SpeedStor
16 Hidden FAT16     63 GNU HURD or Sys  af HFS / HFS+       f2 DOS secondary
17 Hidden HPFS/NTF  64 Novell Netware   b7 BSDI fs          fb VMware VMFS
18 AST SmartSleep   65 Novell Netware   b8 BSDI swap        fc VMware VMKCORE
1b Hidden W95 FAT3  70 DiskSecure Mult  bb Boot Wizard hid  fd Linux raid auto
1c Hidden W95 FAT3  75 PC/IX            bc Acronis FAT32 L  fe LANstep
1e Hidden W95 FAT1  80 Old Minix        be Solaris boot     ff BBT
```

Введите t, чтобы изменить тип. Введите L, чтобы увидеть все доступные типы разделов, а затем введите соответствующий номер (82) в систему EFI. Порядок выполняемых действий указан ниже.



Наконец, создадите корневой раздел в начале вашего диска. Введите n, затем введите 3. Оставьте размер блока по умолчанию, когда он запросит размер раздела, оставьте поле пустым и нажмите <Enter>. Порядок действий указан ниже.



Форматирование разделов

Когда новые разделы созданы, каждый из них необходимо отформатировать в подходящую файловую систему.

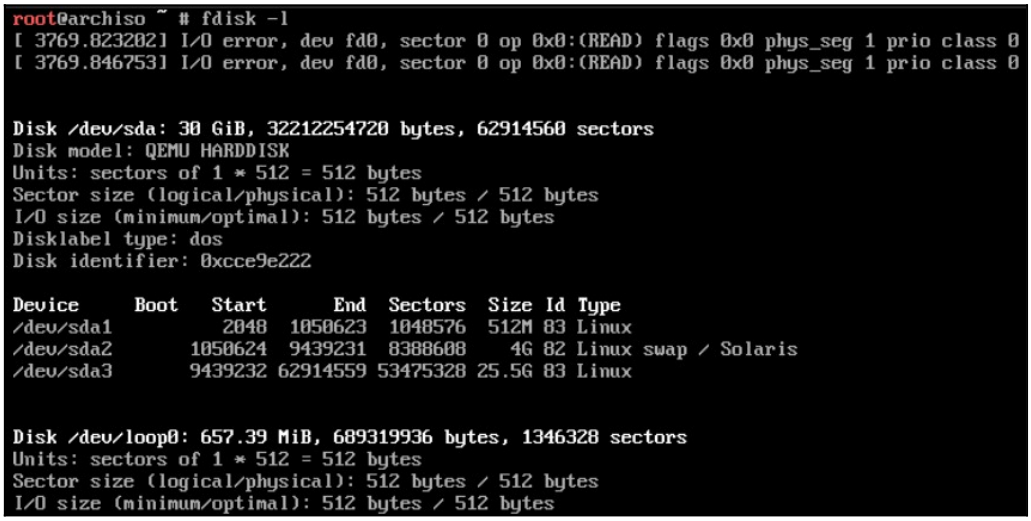
Например, чтобы отформатировать /dev/*корневой_раздел* в ext4, выполните следующую команду: *# mkfs.ext4 /dev/корневой_раздел*

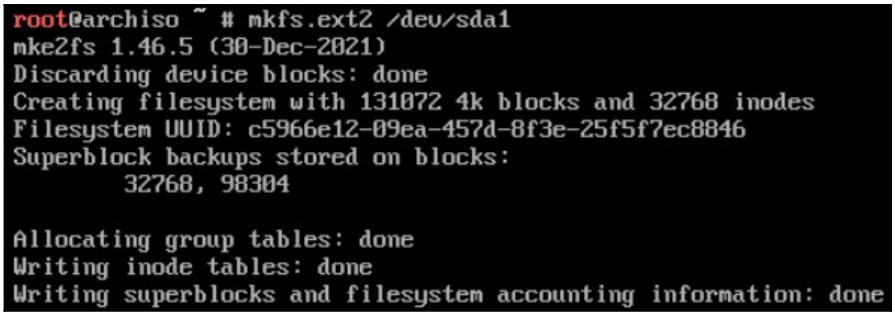
Если вы создали раздел для подкачки (swap), инициализируйте его с помощью утилиты mkswap: *# mkswap /dev/раздел_подкачки*

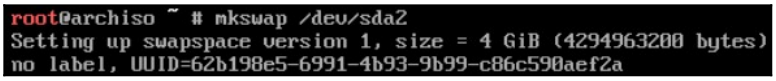
Если вы создали системный раздел EFI, отформатируйте его в FAT32 с помощью mkfs.fat: *# mkfs.fat -F 32 /dev/системный_раздел_efi*

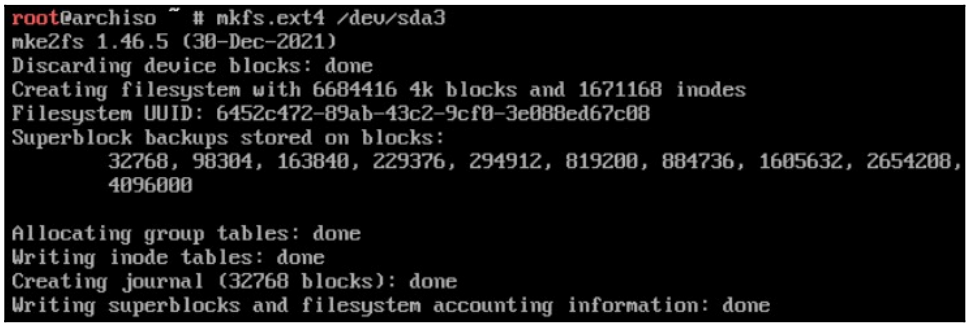
Выполним форматирование наших разделов:

root@archiso ~ # fdisk -l
root@archiso ~ # mkfs.ext2 /dev/vda1
root@archiso ~ # mkswap /dev/vda2
root@archiso ~ # mkfs.ext4 /dev/vda3









Монтирование разделов

После того, как вы отформатировали разделы, используйте команду mount для их монтирования. Корневой раздел «/» должен быть смонтирован в каталоге /mnt.

Если у вас есть дополнительные разделы, то их необходимо смонтировать в соответствующие каталоги на /mnt. Например: раздел /boot должен быть смонтирован на /mnt/boot.

Кроме того, вам нужно будет проинициализировать раздел подкачки (раздел swap).

Выполним монтирование наших разделов:

root@archiso ~ # mount /dev/vda3 /mnt
root@archiso ~ # mkdir /mnt/boot
root@archiso ~ # mount /dev/vda1 /mnt/boot
root@archiso ~ # swapon /dev/vda2

Выбор зеркал

Пакеты Arch Linux должны быть загружены с зеркальных серверов в Интернете. Зеркала определены в /etc/pacman.d/mirrorlist. Зеркало, которое возглавляет список, получает приоритет для загрузки пакета. Это может подходить не для всех ситуаций, так как необходимо учитывать свое географическое положение, доступность зеркал и возможность рассинхронизации.

Используйте reflector, чтобы получить последнее зеркало из состояния зеркала Arch Linux, отфильтровать актуальные зеркала, отсортировать их по скорости и обновить файл списка зеркал.

Сперва необходимо сделать резервную копию существующего списка зеркал:

root@archiso ~ # cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.old

Затем обновите файл списка зеркал с 10 зеркалами по скорости загрузки:

root@archiso ~ # reflector --verbose --latest 10 --sort rate --save /etc/pacman.d/mirrorlist

Установка базовой системы Arch Linux

Теперь пришло время установить базовую систему Arch Linux:

root@archiso ~ # pacman -Syu
root@archiso ~ # pacstrap /mnt/ base linux linux-firmware net-tools networkmanager openssh nano

Создание файла fstab

Файл fstab – это текстовый файл, который содержит информацию о различных файловых системах и устройствах хранения информации в вашем компьютере. Это всего лишь один файл, определяющий, как диск и/или раздел будут использоваться и как будут встроены в остальную систему.

После базовой установки сгенерируйте файл fstab для системы с помощью команды genfstab:

root@archiso ~ # genfstab -U /mnt >> /mnt/etc/fstab

Проверьте записи fstab с помощью приведенной ниже команды:

root@archiso ~ # cat /mnt/etc/fstab

```
root@archiso ~ # cat /mnt/etc/fstab
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
# /dev/sda3
UUID=1dce9633-ca99-4375-b237-fce673d6f1ee       /               ext4            rw,relatime     0 1

# /dev/sda1
UUID=c5966e12-09ea-457d-8f3e-25f5f7ec8846       /boot           ext2            rw,relatime     0 2

# /dev/sda2
UUID=58f73ee0-b887-4096-b1c5-3105470ee858       none            swap            defaults        0 0
```

Конфигурация системы Arch Linux

Для дальнейшей настройки Arch Linux вы должны выполнить chroot[3] для новой системы. chroot изменяет корневой каталог для текущего запущенного процесса и его потомков.

root@archiso ~ # arch-chroot /mnt

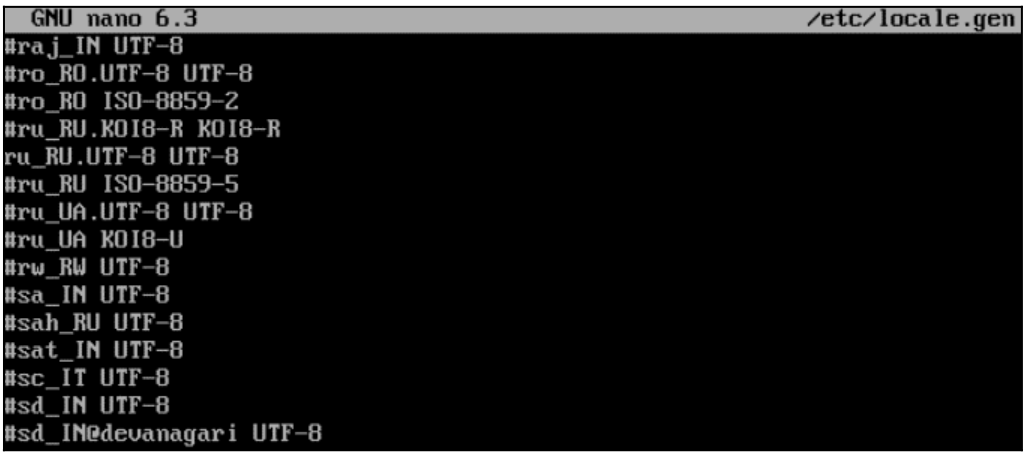
> *Если вдруг у вас произошла какая-либо внештатная ситуация и во время работы у выключилась виртуальная машина, то в дальнейшем вы можете снова причрутиться с диска к системе и продолжить установку ОС. Для этого надо выполнить по инструкции монтирование разделов, а затем снова выполнить команду arch-chroot /mnt.*

Установка языка системы

Вы можете настроить системный язык, раскомментировав нужные языки в файле /etc/locale.gen:

[root@archiso /]# nano /etc/locale.gen

Раскомментируйте строку ru_RU.UTF-8 UTF-8 для русского языка, а затем сгенерируйте локали. Затем установите переменную LANG в файле /etc/locale.conf.



[root@archiso /]# locale-gen
[root@archiso /]# echo "LANG=ru_RU.UTF-8" > /etc/locale.conf

Установка часового пояса

Теперь нужно настроить системный часовой пояс, создав символическую ссылку вашего часового пояса на файл /etc/localtime.

Все доступные часовые пояса находятся в каталоге /usr/share/zoneinfo:

[root@archiso /]# ls /usr/share/zoneinfo/Asia
[root@archiso /]# ln -sf /usr/share/zoneinfo/Asia/Vladivostok /etc/localtime (создание символьной ссылки на файл)

Также установите аппаратные часы в формате UTC:

---

[3] chroot – это операция изменения корневого каталога диска для запущенного процесса и его дочерних процессов. Программа, запущенная в таком окружении, не может получить доступ к файлам вне нового корневого каталога. Это измененное окружение называется chroot jail.

[root@archiso /]# hwclock --systohc --utc

Настройка имени хоста

Поместите системное имя хоста в файл /etc/hostname:

[root@archiso /]# echo "localhost" > /etc/hostname

Установка пароля root

Используйте команду passwd в терминале, чтобы установить пароль root:

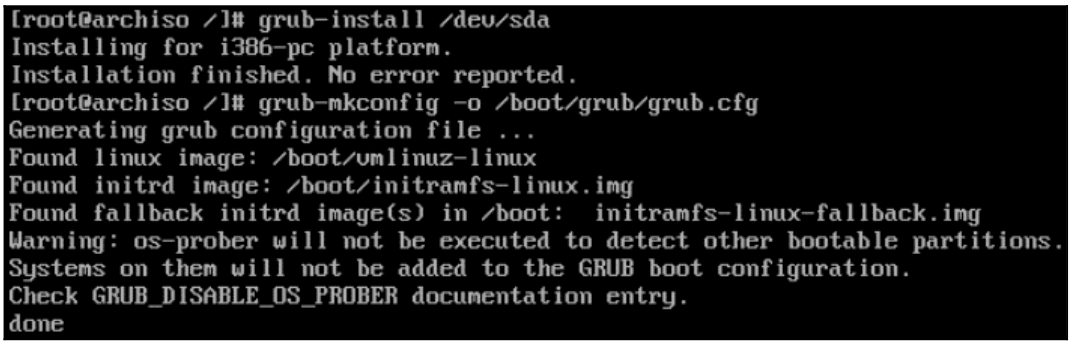
[root@archiso /]# passwd
New password: guest
Retype new password: toor

passwd: password updated successfully

Установка загрузчика GRUB

Arch Linux требует загрузчика для загрузки системы. Вы можете установить загрузчик GRUB, используя приведенные ниже команды:

[root@archiso /]# pacman -S grub
[root@archiso /]# grub-install /dev/vda
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg

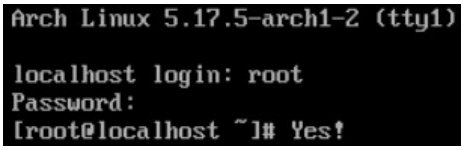


Перезагрузка и проверка авторизации в установленную ОС Arch Linux

Выйдите из системы chroot и перезагрузитесь:

[root@archiso /]# exit
root@archiso ~ # reboot

После завершения перезагрузки вы получите приглашение для входа в Arch Linux. Войдите в систему как пользователь root и используйте для входа пароль, который вы установили во время установки операционной системы.



Смена шрифта по умолчанию

В прошлый раз во время установки мы установили шрифт UniCyrExt_8x16, но это было временное решение, которое сбросилось после установки операционной системы и перезагрузки виртуальной машины. Чтобы выполнить постоянные изменения нужно

вписать переменную FONT в /etc/vconsole.conf. Это переменная специально используется для применения шрифта при загрузке системы для всех консолей:

```
[root@localhost ~]# setfont UniCyrExt_8x16
[root@localhost ~]# nano /etc/vconsole.conf
```

```
FONT=UniCyrExt_8x16
```

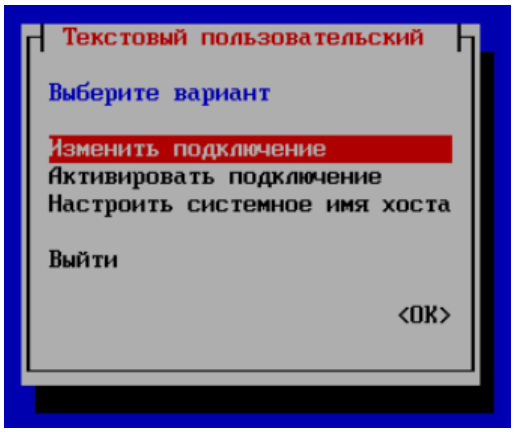
Настройка NetworkManager

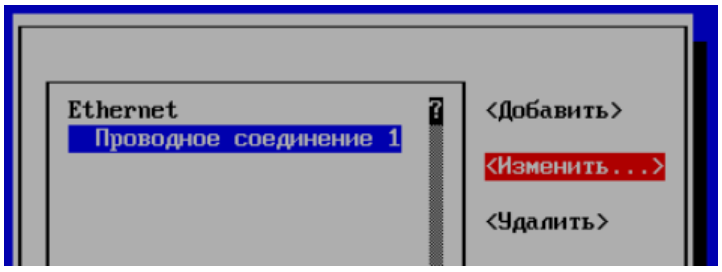
Чтобы при запуске виртуальной машины работали сетевые службы необходимо включить и добавить в автозапуск NetworkManager:

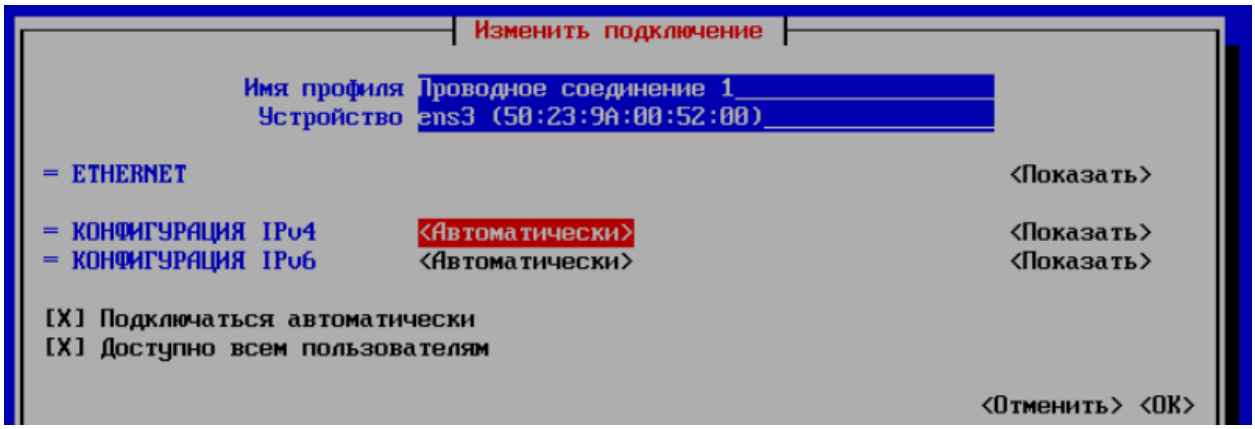
```
[root@localhost ~]# ip link show dev ens3
[root@localhost ~]# systemctl enable NetworkManager
[root@localhost ~]# systemctl start NetworkManager
```

Войдите в Текстовый пользовательский интерфейс настройки NetworkManager и убедитесь, что в пункте «КОНФИГУРАЦИЯ IPv4» стоит «<Автоматически>»:

```
[root@localhost ~]# nmtui
```







[root@localhost ~]# reboot
[root@localhost ~]# ping ya.ru
[ctrl+c]

Создание нового пользователя и добавление пользователя в sudo

Программа sudo (Substitute user and do, дословно «подменить пользователя и выполнить») – это программа для системного администрирования UNIX-систем, позволяющая делегировать те или иные привилегированные ресурсы пользователям с ведением протокола работы. Основная идея – дать пользователям как можно меньше прав, при этом достаточных для решения поставленных задач. Программа поставляется для большинства UNIX и UNIX-подобных операционных систем.

Перед тем как настраивать эту программу, необходимо ее установить и создать пользователя, под которым мы будем выполнять команды:

[root@localhost ~]# pacman -S sudo

Далее мы должны включить группу wheel. Без нее добавление учетной записи администратора в систему невозможно. Нам нужно изменить файл «sudoers». Выполните следующую команду:

[root@localhost ~]# sudo EDITOR=nano visudo

Это запустит файл «/etc/sudoers» с редактором nano. Теперь прокрутите вниз и раскомментируйте группу wheel.

```
## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL:ALL) ALL
```

Группа wheel позволяет создать пользователя с возможностью запуска команд от имени корневого пользователя. root обладает максимальной мощью всей системы, и, если вы используете Linux в течение достаточно долгого времени, вы уже знаете о важности выполнения множества операций по обслуживанию и настройке, требующих root-доступа.

Теперь мы готовы добавить нового пользователя. Команда useradd имеет следующую структуру:

[root@localhost ~]# sudo useradd -m guest (Создание пользователя guest с его собственной домашней папкой (-m)
[root@localhost ~]# sudo passwd guest (Установка пароля пользователю guest)

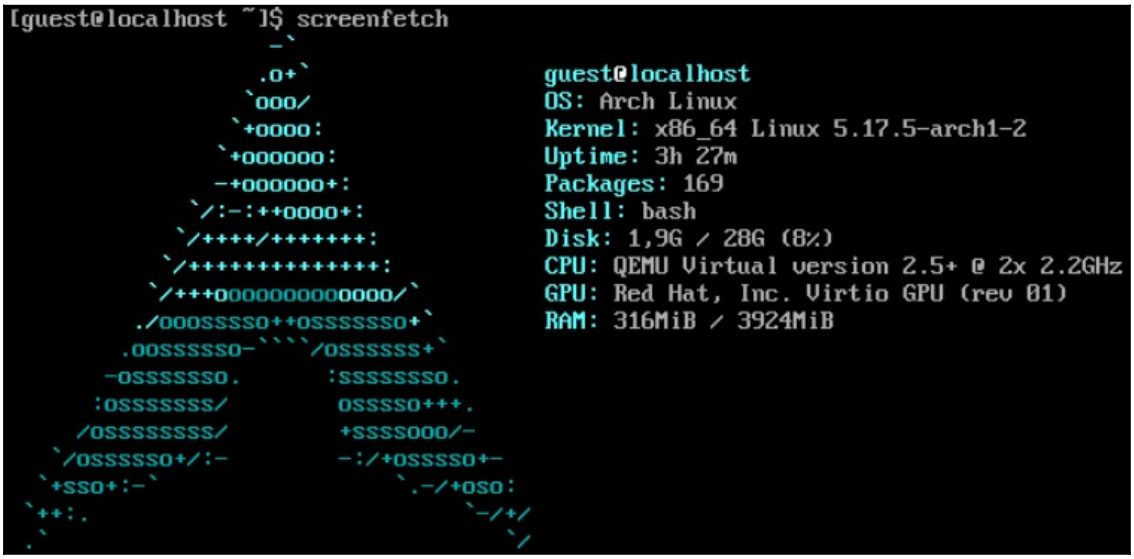
Чтобы добавить пользователя в группу wheel необходимо изменить атрибуты пользователя:

[root@localhost ~]# sudo usermod -g wheel guest

Наконец, попробуйте авторизоваться под гостевым пользователем, и, затем, установить и запустить программу screenfetch:

[root@localhost ~]# login ПАФ
localhost login: guest

Password: guest
[guest@localhost ~]$ sudo pacman -S screenfetch
[guest@localhost ~]$ screenfetch



Поздравляем, вы установили и выполнили базовую конфигурацию Arch Linux!

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: Установка оконного менеджер i3 в дистрибутив Arch Linux

Цель работы: Выполнить конфигурацию оконного менеджера i3, репозиториев yay, AUR и утилит fish, rofi и picom.

– Установка оконного менеджера i3;

– Установка клавиши модификатора доступа и шрифта с поддержкой кириллицы;

– Установка yay и сборка пакета из AUR;

– Использование yay для установки и удаления пакетов AUR;

– Автозапуск i3 при загрузке операционной системы;

– Установка обоев рабочего стола;

– Установка fish, rofi и picom и перенос i3bar наверх;

– Установка polybar.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

i3 – это динамический тайловый оконный менеджер, вдохновленный wmii и нацеленный на разработчиков и опытных пользователей. Этот оконный менеджер функционален и минималистичен, что позволяет легко управлять окнами с помощью

одной лишь только клавиатуры, не вынуждая пользователя постоянно использовать мышь или тачпад.

В отличии от сред рабочего стола (Desktop Environment), таких как GNOME, KDE, Xfce, оконный менеджер предоставляет лишь базовый функционал по управлению окнами. Панель и лаунчер приложений ставятся отдельно и «прикручиваются» к оконному менеджеру с помощью правки конфигурационного файла в текстовом редакторе.

i3 не является единственной в своем классе, есть и другие вроде awesome, sway, и т.д. Любителям тайлинга есть смысл обратить внимание на sway, который стал достаточно популярен. Он содержит все фишки i3, только работает под графическим сервером Wayland вместо стремительно устаревающего и уходящего x.org.

Установка оконного менеджера i3

i3 работает на графическом сервере x.org, поэтому нам потребуется установить его. Также мы установим не сам оригинальный i3wm, а его немного модернизированный форк i3-gaps. Еще сразу нам понадобится программа i3status для отображения системного бара, программа dmenu для запуска установленных программ, которую в последствии заменим на аналог, rofi и xterm для запуска терминала.

```
[guest@localhost ~]$ sudo pacman -Syu
[guest@localhost ~]$ sudo pacman -S xorg-server xorg-xinit i3-gaps i3status dmenu xterm ttf-dejavu ttf-ubuntu-font-family ttf-font-awesome nvidia-utils nvidia-lts
```

После установки создадим и отредактируем файл ~/.xinitrc, записав в него текст «exec i3»:

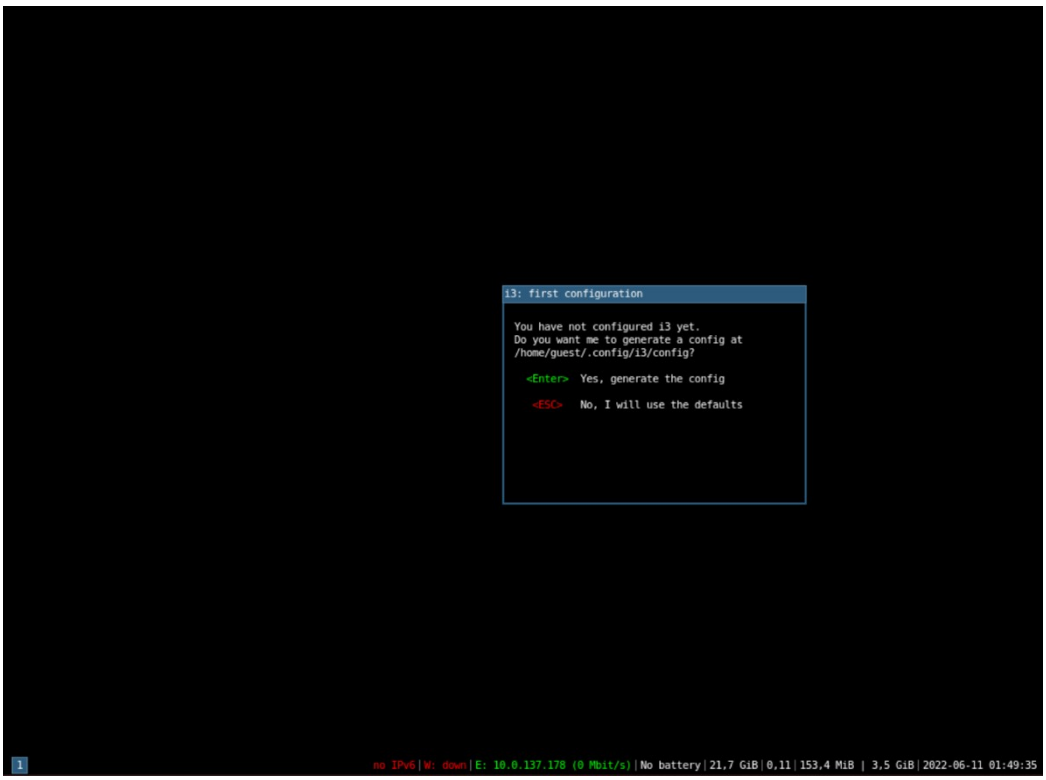
```
[guest@localhost ~]$ echo 'exec i3' >> ~/.xinitrc
```

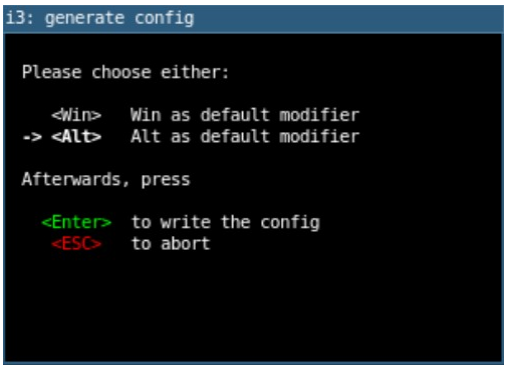
Эта инструкция при запуске графического сервера x.org запустит наш i3wm. И наконец, впервые запустим наш i3:

```
[guest@localhost ~]$ startx
```

Установка клавиши модификатора доступа и шрифта с поддержкой кириллицы

При первом запуске i3 вы увидите рабочий стол с одним диалоговым окном, который предлагает создать конфигурационный файл в домашней директории, согласимся, нажав <Enter>. Затем i3 спросит о том, какую клавишу использовать по умолчанию для модификатора. Выберите <Alt> в качестве дефолтной клавиши <Mod> (Аналог кнопки <Win> в системе Windows). Таким образом нам будет удобнее взаимодействовать с графическим интерфейсом.





Новичку потребуется выучить горячие клавиши для работы с i3. Самое базовое, что нужно знать на данный момент – это:

– <Mod> + <Enter> – запустить терминал;

– <Mod> + <D> – запустить dmenu для запуска любой программы;

– <Mod> + <Shift> + <Q> – закрыть активное окно программы;

– <Mod> + <1>, <Mod> + <2>, <Mod> <…*n*> – переключение рабочих столов от 1 до 9;

– <Mod> + <Shift> + <1>, <Mod> + <Shift> + … n – перемещение активного окна на другой рабочий стол;

– <Mod> + <Shift> + <… ←, →, ↑, ↓> – изменение положения окон в рамках рабочего стола;

– <Mod> + <R> – изменение размера активного окна;

– <Mod> + <Shift> + <C> – «перечитать» конфигурационный файл и применить его изменения;

– <Mod> + <Shift> + <R> – перезапуск i3;

– <Mod> + <Shift> + <E> – выход из i3 с подтверждением;

– <Mod> + <Shift> + <Space> – перевод окна в плавающий режим.

<Mod> – это клавиша модификатора доступа, выбранная ранее. Это основная кнопка в i3, на ней завязано много горячих клавиш, как можно видеть по вышеперечисленному списку. Любые привязки горячих клавиш и остальные настройки i3 хранятся в ~./config/i3/config.

Давайте установим в i3 шрифт из семейства ttf-ubuntu-font-family, чтобы операционная система могла отображать текст с кириллическими символами:

[guest@localhost ~]$ fc-list
[guest@localhost ~]$ sudo nano -c ~/.config/i3/config

```
…
font pango:Ubuntu:style=Medium 10 (Строка 16)
…
```

Теперь установим шрифты с поддержкой кириллицы в терминале xterm.

[guest@localhost ~]$ sudo nano -c ~/.Xresources

```
xterm*faceName: Ubuntu Mono
xterm*faceSize: 10 (Если для вас этот шрифт слишком маленький, вы можете установить значение побольше)
xterm*renderFont: true
```

[guest@localhost ~]$ sudo nano -c ~/.xinitrc

```
#!/bin/sh

xrdb -merge ~/.Xresourses (Если не написать эту строчку, то наши изменения не будут применяться на терминал при загрузке графического сервера x.org)
exec i3
```

[guest@localhost ~]$ sudo nano -c ~/.config/i3/config

```
bindsym $mod+Return exec xterm (Строка 48 | Если не поменять эту строчку, то наши изменения не будут применяться на терминал, когда мы будем запускать ее с помощью горячей клавиши)
```

При старте системы через программу setxkbmap устанавливается смена раскладки клавиатуры.[4] Давайте установим этот пакет и впишем настройки о раскладке клавиатуры в конфигурационный файл:

[guest@localhost ~]$ sudo pacman -S xorg-setxkbmap (При установке x.org он, как правило, устанавливается автоматически)

[4] Установленная в /etc/vconsole.conf по неизвестной мне причине не задействуется в i3.

[guest@localhost ~]$ sudo nano -c ~/.config/i3/config

exec --no-startup-id setxkbmap us,ru -option 'grp:ctrl_shift_toggle' (Строка 188 | Смена раскладки осуществляется нажатием клавиш <Ctrl> + <Shift>)

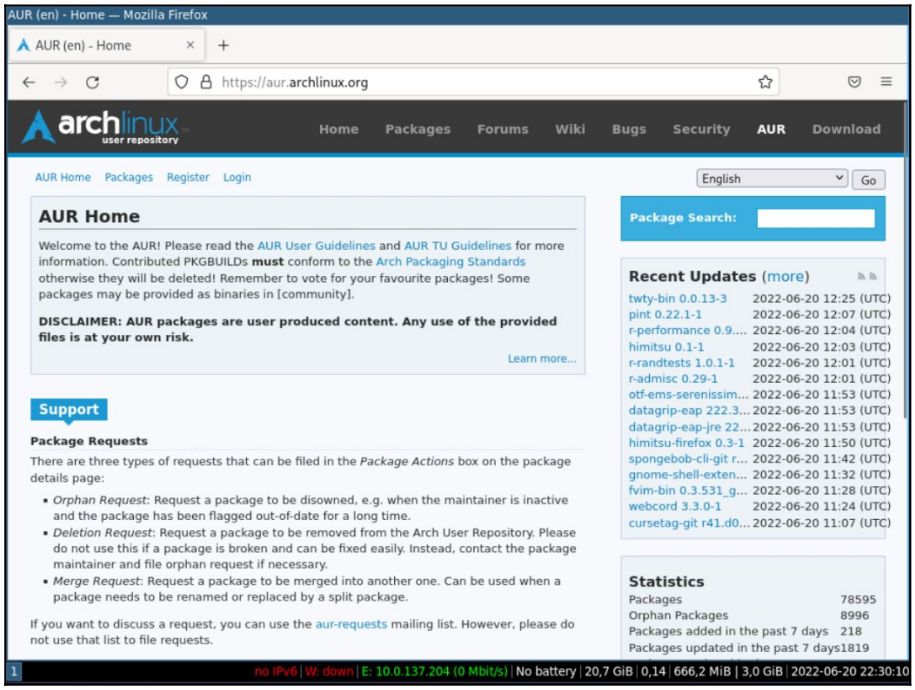
Перед самой командой фигурирует ключевое слово exec, т.е. запуск, и параметр -no-startup-id, который нужен для того, чтобы курсор мыши не превращался в «бесконечно крутящиеся часы». Похожим образом мы можем поставить любую программу или команду на автостарт i3.

Перезагрузите сеанс и проверьте работает ли смена раскладки клавиатуры:

[Alt+Shift+r]
[Alt+Shift]
[guest@localhost ~]$ Привет, мир!

Установка yay и сборка пакета из AUR

Пользовательский репозиторий Arch (AUR) – это место, где вы можете найти программное обеспечение, протестированное не создателями и сопровождающими Arch, а его пользователями. Благодаря этому вы можете получить доступ к большему количеству программного обеспечения, которого нет в официальных репозиториях.



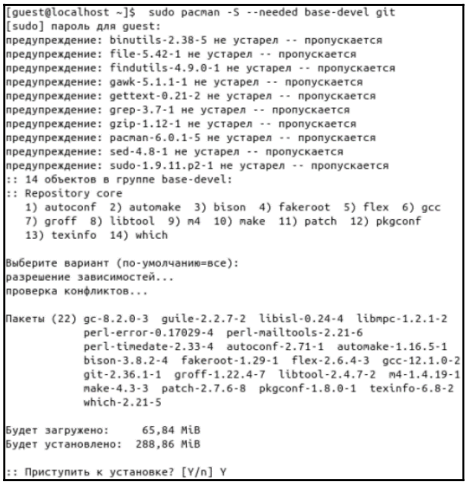
Есть два способа получить доступ к коллекции программного обеспечения AUR. Большинство людей предпочитают использовать AUR Helpers, которые работают аналогично диспетчеру пакетов Arch по умолчанию – pacman. Этот подход автоматизирует извлечение и сборку исходного кода из AUR с использованием знакомых команд. Кроме того, вы можете загрузить пакет прямо из AUR и скомпилировать его вручную.

Есть много помощников AUR, которые вы можете использовать. yay, вероятно, самый популярный сегодня, поэтому мы будем использовать его.

Установите необходимые инструменты разработки и git, если они еще не доступны в вашем дистрибутиве:

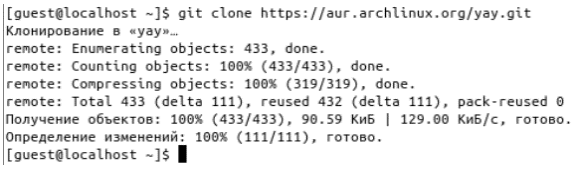
[guest@localhost ~]$ sudo pacman -S --needed base-devel git

Вероятно, вам будет предложено несколько дополнений к программному обеспечению и вам будет предложено выбрать, какое из них вы хотите установить. Перейдите к опции по умолчанию «все», нажав <Enter< на клавиатуре. Далее введите «y» и нажмите <Enter>, когда вас спросят, хотите ли вы продолжить установку.



Чтобы установить yay, сначала скачайте его исходный код.

[guest@localhost ~]$ git clone https://aur.archlinux.org/yay.git

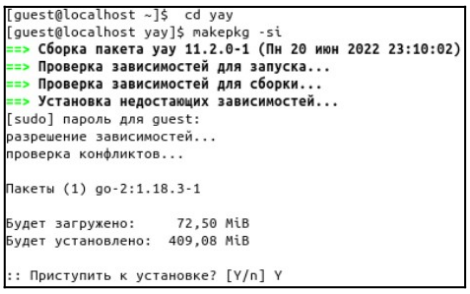


Все, что вы загрузили, будет находиться в подпапке с именем «yay». Войдите в директорию с помощью команды:

[guest@localhost ~]$ cd yay

Находясь в этой папке, соберите пакет с помощью команды:

[guest@localhost yay]$ makepkg -si



Ответьте «y» на вопрос, хотите ли вы также установить встроенный пакет.

```
==> Очистка...
  -> Удаление файлов libtool...
  -> Удаление ненужных файлов...
  -> Удаление статических библиотек...
  -> Удаление отладочной информации из бинарников и библиотек...
  -> Сжатие документации (man и info)...
==> Проверка сборки на ошибки...
==> Создание пакета "yay"...
  -> Создание файла '.PKGINFO'...
  -> Создание файла '.BUILDINFO'...
  -> Создание файла '.MTREE'...
  -> Сжатие пакета...
==> Выход из окружения fakeroot.
==> Завершена сборка пакета yay 11.2.0-1 (Пн 20 июн 2022 23:11:20)
==> Установка пакета 'yay' с помощью 'pacman -U'...
загрузка пакетов...
разрешение зависимостей...
проверка конфликтов...

Пакеты (1) yay-11.2.0-1

Будет установлено:  7,04 MiB

:: Приступить к установке? [Y/n] Y
```

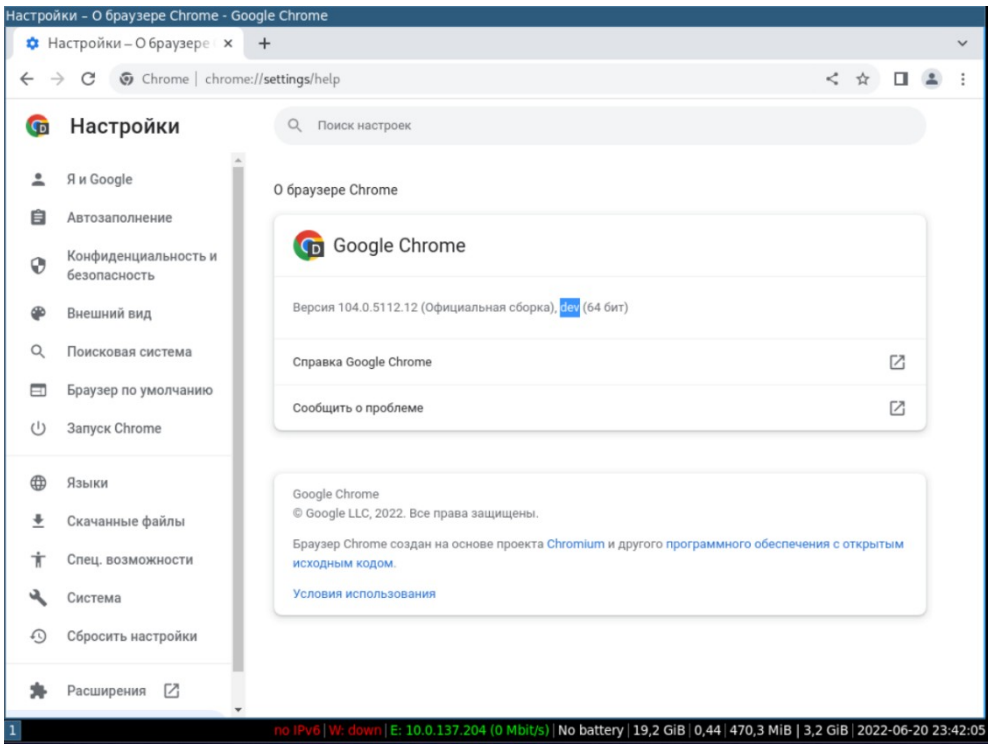
Отлично, мы произвели установку и сборку пакета yay. Применяя этот способ можно скачивать и устанавливать любой пакет с необходимым нам программным обеспечением из репозитория AUR.

Использование yay для установки и удаления пакетов AUR

Раз уж мы установили yay, то давайте им воспользуемся для установки пакетов из AUR. yay использует тот же синтаксис, что и pacman. Таким образом, вы можете легко установить пакеты AUR, используя команду:

[guest@localhost ~]$ yay -S google-chrome-dev

Отличительной чертой помощников AUR, таких как yay, является то, что они также автоматически работают с зависимостями. Это дополнительные подпрограммы, необходимые для всего, что вы пытаетесь установить, и которые в противном случае вам пришлось бы выискивать и устанавливать самостоятельно.



Чтобы удалить что-либо, установленное через yay, вам придется снова использовать yay. Для этого вы можете использовать параметр -R, но лучше, если вы выберете параметр -Rns, который лучше выполняет очистку. Например, чтобы удалить Google Chrome:

[guest@localhost ~]$ yay -Rns google-chrome-dev

Вы также можете удалить лишние пакеты и все остатки от прошлых установок с помощью команды yay:

[guest@localhost ~]$ yay -Yc

Автозапуск i3 при загрузке операционной системы

Поместите следующую информацию в файл инициализации оболочки bash при входе в систему:

[guest@localhost ~]$ sudo nano -c ~/.bash_profile

```
if [ -z "${DISPLAY}" ] && [ "${XDG_VTNR}" -eq 1 ]; then (Строка 6)
  exec startx
fi
```

[guest@localhost ~]$ sudo reboot

Установка обоев рабочего стола

Оконный менеджер i3 не будет изменять настройки вашего дисплея по умолчанию, поэтому вам нужен инструмент для установки обоев, например, программа feh.

feh – это легкий и мощный просмотрщик изображений, который также может управлять фоном рабочего стола для оконных менеджеров, не умеющих делать это самостоятельно:

[guest@localhost ~]$ sudo pacman -S feh

Найдите изображение, которое нравится вам, и сохраните его на своей виртуальной машине. Для скачивания можно использовать, например, вновь установленный браузер Google Chrome. Чтобы установить его в качестве обоев рабочего стола, введите:

[guest@localhost ~]$ sudo nano -c ~/.config/i3/config

```
exec --no-startup-id feh --bg-scale ~/'Загрузки/raccoons.jpg' (Строка 189)
```

[Alt+Shift+e]
[guest@localhost ~]$ startx

Установка fish, rofi и picom и перенос i3bar наверх

fish (friendly interactive shell, т.е. дружелюбная интерактивная оболочка) представляет собой удобную интерактивную оболочку командной строки предназначенную в основном для интерактивного использования. Давайте установим ее:

[guest@localhost ~]$ sudo pacman -S fish

Для того чтобы сделать fish оболочкой по умолчанию, выполните следующую команду:

[guest@localhost ~]$ chsh -s /usr/bin/fish
[guest@localhost ~]$ sudo reboot

Из-за того, что мы ранее использовали оболочку bash для запуска x.org, то после перезагрузки компьютера он автоматически не запустит нам графический интерфейс, так как теперь он запускает оболочку fish. Давайте это исправим.

Добавьте следующее в нижнюю часть конфигурационного файла ~/.config/fish/config.fish.

guest@localhost ~> sudo nano -c ~/.config/fish/config.fish

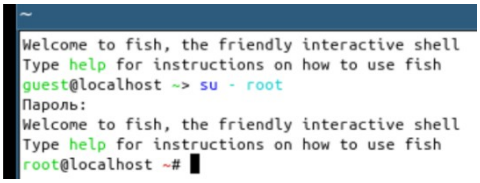
```
# start X at login (Строка 5)
if status --is-login
    if test -z "$DISPLAY" -a $XDG_VTNR -eq 1
        exec startx -- -keeptty
    end
end
```

Давайте в свою очередь также определим следующую функцию в fish, чтобы при использовании команды su запускался fish вместо bash:

guest@localhost ~> sudo nano -c ~/.config/fish/functions/su.fish

```
# make su launch fish
function su
        command su --shell=/usr/bin/fish $argv
end
```

[guest@localhost ~]$ sudo reboot



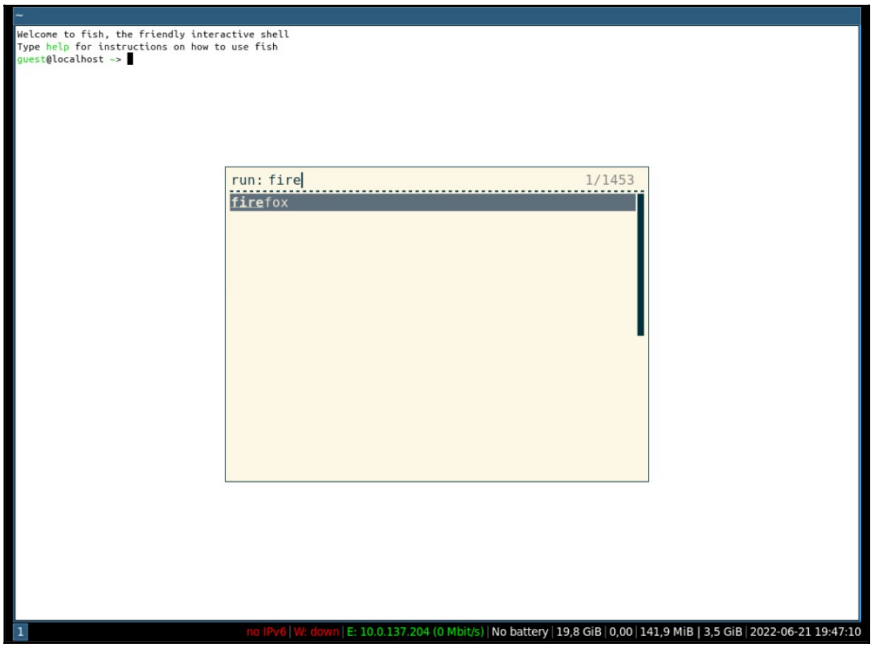
Следующим шагом заменим dmenu на аналог rofi. Концептуально они не отличаются, просто rofi мне расположен не в баре, а плавающей строкой, что удобнее:

guest@localhost ~> sudo pacman -S rofi

Чтобы задействовать его вместо dmenu найдем в конфигурационном файле i3 строку bindsym $mod+d exec dmenu_run и замените ее:

guest@localhost ~> sudo nano -c ~/.config/i3/config

```
bindsym $mod+d exec --no-startup-id rofi -show run (Строка 54)
```



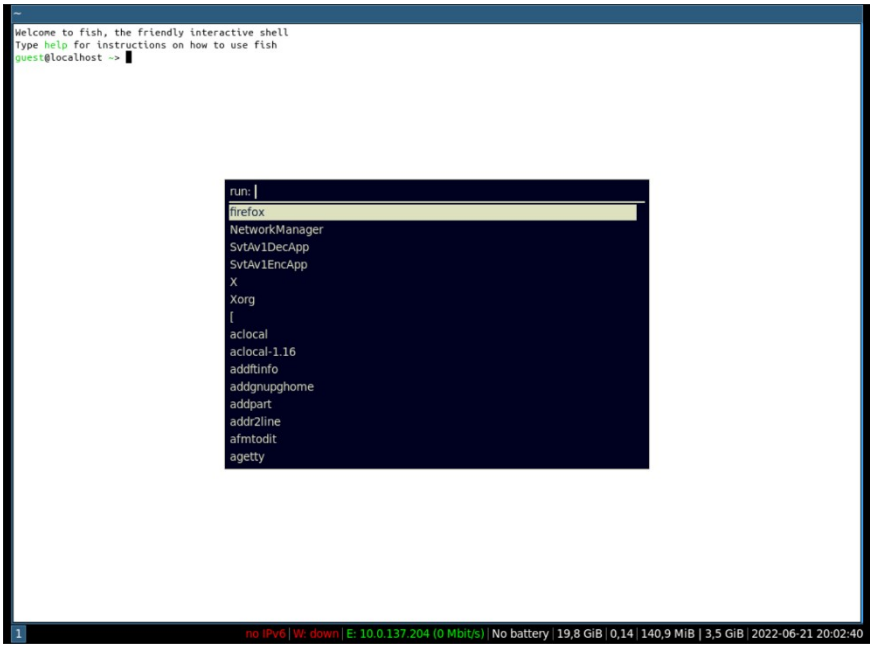
rofi имеет множество встроенных тем, посмотреть и выбрать подходящую для себя можно утилитой rofi-theme-selector, запустив ее в терминале. После выбранную тему установить, как параметр -theme *имя_темы* в строке запуска в конфигурационном файле. В качестве шрифта был выбран UbuntuMono-R с кеглем 10. Давайте воспользуемся этой утилитой и выберем подходящую тему:

guest@localhost ~> rofi-theme-selector

Чтобы применить понравившуюся тему заменим в конфигурационном файле i3 строку bindsym $mod+d exec --no-startup-id rofi -show run на следующее:

guest@localhost ~> sudo nano -c ~/.config/i3/config

```
bindsym $mod+d exec --no-startup-id rofi -combi-modi window#drun#ssh -theme DarkBlue -font "UbuntuMono-R 10" -show run (Строка 54)
```



Чтобы включить пробелы (gaps) между окнами, вам нужно установить некоторые переменные в вашей конфигурации i3:

guest@localhost ~> sudo nano -c ~/.config/i3/config

```
gaps inner 5 (Строка 192)
gaps outer 5
```

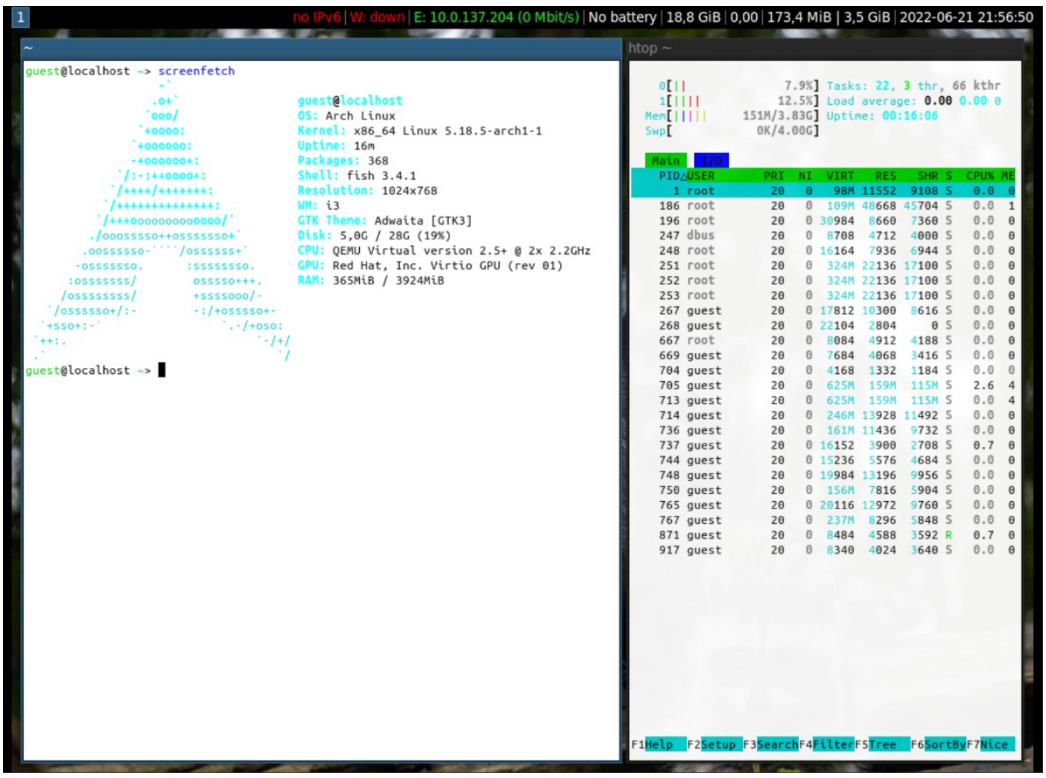
[Alt+Shift+e]

Чтобы перенести статус-бар наверх, то необходимо в секцию bar добавить строку position top, и перезапустить i3 (<Alt> + <Shift> + <R>):

guest@localhost ~> sudo nano -c ~/.config/i3/config

```
bar { (Строка 85)
        position top
        status_command i3status
}
```

[Alt+Shift+e]



picom (форк устаревшего compton) – это легкий композитный менеджер для графического сервера x.org. Нам он нужен в первую очередь для устранения тиринга, если таковой присутствует, а также для создания небольших эффектов плавности переключения рабочих столов, чтобы было покрасивее и повеселее:

guest@localhost ~> sudo pacman -S picom
guest@localhost ~> sudo nano -c ~/.config/i3/config

exec –-no-startup-id picom -–experimental-backends -b (Строка 195)

[Alt+Shift+e]

Установка polybar

polybar – это быстрый и легкий инструмент для создания статус-баров. Он нацелен на легкую персонализацию, используя множество модулей и позволяя, например, отображать рабочие столы, дату или громкость звука. Особенно polybar полезен в оконных менеджерах без панели или с ее ограниченной функциональностью, таких как awesome или i3. Polybar также можно использовать и в окружениях рабочего стола, например, в Plasma:

guest@localhost ~> sudo pacman -S polybar

Давайте скопируем файл настроек polybar в директорию пользователя:

guest@localhost ~> mkdir ~/.config/polybar/
guest@localhost ~> cp /etc/polybar/config.ini ~/.config/polybar/config.ini
(/etc/polybar/config.ini – это место хранения конфигурации polybar по умолчанию; ~/.config/polybar/config.ini – это место хранения конфигурации polybar, которая будет применяться у пользователя guest при запуске программы)
guest@localhost ~> cat ~/.config/polybar/config.ini (Конфигурация polybar по умолчанию)

Чтобы запустить polybar вместе с нашим оконным менеджером, необходимо сначала создать исполняемый файл, содержащий процесс загрузки, /home/guest/.config/polybar/launch.sh:

guest@localhost ~> sudo nano -c /home/guest/.config/polybar/launch.sh

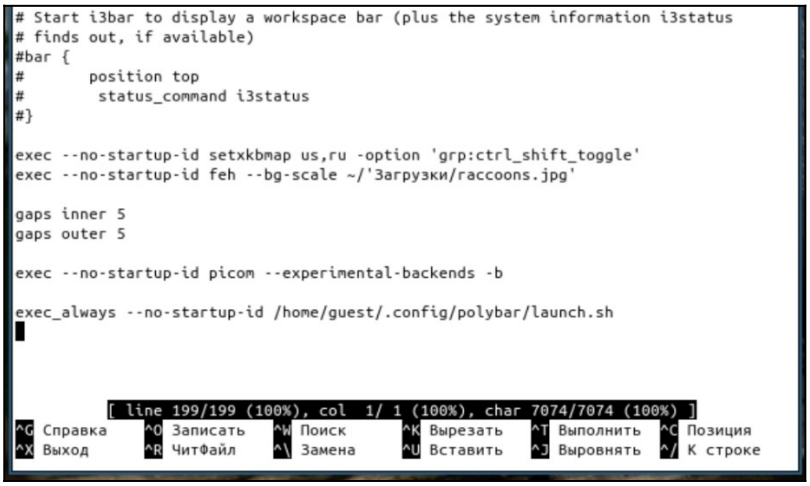
```
#!/bin/bash

#Terminate already running bar instances
killall -q polybar
#If all your bars have ipc enabled, you can also use
#polybar-msg cmd quit

#Launch Polybar, using default config location ~/.config/polybar/config.ini
polybar example 2>&1 | tee -a /tmp/polybar.log & disown

echo "Polybar launched..."
```

guest@localhost ~> sudo chmod ugo+x /home/guest/.config/polybar/launch.sh

Данный скрипт означает, что при перезагрузке оконного менеджера также перезагрузится и polybar.

Затем требуется добавить следующее содержание в конфигурационный файл i3:

guest@localhost ~> sudo nano -c ~/.config/i3/config



[Alt+Shift+e]

Если вы захотите сконфигурировать polybar под себя, то вы можете для этого выполнить изменения в конфигурационном файле. Но лучше для этого использовать заранее разработанные темы. Воспользуемся пакетом polybar-themes.

Во-первых, клонируйте репозиторий с этим пакетом:

guest@localhost ~> git clone --depth=1 https://github.com/adi1090x/polybar-themes.git

Затем перейдите в клонированный каталог и сделайте setup.sh исполняемым файлом:

guest@localhost ~> cd polybar-themes
guest@localhost ~> chmod ugo+x setup.sh

Запустите setup.sh и выберите стиль:

guest@localhost ~> ./setup.sh

Теперь, чтобы выбрать определенную тему, нужно ее добавить в конфигурационный файл i3:

guest@localhost ~> sudo nano -c ~/.config/i3/config

```
exec_always --no-startup-id /home/guest/.config/polybar/launch.sh --blocks
```

[Alt+Shift+e]

Список доступных тем представлен ниже (их необходимо вписать в конфигурационный файл i3 после слов launch.sh):

– --blocks;

– --colorblocks;

– --cuts;

– --docky;

– --forest;

– --grayblocks;

– --hack;

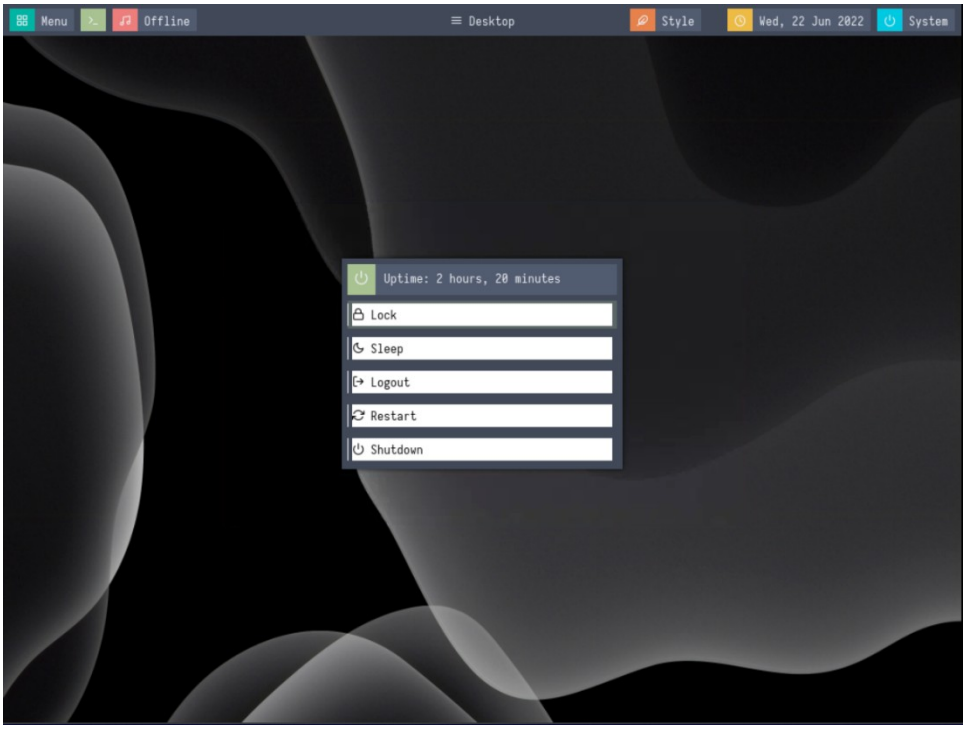
– --material;

– --panels;

– --pwidgets;

– --shades;

– --shapes.



ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №5

Тема: Установка пакетов в Arch Linux

Цель работы: Установить пакеты с программным обеспечением в Debian 11:

– Использование утилиты pacman;

– Графический интерфейс pacman.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Все существующие дистрибутивы Linux для того, чтобы осуществлять операции по обновлению ПО и библиотек ОС в основном используют пакетные менеджеры. Пакетный менеджер является ключевой программой, который зачастую и отличает различие одного дистрибутива Linux от другого.

Пакетный менеджер pacman является основным признаком Arch Linux. pacman предназначен для установки программ и пакетов в этом дистрибутиве. pacman оперирует пакетами, которые, как и в случае с другими пакетными менеджерами представляют из себя бинарные архивы, содержащие некоторые данные для установки.

pacman, как и RPM имеет возможность загрузки части пакета при обновлении. При работе pacman зачастую значительно быстрее обрабатывает транзакции, нежели это делает DPKG или RPM.

pacman, как и RPM имеет возможность загрузки части пакета при обновлении. При работе pacman зачастую значительно быстрее обрабатывает транзакции, нежели это делает dpkg (Debian) или RPM (RHEL).

Установка пакетов Arch Linux, их обновление, а также их удаление как мы уже поняли выше, осуществляются при помощи pacman. Данная утилита достаточно проста. Рассмотрим ее синтаксис:

pacman [опция] <имя пакета>

Вот основные опции программы, которые чаще всего используются при использовании pacman:

– -S – установка программ в Arch Linux;

– -Sw - скачать пакет но не устанавливать;

– -U - установить локальный пакет;

– -s – поиск пакета;

– -i – информация;

– -u – обновить пакеты;

– -y – обновить списки пакетов из репозиториев;

– -dd – игнорирование зависимостей;

– -g – запрос информации о группе;

– -l – вывод пакетов в виде списка;

– -o – вывод в файл;

– -R – удаление;

– -Q – запрос к базе данных установленных пакетов;

– -F – поиск по файлам;

– -D – задание и изменение цели установки уже имеющегося пакета;

– -R - удалить пакет;

– -Rn - удалять резервные копии конфигурационных файлов;

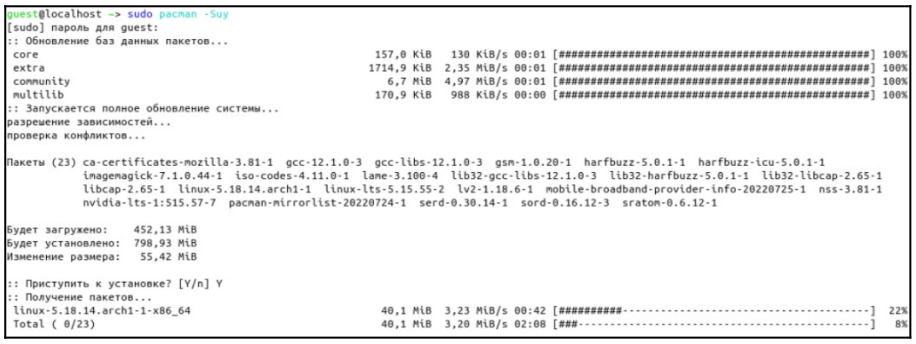
– -Rs - удалять зависимости пакета.

Сначала рассмотрим как выполняется установка программ Arch Linux с помощью pacman из официальных репозиториев, а потом остановимся на пользовательском репозитории AUR. Теперь рассмотрим основные команды pacman:

Использование утилиты pacman

Опции для pacman при использовании комбинируются в одну фразу, таким образом, например, для обновления репозиториев и обновления ОС команда будет выглядеть так:

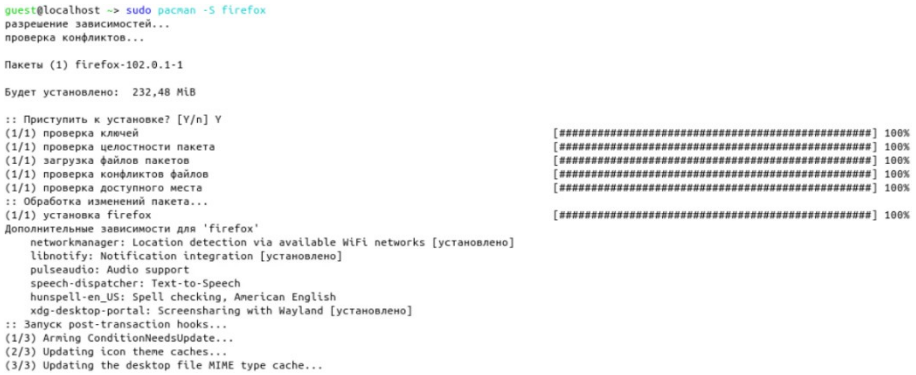
[guest@localhost ~]$ sudo pacman -Suy



Порядок расстановки опций значения не имеет, за исключением только того, что основной ключ, указывающийся большой буквой должен быть первым.

Установка конкретного пакета выполняется так:

[guest@localhost ~]$ sudo pacman -S *[имя пакета]*



Удаление конкретного пакета:

[guest@localhost ~]$ sudo pacman -R *[имя пакета]*

```
guest@localhost ~ [1]> sudo pacman -R firefox
проверка зависимостей...

Пакеты (1) firefox-102.0.1-1

Будет освобождено:  232,48 MiB

:: Удалить эти пакеты? [Y/n] Y
:: Обработка изменений пакета...
(1/1) удаление firefox                                          [################################################] 100%
:: Запуск post-transaction hooks...
(1/3) Arming ConditionNeedsUpdate...
(2/3) Updating icon theme caches...
(3/3) Updating the desktop file MIME type cache...
```

Принудительное удаление конкретного пакета, с игнорированием зависимостей:

[guest@localhost ~]$ sudo pacman -Rdd *[имя пакета]*

```
guest@localhost ~> sudo pacman -Rdd firefox

Пакеты (1) firefox-102.0.1-1

Будет освобождено:  232,48 MiB

:: Удалить эти пакеты? [Y/n] Y
:: Обработка изменений пакета...
(1/1) удаление firefox                                          [################################################] 100%
:: Запуск post-transaction hooks...
(1/3) Arming ConditionNeedsUpdate...
(2/3) Updating icon theme caches...
(3/3) Updating the desktop file MIME type cache...
guest@localhost ~>
```

Поиск пакета в репозиториях:

[guest@localhost ~]$ sudo pacman -Ss *[имя пакета]*

```
community/firefox-developer-edition-i18n-tr 103.0b9-1
    Turkish language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-trs 103.0b9-1
    Triqui language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-uk 103.0b9-1
    Ukrainian language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-ur 103.0b9-1
    Urdu language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-uz 103.0b9-1
    Uzbek language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-vi 103.0b9-1
    Vietnamese language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-xh 103.0b9-1
    Xhosa language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-zh-cn 103.0b9-1
    Chinese (Simplified) language pack for Firefox Developer Edition
community/firefox-developer-edition-i18n-zh-tw 103.0b9-1
    Chinese (Traditional) language pack for Firefox Developer Edition
community/firefox-extension-mailvelope 4.6.0-1 (firefox-addons)
    Browser extension for OpenPGP encryption with Webmail
community/firefox-extension-passff 1.14.0-1 (firefox-addons)
    zx2c4 pass manager addon for firefox
community/firefox-noscript 11.4.6-1 (firefox-addons)
    Extension for firefox which disables javascript
community/firefox-spell-ru 0.4.5-3
    Russian spellchecker dictionary for Firefox
community/firefox-tree-style-tab 3.8.26-1 (firefox-addons)
    Firefox extension to show tabs like a tree
community/firefox-tridactyl 1.22.1-1 (firefox-addons)
    Replace Firefox's control mechanism with one modelled on Vim
community/firefox-ublock-origin 1.43.0-1 (firefox-addons)
    Efficient blocker add-on for various browsers. Fast, potent, and lean
community/otf-fira-mono 2:3.206-4
    Mozilla's monospace typeface designed for Firefox OS
community/otf-fira-sans 1:4.301-2
    Mozilla's sans-serif typeface designed for Firefox OS
community/passff-host 1:1.2.3-1
    PassFF native messaging host application for Firefox, Chromium, Chrome, Vivaldi
community/python-fxa 0.7.7-5
    Python library for interacting with the Firefox Accounts ecosystem
community/ttf-fira-mono 2:3.206-4
    Mozilla's monospace typeface designed for Firefox OS
community/ttf-fira-sans 1:4.301-2
    Mozilla's sans-serif typeface designed for Firefox OS
```

Получение информации о пакете:

[guest@localhost ~]$ sudo pacman -Si *[имя пакета]*

```
guest@localhost ~> sudo pacman -Si firefox
Репозиторий          : extra
Название             : firefox
Версия               : 102.0.1-1
Описание             : Standalone web browser from mozilla.org
Архитектура          : x86_64
URL                  : https://www.mozilla.org/firefox/
Лицензии             : MPL  GPL  LGPL
Группы               : Нет
Предоставляет        : Нет
Зависит от           : gtk3  libxt  mime-types  dbus-glib  ffmpeg  nss  ttf-font  libpulse
Доп. зависимости     : networkmanager: Location detection via available WiFi networks
                       libnotify: Notification integration
                       pulseaudio: Audio support
                       speech-dispatcher: Text-to-Speech
                       hunspell-en_US: Spell checking, American English
                       xdg-desktop-portal: Screensharing with Wayland
Конфликтует с        : Нет
Заменяет             : Нет
Размер загрузки      : 62,61 MiB
Установленный размер : 232,48 MiB
Сборщик              : Jan Alexander Steffens (heftig) <heftig@archlinux.org>
Дата сборки          : Ср 06 июл 2022 22:23:07
Проверен             : MD5  SHA-256  Подпись
```

Поиск только среди установленных пакетов:

[guest@localhost ~]$ sudo pacman -Qs *[имя пакета]*

```
guest@localhost ~ [1]> sudo pacman -Qs htop
local/htop 3.2.1-1
    Interactive process viewer
```

Просмотр списка всех установленных пакетов:

[guest@localhost ~]$ sudo pacman -Qqe *[имя пакета]*

```
htop
i3-gaps
i3status
kdialog
leafpad
lib32-mpg123
libtool
linux
linux-firmware
m4
make
nano
nautilus
net-tools
networkmanager
nvidia-lts
nvidia-utils
openssh
patch
picom
pkgconf
playonlinux
polybar
poppler
ranger
rofi
screenfetch
sudo
texinfo
ttf-dejavu
ttf-font-awesome
ttf-ubuntu-font-family
vim
w3m
which
wine
wine-gecko
wine-mono
winetricks
xorg-server
xorg-xinit
xterm
yay
zenity
```

Просмотр списка файлов пакета:

[guest@localhost ~]$ sudo pacman -Ql *[имя пакета]*

```
guest@localhost ~> sudo pacman -Ql htop
htop /usr/
htop /usr/bin/
htop /usr/bin/htop
htop /usr/share/
htop /usr/share/applications/
htop /usr/share/applications/htop.desktop
htop /usr/share/icons/
htop /usr/share/icons/hicolor/
htop /usr/share/icons/hicolor/scalable/
htop /usr/share/icons/hicolor/scalable/apps/
htop /usr/share/icons/hicolor/scalable/apps/htop.svg
htop /usr/share/man/
htop /usr/share/man/man1/
htop /usr/share/man/man1/htop.1.gz
htop /usr/share/pixmaps/
htop /usr/share/pixmaps/htop.png
```

Очистка кэша:

[guest@localhost ~]$ sudo pacman -Scc *[имя пакета]*

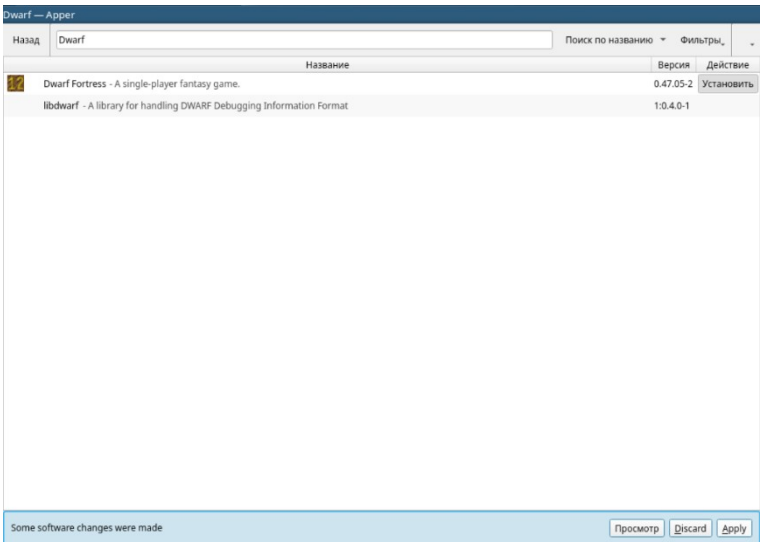
```
guest@localhost ~> sudo pacman -Scc

Каталог для кэша: /var/cache/pacman/pkg/
:: Удалить ВСЕ файлы из кэша? [y/N] y
удаление всех файлов из кэша...

Каталог базы данных: /var/lib/pacman/
:: Удалить неиспользуемые репозитории? [Y/n] y
удаление неиспользуемых репозиториев...
```

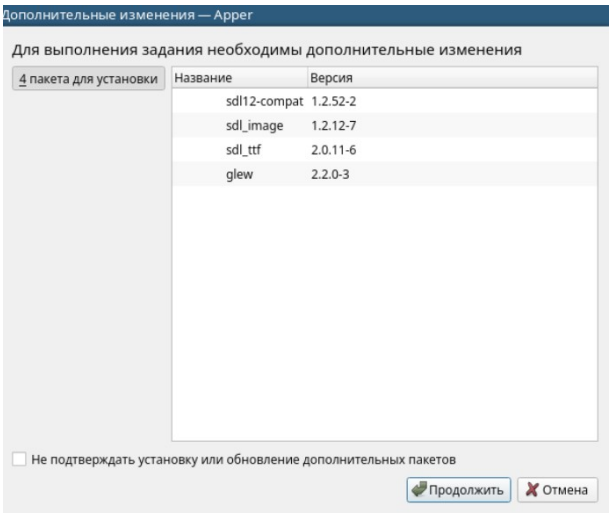
Графический интерфейс pacman

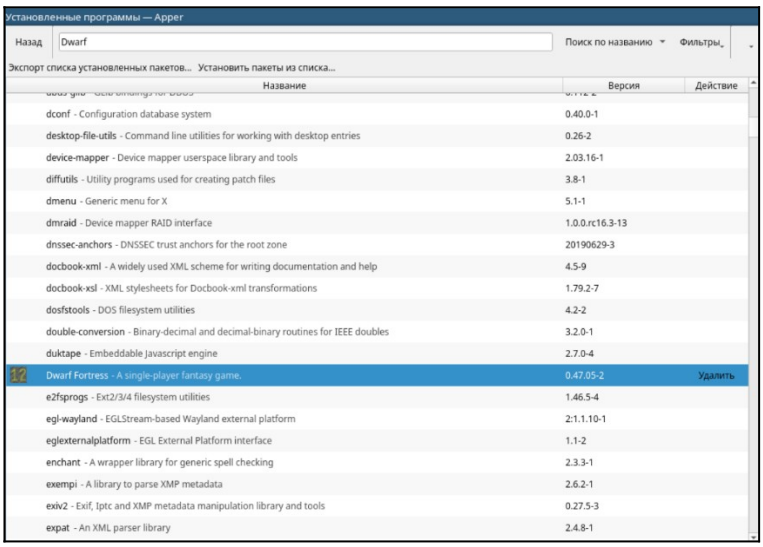
Для любителей пользоваться графическим интерфейсом вместо терминала в Arch Linux есть утилита Apper. Это простой менеджер приложений, чем-то похожий на стандартный менеджер приложений Ubuntu:

[guest@localhost ~]$ yay -S apper

Для установки пакета достаточно нужно найти нужную программу и нажать «Установить». Удаление пакетов выполняется схожим образом.







ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №6

Тема: Основы работы с текстовым редактором vim и файловым менеджером ranger

Цель работы: Изучить базовые возможности работы с vim и ranger.

– Текстовый редактор vim;

– Опции vim;

– Учебник vimtutor;

– Одновременное редактирование нескольких файлов в vim;

– Использование буфера обмена в vim;

– Использование кириллической раскладки при вводе команд в vim;

– Файловый менеджер ranger;

– Установка и запуск ranger;

– Визуализация скрытых файлов в ranger;

– Базовое перемещение и сочетания клавиш в ranger;

– Копирование, перемещение и удаление файлов в ranger;

– Получение предварительного просмотра документа в ranger;

– Предварительный просмотр PDF-файлов и изображений в ranger;

– Создание, доступ и удаление закладок в ranger;

– Выбор файлов в ranger.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Текстовый редактор vim

Опытные пользователи Linux часто используют терминал, потому что так можно намного быстрее выполнить необходимые действия. Во время настройки системы нам довольно часто приходится редактировать различные файлы. Это могут быть настройки программ, какие-нибудь данные или обычные текстовые файлы.

В операционной системе Linux есть несколько текстовых редакторов, которые работают в терминале. Чаще всего новички используют редактор nano, у которого интуитивно понятный интерфейс и есть много функций, свойственных графическим приложениям. Он позволяет вырезать и вставлять, искать и заменять слова и так далее. Его удобно использовать для решения быстрых и простых задач, однако если для системного администратора и программиста в приоритете максимально расширенный функционал, ему все же стоит присмотреться именно к vim.

Программа vim – это текстовый редактор, который можно удобно настроить под себя благодаря расширениям и большому выбору инструментов для пользователя. Он работает в терминале, что дает возможность разработчикам и системным администраторам значительно быстрее выполнять задачи. За счет командных функций vim помогает увеличить продуктивность работы. Это выбор профессионалов, которые используют консоли для быстрого и удобного написания кода, скриптов и текста.

Давайте установим его, если он не установлен по умолчанию:

guest@localhost ~> sudo pacman -Syu
guest@localhost ~> sudo pacman -S vim (или sudo apt install vim в случае дистрибутивов, основанных на Debian)

Текстовый редактор vim может работать в двух режимах. Это и есть его главная особенность. Первый режим, который используется по умолчанию при открытии редактора – это командный. В этом режиме вы можете вводить команды vi, а также использовать символьные клавиши для управления редактором. Второй режим – обычное редактирование текста, он работает так же, как и редактирование текста в nano. Для переключения в командный режим используется клавиша <Esc>. Для переключения в режим редактирования – клавиша <i>.

Опции vim

Начнем мы с запуска программы, а также опций, которые ей можно передать. Синтаксис vim прост: vim *-опция имя_файла* (или vi *-опция имя_файла*)

Простой запуск vim без указания имени файла приведет к созданию пустого файла. А теперь давайте рассмотрим основные опции запуска:

– +*номер* – переместить курсор к указной строке после запуска;

– +/*шаблон* – выполнить поиск по шаблону и переместить курсор к первому вхождению;

– "+*команда*" – выполнить команду после запуска программы;

– -b – двоичный режим, для редактирования исполняемых файлов;

– -d – режим поиска различий в файлах, нужно указать несколько файлов для открытия;

– -g – графический режим;

– -n – не использовать автосохранение для восстановления файла при сбое;

– -R – режим только для чтения;

– -w – сохранить все действия в файл;

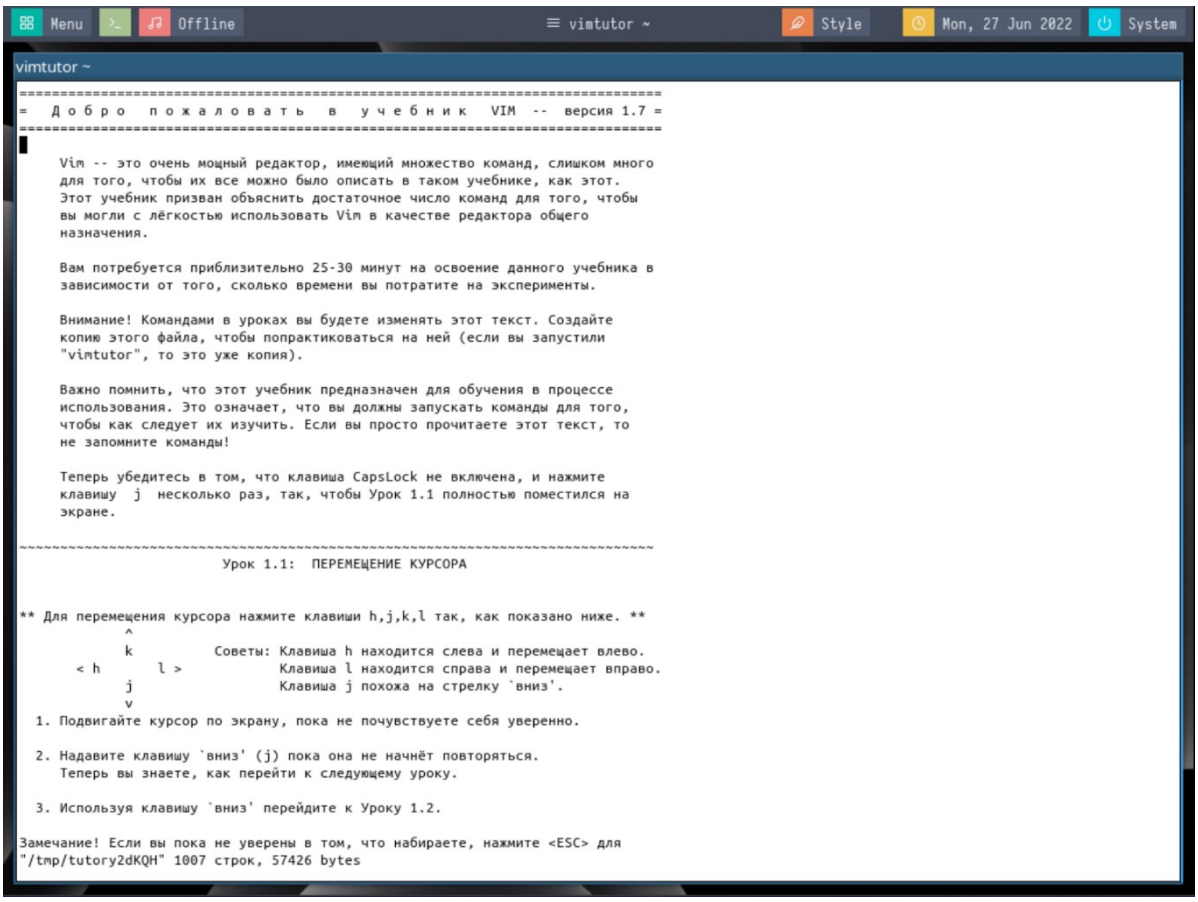
– -x – шифровать файл при записи;

– -C – режим совместимости с vi.

Учебник vimtutor

Есть определенная проблема, заключающаяся в том, что команд и сочетаний клавиш в программе vim очень много и запомнить их без практики невозможно. Именно поэтому в текстовом редакторе vim легче всего научиться работать с помощью курса обучения, встроенного в редактор. Выполнение всех обучающих заданий займет 25-30 минут. Для запуска обучения наберите:

guest@localhost ~> vimtutor



Внимательно прочтите учебник и выполните все уроки, описанные в нем.

```
                        Урок 1.1:  ПЕРЕМЕЩЕНИЕ КУРСОРА


** Для перемещения курсора нажмите клавиши h,j,k,l так, как показано ниже. **
              ^
              k           Советы: Клавиша h находится слева и перемещает влево.
         < h       l >            Клавиша l находится справа и перемещает вправо.
              j                   Клавиша j похожа на стрелку `вниз'.
              v
  1. Подвигайте курсор по экрану, пока не почувствуете себя уверенно.

  2. Нажмите клавишу `вниз' (j) пока она не начнёт повторяться.
     Теперь вы знаете, как перейти к следующему уроку.

  3. Используя клавишу `вниз' перейдите к Уроку 1.2.

Замечание! Если вы пока не уверены в том, что набираете, нажмите <ESC> для
        перехода в обычный режим (Normal mode). После этого перенаберите
        требуемую команду.

Замечание! Обычные клавиши управления курсором (стрелки) также должны
        работать. Однако, клавиши hjkl позволят вам перемещаться
        значительно быстрее, как только вы научитесь ими пользоваться.
```

~~~~~~

Урок 1.2: ЗАВЕРШЕНИЕ РАБОТЫ С VIM

!! ВНИМАНИЕ! Прежде, чем выполнять любой из описанных ниже шагов, прочтите урок целиком !!

1. Нажмите клавишу <ESC> (для того, чтобы удостовериться, что вы в обычном режиме (Normal mode)).

2. Наберите: :q! <ENTER>.
   (Это означает, что вы должны набрать три символа :q! и нажать клавишу ввод.) Это позволит вам выйти из редактора БЕЗ СОХРАНЕНИЯ любых сделанных изменений.

3. Когда вы увидите приглашение командной оболочки, наберите команду, которая привела вас в этот учебник. Это может быть:
   vimtutor <ENTER>

4. Если вы уверены в том, что запомнили эти шаги, выполните шаги от 1 до 3 чтобы выйти и снова запустить редактор.

Замечание! :q! <ENTER> отбрасывает любые сделанные вами изменения. Через несколько уроков вы узнаете как сохранять изменения в файл.

5. Переместите курсор вниз к Уроку 1.3.

~~~~~~

Урок 1.3: РЕДАКТИРОВАНИЕ ТЕКСТА -- УДАЛЕНИЕ

** Находясь в обычном режиме нажмите x для удаления символа под курсором. **

1. Переместите курсор к строке ниже, помеченной --->.

2. Для исправления ошибок, переместите курсор, пока он не окажется над удаляемым символом.

3. Нажмите клавишу x для удаления требуемого символа.

4. Повторите шаги со 2 по 4 пока строка не будет исправлена.

---> От тттопота копытт пппыль ппо пппполю леттитт.

5. Теперь, когда строка откорректирована, переходите к Уроку 1.4.

Замечание! В ходе освоения этого учебника не пытайтесь запоминать, учите в процессе использования.

~~~~~~

Урок 1.4: РЕДАКТИРОВАНИЕ ТЕКСТА -- ВСТАВКА

** Находясь в обычном режиме, нажмите i для вставки текста. **

1. Переместите курсор к первой строке ниже, помеченной --->.

2. Для того, чтобы сделать первую строку идентичной второй, поместите курсор на символ ПЕРЕД которым следует вставить текст.

3. Нажмите i и наберите требуемые добавления.

4. После исправления всех ошибок нажмите <ESC> для возврата в обычный режим. Повторите шаги со 2 по 4, пока фраза не будет исправлена полностью.

---> Часть текста в строке бесследно .
---> Часть текста в этой строке бесследно пропала.

5. Когда освоите вставку текста, переходите к Уроку 1.5.
~~~~~~

~~~~~~

Урок 1.5: РЕДАКТИРОВАНИЕ ТЕКСТА -- ДОБАВЛЕНИЕ

** Находясь в обычном режиме, нажмите A для добавления текста. **

1. Переместите курсор к первой строке ниже, помеченной --->.
   Не имеет значения на каком символе расположен курсор на этой строке.

2. Нажмите A и наберите требуемые добавления.

3. После добавления требуемого текста нажмите <ESC> для возврата в обычный
   режим.

4. Переместите курсор на следующую строку, помеченную ---> и повторите шаги
   со 2 по 4 для исправления этой строки.

---> Часть текста в этой строке бессле
     Часть текста в этой строке бесследно пропала.
---> Здесь также не достаёт час
     Здесь также не достаёт части текста.

5. Когда освоите добавление текста, переходите к Уроку 1.6.

~~~~~~

Урок 1.6: РЕДАКТИРОВАНИЕ ФАЙЛА

** Используйте :wq для сохранения файла и выхода из Vim. **

!! ВНИМАНИЕ! Прежде, чем выполнять любой из описанных ниже шагов, прочтите
урок целиком !!

1. Выйдите из Vim, как вы это узнали в Уроке 1.2: :q!
   Или, если у вас есть доступ к другому терминалу, можете сделать следующее
   в нём.

2. По приглашению командной оболочки введите следующую команду:
   vim tutor <ENTER>
   `vim' -- команда для запуска редактора Vim, а `tutor' -- имя файла для
   редактирования. Используете имя файла, который можно изменять.

3. Вставляйте и удаляйте текст, как вы научились в предыдущих уроках.

4. Сохраните файл с изменениями и выйдите из Vim выполнив: :wq <ENTER>

5. Если вы вышли из vimtutor на шаге 1, перезапустите vimtutor и переходите
   к следующему Резюме.

~~~~~~

РЕЗЮМЕ УРОКА 1

1. Курсор перемещается либо клавишами со стрелками, либо клавишами hjkl.
   h (влево) j (вниз) k (вверх) l (вправо)

2. Для запуска Vim (из приглашения командной оболочки) наберите:
   vim ИМЯ_ФАЙЛА <ENTER>

3. Для завершения работы с Vim наберите:
   <ESC> :q! <ENTER> чтобы отказаться от сохранения изменений.
   Или наберите:
   <ESC> :wq <ENTER> чтобы сохранить изменения.

4. Для удаления символа под курсором в обычном режиме, нажмите: x

5. Чтобы вставить текст перед курсором в обычном режиме, наберите:
   i вводите вставляемый текст <ESC>
   Чтобы добавить текст после курсора:
   a вводите добавляемый текст <ESC>

Замечание! Нажатие <ESC> переместит вас в обычный режим (Normal mode) либо
прервёт нежелательную и частично завершённую команду.

Теперь переходите к Уроку 2.
~~~~~~

~~~~~~

Урок 2.1: КОМАНДЫ УДАЛЕНИЯ

** Наберите dw для удаления участка текста до конца слова. **

1. Нажмите <ESC>, чтобы перейти в обычный режим.

2. Переместите курсор вниз, к строке помеченной --->.

3. Переместите курсор в начало слова, которое следует удалить.

4. Наберите dw для удаления этого слова.

Замечание! Во время набора буквы dw появятся справа в самой нижней строке
экрана. Если вы что-то наберёте неправильно, нажмите <ESC> и начните
сначала.

---> Несколько слов рафинад в этом предложении автокран излишни.

5. Повторите шаги 3 и 4, пока не исправите все ошибки и переходите к
Уроку 2.2.

~~~~~~

Урок 2.2: ДОПОЛНИТЕЛЬНЫЕ КОМАНДЫ УДАЛЕНИЯ

** Наберите d$ для удаления текста до конца строки. **

1. Нажмите <ESC>, чтобы перейти в обычный режим.

2. Переместите курсор вниз, к строке помеченной --->.

3. Переместите курсор к концу правильной строки (ПОСЛЕ первой точки).

4. Наберите d$ для удаления остатка строки.

---> Кто-то набрал окончание этой строки дважды. окончание этой строки дважды.

5. Чтобы лучше разобраться в том, как это происходит, переходите к Уроку 2.3.

~~~~~~

Урок 2.3: КОМАНДЫ И ОБЪЕКТЫ

Многие команды, изменяющие текст, состоят из оператора и объекта. Формат
команды удаления с оператором d следующий:

d   объект

Здесь:
d   - оператор удаления.
объект - над чем должна быть выполнена команда (перечислено ниже).

Краткий список объектов:
w - от курсора до конца слова, включая последующий пробел.
e - от курсора до конца слова, НЕ включая последующий пробел.
$ - от курсора до конца строки.
^ - от курсора до начала строки.

Замечание! Простое нажатие на символ объекта в обычном режиме (Normal mode)
без предварительного оператора переместит курсор так, как указано
в списке объектов.
~~~~~~

~~~~~~

Урок 2.4: ИСПОЛЬЗОВАНИЕ СЧЁТЧИКА ДЛЯ ПЕРЕМЕЩЕНИЯ

** Ввод числа перед оператором перемещения приведёт к его повторению
заданное количество раз. **

1. Переместите курсор к началу строки отмеченной ---> ниже.

2. Наберите 2w для перемещения курсора вперёд к началу второго слова.

3. Наберите 3e для перемещения курсора вперёд к концу третьего слова.

4. Наберите 0 (нуль) для перемещения к началу строки.

5. Повторите шаги 2 и 3 с различными числами.

---> Обычная строка из слов для вашего перемещения по ней.

6. Переходите к Уроку 2.5.

~~~~~~

Урок 2.5: ИСПОЛЬЗОВАНИЕ СЧЁТЧИКА ДЛЯ УДАЛЕНИЯ

** Ввод числа перед оператором приведёт к его повторению
заданное количество раз. **

Добавьте число перед перед объектом в комбинацию оператора удаления и
перемещения указанную выше для удаления указанного количества объектов:
d   число   объект

1. Переместите курсор к первому слову из прописных букв в отмеченной --->
строке ниже.

2. Наберите d2w для удаления двух слов из прописных букв.

3. Повторите шаги 1 и 2 с другими числами для удаления последовательных слов
из прописных букв одной командой.

---> эта АБВ ГД строка ЕЖЗИ КЛ МНО из слов П РС ТУФ очищена.

~~~~~~

Урок 2.6: ОПЕРАЦИИ СО СТРОКАМИ

** Наберите   dd   для удаления целой строки. **

В связи с частой необходимостью удаления целой строки, создатели Vi решили
для упрощения сделать возможным удаление строки набором двух d.

1. Переместите курсор вниз, ко второй строке фразы.

2. Наберите dd для удаления строки.

3. Теперь переместитесь к четвёртой строке.

4. Наберите 2dd для удаления двух строк.

---> 1) Летом я хожу на стадион,
---> 2) О, как внезапно кончился диван!
---> 3) Я болею за ``Зенит'', ``Зенит'' --- чемпион!
---> 4) Печально я гляжу на наше поколение!
---> 5) Его грядущее иль пусто иль темно...
---> 6) Я сижу на скамейке в ложе `Б'
---> 7) И играю на большой жестяной трубе.

~~~~~~

Урок 2.7: КОМАНДА `ОТМЕНА'

** Нажмите u для отмены результата работы предыдущей команды, U для отмены
исправлений во всей строке. **

1. Переместите курсор вниз, к строке помеченной --->, и установите его на
первую ошибку.

2. Нажмите x для удаления первого неправильного символа.

3. Теперь нажмите u для отмены (отката) последней выполненной команды.

4. Исправьте все ошибки в строке, используя команду x .

5. Теперь нажмите заглавную U для того, чтобы вернуть всю строку
в исходное состояние.

6. Нажмите u несколько раз для отмены команды U и предыдущих команд.

7. Нажмите теперь CTRL-R (т.е. удерживайте клавишу CTRL нажатой в момент
нажатия клавиши R) несколько раз для возврата команд (откат отката).

---> Испрравьте оошибки в этойй строке и вернитте их сс помоцьью `отмены'.

8. Это были очень полезные команды. Далее переходите к Резюме Урока 2.

~~~~~~

РЕЗЮМЕ УРОКА 2

1. Для удаления текста от курсора до конца слова наберите: dw

2. Для удаления текста от курсора до конца строки наберите: d$

3. Для удаления всей строки наберите: dd

4. Для повтора перемещения введите количество перед командой: 2w

5. Формат команды в обычном режиме имеет вид:

   [число] команда объект ИЛИ команда [число] объект
   где:
   [число] - сколько раз повторить выполнение команды, опционально
   команда - что выполнить, например d для удаления
   объект - на что должна воздействовать команда, например w (слово),
             $ (до конца строки), и т.д.

6. Для перехода к началу строки используйте нуль: 0

7. Для отмены (отката) предшествующих действий наберите: u (строчная u)
   Для отмены (отката) всех изменений в строке наберите: U (прописная U)
   Для отмены отката наберите: CTRL-R

~~~~~~

Урок 3.1: КОМАНДА ВСТАВКИ

** Наберите p для вставки последнего удалённого текста после курсора. **

1. Переместите курсор вниз, к строке помеченной --->.

2. Наберите dd для удаления строки и её сохранения в буфере Vim'a.

3. Переместите курсор к строке НАД тем местом, куда следует вставить
   удалённую строку.

4. Находясь в обычном режиме наберите p для вставки строки ниже курсора.

5. Повторите шаги со 2 по 4, пока не расставите все строки в нужном порядке.

---> г) И лучше выдумать не мог.
---> б) Когда не в шутку занемог,
---> в) Он уважать себя заставил
---> а) Мой дядя самых честных правил

~~~~~~

Урок 3.2: КОМАНДА ЗАМЕНЫ

** Наберите r и символ, заменяющий символ под курсором. **

1. Переместите курсор вниз, к строке помеченной --->.

2. Установите курсор так, чтобы он находился над первой ошибкой.

3. Наберите r и затем символ, исправляющий ошибку.

4. Повторите шаги 2 и 3, пока первая строка не будет исправлена как вторая.

---> В момегт набтра этой чтроки кое0кто с трудом попвдал по клваишам!
---> В момент набора этой строки кое-кто с трудом попадал по клавишам!

5. Теперь переходите к Уроку 3.3.

Замечание! Помните, что вы должны учиться в процессе работы, а не просто
       запоминая.

~~~~~~

Урок 3.3: КОМАНДА ИЗМЕНЕНИЯ

** Для изменения конечной части слова наберите ce . **

1. Переместите курсор вниз, к строке помеченной --->.

2. Расположите курсор над буквой `o' в слове `сола'.

3. Наберите ce и исправьте слово (в данном случае, наберите `лов').

4. Нажмите <ESC> и переходите к следующей ошибке (к первому символу, который
   надо изменить).

5. Повторите шаги 3 и 4 пока первое предложение не станет идентичным второму.

---> Несколько сола в эьгц строке тпгшцбь редалзкуюиесвх.
---> Несколько слов в этой строке требуют редактирования.

Обратите внимание, что ce не только удаляет слово, но и переводит вас в
режим вставки.

~~~~~~

Урок 3.4: ПРОДОЛЖАЕМ ИЗМЕНЯТЬ С КОМАНДОЙ c

** Команда замены используется с теми же объектами, что и команда удаления. **

1. Команда изменения применяется таким же образом, как и команда удаления.
   Её формат таков:

      [число] c объект   ИЛИ   c [число] объект

2. Объекты также совпадают: w (слово), $ (конец строки) и т.п.

3. Переместите курсор вниз, к строке помеченной --->.

4. Перейдите к первой ошибке.

5. Наберите c$ и отредактируйте первую строку так, чтобы она совпадала со
   второй, после чего нажмите <ESC>.

---> Конец этой строки нуждается в помощи, чтобы стать похожим на второй.
---> Конец этой строки нуждается в помощи команды c$.

Замечание! Клавиша Backspace может использоваться для исправления при наборе.

~~~~~~

РЕЗЮМЕ УРОКА 3

1. Для вставки текста, который только что был удалён, наберите p . Эта
   команда вставит удалённый текст ПОСЛЕ курсора (если была удалена строка,
   то она будет помещена в строке под курсором).

2. Для замены символа под курсором наберите r и затем заменяющий символ.

3. Команда изменения позволяет вам изменить указанный объект от курсора до
   окончания перемещения. Например, наберите ce для замены от курсора до
   конца слова, c$ для изменения до конца строки.

4. Формат команды изменения таков:

      [число] c объект   ИЛИ   c [число] объект

Теперь переходите к следующему уроку.

~~~~~~

Урок 4.1: ИНФОРМАЦИЯ О ФАЙЛЕ И РАСПОЛОЖЕНИИ В НЕМ

** Наберите CTRL-g чтобы увидеть ваше месторасположение в файле и
информацию о файле. Наберите G (SHIFT-G) для перемещения к заданной
строке в файле. **

Замечание! Прочитайте весь урок прежде чем выполнять любые команды!

1. Удерживая клавишу Ctrl нажмите g . Внизу экрана появится строка статуса
   с именем файла и номером строки, в которой вы находитесь. Запомните номер
   строки, он потребуется на Шаге 3.

2. Удерживая клавишу Shift нажмите g для перемещения к концу файла.

3. Наберите номер строки, в которой вы находились и затем Shift-G. Это
   вернёт вас к строке, в которой вы были, когда в первый раз нажали Ctrl-g.

4. Если вы запомнили все вышесказанное, выполните шаги с 1 по 3.

~~~~~~

Урок 4.2: КОМАНДА ПОИСКА

** Наберите / и затем введите искомую фразу. **

1. В обычном режиме (Normal mode) наберите символ / . Обратите внимание,
   что он вместе с курсором появится внизу экрана, как это происходит с
   командой : .

2. Теперь наберите 'ошшшибка' <ENTER>. Это то слово, которое вы будете
   искать.

3. Для того, чтобы повторить поиск, просто нажмите n .
   Для поиска этой же фразы в обратном направлении, нажмите Shift-N .

4. Если вы желаете сразу искать в обратном направлении, используйте
   команду ? вместо / .

5. Для того, чтобы вернуться туда, откуда вы начали поиск нажмите Ctrl-O.
   (Удерживая нажатой клавишу Ctrl нажмите o). Повторите несколько раз
   для дальнейшего перехода. Для перехода вперёд используйте Ctrl-I .

---> "ошшшибка" это не способ написания слова `ошибка'; ошшшибка это ошибка.

Замечание! Если при поиске будет достигнут конец файла, то поиск будет продолжен
с начала.

~~~~~~

Урок 4.3: ПОИСК ПАРНЫХ СКОБОК

** Наберите % для поиска парных),] или } . **

  1. Поместите курсор над любой из (, [или { в строке ниже, помеченной --->.

  2. Теперь наберите символ % .

  3. Курсор должен перескочить на парную скобку.

  4. Наберите % для возврата курсора назад к первой скобке.

---> Это (строка, содержащая такие (, такие [ ] и такие { } скобки.))

Замечание! Это очень удобно при отладке программ с пропущенными скобками!

~~~~~~

Урок 4.4: СПОСОБ ИСПРАВЛЕНИЯ ОШИБОК

** Наберите :s/было/стало/g для замены 'было' на 'стало'. **

  1. Переместите курсор вниз, к строке помеченной --->.

  2. Наберите :s/уводю/увожу <ENTER> . Обратите внимание на то, что эта
     команда заменит только первое найденное вхождение в строке.

  3. Теперь наберите :s/уводю/увожу/g , добавленная в конце g означает
     подстановку глобально во всей строке. Это заменит все найденные в строке
     вхождения.

---> Я уводю к отверженным селеньям, я уводю сквозь вековечный стон, я уводю к
     забытым поколеньям.

  4. Для замены всех вхождений последовательности символов между двумя
     строками,
     наберите :#,#s/было/стало/g где #,# -- номера этих строк.
     Наберите :%s/было/стало/g   для замены всех вхождений во всем файле.
     Наберите :%s/было/стало/gc   для поиска всех вхождений во всем файле и
                                   запроса подтверждения замены.

~~~~~~

РЕЗЮМЕ УРОКА 4

  1. Ctrl-g показывает ваше положение в файле и информацию о нем.
     Shift-G перемещает вас в конец файла. Номер, за которым следует Shift-G
     позволяет перейти к строке с этим номером.
     gg перемещает вас к первой строке файла.

  2. Нажатие / и затем ввод строки позволяет произвести поиск этой строки
     ВПЕРЁД по тексту.
     Нажатие ? и затем ввод строки позволяет произвести поиск этой строки
     НАЗАД по тексту.
     После поиска наберите n для перехода к следующему вхождению искомой
     строки в том же направлении или Shift-N для перехода в противоположном
     направлении.

  3. Нажатие % , когда курсор находится на (,),[,],{, или } позволяет найти
     парную скобку.

  4. Для подстановки `стало' вместо первого `было' в строке, наберите
        :s/было/стало
     Для подстановки `стало' вместо всех `было' в строке, наберите
        :s/было/стало/g
     Для замены в интервале между двумя строками, наберите
        :#,#s/было/стало/g
     Для замены всех вхождений `было' на `стало' в файле, наберите
        :%s/было/стало/g
     Чтобы редактор каждый раз запрашивал подтверждение, добавьте 'c'
        :%s/было/стало/gc
~~~~~~

~~~~~~

Урок 5.1: КАК ВЫПОЛНИТЬ ВНЕШНЮЮ КОМАНДУ

** Наберите :! и затем внешнюю команду, которую следует выполнить. **

1. Наберите уже знакомую вам команду : для установки курсора в командную строку редактора. Это позволит вам ввести команду.

2. Теперь наберите символ ! (восклицательный знак). Это позволит выполнить внешнюю команду, используя командную оболочку.

3. Для примера наберите ls после ! и нажмите <ENTER>. Команда выведет список файлов в текущем каталоге, точно также, как если бы вы ввели эту команду в приглашении оболочки. Или попробуйте :!dir , если команда ls не сработала.

Замечание! Таким способом можно выполнить любую внешнюю команду с указанием аргументов.

Замечание! Все команды, начинающиеся с : , должны завершаться нажатием <ENTER>. Далее на это не всегда будет обращаться особое внимание.

~~~~~~

Урок 5.2: КАК ЗАПИСАТЬ ФАЙЛ

** Для сохранения изменений, произведённых в файле,
наберите :w ИМЯ_ФАЙЛА. **

1. Наберите :!dir или :!ls для получения списка файлов в текущем каталоге. Как вам уже известно, после ввода команды надо нажать <ENTER>.

2. Придумайте название для файла, которое ещё не существует, например TEST.

3. Теперь наберите :w TEST (где TEST -- это имя файла, придуманное вами.)

4. Команда сохранит весь этот файл (Учебник по Vim) под именем TEST. Чтобы удостовериться в этом, снова наберите :!dir или :!ls и просмотрите каталог.

Замечание! Если вы выйдете из Vim и затем запустите его снова с файлом TEST (т.е. выполните vim TEST), этот файл будет точной копией учебника в тот момент, когда вы его сохранили.

5. Теперь удалите этот файл, набрав для MS-DOS   :!del TEST
для Unix   :!rm TEST

~~~~~~

Урок 5.3: ВЫБОРОЧНОЕ СОХРАНЕНИЕ

** Для сохранения части файла, наберите v выберите часть
и сохраните её :w ИМЯ_ФАЙЛА **

1. Переместите курсор к этой строке.

2. Нажмите v и переместите курсор ниже к пятому шагу. Обратите внимание, что текст подсвечен.

3. Нажмите : и внизу экрана появится :'<,'> .

4. Введите w TEST (где TEST -- имя файла, который ещё не существует).
До нажатия <ENTER>, проверьте что внизу экрана написано :'<,'>w TEST.

5. Vim запишет выбранные строки в файл TEST. Как и прежде, убедитесь в наличии этого файла командой :!dir или :!ls . НЕ УДАЛЯЙТЕ этот файл, он потребуется в следующем уроке.

Замечание! Нажатие v начинает визуальный выбор. Вы можете перемещать курсор для изменения выбора. Затем для выбранного фрагмента можно выполнить какой-то оператор, например, удалить нажатием d.
~~~~~~

~~~~~~

Урок 5.4: ЧТЕНИЕ И ОБЪЕДИНЕНИЕ ФАЙЛОВ

** Для вставки содержимого из файла, наберите :r ИМЯ_ФАЙЛА **

1. Установите курсор над этой строкой.

Замечание! После выполнения Шага 2 вы увидите текст из Урока 5.3. Переместитесь ВНИЗ по тексту до этого урока.

2. Теперь прочитайте ваш файл TEST, используя команду :r TEST , где TEST -- это имя файла.

3. Для проверки что содержимое файла было вставлено, переместитесь по тексту и удостоверьтесь, что теперь в нём две копии Урока 5.3: исходная и из файла TEST.

Замечание! Вставить можно и вывод внешней команды. Например, :r !ls прочитает вывод команды ls и вставит его ниже курсора.

~~~~~~

РЕЗЮМЕ УРОКА 5

1. :!команда исполняет внешнюю команду.

    Некоторые полезные примеры:
      (MS-DOS)   (Unix)
       :!dir   :!ls   -- вывести список файлов в каталоге.
       :!del ИМЯ   :!rm ИМЯ   -- удалить файл по имени.

2. :w ИМЯ_ФАЙЛА записывает текущий редактируемый в Vim файл на диск под указанным именем.

3. v перемещение :w ИМЯ_ФАЙЛА сохраняет визуально выбранные строки в файл с указанным именем.

4. :r ИМЯ_ФАЙЛА считывает с диска файл с указанным именем и помещает его ниже курсора.

5. :r !dir читает вывод команды dir и помещает его ниже курсора.

~~~~~~

Урок 6.1: КОМАНДА СОЗДАНИЯ

** Наберите o чтобы создать пустую строку под курсором и перейти в режим вставки (Insert mode) **

1. Переместите курсор вниз, к строке помеченной --->.

2. Наберите o (в нижнем регистре) для того, чтобы создать пустую строку НИЖЕ курсора и перейти в режим вставки (Insert mode).

3. Теперь наберите какой-нибудь текст и нажмите <ESC> для выхода из режима вставки.

---> После нажатия o курсор перейдёт на новую пустую строку в режиме вставки.

4. Для создания строки ВЫШЕ курсора, просто наберите заглавную O , вместо строчной o . Попробуйте проделать это с нижеследующей строкой.

---> Создайте новую строку над этой, поместив на неё курсор и нажав Shift-O.

~~~~~~

Урок 6.2: КОМАНДА ДОБАВЛЕНИЯ

** Наберите a , чтобы вставить текст ПОСЛЕ курсора. **

1. Переместите курсор вниз, в начало первой строки помеченной --->.

2. Набирайте e пока курсор не переместиться на конец стро .

3. Наберите a (в нижнем регистре) для добавления текста ПОСЛЕ символа, находящегося под курсором. (Заглавная A позволяет добавить в конец строки.)

4. Допишите слово так, как показано в строке ниже. Нажмите <ESC> для выхода из режима вставки (Insert mode).

5. Используйте e для перехода к концу следующего незавершённого слова и повторите шаги 3 и 4.

---> Эта стро позволит вам попрактиков в добавле текста.
---> Эта строчка позволит вам попрактиковаться в добавлении текста.

Замечание! a , i и A переводят в один и тот же режим вставки, различие только в том, где вставляются символы.

~~~~~~

Урок 6.3: ЕЩЁ ОДИН СПОСОБ ЗАМЕНЫ

** Наберите заглавную R для замены более чем одного символа. **

1. Переместите курсор вниз, к первой строке помеченной --->, и в начало
   первого слова xxx.

2. Теперь нажмите R и введите число, указанное ниже во второй строке, чтобы
   заменить xxx.

3. Нажмите <ESC> для выхода из режима замены. Заметьте, что остаток строки
   не был изменён.

4. Повторите эти шаги для замены оставшихся xxx.

---> Добавление 123 к xxx даёт xxx.
---> Добавление 123 к 456 даёт 579.

Замечание! Режим замены похож на режим вставки, но каждый введённый символ
удаляет существующий.

~~~~~~

Урок 6.4: КОПИРОВАНИЕ И ВСТАВКА ТЕКСТА

** Используйте команду y для копирования и p для вставки **

1. Переместите курсор вниз, к первой строке помеченной --->, и после `а)'.

2. Переключитесь в режим визуального выбора нажав v и переместите курсор
   перед словом `первый'.

3. Введите y для копирования подсвеченного текста.

4. Переместите курсор в конец следующей строки комбинацией j$.

5. Нажмите p для вставки текста. Затем введите `второй' и нажмите <ESC>.

---> а) Этот элемент первый.
      б)

Замечание! Также возможно использовать yw (команду y с оператором w) для
копирования одного слова.

~~~~~~

Урок 6.5: УСТАНОВКА ПАРАМЕТРОВ

** Установка параметра для игнорирования регистра при поиске или замене **

1. Найдите слово `игнорировать', набрав: /игнорировать <ENTER>.
   Повторите поиск несколько раз, нажимая клавишу n .

2. Установите параметр `ic' (игнорировать регистр), набрав: :set ic

3. Теперь снова несколько раз сделайте поиск слова `игнорировать',
   нажимая: n
   Заметьте, что теперь находятся `Игнорировать' и `ИГНОРИРОВАТЬ'.

4. Установите параметры `hlsearch' и `incsearch': :set hls is

5. Теперь опять введите команду поиска и посмотрите, что получится:
                        /игнорировать <ENTER>

6. Для возвращения учёта регистра при поиске наберите: :set noic

Замечание! Для отключения подсветки совпадений наберите: :nohlsearch

Замечание! Если вы хотите игнорировать регистр только для одного поиска,
используйте \c в команде поиска: /игнорировать\c <ENTER>
~~~~~~

~~~~~~

РЕЗЮМЕ УРОКА 6

1. Нажмите o для создания строки НИЖЕ курсора и перехода в режим вставки.
   Нажмите O для создания строки ВЫШЕ курсора.

2. Нажмите a для вставки текста ПОСЛЕ курсора.
   Нажмите A для вставки текста в конец строки.

3. Команда e подводит курсор к концу слова.

4. Команда y копирует текст, p -- вставляет скопированный текст.

5. Нажатие заглавной R переводит в режим замены до нажатия клавиши <ESC> .

6. Наберите `:set xxx' для включения параметра `xxx', некоторые параметры:

      `ic'    `ignorecase'   игнорирование регистра при поиске
      `is'    `incsearch'   отображение частичных совпадений при поиске
      `hls'   `hlsearch'   подсветка всех совпадений при поиске

7. Добавьте `no' перед параметром для его отключения: :set noic

~~~~~~

Урок 7.1: ВСТРОЕННАЯ СПРАВКА

** Используйте встроенную справочную систему **

Vim обладает мощной встроенной справочной системой. Для начала попробуйте
один из трёх вариантов:
      - нажмите клавишу <HELP> (если таковая имеется на клавиатуре)
      - нажмите клавишу <F1> (если таковая имеется на клавиатуре)
      - наберите   :help <ENTER>

Прочитайте текст в окне справки для получения представления о том как
работает справка.
Нажмите CTRL-W CTRL-W для перехода от окна к окну.
Наберите   :q <ENTER>   чтобы закрыть окно справки.

Вы можете найти справку для любого понятия или команды, задав соответствующий
аргумент команде `:help'. Попробуйте следующее (не забудьте нажать <ENTER>):

      :help w
      :help c_CTRL-D
      :help insert-index
      :help user-manual

~~~~~~

Урок 7.2: СОЗДАНИЕ СТАРТОВОГО СЦЕНАРИЯ

** Включим возможности Vim **

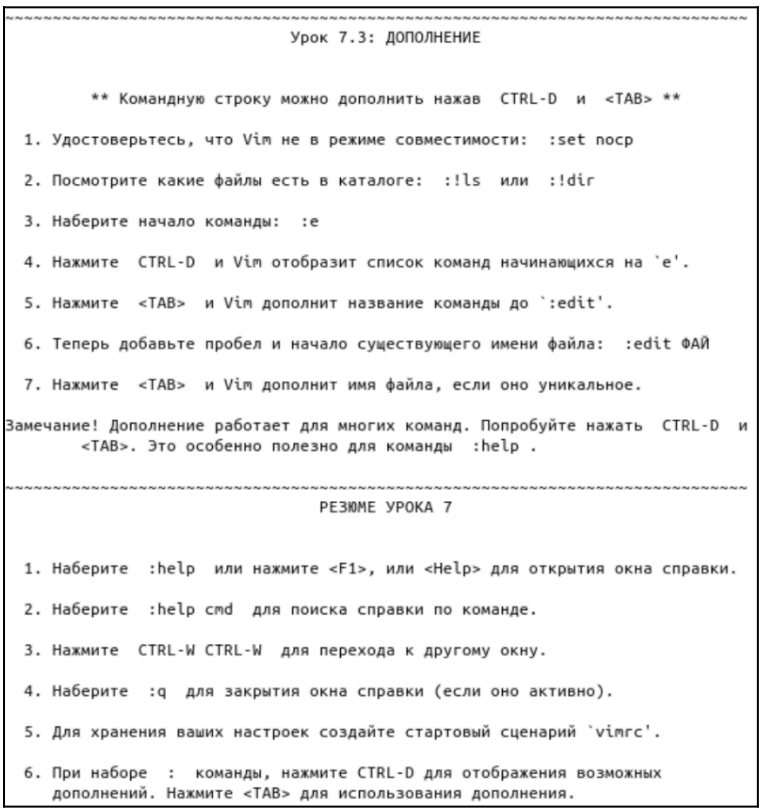
Vim имеет намного больше возможностей, чем Vi, но большинство из них по
умолчанию выключены. Для использования больших возможностей вам следует
создать файл `vimrc'.

1. Отредактируйте новый файл `vimrc'. Его расположение зависит от
   используемой системы:
      :e ~/.vimrc   для Unix
      :e ~/_vimrc   для MS-Windows

2. Теперь прочитайте пример файла `vimrc':
      :r $VIMRUNTIME/vimrc_example.vim

3. Запишите созданный вами новый файл `vimrc':
      :w

Теперь при следующем запуске Vim будет включена подсветка синтаксиса. Все
настройки, предпочитаемые вами, могут быть добавлены в файл `vimrc'.
Для дальнейшей информации наберите :help vimrc-intro
~~~~~~



```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                              Урок 7.3: ДОПОЛНЕНИЕ


       ** Командную строку можно дополнить нажав  CTRL-D  и  <TAB> **

  1. Удостоверьтесь, что Vim не в режиме совместимости:  :set nocp

  2. Посмотрите какие файлы есть в каталоге:  :!ls  или  :!dir

  3. Наберите начало команды:  :e

  4. Нажмите  CTRL-D  и Vim отобразит список команд начинающихся на `e'.

  5. Нажмите  <TAB>  и Vim дополнит название команды до `:edit'.

  6. Теперь добавьте пробел и начало существующего имени файла:  :edit ФАЙ

  7. Нажмите  <TAB>  и Vim дополнит имя файла, если оно уникальное.

Замечание! Дополнение работает для многих команд. Попробуйте нажать  CTRL-D  и
       <TAB>. Это особенно полезно для команды  :help .

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                РЕЗЮМЕ УРОКА 7


  1. Наберите  :help  или нажмите <F1>, или <Help> для открытия окна справки.

  2. Наберите  :help cmd  для поиска справки по команде.

  3. Нажмите  CTRL-W CTRL-W  для перехода к другому окну.

  4. Наберите  :q  для закрытия окна справки (если оно активно).

  5. Для хранения ваших настроек создайте стартовый сценарий `vimrc'.

  6. При наборе  :  команды, нажмите CTRL-D для отображения возможных
     дополнений. Нажмите <TAB> для использования дополнения.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

  На этом завершается Учебник Vim. Он был предназначен дать общее представление
  о редакторе Vim, достаточное для того, чтобы с лёгкостью использовать его.
  Учебник далёк от полноты, поскольку Vim имеет очень много команд. Прочитайте
  теперь руководство пользователя: `:help user-manual'.

  Для дальнейшего чтения рекомендуется книга:
        Vim - Vi Improved, автор: Steve Oualline, издатель: New Riders

  Эта книга полностью посвящена Vim. Особенно полезна она будет новичкам.
  Содержит множество примеров и иллюстраций.
  См. https://iccf-holland.org/click5.html

  Следующая книга более почтенного возраста и посвящена больше Vi, чем Vim,
  однако также рекомендуется:
        Learning the Vi Editor, автор: Linda Lamb,
                                издатель: O'Reilly & Associates Inc.

  Это хорошая книга, чтобы узнать всё, что только можно сделать в Vi. Шестое
  издание также включает информацию о Vim.

  Этот учебник написал Michael C. Pierce и Robert K. Ware, Colorado School of
  Mines с использованием идей, которые предложил Charles Smith, Colorado State
  University. E-mail: bware@mines.colorado.edu.

  Доработано для Vim Брамом Моленаром (Bram Moolenaar).

  Перевод:
        Андрей Киселев <a_kissel@eudoramail.com>, 2002.
        Сергей Алёшин <alyoshin.s@gmail.com>, 2014.

  Translators:
        Andrey Kiselev <a_kissel@eudoramail.com>, 2002.
        Sergey Alyoshin <alyoshin.s@gmail.com>, 2014.
```

Одновременное редактирование нескольких файлов в vim

Чтобы открыть несколько файлов, просто передайте их в параметры при запуске программы:

guest@localhost ~> vim one two three

Редактор vim откроет первый файл, для переключения ко второму используйте команду :n, чтобы вернутся назад :N.

С помощью команды :buffers вы можете посмотреть все открытые файлы, а командой :buffer 3 переключится на третий файл.

```
:buffers
  1      "one"                          строка 1
  2 %a   "two"                          строка 1
  3 #    "three"                        строка 1
Нажмите ENTER или введите команду для продолжения
```

Использование буфера обмена в vim

Текстовый редактор vim имеет свой буфер обмена. Например, вам нужно скопировать в четыре строки и вставить их в другое место программы, для этого выполните такую последовательность действий:

– Создайте в vim файл под названием clipboard (guest@localhost ~> vim clipboard);

– Введите в пустой файл следующие данные:

– Нажмите <Esc>, чтобы перейти в командный режим;

– Наберите 4yy чтобы скопировать четыре строки;

– Переместите курсор в место, где нужно вставить эти строки;

– Нажмите p для вставки.

Также можно использовать выделение vim, чтобы скопировать строки. Выделите текст с помощью v, а затем нажмите y, чтобы скопировать.

В качестве проверки знаний попробуйте скопировать написанную вами строку из файла one в файл two, используя общий буфер обмена.

Использование кириллической раскладки при вводе команд в vim

Сама кириллица в программе vim работает просто отлично, но есть один нюанс. Когда включена кириллица в системе, все команды vim не работают, так как они не приспособлены для кириллицы.

Переключать каждый раз раскладку, когда работаете в командном режиме тоже не очень удобно, поэтому открываем файл ~/.vimrc и добавляем туда такие строки:

guest@localhost ~> vim ~/.vimrc

```
set keymap=russian-jcukenwin
set iminsert=0
set imsearch=0
```

[esc]
[:]
wq
guest@localhost ~> vim one
[esc]
[i]
[ctrl+6]

```
Привет, мир!
```

[esc]
[:]
wq
guest@localhost ~> cat one

Теперь раскладка клавиатуры в командном режиме переключается по <Ctrl> + <6> и <Ctrl> + <Shift> + <6> и все команды работают.

В этом гайде мы рассмотрели, как пользоваться текстовым редактором vim. Это еще далеко не все его возможности, но теперь вы можете уверенно обращаться с редактором и забыть о nano.

Файловый менеджер ranger

Программа ranger – это минималистичный консольный файловый менеджер, чье взаимодействие частично реализовано в стиле vim. Каждый каталог в ranger отображается как отдельная панель. При переходе в новый каталог, он открывается (как следующий уровень иерархии) справа от текущего каталога. Текущий список файлов в каталоге отображается в самой правой части (в правой панели).

Основные возможности и особенности программы ranger:

– Минималистичный интерфейс;

– Быстрое переключение между директориями и просмотр файлов;

– Просмотр содержимого файлов;

– Просмотр изображений прямо в файловом менеджере (может потребоваться установка дополнительных пакетов);

– Поддержка основных операций с файлами (создание, копирование, перемещение, удаление, chmod и так далее);

– Массовое переименование файлов;

– Консоль в стиле vim;

– Поддержка горячих клавиш из vim;

– Поддержка вкладок;

– Поддержка закладок;

– Поставляется с утилитой rifle, которая предназначена для открытия (запуска) файлов. Она автоматически определяет программу, которая требуется для открытия выбранного файла;

– Поддержка цветовых схем;

– Поддержка UTF-8;

– Изменение директории оболочки после закрытия программы;

– Поддержка плагинов.

Установка и запуск ranger

Давайте установим эту программу:

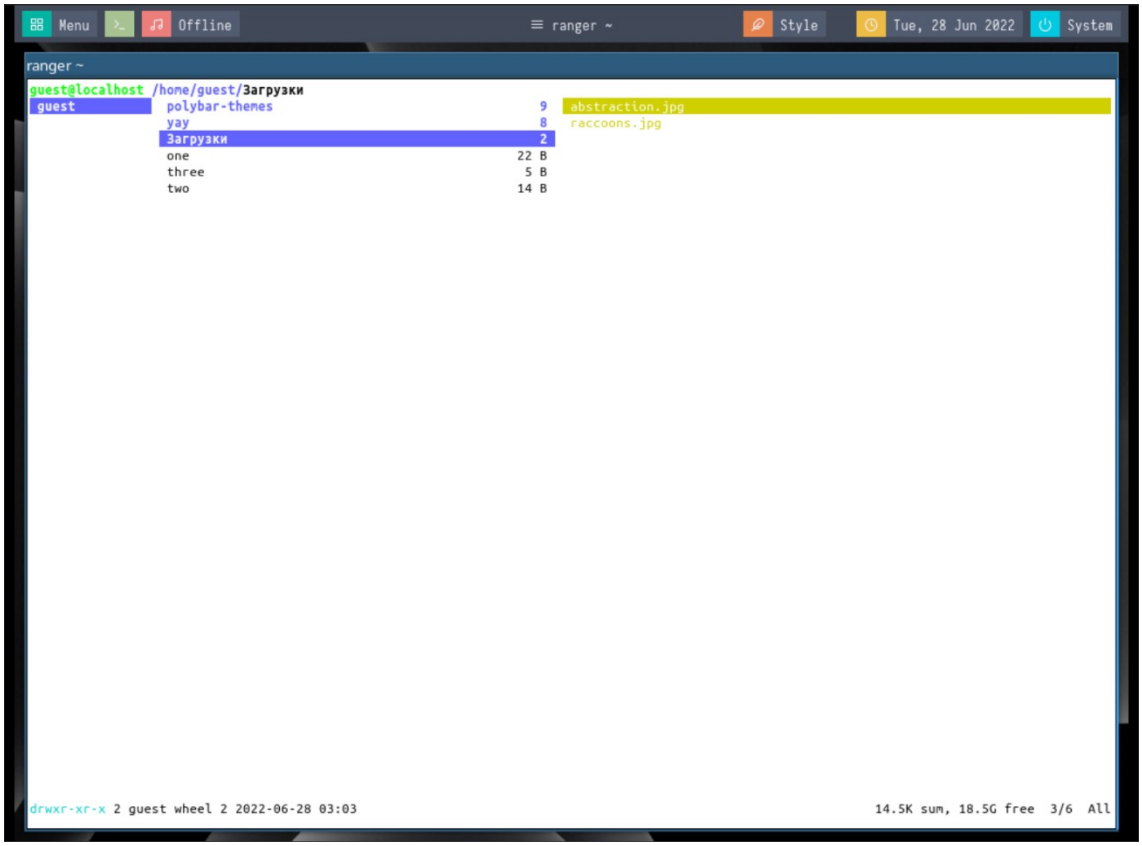
guest@localhost ~> sudo pacman -S ranger (или sudo apt install ranger в случае дистрибутивов, основанных на Debian)

В качестве запасного варианта можете установить еще графический файловый менеджер Nautilus на случай, если вы не сможете что-нибудь сделать в ranger:

guest@localhost ~> yay -S Nautilus

После установки вы можете запустить утилиту из терминала, используя следующую команду:

guest@localhost ~> ranger



[q] (Нажмите, чтобы выйти из ranger. Убедитесь, что у вас стоит английская раскладка клавиатуры, иначе горячие клавиши работать не будут.)

Файловый менеджер использует наш текущий рабочий каталог в качестве отправной точки (в нашем случае /home/guest). Однако мы можем явно указать начальный каталог, передав его в качестве аргумента при вызове файлового менеджера; чтобы открыть рейнджер и использовать каталог ~/Загрузки в качестве отправной точки, мы должны запустить:

guest@localhost ~> ranger ~/Загрузки
[q]

Одна важная вещь, которую мы должны сделать при первом использовании рейнджера, это скопировать файлы конфигурации по умолчанию в локальный каталог ~/.config/ranger; таким образом мы сможем персонализировать рейнджера без нужды в привилегиях администратора. Мы можем выполнить действие, запустив:

guest@localhost ~> ranger --copy-config=all

Команда вернет следующий вывод, который подтверждает, что файлы были скопированы:

```
creating: /home/guest/.config/ranger/rifle.conf
creating: /home/guest/.config/ranger/commands.py
creating: /home/guest/.config/ranger/commands_full.py
creating: /home/guest/.config/ranger/rc.conf
creating: /home/guest/.config/ranger/scope.sh
```

Какой файл за что отвечает? Ответ дан в таблице ниже:

Файл	**Функция**
rifle.conf	Конфигурация для rifle. В этом файле вы определяете какой тип файла какой программой открывать.
commands.py	Модуль Python, определяющий консольные команды ranger.
commands_full.py	Игнорируется: используется только в качестве референса для пользовательских команд.
rc.conf	Содержит сочетания клавиш и настройки ranger.
scope.sh	Определяет, как переопределить просмотрщики для выбранных типов файлов.

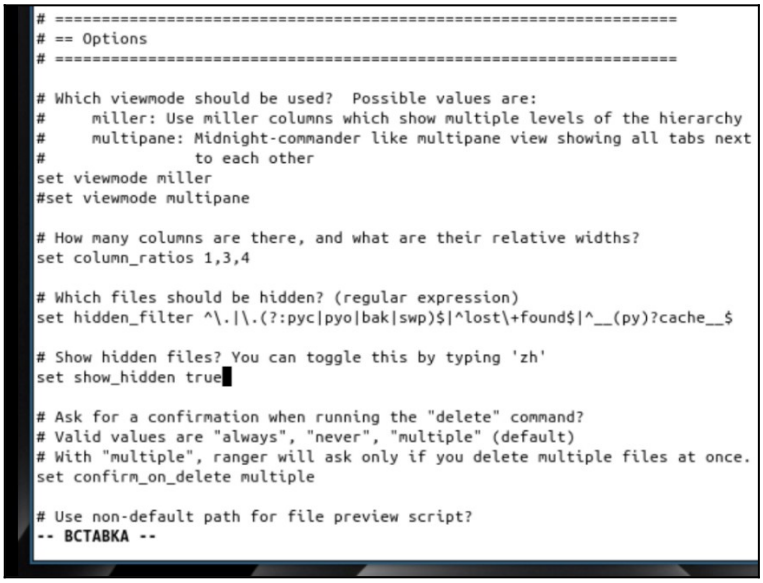
Визуализация скрытых файлов в ranger

По умолчанию скрытые файлы и каталоги не отображаются (это те, имя которых начинается с точки). Чтобы визуализировать их, мы должны написать соответствующую команду. Как и в vim, мы нажимаем : и пишем следующее:

```
[:]
set show_hidden true
[q]
```

Приведенной выше командой мы устанавливаем для параметра show_hidden значение true. Изменение вступит в силу, как только мы нажмем <Enter>, однако оно не сохранится, когда приложение будет закрыто. Чтобы сделать эти и другие настройки постоянными, мы должны прописать их в «основном» конфигурационном файле ~/.config/ranger/rc.conf:

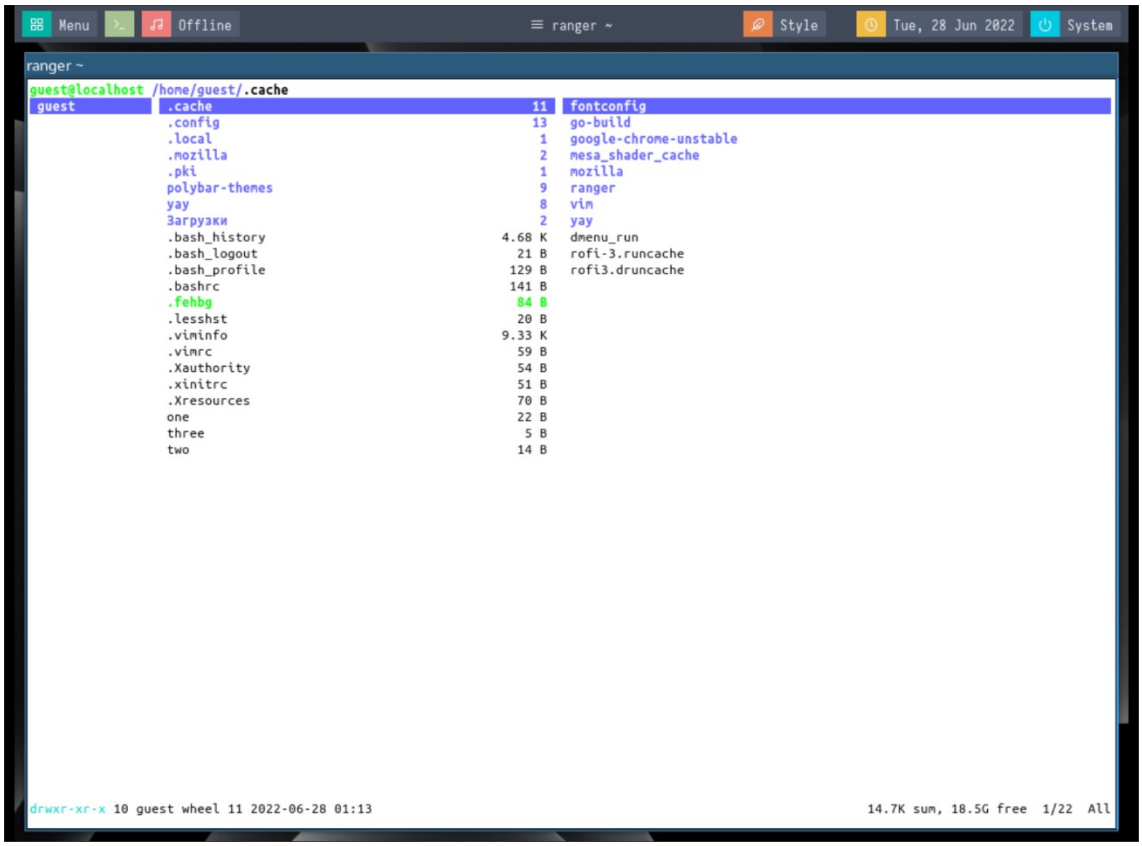
```
guest@localhost ~> vim ~/.config/ranger/rc.conf
```



[:]
wq

Базовое перемещение и сочетания клавиш в ranger

Комбинации клавиш ranger вдохновлены текстовым редактором vim; для перемещения вверх и вниз по списку каталогов и файлов, например, мы можем использовать клавиши k и j соответственно. После выбора каталога его содержимое отображается в крайнем правом столбце ranger. Вместо этого текущий родительский каталог отображается в крайнем левом:



Чтобы войти в каталог, после того как он выбран, мы можем нажать клавишу <Enter>, нажать l или использовать клавишу со стрелкой вправо. Вместо этого, чтобы перейти в его родительский каталог, мы можем нажать клавишу h[5]. Точно так же, как мы это делаем в vim, мы можем указать количество движений, которые нужно выполнить, перед клавишей: например, чтобы сдвинуться вниз на два раза, мы можем нажать 2j. Нажав G, мы переместимся в конец списка, а нажав вместо этого gg мы переместимся наверх.

Копирование, перемещение и удаление файлов в ranger

Чтобы скопировать файл при использовании ranger, все, что нам нужно сделать, это выбрать файл и нажать yy.

Чтобы в свою очередь вставить файл, вместо этого мы можем нажать p.

Чтобы переместить файлы, мы нажимаем dd, чтобы «вырезать» их, а затем p, чтобы вставить их в нужное место.

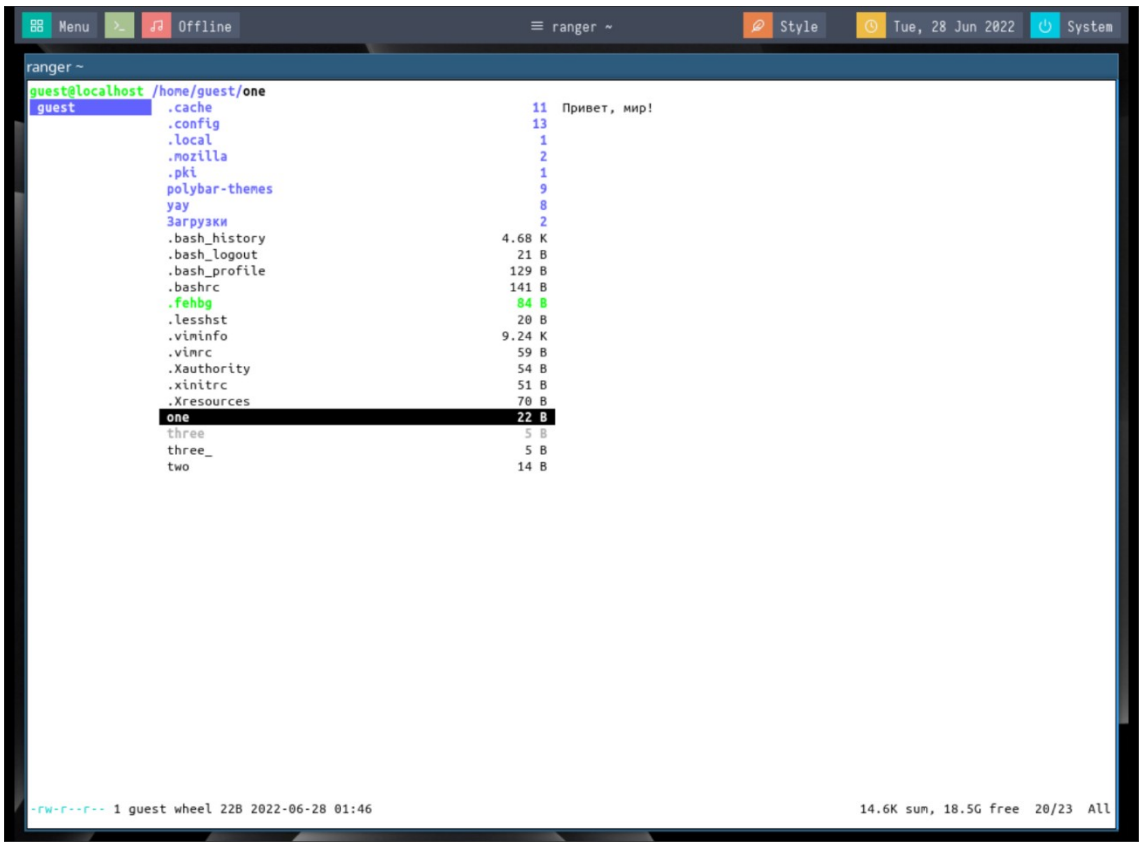
Наконец, делаем удаление файла, нажимаем dD.

Все эти действия также можно запустить, выполнив соответствующие команды: :copy, :paste, :rename *новое_имя* и :delete соответственно.

Получение предварительного просмотра документа в ranger

Как мы видели, когда мы выбираем каталог, мы можем визуализировать его содержимое в крайнем левом столбце приложения. Таким же образом мы можем получить предварительный просмотр существующего текстового файла, когда он выбран. На картинке ниже мы видим содержимое файла ~/one:

[5] Как мы помним, h и l – это клавиши, используемые в vim для перемещения влево и вправо в теле документа в обычном режиме.



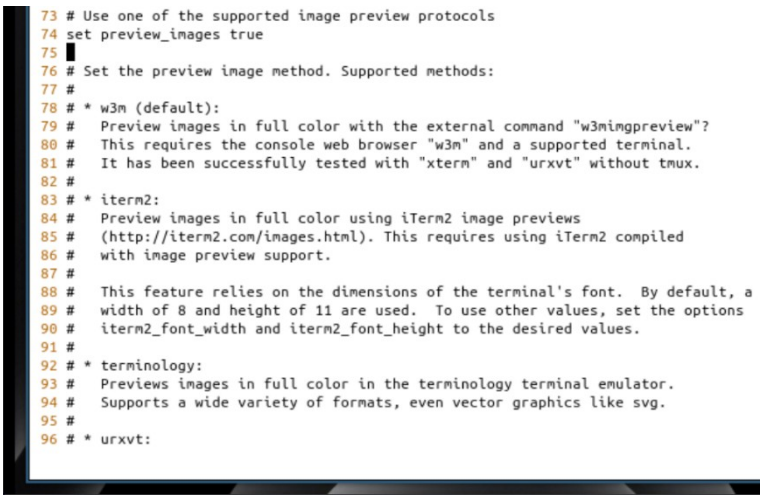
На определенных эмуляторах терминала, таких как «rxvt-unicode» или «xterm» (наш случай), а также с помощью специальных внешних программ и утилит ranger может отображать предварительный просмотр файлов других типов. Давайте посмотрим на некоторые примеры.

Предварительный просмотр PDF-файлов и изображений в ranger

По умолчанию файлы формата PDF «предварительно просматриваются» как текст; с установленными пакетами pdftoppm (часть пакета poppler-utils, в Arch Linux он называется poppler) и w3m-img (в Arch Linux он называется w3m), однако ranger может просматривать их как изображения.

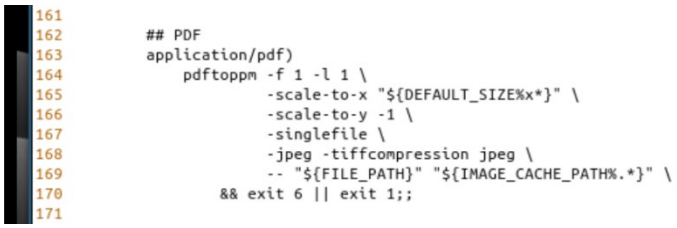
Чтобы эта функция работала, для параметра preview_images должно быть установлено значение true, и мы должны внести некоторые изменения в файл scope.sh. Этот файл представляет собой простой сценарий оболочки, используемый для определения того, как обрабатывать различные расширения файлов. Что мы хотим сделать, так это раскомментировать строки с 163 по 170:

```
guest@localhost ~> sudo pacman -S poppler
guest@localhost ~> sudo pacman -S w3m
guest@localhost ~> vim ~/.config/ranger/rc.conf
[:]
set number
```



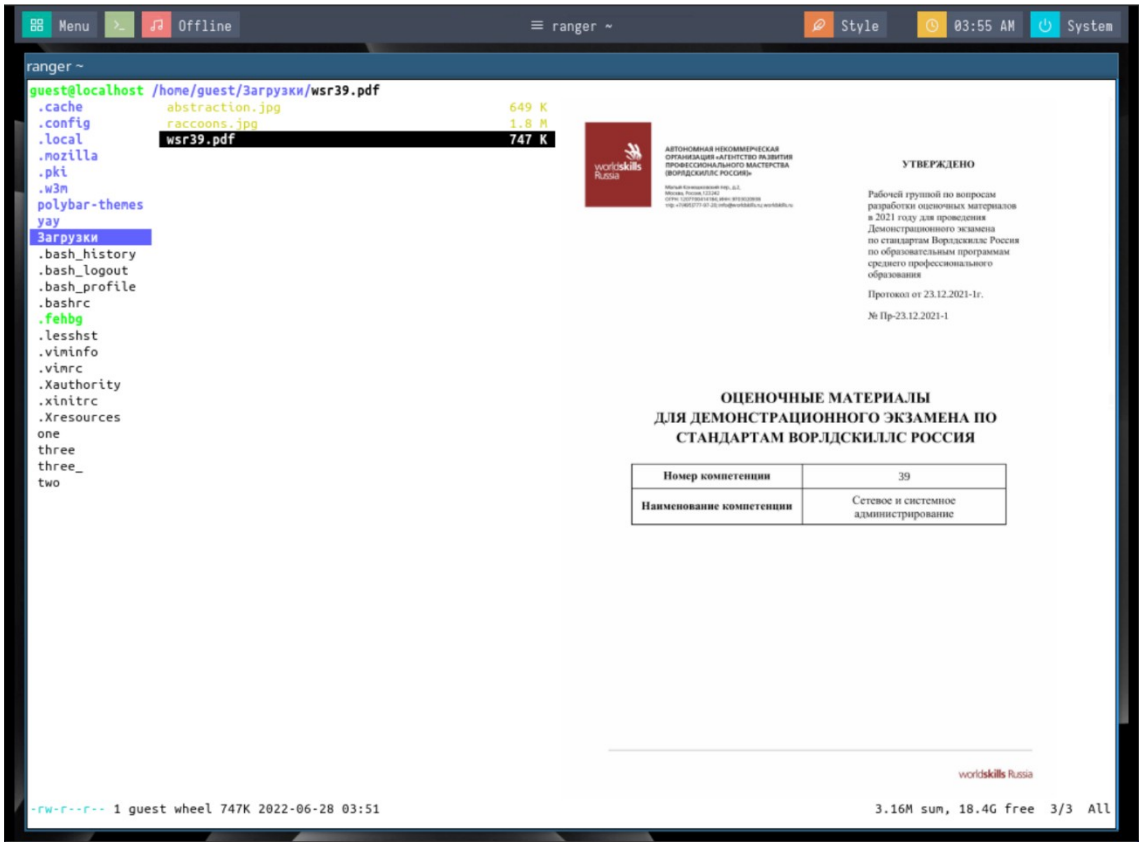
```
# Use one of the supported image preview protocols
set preview_images true

# Set the preview image method. Supported methods:
#
# * w3m (default):
#   Preview images in full color with the external command "w3mimgpreview"?
#   This requires the console web browser "w3m" and a supported terminal.
#   It has been successfully tested with "xterm" and "urxvt" without tmux.
#
# * iterm2:
#   Preview images in full color using iTerm2 image previews
#   (http://iterm2.com/images.html). This requires using iTerm2 compiled
#   with image preview support.
#
#   This feature relies on the dimensions of the terminal's font.  By default, a
#   width of 8 and height of 11 are used.  To use other values, set the options
#   iterm2_font_width and iterm2_font_height to the desired values.
#
# * terminology:
#   Previews images in full color in the terminology terminal emulator.
#   Supports a wide variety of formats, even vector graphics like svg.
#
# * urxvt:
```

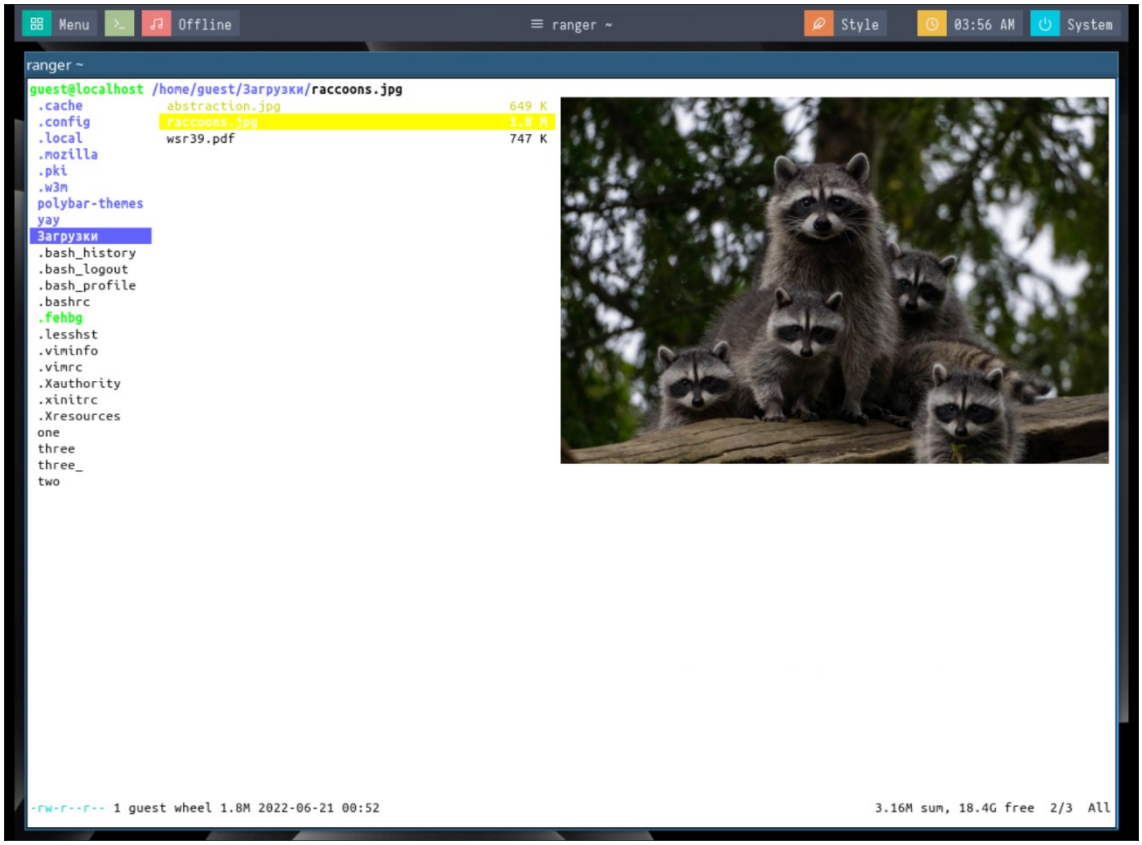
guest@localhost ~> vim ~/.config/ranger/rc.conf
[:]
set number



```

        ## PDF
        application/pdf)
            pdftoppm -f 1 -l 1 \
                     -scale-to-x "${DEFAULT_SIZE%x*}" \
                     -scale-to-y -1 \
                     -singlefile \
                     -jpeg -tiffcompression jpeg \
                     -- "${FILE_PATH}" "${IMAGE_CACHE_PATH%.*}" \
                && exit 6 || exit 1;;

```

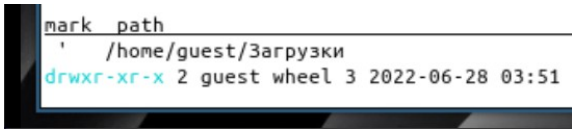
Если все настроено правильно, первая страница PDF-файла должна отображаться в качестве предварительного просмотра, когда мы ее выбираем. С установленным пакетом w3m-img ranger также будет отображать превью изображений. Скачайте любой PDF-файл, чтобы это проверить.


Menu
Offline
ranger ~
Style
03:55 AM
System
ranger ~
guest@localhost /home/guest/Загрузки/wsr39.pdf
.cache
.config
.local
.mozilla
.pki
.w3m
polybar-themes
yay
Загрузки
.bash_history
.bash_logout
.bash_profile
.bashrc
.fehbg
.lesshst
.viminfo
.vimrc
.Xauthority
.xinitrc
.Xresources
one
three
three_
two
abstraction.jpg 649 K
raccoons.jpg 1.8 M
wsr39.pdf 747 K
worldskills Russia
АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ «АГЕНТСТВО РАЗВИТИЯ ПРОФЕССИОНАЛЬНОГО МАСТЕРСТВА (ВОРЛДСКИЛЛС РОССИЯ)»
УТВЕРЖДЕНО
Рабочей группой по вопросам разработки оценочных материалов в 2021 году для проведения Демонстрационного экзамена по стандартам Ворлдскиллс Россия по образовательным программам среднего профессионального образования
Протокол от 23.12.2021-1г.
№ Пр-23.12.2021-1
ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ДЛЯ ДЕМОНСТРАЦИОННОГО ЭКЗАМЕНА ПО СТАНДАРТАМ ВОРЛДСКИЛЛС РОССИЯ
Номер компетенции 39
Наименование компетенции Сетевое и системное администрирование
worldskills Russia
-rw-r--r-- 1 guest wheel 747K 2022-06-28 03:51
3.16M sum, 18.4G free 3/3 All


Menu
Offline
ranger ~
Style
03:56 AM
System
ranger ~
guest@localhost /home/guest/Загрузки/raccoons.jpg
.cache
.config
.local
.mozilla
.pki
.w3m
polybar-themes
yay
Загрузки
.bash_history
.bash_logout
.bash_profile
.bashrc
.fehbg
.lesshst
.viminfo
.vimrc
.Xauthority
.xinitrc
.Xresources
one
three
three_
two
abstraction.jpg 649 K
raccoons.jpg 1.8 M
wsr39.pdf 747 K
-rw-r--r-- 1 guest wheel 1.8M 2022-06-21 00:52
3.16M sum, 18.4G free 2/3 All

Создание, доступ и удаление закладок в ranger

Возможность создавать закладки важна во всех приложениях для управления файлами. Чтобы создать закладку в ranger, все, что нам нужно сделать, это нажать клавишу m, а затем букву или цифру, которую мы хотим связать с каталогом, находясь в нем. Давайте посмотрим пример. Предположим, мы хотим связать ключ 1 с каталогом ~/Загрузки. Оказавшись внутри, нажимаем m; отобразится список текущих закладок:



Далее нажимаем 1 (это цифра, которую мы хотим связать с нашим каталогом), чтобы создать закладку.

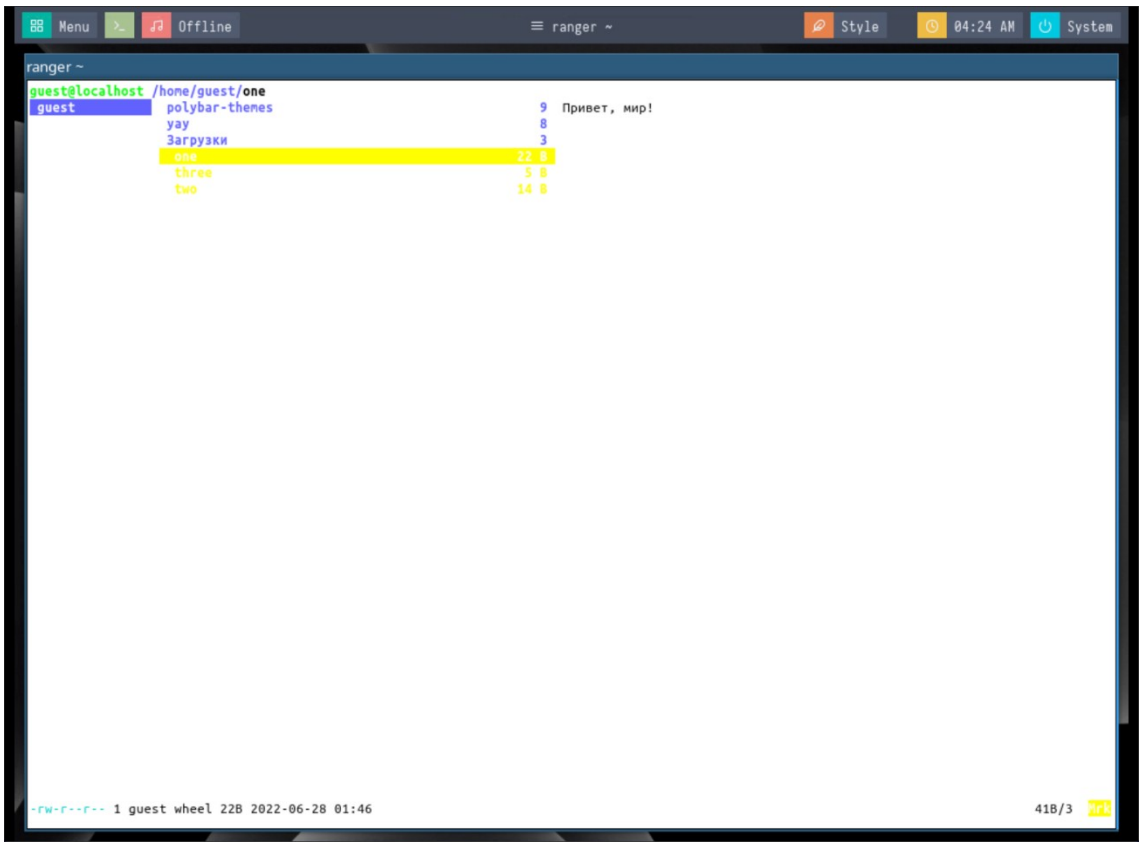
После создания закладки, чтобы получить к ней доступ, мы должны нажать клавишу `, а затем букву (или цифру), связанную с закладкой, поэтому в данном случае `1.

Чтобы удалить существующую закладку, все, что нам нужно сделать, это нажать клавиши um, а затем клавишу, связанную с закладкой, которую мы хотим удалить. Чтобы удалить закладку, которую мы установили выше, например, мы должны нажать umm.

Закладки можно сохранять мгновенно или при выходе из ranger. Параметр, управляющий этим поведением, называется autosave_bookmarks и принимает логическое значение. Обычно по умолчанию установлено значение true.

Выбор файлов в ranger

Чтобы выбрать один или несколько файлов при работе в ranger, все, что нам нужно сделать, это «отметить» их, нажав клавишу <Space>. Как только мы это сделаем, в правом нижнем углу появится желтый символ Mrk, а выбранные файлы будут выделены.



После того, как файлы выбраны, мы можем применить действие ко всем из них одновременно. Например, чтобы удалить их, мы должны ввести команду удаления или нажать клавиши dD. При удалении файлов появится запрос и попросит нас подтвердить действие.

В этом гайде мы рассмотрели, как пользоваться файловым менеджером ranger. Мы только поверхностно рассмотрели использование ranger, но этого достаточно для выполнения большинства операций над файлами.

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №7

Тема: Основы работы с Wine

Цель работы: Изучить основные принципы функционирования Wine и его команды.

– Как установить и запустить программу в Wine;

– Передача аргументов командной строки Windows;

– Запуск файлов .msi;

– Графическая среда Wine в стиле Explorer (Wine File Manager);

– Запуск файла .bat в Wine;

– Встроенные в Wine программы;

– Удаление программы из Wine;

– Файловая система Wine и WINEPREFIX;

– Создание 32-битного префикса Wine в 64-битной системе и запуск 32-битных программ;

– Запуск двух программ, как если бы они были на разных компьютерах;

– Запуск приложения в Wine на виртуальном рабочем столе;

– Особенности работы реестра и regedit в Wine;

– Использование regedit.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Wine – это свободное программное обеспечение, позволяющее запускать программы Windows прямо на Linux без использования виртуальных машин. Название WINE – рекурсивный акроним и расшифровывается «Wine Is Not Emulator» – «WINE – это не эмулятор». Имеется в виду, что Wine – это не эмулятор компьютера или виртуальная машина.

Вместо этого Wine – это набор программ и библиотек, которые позволяют запускать Windows приложения в Linux без использования виртуализации. Конечно, Windows программы можно запустить в Linux с использованием VirtualBox, но такой вариант потребует запуска полноценной операционной системы с соответствующими затратами ресурсов, особенно оперативной памяти – для работы Windows необходимо несколько гигабайт памяти, а также место на диске для установки – несколько десятков гигабайт. При этом важным требованием для работы VirtualBox и аналогичных виртуальных компьютеров является то, чтобы ваш центральный процессор поддерживал виртуализацию.

Wine позволяет обойти все эти ограничения – для запуска программ Windows не нужно устанавливать эту операционную систему и приложениям для работы требуется всего несколько десятков мегабайт оперативной памяти.

Но у Wine есть и недостатки — не все приложения работают хорошо или вообще работают. Тем не менее огромное количество Windows программ прекрасно запускаются и работают в Linux благодаря Wine.

Давайте установим его, если он не установлен по умолчанию. Сначала откройте текстовый файл /etc/pacman.conf:

guest@localhost ~> sudo nano /etc/pacman.conf

В нем найдите и раскомментируйте строки (убедитесь, что раскомментировали обе строки, иначе изменения не вступят в силу):

```
[multilib]
Include = /etc/pacman.d/mirrorlist
```

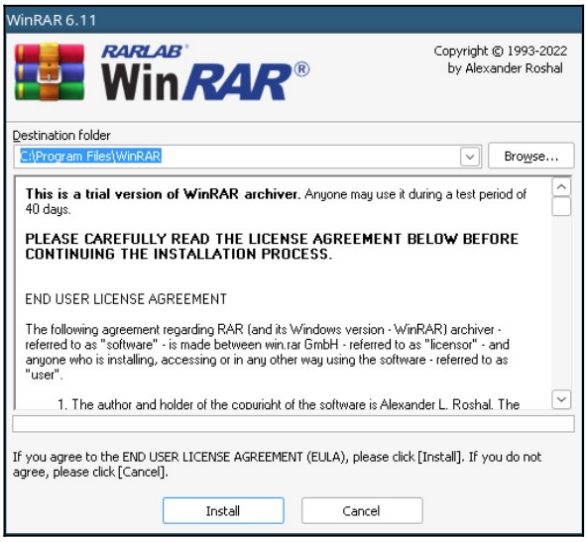
Выполните установку:

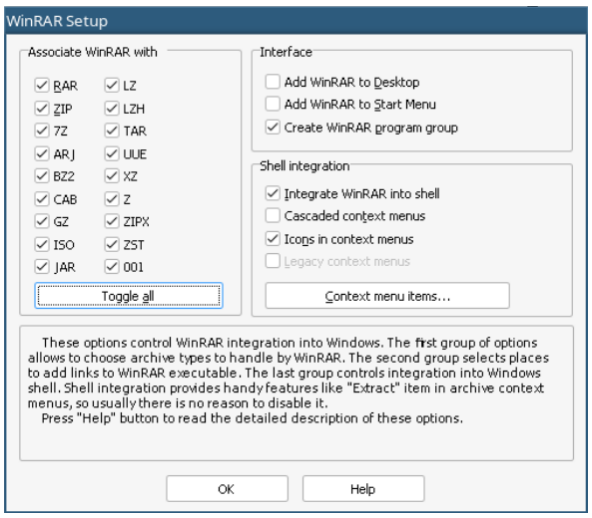
```
guest@localhost ~> sudo pacman -Syu
guest@localhost ~> sudo pacman -S wine wine_gecko wine-mono lib32-mpg123 lib32-gnutls
```

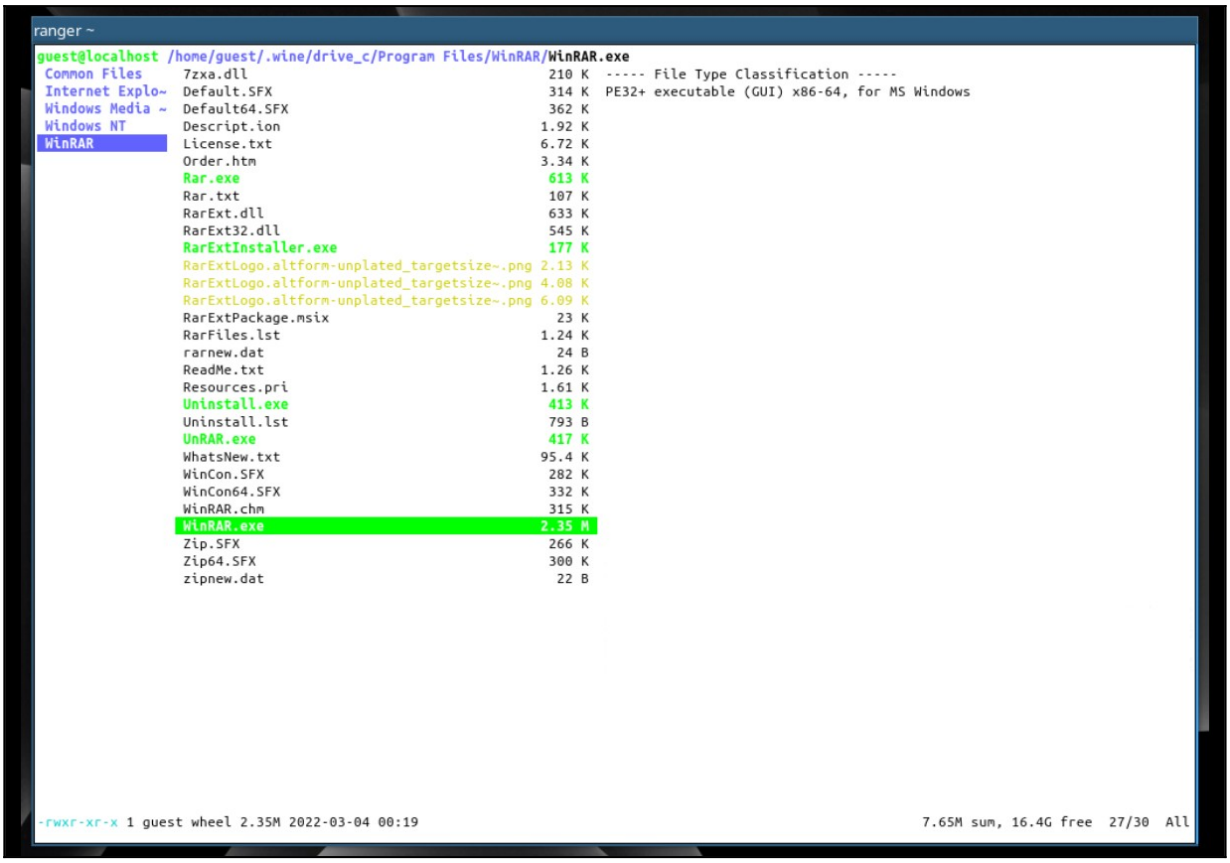
(или sudo apt install wine winbind в случае дистрибутивов, основанных на Debian)

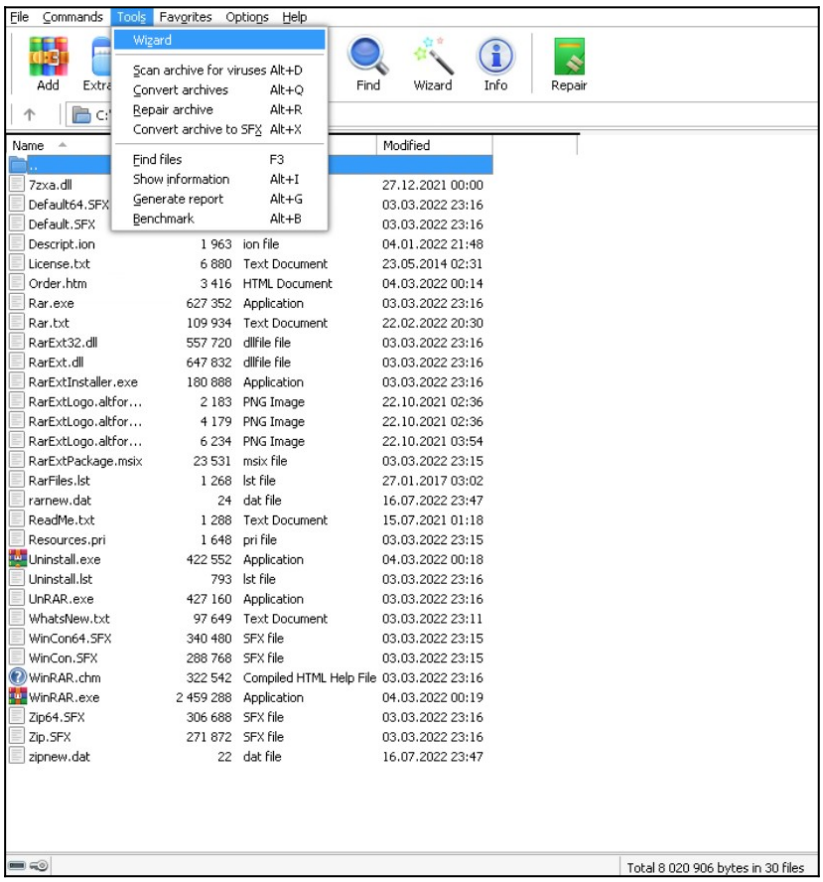
Как установить и запустить программу в Wine

Запуск двойным кликом по исполнимому файлу. Для запуска Windows приложений в Linux обычно достаточно дважды кликнуть по ним в файловом менеджере. Давайте попробуем запустить какое-либо приложение на примере WinRAR:




WinRAR Setup
Associate WinRAR with
RAR
ZIP
7Z
ARJ
BZ2
CAB
GZ
ISO
JAR
LZ
LZH
TAR
UUE
XZ
Z
ZIPX
ZST
001
Toggle all
Interface
Add WinRAR to Desktop
Add WinRAR to Start Menu
Create WinRAR program group
Shell integration
Integrate WinRAR into shell
Cascaded context menus
Icons in context menus
Legacy context menus
Context menu items...
These options control WinRAR integration into Windows. The first group of options allows to choose archive types to handle by WinRAR. The second group selects places to add links to WinRAR executable. The last group controls integration into Windows shell. Shell integration provides handy features like "Extract" item in archive context menus, so usually there is no reason to disable it.
Press "Help" button to read the detailed description of these options.
OK
Help


ranger ~
guest@localhost /home/guest/.wine/drive_c/Program Files/WinRAR/WinRAR.exe
Common Files
Internet Explo~
Windows Media ~
Windows NT
WinRAR
7zxa.dll 210 K ----- File Type Classification -----
Default.SFX 314 K PE32+ executable (GUI) x86-64, for MS Windows
Default64.SFX 362 K
Descript.ion 1.92 K
License.txt 6.72 K
Order.htm 3.34 K
Rar.exe 613 K
Rar.txt 107 K
RarExt.dll 633 K
RarExt32.dll 545 K
RarExtInstaller.exe 177 K
RarExtLogo.altform-unplated_targetsize~.png 2.13 K
RarExtLogo.altform-unplated_targetsize~.png 4.08 K
RarExtLogo.altform-unplated_targetsize~.png 6.09 K
RarExtPackage.msix 23 K
RarFiles.lst 1.24 K
rarnew.dat 24 B
ReadMe.txt 1.26 K
Resources.pri 1.61 K
Uninstall.exe 413 K
Uninstall.lst 793 B
UnRAR.exe 417 K
WhatsNew.txt 95.4 K
WinCon.SFX 282 K
WinCon64.SFX 332 K
WinRAR.chm 315 K
WinRAR.exe 2.35 M
Zip.SFX 266 K
Zip64.SFX 300 K
zipnew.dat 22 B
-rwxr-xr-x 1 guest wheel 2.35M 2022-03-04 00:19
7.65M sum, 16.4G free 27/30 All



Запуск в командной строке. Поскольку программы Windows часто ищут файлы относительного того места, откуда они были запущены, при использовании командной строки вы должны запускать их очень специфическим способом: «сменить каталог» на папку, в которой расположена программа, и запустить файл .exe, используя только его имя файла. Например:

```
guest@localhost ~> cd ~/.wine/drive_c/Program\ Files/
guest@localhost ~> wine WinRAR.exe
```

Передача аргументов командной строки Windows

Если вы используете программу с аргументами в Windows, например:

```
ping -h
ping -l 128 1.1.1.1
```

В таком случае вы можете сделать то же самое в Wine, запустив:

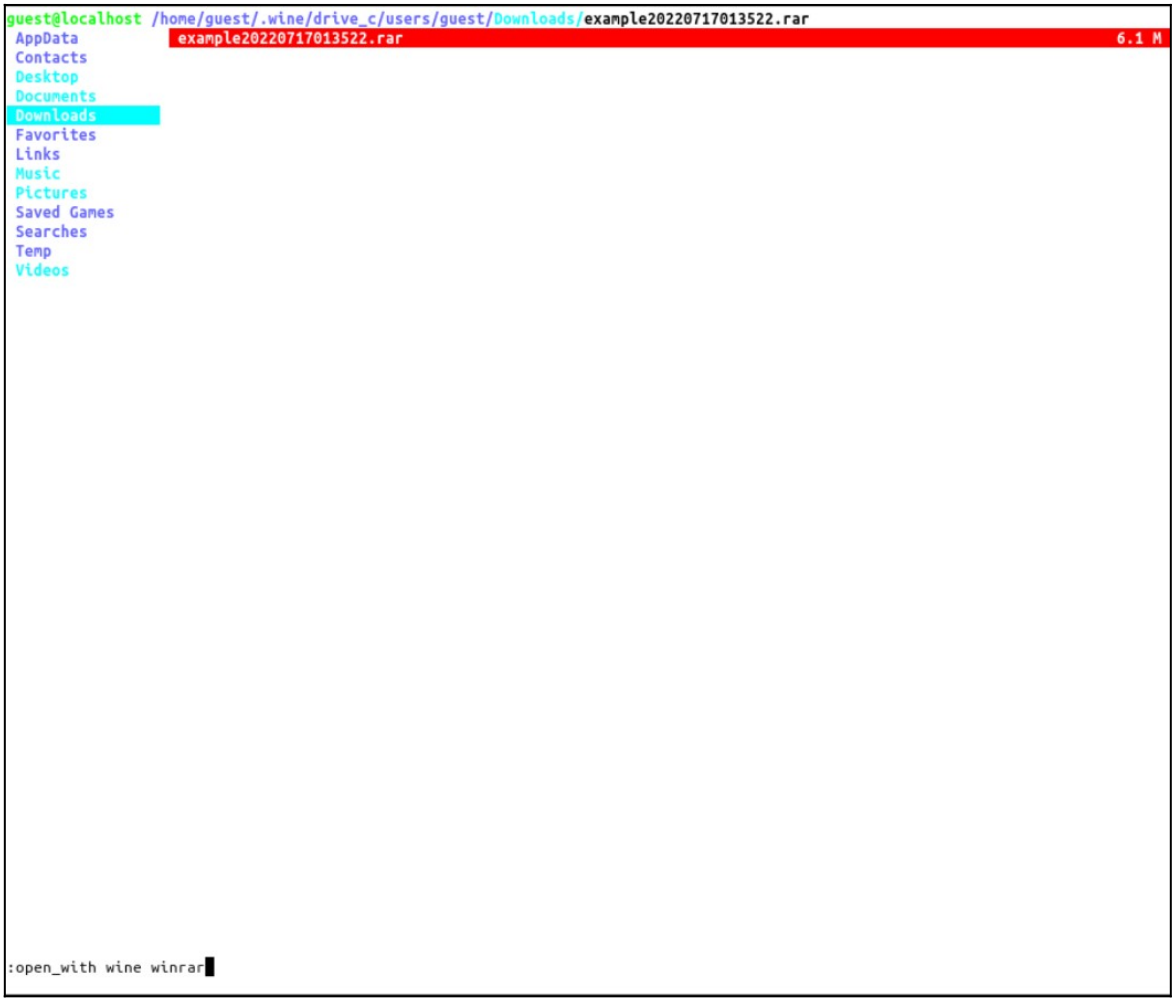
```
guest@localhost ~> wine ping -l 128 1.1.1.1
```

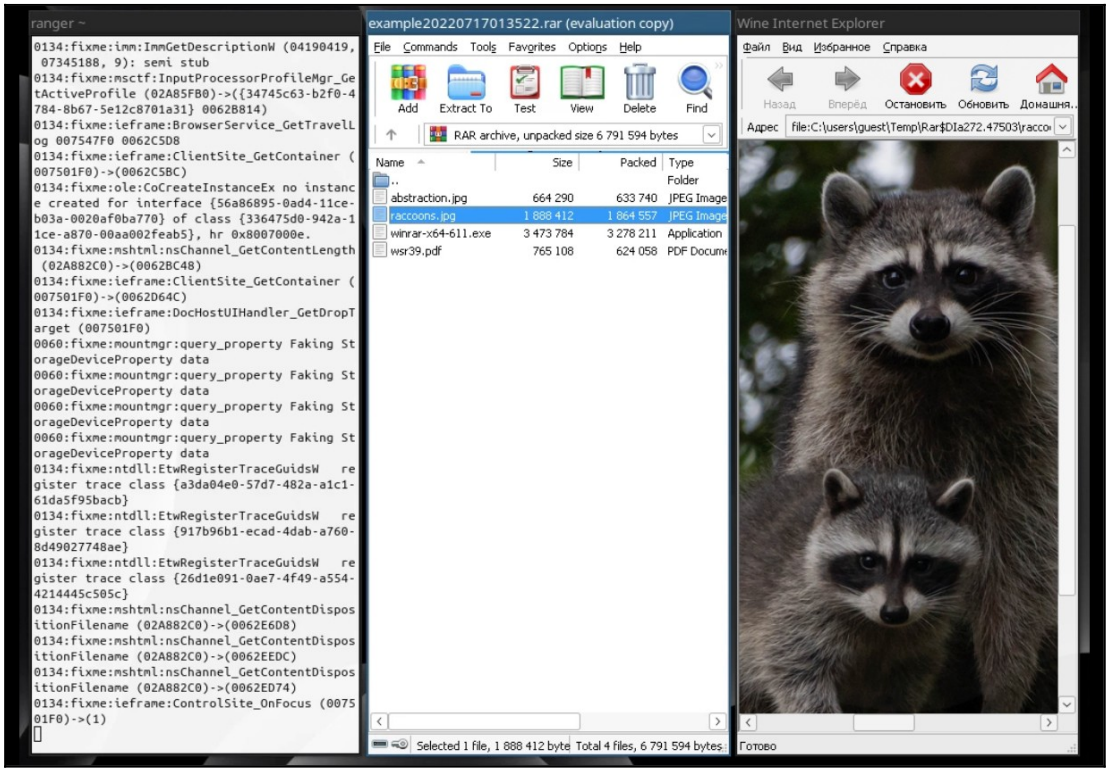
Обратите внимание, однако, что вам может потребоваться экранировать некоторые специальные символы обратной косой чертой из-за того, как они обрабатываются в оболочке Linux. Например, команда:

```
guest@localhost ~> “C:\Program Files\WinRAR\winrar.exe” a -ag “C:\Users\guest\
Downloads\example.rar” “C:\Users\guest\Downloads\example”
```

становится:

guest@localhost ~> wine ~/.wine/drive_c/Program\ Files/winrar.exe a -ag C:\\users\guest\\downloads\\example.rar ~/Загрузки





Запуск файлов .msi

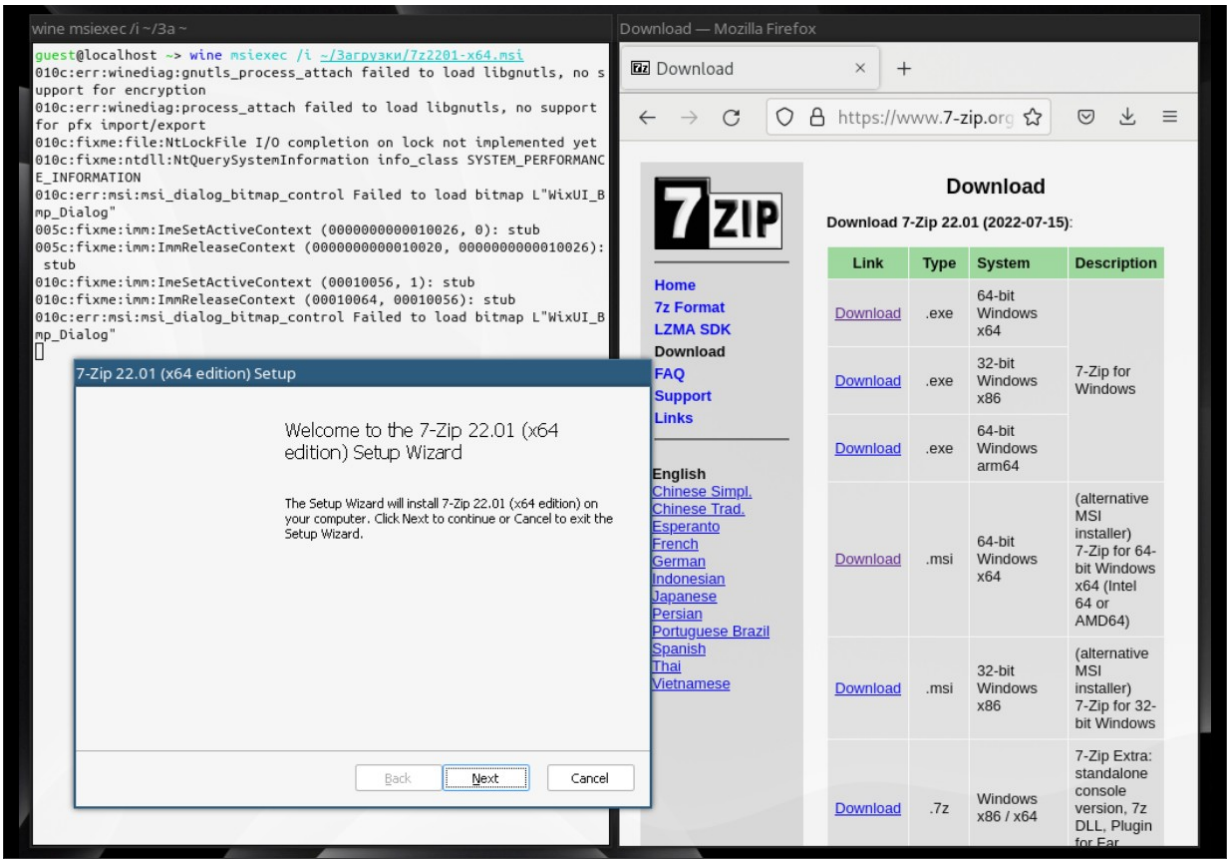
Windows Installer (msiexec.exe, раннее известный как Microsoft Installer (MSI)) – это подсистема Microsoft Windows, обеспечивающая установку программ (инсталлятор). Является компонентом Windows, начиная с Windows 2000; может доустанавливаться и на более ранние версии Windows. Вся необходимая для установки информация (иногда и вместе с устанавливаемыми файлами) содержится в установочных пакетах (installation packages), имеющих расширение .msi.

Файлы MSI нельзя запускать напрямую; вам нужно использовать либо программу Wine msiexec:

guest@localhost ~> wine msiexec /i ~/Загрузки/7z2201-x64.msi

либо запуск Wine с терминала:

guest@localhost ~> wine start /exec ~/Загрузки/7z2201-x64.msi

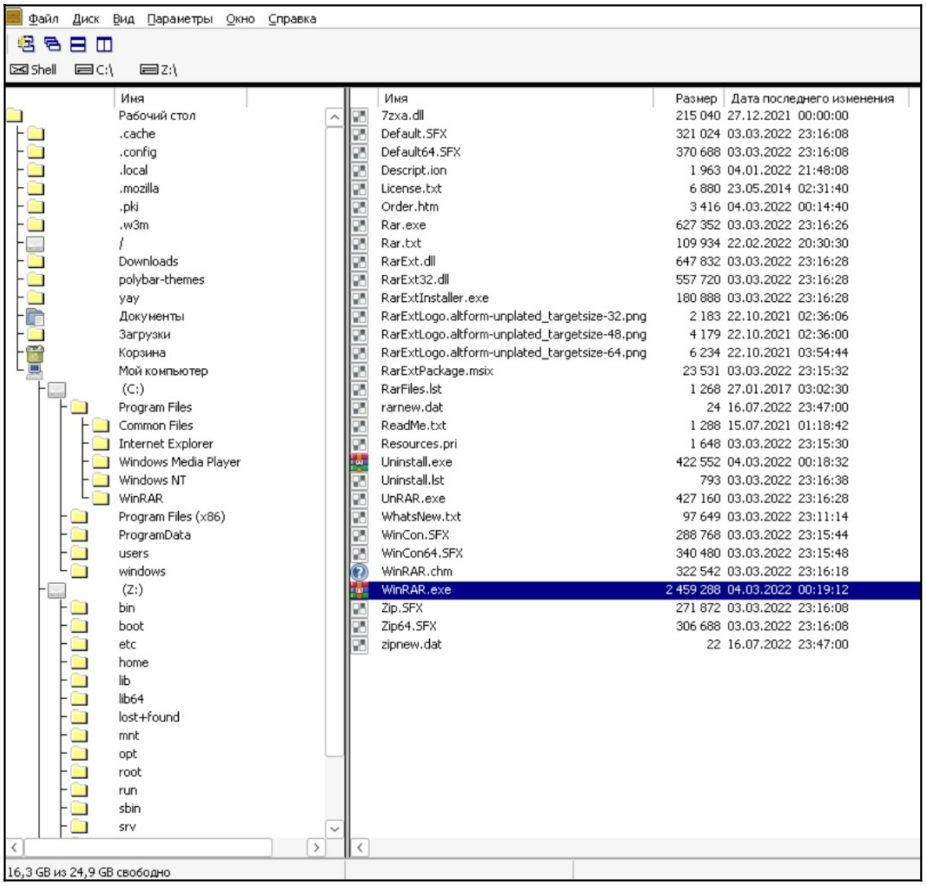


Графическая среда Wine в стиле Explorer (Wine File Manager)

Если вы предпочитаете использовать графический интерфейс для управления файлами, возможно, вам стоит рассмотреть возможность использования winefile. Это приложение winelib поставляется с Wine и может быть найдено с другими программами Wine. Это полезный способ просмотреть конфигурацию вашего диска и найти файлы, а также вы можете запускать программы прямо из winefile. Обратите внимание, что многие функции еще не реализованы.

Давайте попробуем в Wine File Manager настроить «Избранное» («Favorites») и добавить папки. Wine File Manager – это аналог проводника Windows. Его можно открыть следующей командой:

guest@localhost ~> winefile



Там вы можете увидеть несколько ярлыков и диски. Среди ярлыков вы найдете:

– My Computer;

– Documents;

– Trash;

– / (корень файловой системы Linux).

В «My Computer» собраны все диски, подключенные к Linux. Диск «C:» это то, что размещено в папке ~/.wine/drive_c/. Диск «Z:» – это корневой диск файловой системы

Linux. Другие буквы – это подключенные к Linux флешки и диски. Корневым элементов ярлыков является «Desktop», то есть рабочий стол.

> *Обратите внимание, имеется ввиду рабочий стол Linux, а не Windows! Если вы работаете с Wine в системе, не поддерживающей ярлыки на рабочем столе, например i3, то в таком случае рабочим столом в Wine File Manager у вас будет ваша домашняя папка «~», вместо «*~/Desktop».

То есть если вы хотите, чтобы в Wine File Manager была видна новая папка, то создайте ее на рабочем столе вашего Linux, например:

guest@localhost ~> mkdir ~/Favorites или mkdir ~/Desktop/Favorites (Если ваша DE поддерживает ярлыки на рабочем столе)

В эту папку вы можете скопировать любые файлы для быстрого доступа.

Также вы можете создавать ярлыки в этой папке на файлы и программы как в файловой системе Wine, так и за ее пределами.

Команда для создания ярлыка:

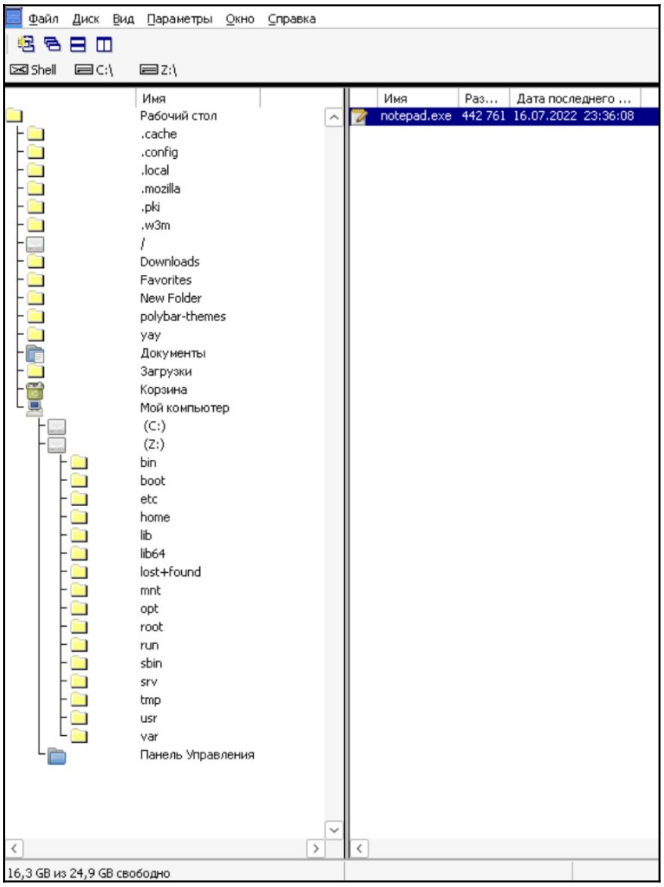
ln -s файл_или_директория целевой_файл_или_директория

Например, следующая команда создаст в папке ~/Desktop/Favorites/ ссылку на файл ~/.wine/drive_c/windows/notepad.exe:

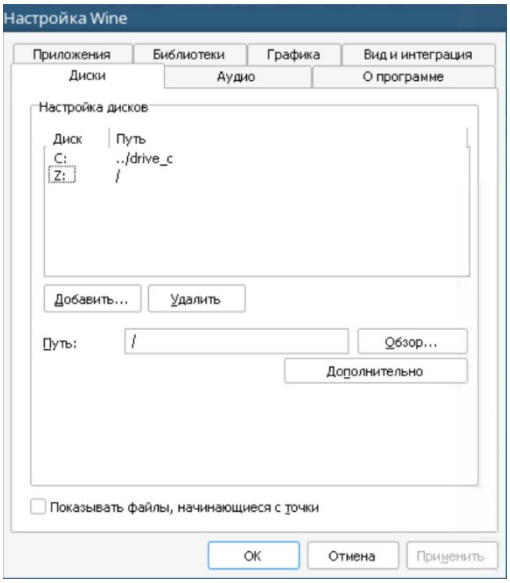
guest@localhost ~> ln -s ~/.wine/drive_c/windows/notepad.exe ~/Favorites/ или ~/Desktop/Favorites
guest@localhost ~> ln -s ~/.wine/drive_c/Program\ Files/WinRAR/WinRAR.exe ~/Favorites/ или ~/Desktop/Favorites

Если вы хотите изменить буквы дисков, то запустите «Wine configuration»:

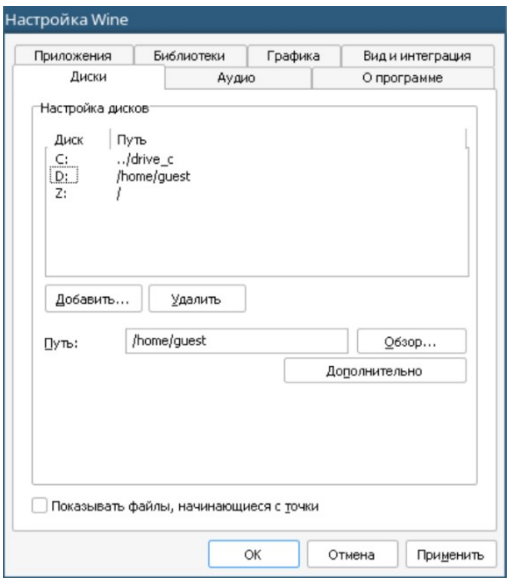
guest@localhost ~> winecfg



И перейдите на вкладку «Drives» («Диски») для настройки отображения дисков в Wine File Manager:



Попробуйте настроить диски таким образом, чтобы к диску «D» был примонтирован пусть с домашней папкой нашего пользователя «/home/guest»:



Запуск файла .bat в Wine

Перво-наперво давайте создадим .bat файл, который будет делать echo-запрос до сервера 1.1.1.1:

guest@localhost ~> touch ping.bat
guest@localhost ~> nano ping.bat

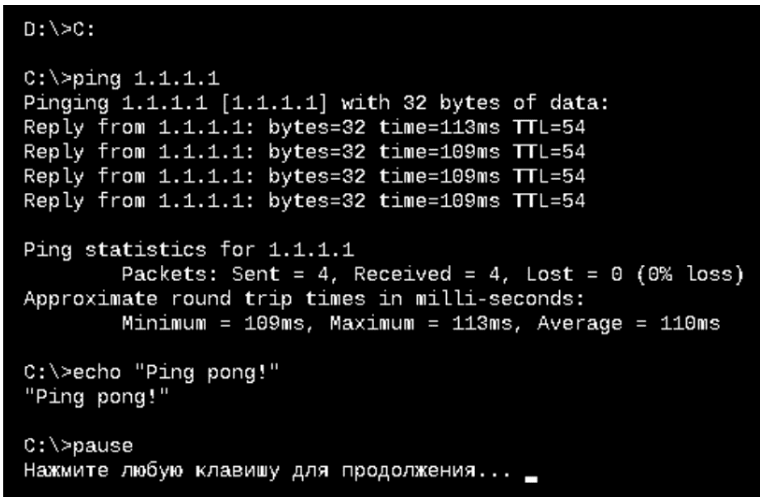
```
C:
ping 1.1.1.1
echo "Ping pong!"
pause
```

Вы можете запустить файл .bat разными способами:

1. Выполните команду:

guest@localhost ~> winefile

Найдите файл .bat и запустите его двойным кликом.



2. Выполните команду:

guest@localhost ~> wine explorer

Найдите в окне «Проводника» нужный вам файл.

Либо выполните следующую команду, чтобы сразу попасть в нужную папку:

guest@localhost ~> wine explorer ~/

Когда найдете файл .bat, то дважды кликните на него.

3. Вы можете запустить файл .bat в командной строке, используя конструкцию вида:

guest@localhost ~> wine start /exec ~/ping.bat

Если вы используете путь до файла в файловой системе Linux, то обязательно нужно указать опцию /unix.

Встроенные в Wine программы

В Wine имеется несколько встроенных «стандартных программ Windows»:

– «Блокнот»;

– «Редактор реестра»;

– «Проводник»;

– «Диспетчер задач»;

– Игра «Сапер»;

– Браузер «Internet Explorer»;

– «Командная строка Windows»;

– Установщик MSI файлов;

– «Панель управления»;

– Программа для просмотра файлов .chm в Linux.

1. «Блокнот». Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> notepad или wine notepad

2. «Редактор реестра». Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> regedit или wine regedit

3. «Проводник». Чтобы открыть проводник, выполните команду:

guest@localhost ~> wine regedit

В проводнике вы можете открыть «Мой компьютер» – там будут ссылки на все диски в Linux. Диск «C:» это то, что размещено в папке ~/.wine/drive_c/. Диск «Z:» – это корневой диск файловой системы Linux. Другие диски – это подключенные к Linux флешки и диски.

В проводнике вы можете запускать файлы .bat двойным кликом, а также управлять файлами (перемещать, создавать, удалять их) в графическом интерфейсе.

4. «Диспетчер задач». Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> wine taskmgr

5. Игра «Сапер». Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> winemine

6. Браузер «Internet Explorer». На самом деле, это не настоящий «Internet Explorer» – это браузер на основе Firefox. Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> wine iexplore

При желании, можно установить старые версии «Internet Explorer», но даже Windows в последних версиях отказалась от использования «Internet Explorer» и перешла на веб-браузер на основе «Chrome».

7. «Командная строка Windows». Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> wine cmd

Выполните следующую команду для дополнительной информации по перечисленным командам:

help *команда*

```
D:\>help
Встроенные команды CMD:
ASSOC           Показывает или изменяет сопоставления типов файлов
ATTRIB          Показывает или изменяет DOS-атрибуты файла
CALL            Вызывает один bat-файл из другого
CD (CHDIR)      Изменяет текущий каталог
CHOICE          Ждёт выбора из списка
CLS             Очищает экран консоли
COPY            Копирует файл(ы)
CTTY            Изменяет устройство ввода/вывода
DATE            Показывает или изменяет системную дату
DEL (ERASE)     Удаляет файл или несколько файлов
DIR             Выводит содержимое каталога
ECHO            Выводит текст непосредственно в консоль
ENDLOCAL        Заканчивает действие локальных изменений окружения
FTYPE           Выводит или изменяет команды открытия, связанные с типами файлов
HELP            Показывает краткую подсказку по команде
MD (MKDIR)      Создаёт каталог
MKLINK  Создаёт символическую ссылку
MORE            Выводит данные по страницам
MOVE            Перемещает файл, несколько файлов или дерево каталогов
PATH            Показывает или изменяет путь поиска программ
PAUSE           Останавливает исполнение bat-файла
POPD            Восстанавливает предыдущий текущий каталог, сохранённый с
                помощью
PROMPT          Изменяет приглашение командной строки
PUSHD           Сохраняет текущий каталог и переходит в другой
REN (RENAME)    Переименовывает файл
RD (RMDIR)      Удаляет каталог
SET             Показывает или изменяет переменные окружения
SETLOCAL        Начинает действие локальных изменений окружения
START           Запускает программу, или открывает файл в соответствующей
                программе
TIME            Показывает или изменяет текущее системное время
TITLE           Устанавливает заголовок окна cmd для текущей сессии
TYPE            Выводит содержимое текстового файла
VER             Показывает текущую версию CMD
VOL             Показывает метку тома дискового устройства
XCOPY           Копирует файлы или деревья каталогов
EXIT            Закрывает CMD

Выполните HELP <команда> для дополнительной информации по перечисленным командам.
```

8. Установщик MSI файлов. Эту команду мы уже применяли ранее. Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> msiexec

9. «Панель управления». Чтобы запустить эту программу выполните следующую команду:

guest@localhost ~> wine control

10. «Программа для просмотра файлов .chm в Linux». В Wine присутствует встроенная программа hh.exe, которая может открывать файлы с расширением .chm. Файлы .chm, еще их называют MS Windows HtmlHelp Data – это обычно файлы со справкой по использованию программы, с которой они распространяются.

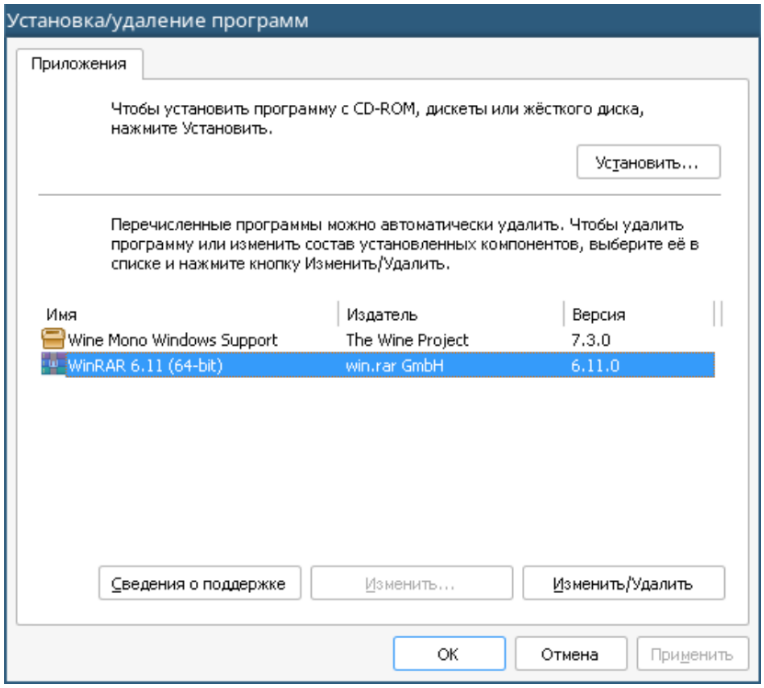
После установки Wine вы можете открыть файлы .chm двойным кликом. Еще один способ – это использовать командную строку:

guest@localhost ~> wine hh.exe "/home/guest/.wine/drive_c/Program Files/WinRAR/WinRAR.chm"

Удаление программы из Wine

Чтобы удалить программу из Wine, перейдите в настройки «Установка/удаление программ», для этого выполните:

guest@localhost ~> wine uninstaller



Далее выберите программу для удаления и нажмите кнопку «Изменить/удалить». Ярлыки, созданные во время установки, будут удалены автоматически.

Файловая система Wine и WINEPREFIX

По умолчанию физически файлы размещены в папке «~/.wine». Это значит, что если вы хотите «переустановить Windows», т.е. полностью удалить все настройки и

установленные программы, то вы можете удалить эту папку. Но помните, что сделанные в меню и на рабочем столе ярлыки останутся, поскольку расположены в других местах.

При использовании WINEPREFIX, будет использоваться другая папка для хранения файлов. По умолчанию, Wine также хранит файлы настроек и установленные приложения Windows в каталоге «~/.wine». Этот каталог называется префиксом Wine (Wine prefix). Он создается и обновляется автоматически по необходимости при запуске программ Windows и программ настройки Wine, например winecfg. Каталог префикса также содержит стандартную структуру корневого раздела каталогов Windows, которая представляется программам Windows как диск «C:».

Вы можете изменить место расположения префикса, создав переменную окружения WINEPREFIX с указанием нового пути. Это полезно, когда вам необходимо использовать различное окружение для разных приложений Windows. При запуске приложения Windows новый префикс будет автоматически создан на указанном в WINEPREFIX месте, если его до этого не существовало.

Для примера, если вы запускаете одно приложение со следующим префиксом:

```
guest@localhost ~> env WINEPREFIX=/home/guest/.win-a wine ~/.wine/drive_c/Program\
Files/WinRAR/WinRAR.exe
```

а другое – с таким:

```
guest@localhost ~> env WINEPREFIX=/home/guest/.win-b wine ~/.wine/drive_c/Program\
Files/7-Zip/7zFM.exe
```

В таком случае у каждой программы будет свой раздел «C:», соответственно, своя копия всех настроек и реестра. Таким образом, обе программы будут запущены в полностью изолированных друг от друга средах.

> *Примечание! Тем не менее, префиксы Wine не являются песочницами. Программы, запущенные в Wine могут также получать доступ к оставшейся части системы (например, раздел «Z:» обычно соответствует корню файловой системы «/»).*

Для создания префикса без запуска каких-либо средств настройки Wine или приложений Windows вы можете использовать команду:

```
guest@localhost ~> env WINEPREFIX=/home/guest/.customprefix wineboot -u
```

Вы можете изменить префикс, который использует Wine, изменив переменную среды WINEPREFIX (за пределами Wine). Для этого запустите в терминале что-то вроде следующей команды:

```
guest@localhost ~> export WINEPREFIX=/home/guest/.wine-new
guest@localhost ~> wine winecfg
```

Чтобы использовать префикс по умолчанию, просто установите значение WINEPREFIX на «~/.wine».

```
guest@localhost ~> export WINEPREFIX=/home/guest/.wine
```

guest@localhost ~> wine winecfg

В качестве альтернативы вы можете указать префикс Wine в каждой команде, например:

guest@localhost ~> WINEPREFIX=/home/guest/.wine wine winecfg

Вы можете переименовывать, перемещать, копировать и удалять префиксы, не затрагивая другие, и каждый префикс имеет свой собственный экземпляр wineserver.

Создание 32-битного префикса Wine в 64-битной системе и запуск 32-битных программ

В настоящее время существует ряд серьезных ошибок, которые не позволяют многим 32-битным приложениям работать с 64-битным префиксом Wine. Чтобы обойти это, вы можете создать новый 32-битный префикс Wine, используя переменную среды WINEARCH. В терминале введите следующую команду:

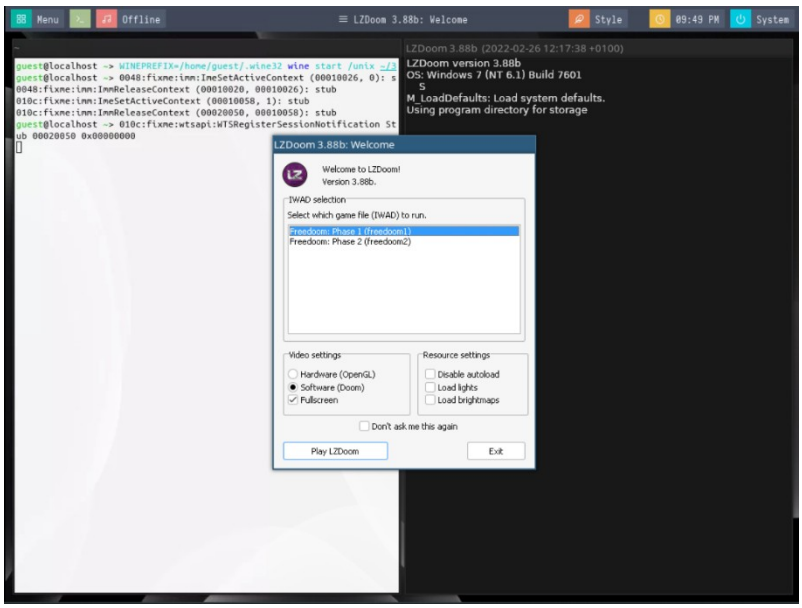
guest@localhost ~> WINEARCH=win32 WINEPREFIX=/home/guest/.wine32 winecfg

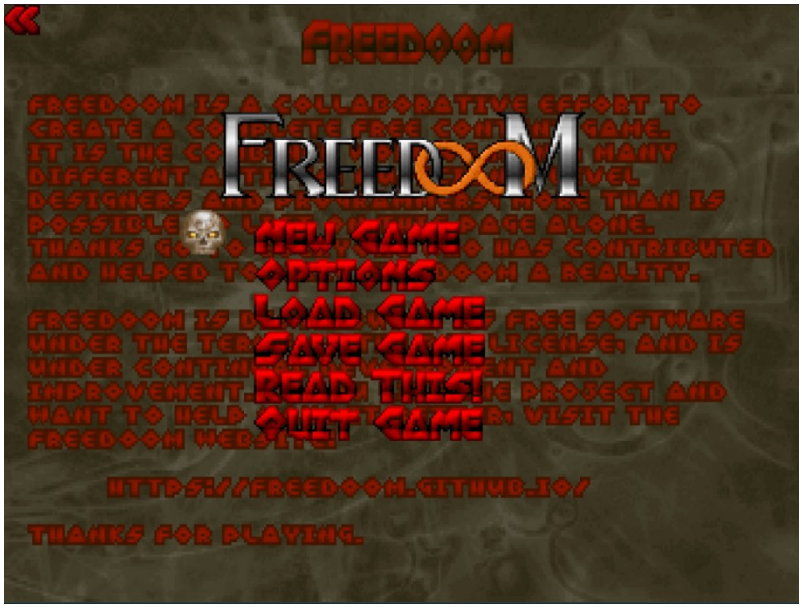
Используйте фактический путь к WINEPREFIX. Не используйте существующий каталог для нового WINEPREFIX: Wine должен его создать сам.

После создания 32-битного WINEPREFIX вам больше не нужно указывать WINEARCH в командной строке, чтобы использовать его, так как архитектура существующего WINEPREFIX не может быть изменена. Обратите внимание, что если вы хотите использовать WINEPREFIX не тот, который установлен по умолчанию (~/.wine), то вам нужно указать переменную WINEPREFIX при установке чего-либо (включая Winetricks):

guest@localhost ~> WINEPREFIX=/home/guest/.wine32 wine start /unix ~/Загрузки/LZDoom_3.88d_x86/lzdoom.exe

> Заранее скачайте 32-битную версию какой-либо программы, чтобы проверить сможет ли она запуститься. В примере будет установлен и запущен LZDoom с набором карт от FreeDoom. Набор карт необходимо будет распаковать в директорию LZDoom. Скачать и установить их можно по следующим ссылкам:
> https://github.com/freedoom/freedoom/releases/download/v0.12.1/freedoom-0.12.1.zip
> https://github.com/drfrag666/gzdoom/releases/download/3.88b/LZDoom_3.88b_x86.zip





Запуск двух программ, как если бы они были на разных компьютерах

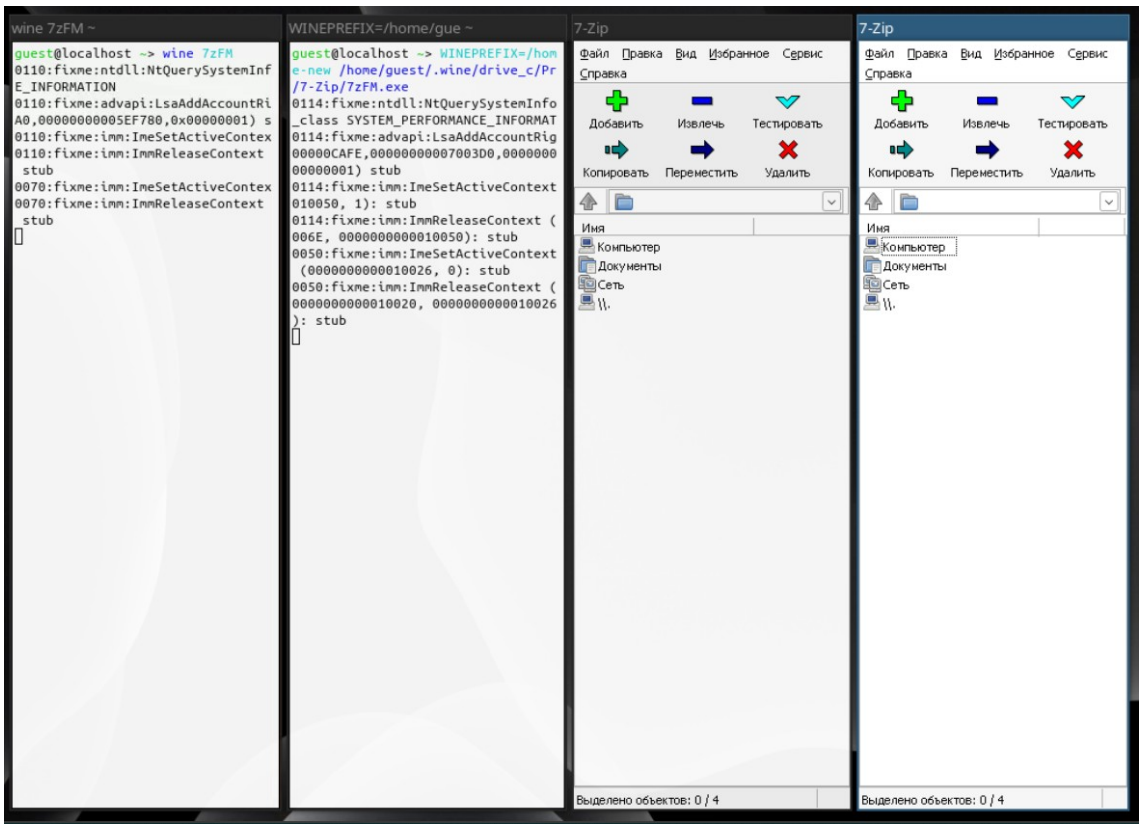
Допустим у вас есть серверная и клиентская программы. И одна не будет работать в присутствии другой. Здесь вам поможет использование разных WINEPREFIX, поскольку они фактически имитируют два компьютера с Windows.

Запустите первую программу как обычно:

guest@localhost ~> wine 7zFM

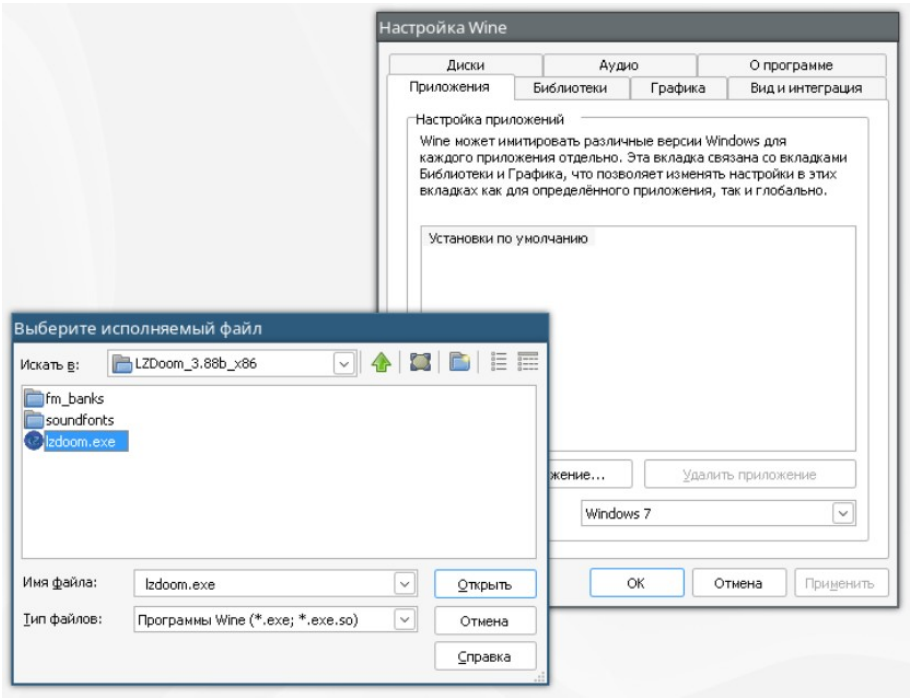
Вторую нужно запускать с другим префиксом, поэтому нам нужно изменить переменную среды WINEPREFIX:

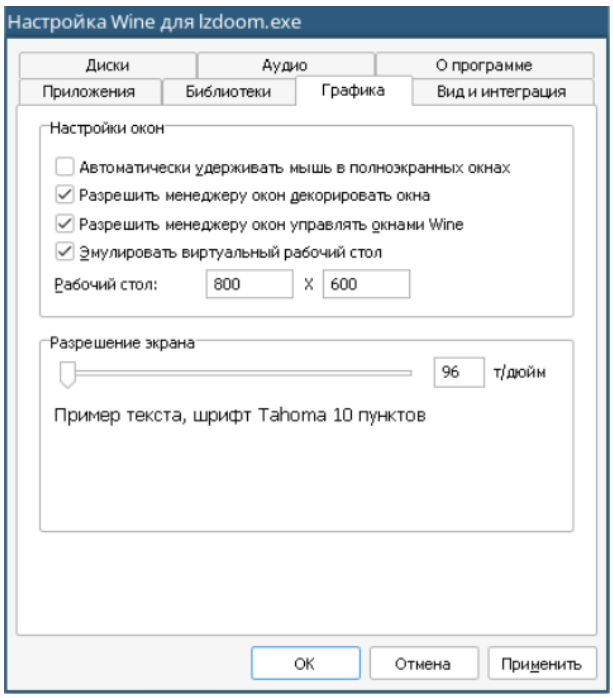
guest@localhost ~> WINEPREFIX=/home/guest/.wine-new wine /home/guest/.wine/drive_c/Program\ Files/7-Zip/7zFM.exe



Запуск приложения в Wine на виртуальном рабочем столе

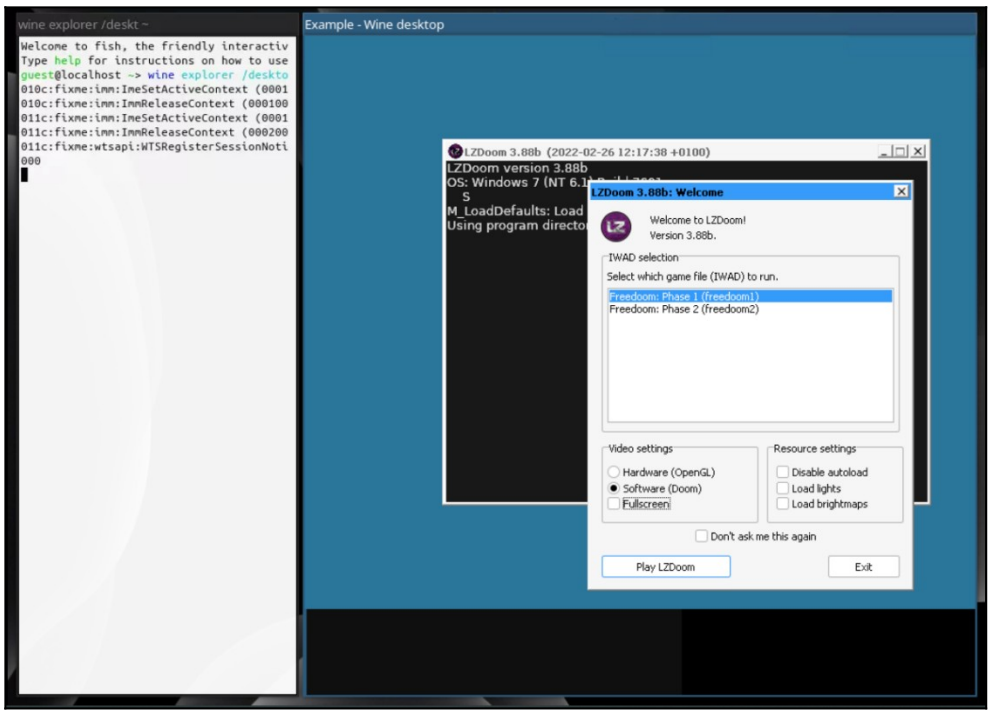
Вы можете заставить Wine запускать приложение на виртуальном рабочем столе с помощью winecfg**.** Добавьте приложение на вкладке «Приложения», а затем на вкладке «Графика» включите «Эмулировать виртуальный рабочий стол».





Вы также можете использовать следующую команду:

guest@localhost ~> wine explorer /desktop=Example,800x600 ~/Загрузки/LZDoom_3.88d_x86/lzdoom.exe



Изменение имени окна позволит одновременно открывать несколько рабочих столов.

Особенности работы реестра и regedit в Wine

Все настройки, которые вы изменяете в winecfg, за исключением настроек диска, в конечном итоге сохраняются в реестре. В Windows это центральный репозиторий для конфигурации приложений и операционной системы. Точно так же Wine реализует реестр, и некоторые настройки, которых нет в winecfg, могут быть изменены в нем.

Структура реестра. Реестр Windows представляет собой сложную древовидную структуру, и даже большинство программистов Windows не полностью осведомлены о том, как устроен реестр с его различными «кустами» и многочисленными связями между ними; полное его описание выходит за рамки нашей темы. Но вот основные ключи реестра, о которых вам стоит знать:

– HKEY_LOCAL_MACHINE – этот основной корневой ключ (в win9x он хранится в скрытой файловой системе system.dat) содержит все, что относится к текущей установке Windows. Часто это сокращенно называют HKLM;

– HKEY_USERS – этот основной корневой ключ (в Win9x он хранится в скрытом файле user.dat) содержит данные конфигурации для каждого пользователя установки;

– HKEY_CLASSES_ROOT – это ссылка на HKEY_LOCAL_MACHINE\Software\Classes. Он содержит данные, описывающие такие вещи, как ассоциации файлов, обработчики документов OLE[6] и классы COM[7];

– HKEY_CURRENT_USER – это ссылка на HKEY_USERS*имя_вашего_пользователя*, т.е. на вашу личную конфигурацию.

Файлы реестра. Теперь вы, вероятно, задаетесь вопросом, как это переводится в структуру Wine. Описанный выше макет реестра на самом деле находится в трех разных файлах в каталоге «~/.wine» каждого пользователя:

– system.reg – этот файл содержит HKEY_LOCAL_MACHINE;

– user.reg – этот файл содержит HKEY_CURRENT_USER;

– userdef.reg – этот файл содержит HKEY_USERS\.Default, т.е. пользовательские настройки по умолчанию;

Эти файлы автоматически создаются при первом использовании Wine. Набор глобальных настроек хранится в файле wine.inf и обрабатывается программой rundll32.exe. При первом запуске Wine файл wine.inf обрабатывается для заполнения

---

[6] Object Linking and Embedding – это технология связывания и внедрения объектов в другие документы и объекты в продуктах Microsoft. В 1996 году они переименовали технологию в ActiveX.

[7] Component Object Model – это технологический стандарт от компании Microsoft, предназначенный для создания ПО на основе взаимодействующих компонентов, каждый из которых может использоваться во многих программах одновременно.

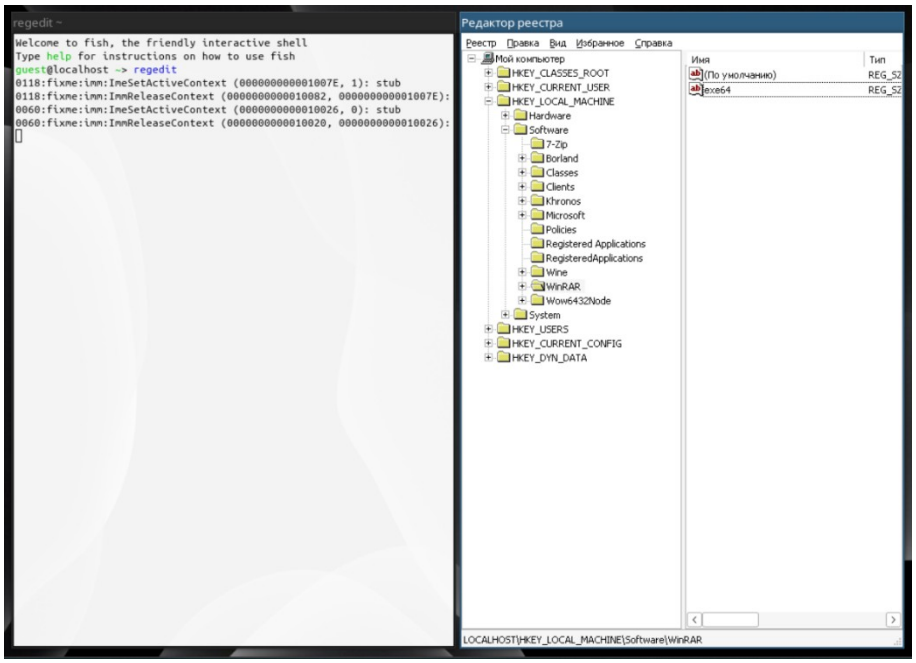
исходного реестра. Реестр также обновляется автоматически при изменении wine.inf, например, при обновлении до более новой версии Wine.

Не рекомендуется редактировать эти файлы для изменения реестра, поскольку они управляются внутри Wine. Используйте regedit.exe, reg.exe или любую программу, которая использует стандартные функции реестра.

Использование regedit

Легкий способ получить доступ к реестру и изменить его с помощью инструмента regedit:

guest@localhost ~> regedit



Подобно программе Windows, которую он заменяет, regedit служит для обеспечения представления системного уровня реестра, содержащего все ключи. Когда вы запустите ее, вы сразу заметите, что загадочные ключи, отображаемые в текстовом файле, организованы в виде иерархии.

Чтобы перемещаться по реестру, нажимайте пункты слева, чтобы перейти к подпунктам. Чтобы удалить ключ, щелкните его и выберите «Удалить» в меню «Правка». Чтобы добавить ключ или значение, найдите, куда вы хотите поместить его, и выберите «Создать» в меню «Правка». Точно так же вы изменяете существующий ключ, выделяя его на правой панели окна и выбирая «Изменить» в меню «Правка». Другой способ выполнить те же действия – щелкнуть правой кнопкой мыши ключ или значение.

Особый интерес для пользователей Wine представляют настройки, хранящиеся в HKEY_CURRENT_USER\Software\Wine. Большинство настроек, которые вы изменяете в winecfg, записываются в эту область реестра.

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №8

Тема: Помощники в установке программ и библиотек в Wine

Цель работы: Попробовать проинсталлировать Windows-программы с использованием помощников Wine.

– Использование Winetricks;

– Использование Bottles.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Существуют несколько программ-помощников, которые помогут более удобно работать с таким инструментом как Wine. В этом разделе мы рассмотрим две программы из них: Winetricks и Bottles.

Использование Winetricks

Winetricks – это простой способ решения различных задач в Wine связанных с установкой программ и игр и настройкой Wine.

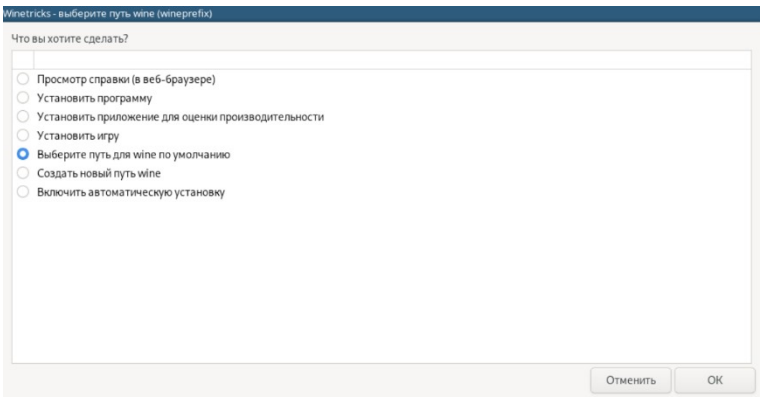
Winetricks – это вспомогательный скрипт для загрузки и установки различных распространяемых библиотек времени выполнения, необходимых для запуска некоторых программ в Wine. Они могут включать замену компонентов Wine с использованием библиотек с закрытым исходным кодом.

У Winetricks есть графическое меню поддерживаемых игр/приложений, для которых он может выполнять последовательные действия. Он также позволяет устанавливать недостающие библиотеки DLL и настраивать различные параметры Wine. Вы можете выбрать префикс, в который вы хотите установить приложение или изменить настройку.

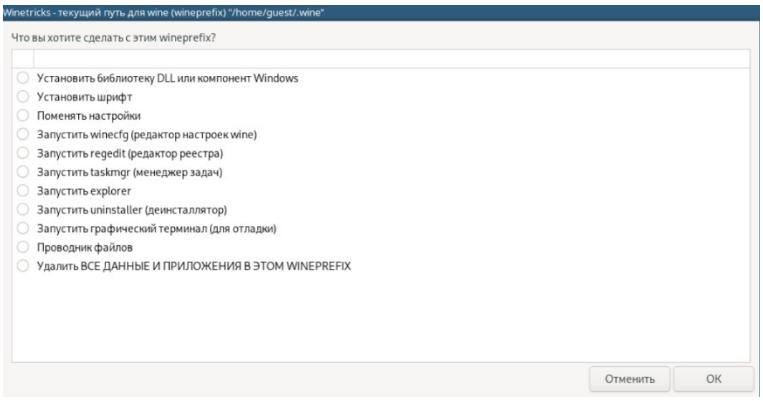
Выполните установку:

guest@localhost ~> sudo pacman -Syu
guest@localhost ~> sudo pacman -S winetricks zenity kdialog (или sudo apt install winetricks zenity kdialog в случае дистрибутивов, основанных на Debian)

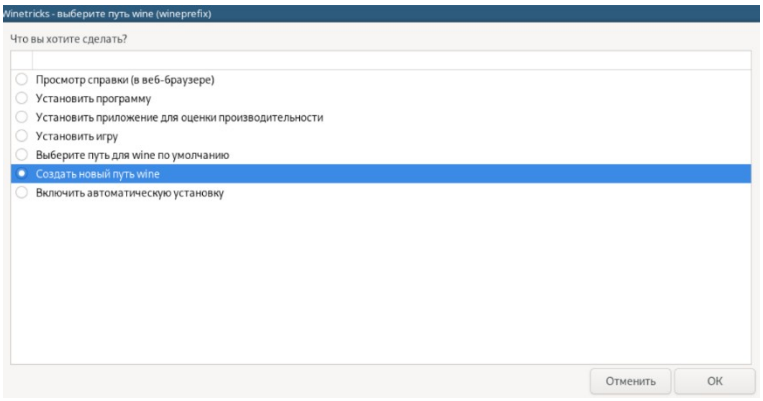
У программы есть графический интерфейс. В открывшемся окне вы можете сразу выполнить действия в префиксе по умолчанию или создать новый префикс.



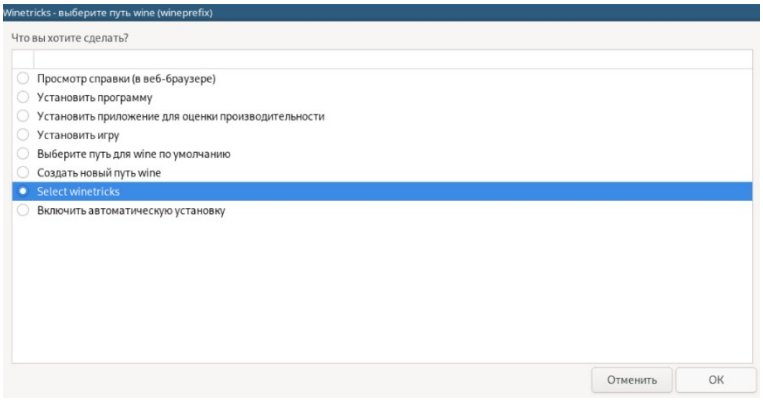
Пример действий после выбора префикса.



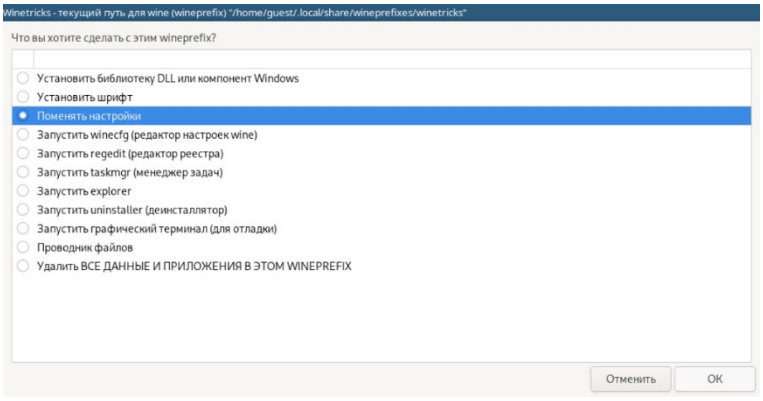
Давайте создадим новый префикс. Чтобы это сделать – достаточно выбрать его разрядность и название.

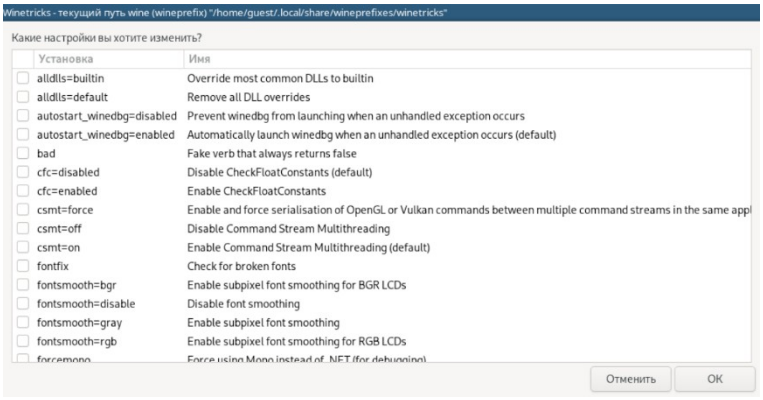


Новый префикс добавлен в список и теперь вы можете выбрать его для установки программ и выполнения других действий. Нажмите «Отменить», чтобы вернуться обратно в меню выбора пути Winetricks и снова выберите ваш свежесозданный префикс «winetricks».

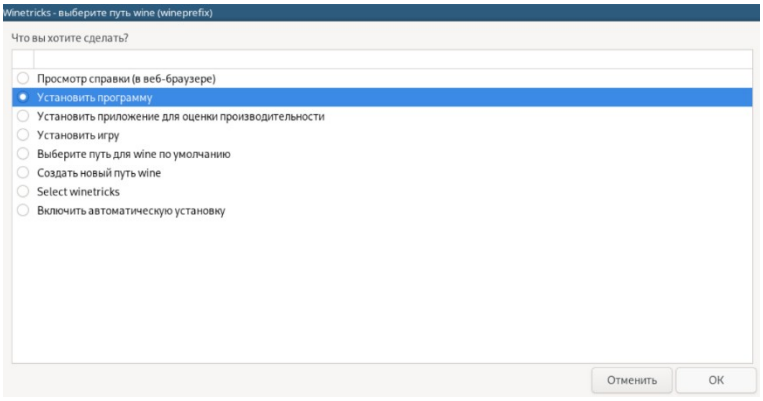


Пример настроек, которые вы можете поменять для выбранного префикса.

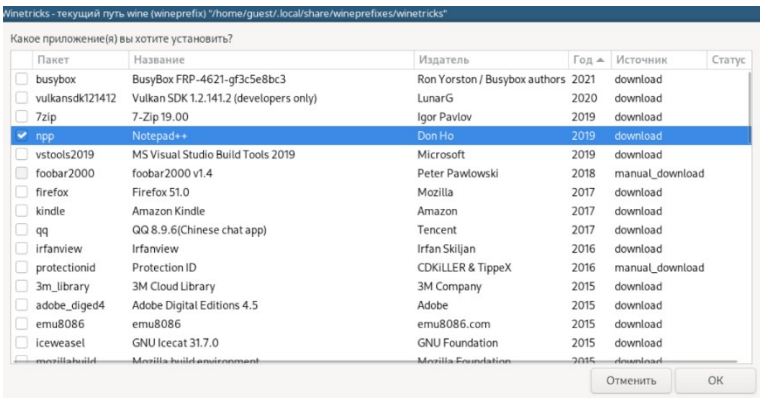


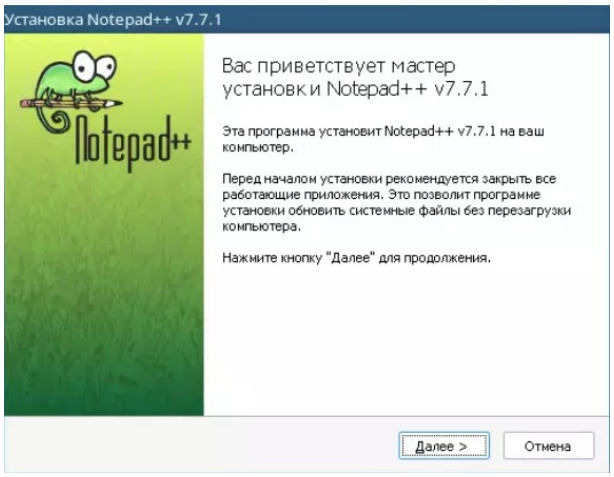


Пример программ для установки.



Давайте установим какую-нибудь программу через Winetricks, например Notepad++.





В Winetricks выполняемые по установке действия называются verbs, то есть буквально «глаголы». Каждый глагол устанавливает приложение или изменяет настройки. В дальнейшем в качестве verb будут использоваться слова «действие» и «глагол».

Синтаксис Winetricks:
guest@localhost ~> winetricks *[опции] [команда|действие|путь_до_глагола]*

Опции Winetricks:

--country=CC	Установить код страны на CC и не определять ваш IP-адрес
-f, --force	Не проверять, были ли уже установлены пакеты
--gui	Показывать диагностику в графическом интерфейсе даже при управлении из командной строки
--isolate	Установить каждое приложение или игру в отдельный префикс (WINEPREFIX)
--self-update	Обновить это приложение до последней версии
--update-rollback	Откатить последнее самообновление
-k, --keep_isos	Кэшировать образы ISO (позволяет последующую установку без диска)
--no-clean	Не удалять временные папки (полезно для отладки)
-q, --unattended	Не задавать никаких вопросов, просто установить автоматически
-r, --ddrescue	Повторите попытку при кэшировании поцарапанных дисков
-t --torify	Запускать загрузки с выходом в сеть через Tor, если возможно
--verify	Запустить (автоматизированные) тесты GUI для глаголов, если доступно
-v, --verbose	Отображать все команды по мере их выполнения
-h, --help	Показать справку и выйти
-V, --version	Показать версию и выйти

Команды Winetricks:

list	вывести список категорий
list-all	вывести все категории и их глаголы
apps list	вывести все глаголы категории 'applications' (приложения)
benchmarks list	вывести все глаголы категории 'benchmarks' (измерители производительности)
dlls list	вывести все глаголы категории 'dlls' (файлы библиотек dll)
fonts list	вывести все глаголы категории 'fonts' (шрифты)
games list	вывести все глаголы категории 'games' (игры)
settings list	вывести все глаголы категории 'settings' (настройки)
list-cached	вывести список кэшированных и готовых к установке глаголов
list-download	вывести глаголы, которые загружаются автоматически
list-manual-download	вывести глаголы, для загрузки которых нужна помощью пользователя
list-installed	показать уже установленные глаголы
arch=32|64	создать wineprefix с 32 или 64 битами, эта опция должна быть передана перед prefix=foobar и не будет работать в случае префикса wineprefix по умолчанию.
prefix=foobar	выбрать WINEPREFIX=/home/$USER/.local/share/wineprefixes/foobar
annihilate	Удалить ВСЕ ДАННЫЕ И ПРИЛОЖЕНИЯ ВНУТРИ ЭТОГО ПРЕФИКСА

```
guest@localhost ~> winetricks --help
Usage: /usr/bin/winetricks [options] [command|verb|path-to-verb] ...
Executes given verbs.  Each verb installs an application or changes a setting.

Options:
    --country=CC      Set country code to CC and don't detect your IP address
-f, --force           Don't check whether packages were already installed
    --gui             Show gui diagnostics even when driven by commandline
    --gui=OPT         Set OPT to kdialog or zenity to override GUI engine
    --isolate         Install each app or game in its own bottle (WINEPREFIX)
    --self-update     Update this application to the last version
    --update-rollback Rollback the last self update
-k, --keep_isos       Cache isos (allows later installation without disc)
    --no-clean        Don't delete temp directories (useful during debugging)
-q, --unattended      Don't ask any questions, just install automatically
-r, --ddrescue        Retry hard when caching scratched discs
-t  --torify          Run downloads under torify, if available
    --verify          Run (automated) GUI tests for verbs, if available
-v, --verbose         Echo all commands as they are executed
-h, --help            Display this message and exit
-V, --version         Display version and exit

Commands:
list                  list categories
list-all              list all categories and their verbs
apps list             list verbs in category 'applications'
benchmarks list       list verbs in category 'benchmarks'
dlls list             list verbs in category 'dlls'
fonts list            list verbs in category 'fonts'
games list            list verbs in category 'games'
settings list         list verbs in category 'settings'
list-cached           list cached-and-ready-to-install verbs
list-download         list verbs which download automatically
list-manual-download  list verbs which download with some help from the user
list-installed        list already-installed verbs
arch=32|64            create wineprefix with 32 or 64 bit, this option must be
                      given before prefix=foobar and will not work in case of
                      the default wineprefix.
prefix=foobar         select WINEPREFIX=/home/guest/.local/share/wineprefixes/foobar
annihilate            Delete ALL DATA AND APPLICATIONS INSIDE THIS WINEPREFIX
```

При запуске без параметров Winetricks отображает графический интерфейс со списком доступных пакетов. Если вам известны имена пакетов, которые вы хотите установить, вы можете добавить их в команду Winetricks, и процесс установки сразу же начнётся. Например, команда

guest@localhost ~> winetricks corefonts vcrun6

установит пакеты corefonts и vcrun6.

Как и все команды Wine, Winetricks знает о переменной среды WINEPREFIX. Это полезно для использования Winetricks с разными расположениями префиксов Wine. Например,

guest@localhost ~> env WINEPREFIX=/home/guest/.wine-new sh winetricks mfc40[8]

устанавливает пакет mfc40 в папку ~/.wine-new.

Использование Bottles

Bottles — это бесплатная программа с открытым исходным кодом, которое упрощает создание «префиксов» вина, подобных тем, что используются в операционных системах Windows. Слоган программного обеспечения – «Запустите Windows в бутылке». Bottles работают с «префиксами» Wine, которые имитируют иерархию файловой системы операционной системы Windows и могут использоваться для хранения программного обеспечения, предназначенного только для этой платформы.

Еще одна особенность использования Bottles – поддержка большого количества компьютерных игр для Windows. После установки программы вы получите доступ к

[8] Microsoft Foundation Classes (MFC) – это библиотека на языке C++, разработанная Microsoft и призванная облегчить разработку GUI-приложений для Microsoft Windows путём использования богатого набора библиотечных классов.

популярным игровым магазинам, например, Epic Games Store, EA Launcher, Battle.net и т.д.

Особенности и функционал Bottles:

– Создание «бутылок», используя предварительно настроенные среды, или создавайте свои собственные;

– Запуск исполняемых файлов (.exe или .msi) в «бутылках» прямо из контекстного меню вашего файлового менеджера;

– Автоматическое обнаружение приложений, установленных в ваших «бутылках»;

– Быстрое добавление переменных среды;

– Переопределение библиотек DLL непосредственно из настроек для каждой «бутылки»;

– Различная оптимизация производительности в играх (esync, fsync, DXVK, кэш, компилятор шейдеров и т.д.);

– Автоматическая установка и управление исполняющими программами Wine и Proton;

Автоматический восстановление «бутылок» в случае их поломки;

– Интегрированный установщик зависимостей на основе репозитория сообщества;

– Встроенный диспетчер задач для процессов Wine;

– Доступ к ProtonDB и WineHQ;

– Система переноса вашей конфигурации на новые версии Bottles;

– Резервное копирование и импорт «бутылок»;

– Импорт префиксов Wine из других менеджеров.

Лучший способ установить Bottles – это использовать менеджер пакетов Flatpak, который по умолчанию не устанавливается в большинстве дистрибутивов.

Особенностью Flatpak является их одноименный универсальный формат пакетов. Все зависимости программы уже находятся в самом пакете, именно такие, какие надо и их не нужно устанавливать отдельно. Поэтому пакеты Flatpak могут быть установлены в любом дистрибутиве. Преимуществом подобного распространения ПО заключается в том, что Flatpak помогает избежать конфликтов с зависимостями.

Необходимо выполнить это задание в дистрибутиве Debian! На виртуальной машине в Arch Linux по неизвестной причине работа Bottles происходит с ошибками и чудовищно медленно.

Давайте сперва установим этот пакетный менеджер:

guest@localhost:~$ sudo apt install flatpak

Пользователям, впервые устанавливающим Flatpak рекомендуется перезагрузить систему. Невыполнение этого требования может привести к возникновению проблем, например, пути для иконок не создаются.

guest@localhost:~$ sudo reboot

Затем вам нужно включить Flatpak:

guest@localhost:~$ flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo

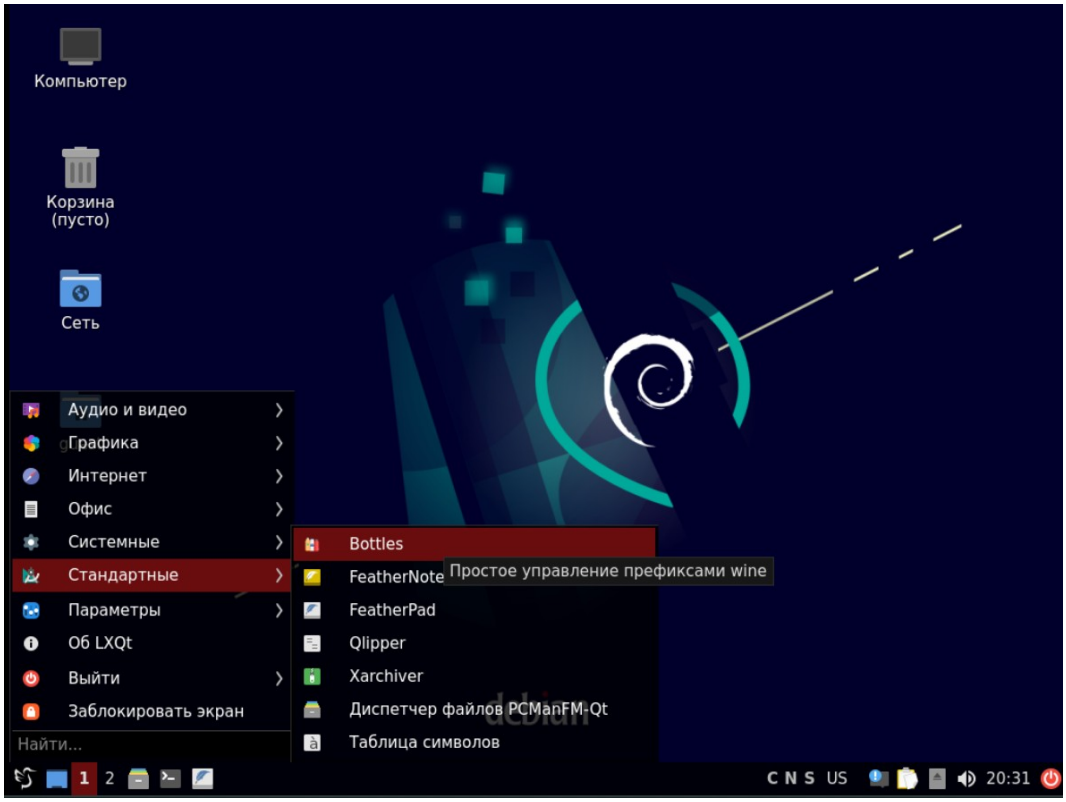
Теперь установите Bottles с помощью следующей команды Flatpak:

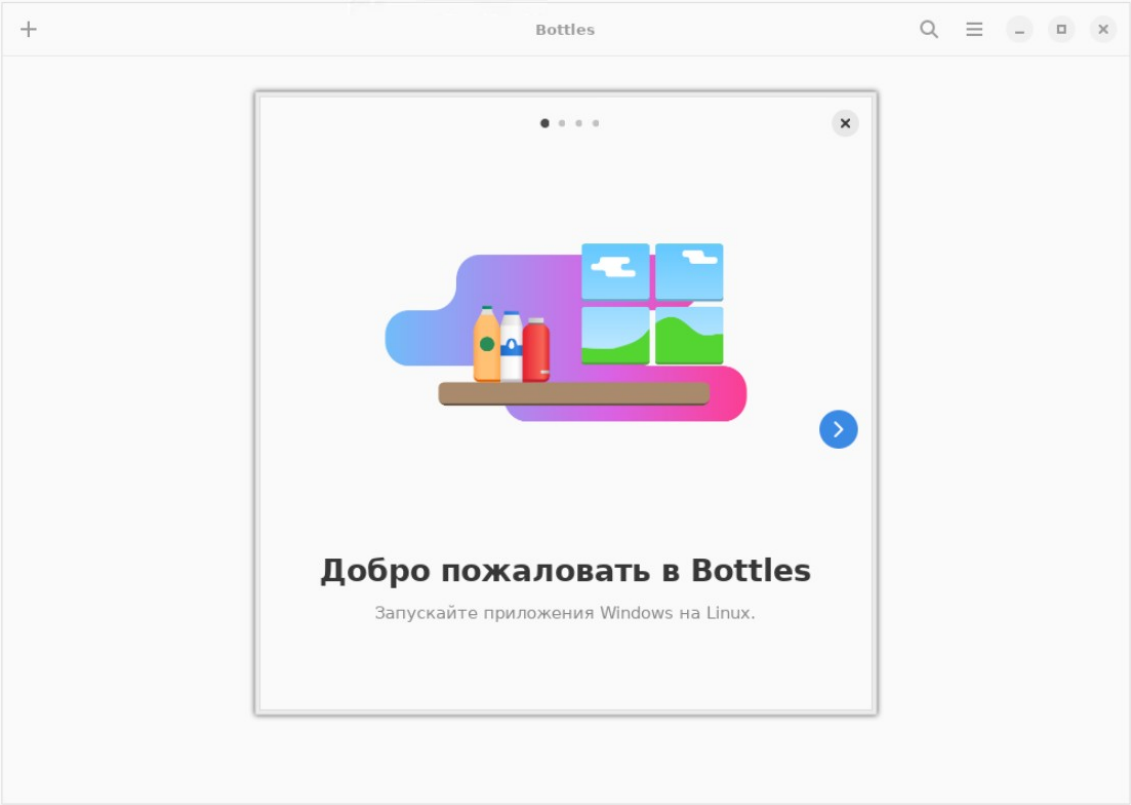
guest@localhost:~$ flatpak install flathub com.usebottles.bottles

Приложение Bottles можно запустить сразу с вашего терминала с помощью следующей команды:

guest@localhost:~$ flatpak run com.usebottles.bottles

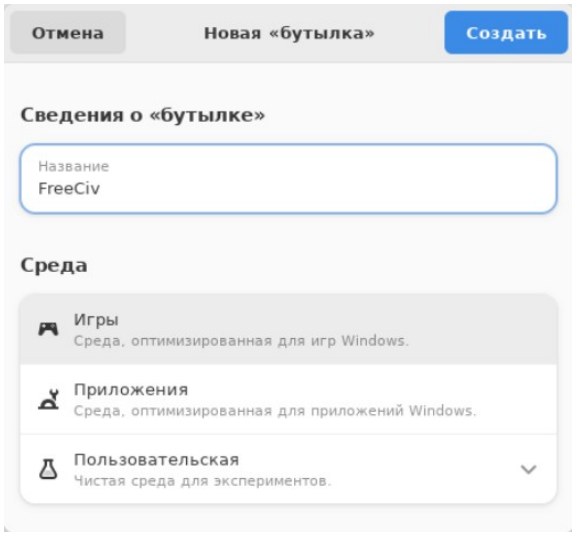
При установке Flatpak в стандартных средах рабочего стола (GNOME, KDE, Xfce) у вас появится ярлык в меню выбора программ и вам не нужно будет запускать его через терминал.





При первом запуске Bottles вы пройдете мастер быстрой предварительной установки, чтобы закончить установку с графическим интерфейсом. В целом это должно занять не более нескольких минут.

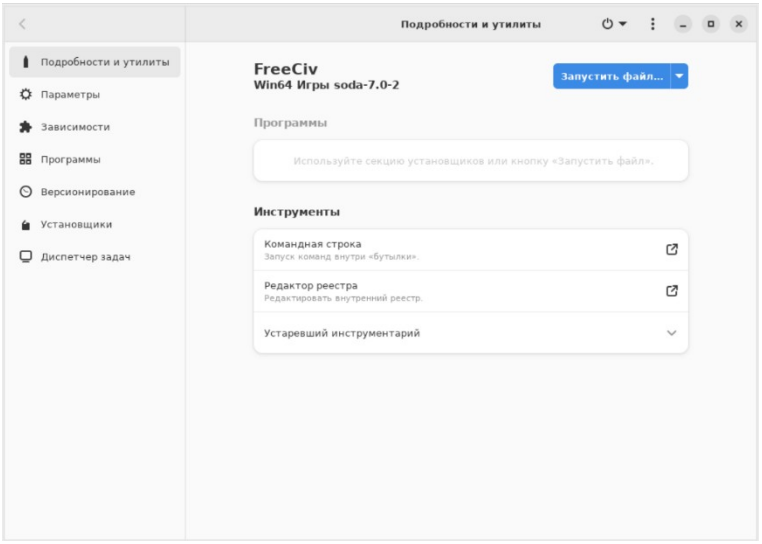
Сначала создайте и назовите вашу среду.

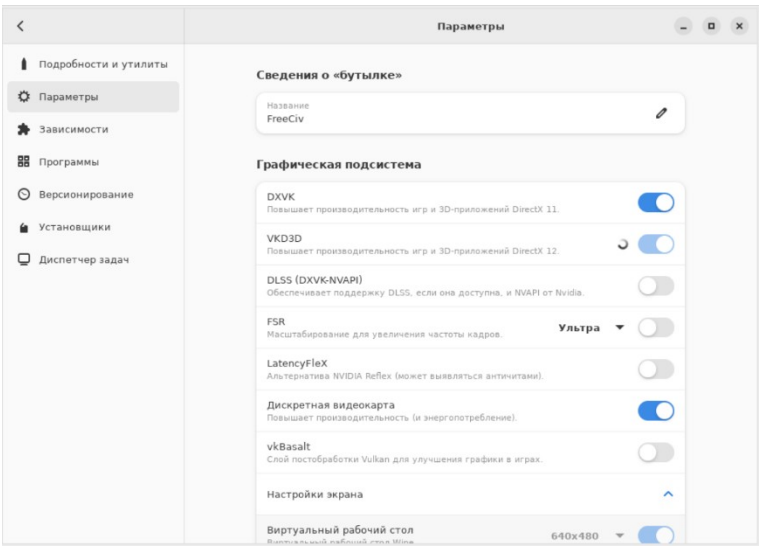


Далее вам нужно подождать, пока будет создана бутылка; она в этот момент будет устанавливать все зависимости Window.

После завершения вы получите информацию о «бутылках». Вы можете настроить параметры, зависимости, программы, установщики и многое другое.

Пример параметров «бутылки» изображен ниже.





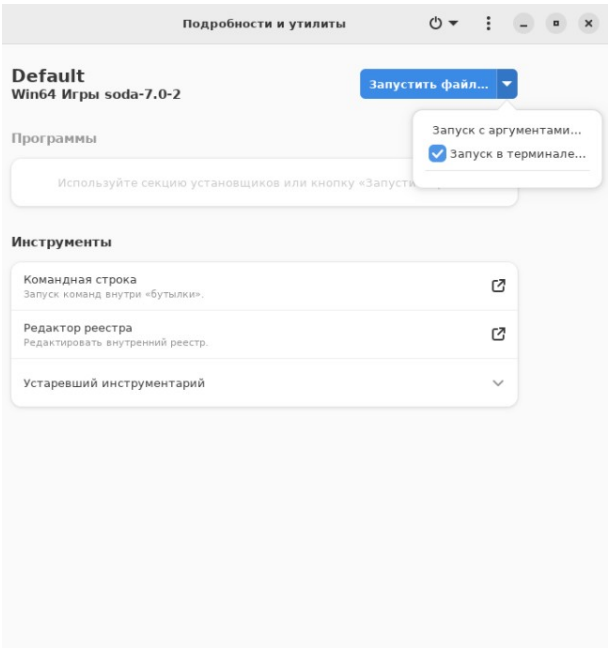
Давайте попробуем самостоятельно установить и запустить Windows-программу в Bottles.

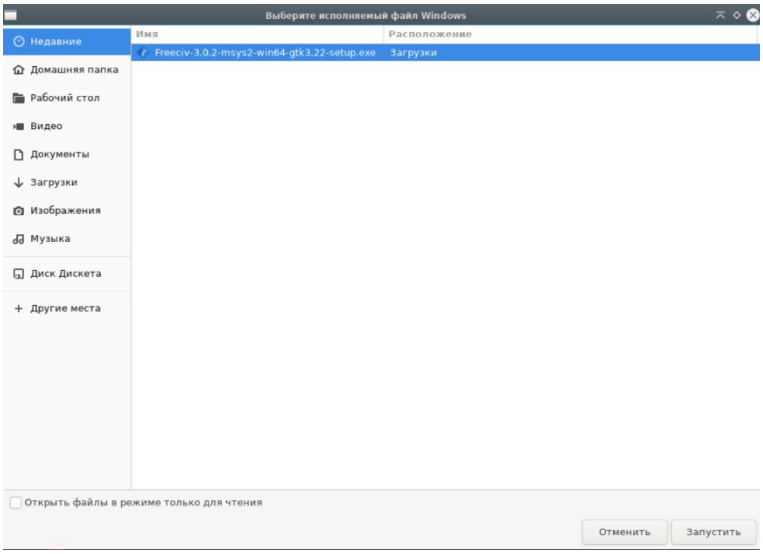
Заранее скачайте какую-либо программу, чтобы проверить сможет ли она запуститься. В примере будет установлен и запущен FreeCiv. Скачать и установить его можно по следующим ссылке: http://prdownloads.sourceforge.net/freeciv/Freeciv-3.0.2-msys2-win64-gtk3.22-setup.exe?download

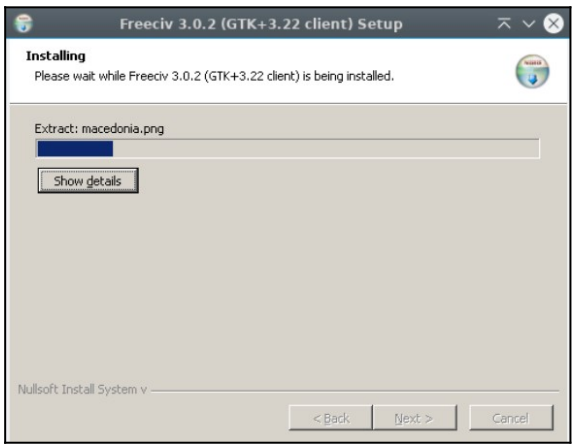
После скачивания скопируйте исполняемый файл в директорию «/home/guest»:

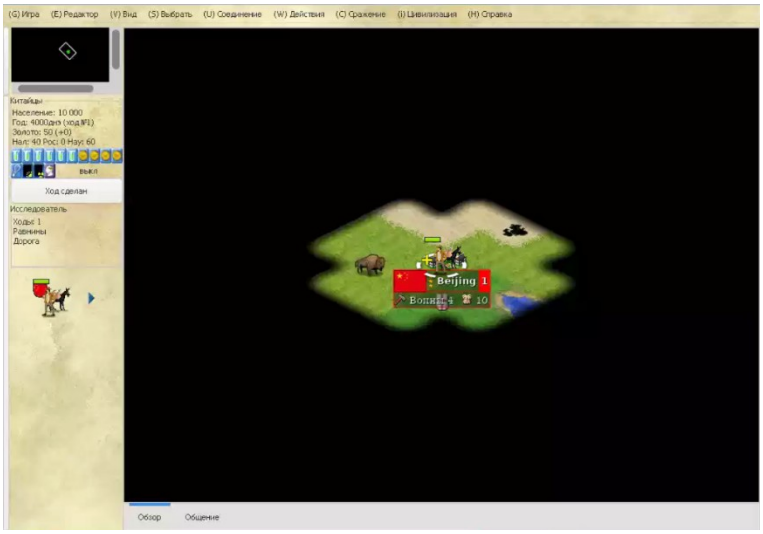
guest@localhost:~$ cp Freeciv-3.0.2-msv2-win64-gtk3.22-setup.exe /home/guest/ (Название скачанного файла может отличаться)

Затем запустите установщик FreeCiv через кнопку «Запустить файл…».









Лучший способ обновить программное обеспечение в Flatpak – это использовать следующую команду терминала:

guest@localhost:~$ flatpak update

По умолчанию Flatpak должен автоматически проверять наличие обновлений.

Если вы больше не хотите больше использовать Bottles, используйте следующую команду для удаления программного обеспечения:

guest@localhost:~$ flatpak uninstall --delete-data com.usebottles.bottles

Затем выполните следующую команду для удаления остаточных файлов:

guest@localhost:~$ flatpak remove –unused

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №9

Тема: Обзор схемы стенда сети предприятия на основе GNU/Linux

Цель работы: Построить рабочую структурную сеть стенда сети предприятия для дальнейшего масштабирования вычислительной сети в последующих работах.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Если компьютеры, подключенные к одному коммутатору не пингуются между собой, то попробуйте сначала перезапустить все виртуальные машины кроме самого коммутатора. Далее попробуйте выполнить какие-либо активные действия в коммутаторе, например, команды:
switch>en
switch#sh ip int br
чтобы «разбудить» виртуальную машину. Если и это не помогло, тогда полностью сотрите конфигурацию коммутатора (кнопка «Wipe») и перезапустить его.

Имейте в виду, что Cisco vIOS Router и Cisco IOL L2 (Switch) значительно уменьшают пропускную способность, поэтому если вы хотите что-то быстро скачать на виртуальную машину, то лучше это делать, подключая компьютер напрямую к облаку.

В схеме сети предприятия, внутри которой будет проводится дальнейшее администрирование компьютерных систем, присутствуют две виртуальные машины с Cisco IOS, три виртуальные машины с GNU\Linux Debian 11 и одна виртуальная машина с Windows 10. Пароль по умолчанию на Debian 11 – «guest», а на Windows 10 – «gu3st~».

Общая схема стенда будет выглядеть следующим образом.

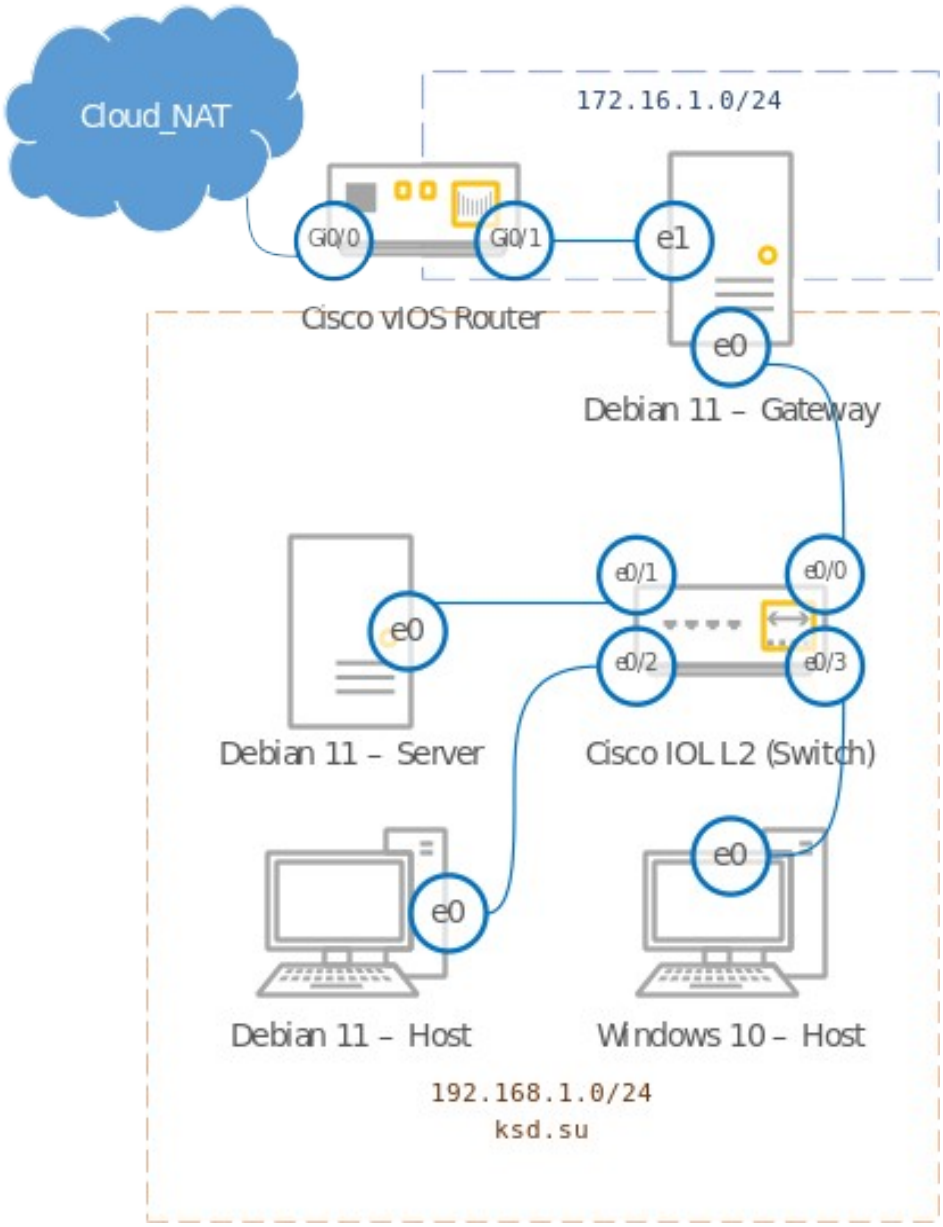


Схема сетевых интерфейсов сети предприятия

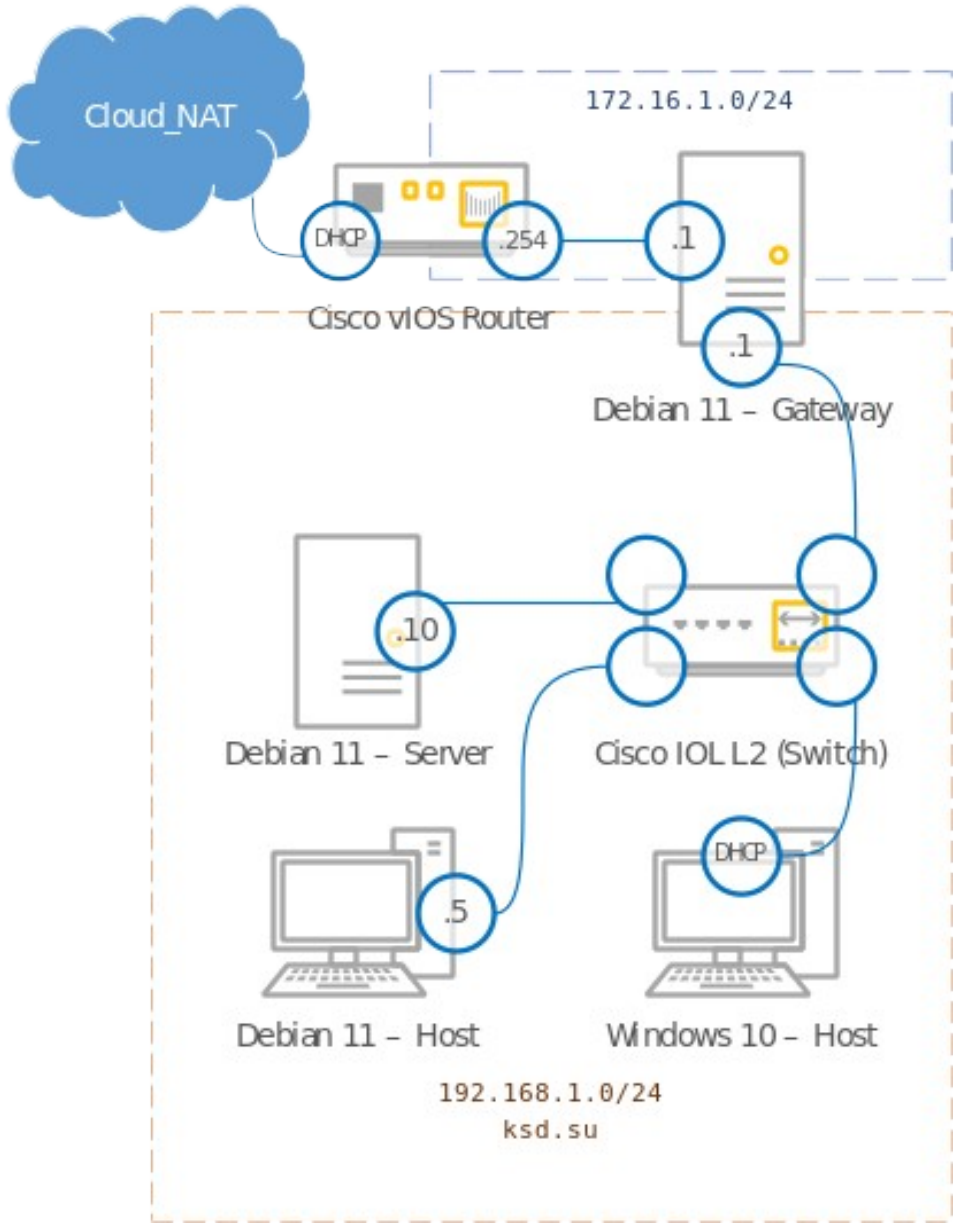


Схема IP-адресов сети предприятия

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №10

Тема: Настройка шлюза сети предприятия

Цель работы: Сконфигурировать виртуальную машину в качестве шлюза сети предприятия:

– Настройка сетевых интерфейсов и установка IP-адреса, полученного по DHCP и статического IP-адреса;

– Проверка сетевых интерфейсов и доступа к Интернету;

– Настройка маскирующего NAT (PAT);

– Переименование маршрутизатора;

– Установка пароля на консоль;

– Настройка сетевых интерфейсов и установка статических IP-адресов ;

– Изменение DNS-сервера по умолчанию;

– Изменение имени хоста (Hostname);

– Включение пересылки пакетов (IP Forwarding);

– Настройка статического маршрута по умолчанию;

– Создание скрипта для добавления статического маршрута по умолчанию и внесение его в автозапуск;

– Запуск скрипта при запуске, как только будет готово к использованию интернет-соединение;

– Проверка сетевых интерфейсов и доступа к Интернету.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Входим на виртуальную машину Cisco vIOS Router в режиме ручной конфигурации и прописываем в терминале следующие команды.

Настройка сетевых интерфейсов и установка IP-адреса, полученного по DHCP и статического IP-адреса

```
Router>enable
Router#show ip interface brief
Router#configure terminal
Router(config)#ip routing
Router(config)#interface gigabitEthernet0/0
Router(config-if)#ip address dhcp
Router(config-if)#ip nat outside
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface gigabitEthernet0/1
Router(config-if)#ip address 172.16.1.254 255.255.255.0
Router(config-if)#ip nat inside
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit
Router#write memory
Router#reload
```

Проверка сетевых интерфейсов и доступа к Интернету

```
Router>enable
Router#show ip interface brief
Router#ping ya.ru
```

Настройка маскирующего NAT (PAT)[9]

```
Router#configure terminal
Router(config)#ip route 0.0.0.0 0.0.0.0 10.0.137.1
```

[9] Network Address Translation, Port Address Translation

Router(config)#ip route 192.168.1.0 255.255.255.0 172.16.1.1
Router(config)#ip access-list standard FOR-NAT
Router(config-std-nacl)#permit 172.16.1.0 0.0.0.255
Router(config-std-nacl)#permit 192.168.1.0 0.0.0.255
Router(config-std-nacl)#end
Router#show running-config
Router#configure terminal
Router(config)#ip nat inside source list FOR-NAT interface gigabitEthernet0/0 overload
Router(config)#end
Router#show ip nat translations (Введите повторно после того, как вы сделаете команду «ping» в Debian 11 – Gateway)

Переименование маршрутизатора

Как маршрутизаторы, так и коммутаторы имеют имя (hostname). Имя это по большей части используется для удобства администратора, чтобы можно было отличать множество устройств друг от друга. Изначально все маршрутизаторы имеют имя «Router», а коммутаторы – «Switch», что не сильно удобно. Имя задается командой hostname, которую можно вести только в режиме глобальной конфигурации:

Router#configure terminal
Router(config)#hostname cisco-vios-router

Установка пароля на консоль

Для защиты устройств Cisco от несанкционированного доступа используется несколько видов паролей. Мы рассмотрим настройку пароля на консоль и на доступ в привилегированный режим работы устройства. Пароли настраиваются одинаковым образом для маршрутизаторов и коммутаторов.

Пароль на консоль. При подключении к устройству через консольный провод необходимо ввести пароль. По умолчанию пароль на консоль отсутствует. Надо понимать, что физическая безопасность устройства наиболее важный аспект защиты, так как имея физический доступ к консольному порту, даже не зная пароля его можно сбросить**.**

cisco-vios-router(config)#line console 0
cisco-vios-router(config-line)#password guest
cisco-vios-router(config-line)#login
cisco-vios-router(config-line)#end
cisco-vios-router#exit
Password: guest

Пароль на telnet и ssh. Доступ по протоколам telnet или ssh может быть осуществлен только после того**,** как на устройстве настроен какой-то IP-адрес, а также заданы пароли. В этом важное отличие от доступа по консоли. Если пароли не заданы, то по консоли можно зайти без пароля, а по **telnet** или **ssh** зайти нельзя – будет выдано сообщение, что пока нет пароля, удаленный вход запрещен. Задается пароль следующим образом:

cisco-vios-router>enable

```
cisco-vios-router#configure terminal
cisco-vios-router(config)#line vty 0 15
cisco-vios-router(config-line)#password guest
cisco-vios-router(config-line)#login
cisco-vios-router(config-line)#exit
```

Процедура аналогична настройке пароля на консоль, только действия выполняются не в режиме конфигурирования консоли (con 0), а в режиме настройки виртуальных терминалов (vty 0 4), где цифры «0» и «4» следует трактовать как «Перейти в подрежим конфигурирования всех виртуальных терминалов с нулевого по четвертый». Обычно для telnet используются именно эти 5 виртуальных терминалов. Если один терминал занят подключением, то человек подключится к следующему свободному. Этот же пароль будет работать и для доступа по ssh, если сам ssh настроен.

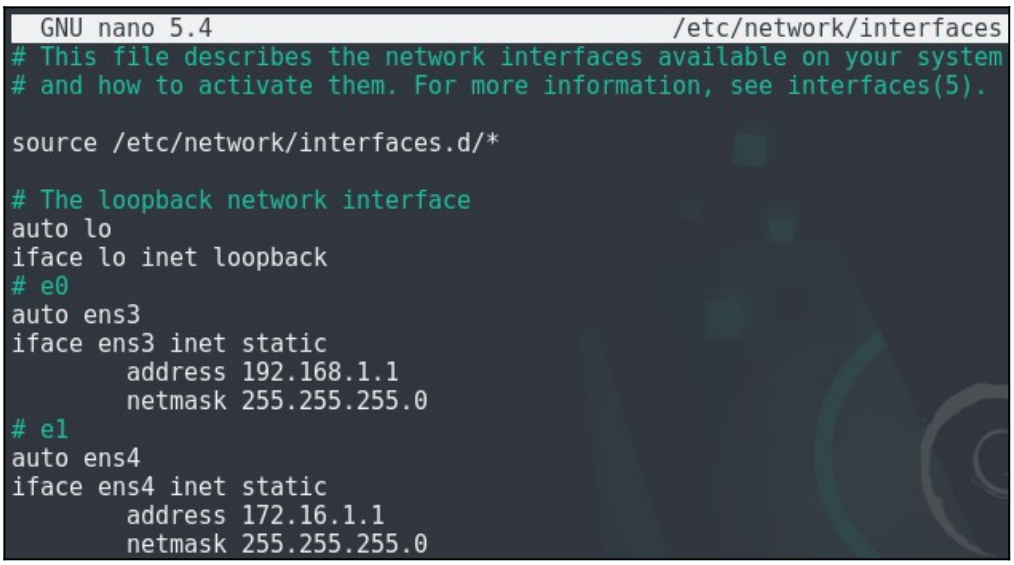
Пароль на привилегированный режим. При входе на устройство, независимо от того, делаем мы это через ssh, telnet или через консоль, мы попадаем в пользовательский режим. Далее можно осуществить переход в привилегированный**.** Если задан пароль на привилегированный режим, то его потребуется ввести, если не задан – то все зависит от того способа, по которому мы подключились к устройству. При подключении по консоли и отсутствующем пароле на enable, переход в привилегированный режим произойдет без ввода пароля, если же доступ осуществляется через telnet или ssh, то без пароля на enable, нас в этот режим не пустят если пароль не задан.

```
cisco-vios-router(config)#enable secret secret
cisco-vios-router#do write memory
cisco-vios-router#do reload
```

Входим на виртуальную машину Debian 11 – Gateway.

Настройка сетевых интерфейсов и установка статических IP-адресов

```
guest@localhost:~$ su -
Пароль: guest
root@localhost:~# ip address
root@localhost:~# nano /etc/network/interfaces
```



Также можно выключить службу connman, чтобы настройка статических IP-адресов применялась сразу после перезагрузки операционной системы:

root@localhost:~# systemctl disable connman.service
root@localhost:~# systemctl stop connman.service
root@localhost:~# systemctl disable avahi-daemon.service
root@localhost:~# systemctl stop avahi-daemon.service
root@localhost:~# systemctl restart networking.service
root@localhost:~# systemctl reboot

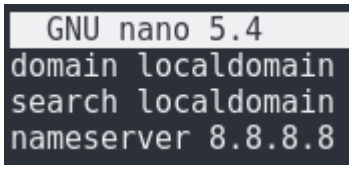
Если после перезагрузки виртуальной машины конфигурация не применилась, то снова перезагрузите службу networking.service с помощью команды:

root@localhost:~# systemctl restart networking.service

Изменение DNS-сервера по умолчанию

Измените в конфигурационном файле IP-адрес DNS-сервера по умолчанию:

root@localhost:~# nano /etc/resolv.conf



Изменение имени хоста (Hostname)

root@localhost:~# hostnamectl set-hostname debian-11-gateway

Включение пересылки пакетов (IP Forwarding)

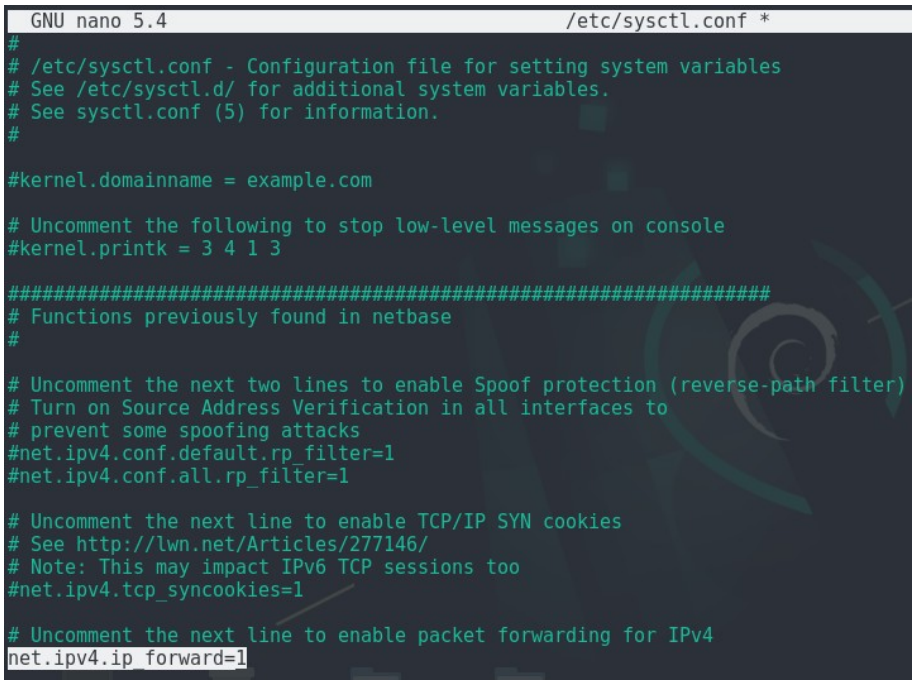
По умолчанию в большинстве дистрибутивов IP Forwarding выключен, но пересылка пакетов может понадобится если на сервере будет подниматься VPN или, например, это будет маршрутизатор, как в нашем случае.

Проверить включен ли IP Forwarding можно так:

root@ldebian-11-gateway:~# cat /proc/sys/net/ipv4/ip_forward
root@debian-11-gateway:~# echo 1 > /proc/sys/net/ipv4/ip_forward (Временное включение до перезагрузки)

Раскомментируйте строку «net.ipv4.ip_forward=1», чтобы Debian 11 – Gateway после перезагрузки всегда включал пересылку пакетов по умолчанию:

root@debian-11-gateway:~# nano /etc/sysctl.conf

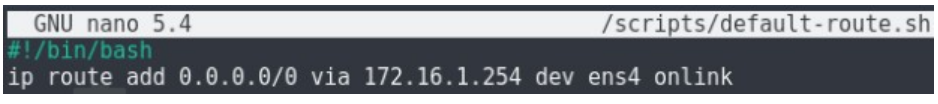


Настройка статического маршрута по умолчанию

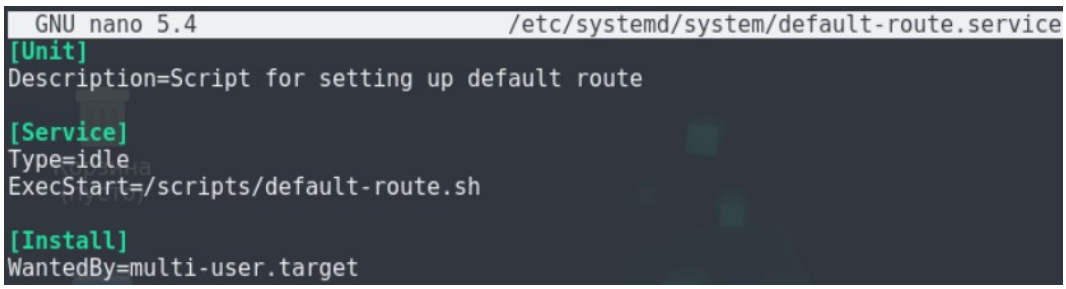
guest@debian-11-gateway:~$ su -
Пароль: guest
root@debian-11-gateway:~# ip route show
root@debian-11-gateway:~# ip route add 0.0.0.0/0 via 172.16.1.254 (Временное включение до перезагрузки)

Создание скрипта для добавления статического маршрута по умолчанию и внесение его в автозапуск

root@debian-11-gateway:~# mkdir /scripts
root@debian-11-gateway:~# cd /scripts
root@debian-11-gateway:/scripts# touch default-route.sh
root@debian-11-gateway:/scripts# chmod ugo+x default-route.sh
root@debian-11-gateway:/scripts# nano default-route.sh (Необходимо заполнить скрипт самостоятельно)



root@debian-11-gateway:/scripts# touch /etc/systemd/system/default-route.service
root@debian-11-gateway:/scripts# nano /etc/systemd/system/default-route.service (Необходимо заполнить юнит самостоятельно)



Обратите внимание, что вам придется перезапускать службу «default-route.service», каждый раз, когда вы вручную перезапускаете «networking.service»!

Перезагрузите systemd с помощью systemctl daemon-reload:

root@debian-11-gateway:/scripts# systemctl daemon-reload

Запуск скрипта при запуске, как только будет готово к использованию интернет-соединение

Поместите ваш созданный скрипт в /etc/network/if-up.d. Он будет автоматически запускаться каждый раз, когда появляется сетевой интерфейс:

root@debian-11-gateway:/scripts# cp default-route.sh /etc/network/if-up.d

Проверьте функционирование вашего созданного юнита systemd:

root@debian-11-gateway:/scripts# systemctl status default-route.service
root@debian-11-gateway:/scripts# systemctl enable default-route.service
root@debian-11-gateway:/scripts# systemctl restart default-route.service
root@debian-11-gateway:/scripts# ip route
root@debian-11-gateway:/scripts# systemctl reboot
guest@debian-11-gateway:~$ su -
Пароль: guest
root@debian-11-gateway:~# systemctl status default-route.service
root@debian-11-gateway:~# ip route show

Проверка сетевых интерфейсов и доступа к Интернету

root@debian-11-gateway:~# ip address
root@debian-11-gateway:~# ping ya.ru
[ctrl+c]

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №11

Тема: Настройка виртуальной машины как программного коммутатора

Цель работы: Сконфигурировать виртуальную машину в качестве программного коммутатора:

– Базовая подготовка виртуальной машины Debian 11 – Switch;

– Настройка сетевого моста;

– Проверка работы сетевого моста и доступа к Интернету.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

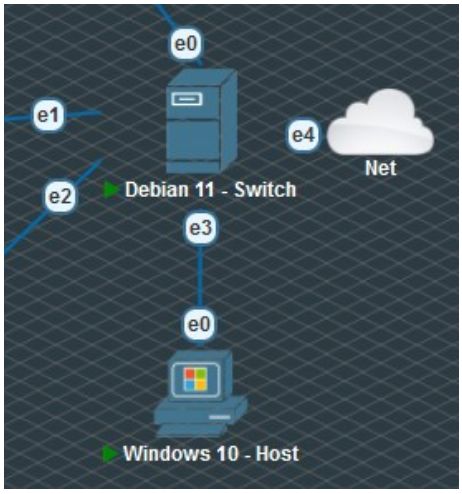
В некоторых случаях имеет смысл заменить наш коммутатор Cisco IOL L2 Switch на виртуальную машину с установленной на ней ОС GNU/Debian. Одна из таких причин – это лицензирование прошивок Cisco. Из-за того, что у этих прошивок предназначены только для внутренних обучающих целей, то у них стоит лимит на внутреннюю пропускную способность программного коммутатора. Чтобы этого избежать можно собрать собственный коммутатор на основе программных мостов в Linux.

В терминологии Ethernet «коммутатор» (switch) и «мост» (bridge) – это синонимы. Термин switch придумали, чтобы отличать многопортовые коммутаторы от первых двухпортовых мостов. Программные реализации коммутатора Ethernet по традиции называют мостами, независимо от числа портов.

Вот и в Linux программный коммутатор называется мостом. Производительность таких коммутаторов не слишком высока – ни о каких десяти гигабитах речь не идет. Кроме того, встроенная функциональность ядра Linux для этой цели достаточно ограниченна по сравнению с новыми проектами вроде Open vSwitch. Тем не менее в ряде случаев она может оказаться полезной.

Сетевой мост (bridge, бридж) – это способ соединения двух сегментов Ethernet на канальном уровне, т.е. без использования протоколов более высокого уровня, таких как IP. Пакеты передаются на основе Ethernet-адресов, а не IP-адресов (как в маршрутизаторе). Поскольку передача выполняется на канальном уровне (уровень 2 модели OSI), все протоколы более высокого уровня прозрачно проходят через мост. Термины коммутатор, мост и бридж могут использоваться как взаимознаменяемые.

Удаляем виртуальную машину Cisco IOL L2 Switch и заменяем его на машину с названием «Debian 11 – Switch». Затем добавим в эту виртуальную машину 5 сетевых интерфейсов и входим в нее.



Базовая подготовка виртуальной машины Debian 11 – Switch

Первым делом переименуем виртуальную машину:

```
guest@localhost:~$ su -
Пароль: guest
root@localhost:~# hostnamectl set-hostname debian-11-switch
```

Введите команду apt для установки bridge-utils (для скачивания пакетов подключите виртуальную машину к облаку «Net» [Cloud_NAT] напрямую):

```
root@localhost:~# apt update
root@localhost:~# apt install bridge-utils
```

* Не забудьте удалить облако после того, как вы скачаете утилиту bridge-utils

Настройка сетевого моста

Вам нужно отредактировать файл /etc/network/interface. Тем не менее, рекомендуется поместить новый конфиг в каталог /etc/network/interface.d/. Процедура настройки сетевого моста выглядит следующим образом.

1. Используйте IP-команду:

```
root@localhost:~# ip -f inet a s
```

2. Впишите в файл /etc/network/interface следующее:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

iface ens3 inet manual
iface ens4 inet manual
iface ens5 inet manual
iface ens6 inet manual
```

4. Создайте текстовый файл с помощью текстового редактора:

```
root@localhost:~# nano /etc/network/interfaces.d/br0
```

```
# static ip config file for br0 #
auto br0
iface br0 inet static
        bridge_ports ens3 ens4 ens5 ens6
                address 192.168.1.2
                broadcast 192.168.1.255
                netmask 255.255.255.0
                gateway 192.168.1.1
```

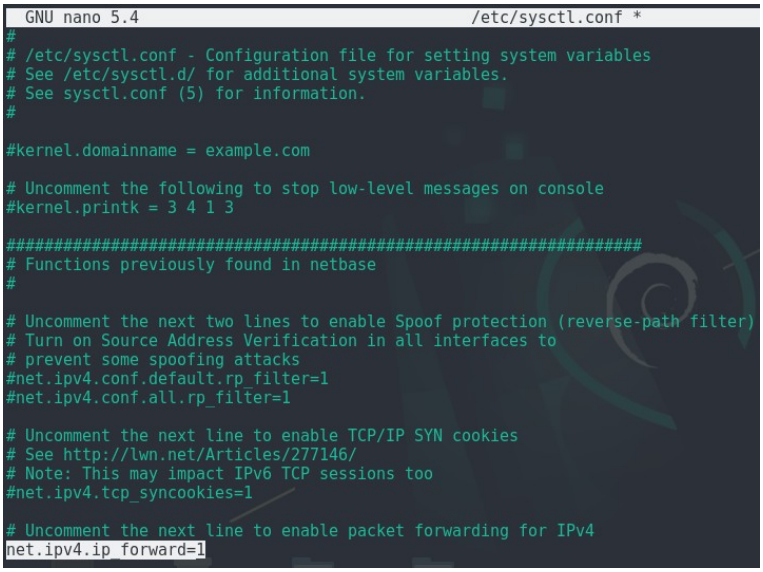
5. Сохраните и закройте конфигурационные файлы в nano.

6. Проверить включен ли IP Forwarding можно так:

root@ldebian-11-switch:~# cat /proc/sys/net/ipv4/ip_forward
root@debian-11-switch:~# echo 1 > /proc/sys/net/ipv4/ip_forward (Временное включение до перезагрузки)

Раскомментируйте строку «net.ipv4.ip_forward=1», чтобы Debian 11 – Switch после перезагрузки всегда включал пересылку пакетов по умолчанию:

root@debian-11-gateway:~# nano /etc/sysctl.conf



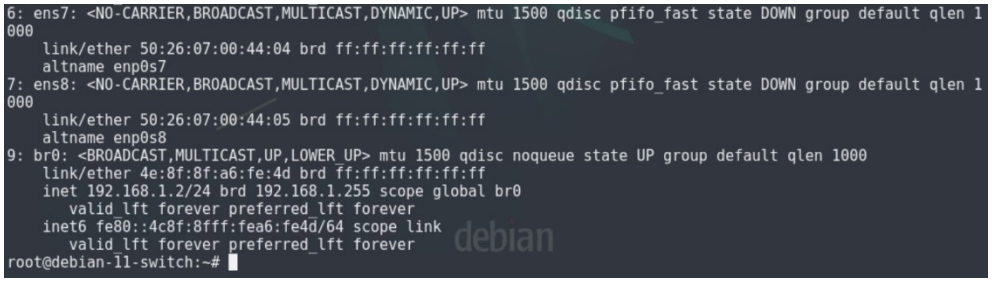
7. Перед перезапуском сетевой службы убедитесь, что брандмауэр в операционной системе отключен. В нашем случае брандмауэр по умолчанию отключен в Debian. После перезапуска службы необходимо обновить правило брандмауэра для интерфейса br0. Введите следующее, чтобы перезапустить сетевую службу:

root@localhost:~# systemctl restart networking
root@localhost:~# systemctl status networking

Проверка работы сетевого моста и доступа к Интернету

Найдите новый интерфейс br0 и таблицу маршрутизации с помощью команды ip:

root@localhost:~# ip a s



root@localhost:~# ip r

```
root@debian-11-switch:~# dhclient -r
root@debian-11-switch:~# ip r
default dev ens3 scope link
default via 192.168.1.1 dev br0 onlink
169.254.0.0/16 dev ens3 proto kernel scope link src 169.254.4.22
169.254.0.0/16 dev ens6 proto kernel scope link src 169.254.195.159
169.254.0.0/16 dev ens5 proto kernel scope link src 169.254.85.202
169.254.0.0/16 dev ens4 proto kernel scope link src 169.254.249.112
192.168.1.0/24 dev br0 proto kernel scope link src 192.168.1.2
root@debian-11-switch:~#
```

Вы также можете использовать команду brctl для просмотра информации о ваших мостах:

root@localhost:~# brctl show

Показать текущие мосты:

root@localhost:~# bridge link

```
root@debian-11-switch:~# brctl show
bridge name     bridge id               STP enabled     interfaces
br0             8000.4e8f8fa6fe4d       no              ens3
                                                        ens4
                                                        ens5
                                                        ens6
root@debian-11-switch:~# bridge link
2: ens3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 master br0 state forwarding priority 32 cost 100
3: ens4: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 master br0 state forwarding priority 32 cost 100
4: ens5: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 master br0 state forwarding priority 32 cost 100
5: ens6: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 master br0 state forwarding priority 32 cost 100
root@debian-11-switch:~#
```

Входим на виртуальную машину Debian 11 – Host.

root@localhost:~# ping -c 2 ya.ru

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №12

Тема: Развертывание сервера DHCP

Цель работы: Сконфигурировать виртуальную машину в качестве сервера DHCP на основе dnsmasq:

– Изменение имени хоста (Hostname);

– Настройка сетевых интерфейсов и установка статических IP-адресов ;

– Изменение DNS-сервера по умолчанию;

– Проверка сетевых интерфейсов и доступа к Интернету;

– Обновление локального индекса пакетов;

– Установка пакета с программой dnsmasq и проверка его на работоспособность;

– Настройка dnsmasq как DHCP-сервера;

– Настройка сетевого адаптера на работу по DHCP;

– Разрешение ICMP-трафика в Брандмауэре Windows;

– Проверка сетевых интерфейсов и доступа к Интернету.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

DHCP (Dynamic Host Configuration Protocol – протокол динамической настройки узла) – это прикладной протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

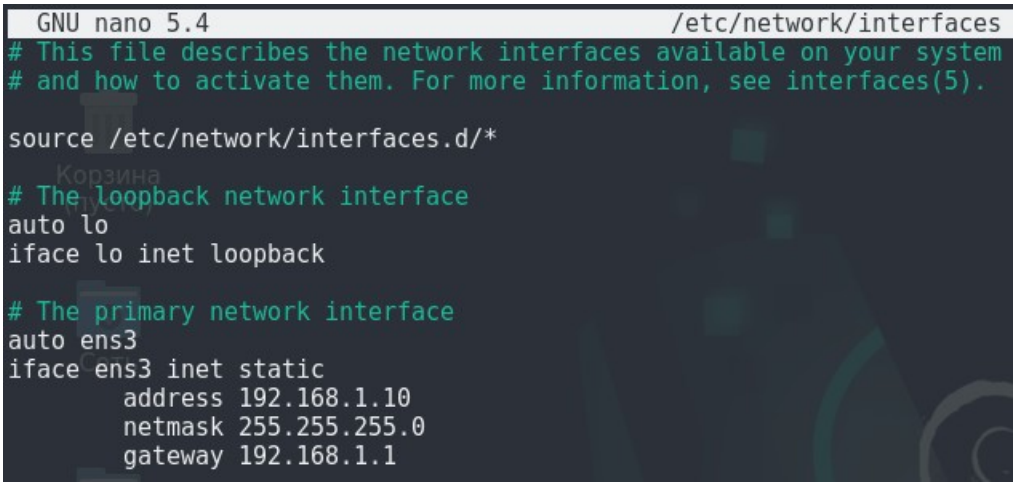
Входим на виртуальную машину Debian 11 – Server.

Изменение имени хоста (Hostname)

guest@localhost:~$ su -
Пароль: guest
root@localhost:~# hostnamectl set-hostname debian-11-server

Настройка сетевых интерфейсов и установка статических IP-адресов

root@debian-11-server:~# ip address
root@debian-11-server:~# nano /etc/network/interfaces



root@debian-11-server:~# systemctl disable avahi-daemon.service
root@debian-11-server:~# systemctl restart networking.service
root@debian-11-server:~# systemctl reboot

Также можно выключить службу connman, чтобы настройка статических IP-адресов применялась сразу после перезагрузки операционной системы:

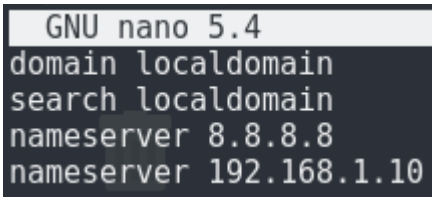
guest@debian-11-server:~$ su -
root@debian-11-server:~# systemctl disable connman.service
root@debian-11-server:~# systemctl stop connman.service

Если после перезагрузки виртуальной машины конфигурация не применилась, то снова перезагрузите службу networking.service с помощью команды:

root@debian-11-server:~# systemctl restart networking.service
root@debian-11-server:~# ip address

Изменение DNS-сервера по умолчанию

root@debian-11-server:~# nano /etc/resolv.conf



Проверка сетевых интерфейсов и доступа к Интернету

root@debian-11-server:~# ip address
root@debian-11-server:~# ping 192.168.1.10
[ctrl+c]
root@debian-11-server:~# ping 192.168.1.1
[ctrl+c]
root@debian-11-server:~# ping 172.16.1.1
[ctrl+c]
root@debian-11-server:~# ping 172.16.1.254
[ctrl+c]
root@debian-11-server:~# ping ya.ru
[ctrl+c]

Обновление локального индекса пакетов

guest@debian-11-server:~$ su –
Пароль: guest
root@debian-11-server:~# apt update

Установка пакета с программой dnsmasq и проверка его на работоспособность

dnsmasq – это легковесный и быстроконфигурируемый DNS-, DHCP- и TFTP-сервер, предназначенный для обеспечения доменными именами и связанными с ними сервисами небольших сетей. Может обеспечивать именами локальные машины, которые не имеют глобальных DNS-записей. DHCP-сервер интегрирован с DNS-сервером и дает машинам с IP-адресом доменное имя, сконфигурированное ранее в конфигурационном файле. Поддерживает привязку IP-адреса к компьютеру или автоматическую настройку IP-адресов из заданного диапазона и BOOTP для сетевой загрузки бездисковых машин.

Разработчики позиционируют программу для использования в домашних сетях, использующих NAT, однако dnsmasq также применим в малых сетях организаций.

root@debian-11-server:~# apt install dnsmasq
root@debian-11-server:~# systemctl status dnsmasq.service

Настройка dnsmasq как DHCP-сервера

Настройка конфигурационного файла dnsmasq выполняется следующим образом:

root@debian-11-server:~# cp /etc/dnsmasq.conf /etc/dnsmasq.conf.old (Сохранение конфигурационного файла по умолчанию, чтобы в случае ошибок откатиться обратно к нему)
root@debian-11-server:~# nano -c /etc/dnsmasq.conf

Программа dnsmasq обычно прослушивает все ваши сетевые интерфейсы. Если вы хотите использовать определенный интерфейс, определите его сейчас, добавив или раскомментировав следующую строку в файле конфигурации:

interface=ens3 (Строка 106)

Сервер DHCP включается путем указания диапазона IP-адресов DHCP:

dhcp-range=192.168.1.15,192.168.1.20,255.255.255.0,6h (Строка 157)

Необходимо выставить в dnsmasq IP-адреса от 192.168.1.15 до 192.168.1.20 в подсети 255.255.255.0. Срок аренды выданных IP-адресов составляет шесть часов, после чего клиентам необходимо будет запросить продление аренды.

Если в конфигурационном файле не настраивать дополнительные опции, то при таком использовании ваш сервер dnsmasq станет шлюзом по умолчанию для ваших устройств, когда они получат IP-адрес. В нашем случае так быть не должно, поэтому вам необходимо изменить это, чтобы dnsmasq перестал предлагать себя в качестве шлюза:

dhcp-option=3,192.168.1.1 (Строка 331)

Укажите, что это единственный DHCP-сервер:

dhcp-authoritative (Строка 548)

Впишите следующую строку, чтобы разрешить ведение логов DHCP-сервера:

log-dhcp (Строка 664)

Когда вы закончите настройку dnsmasq, проверьте его конфигурацию, чтобы убедиться, что ваши изменения действительны:

root@debian-11-server:~# dnsmasq --test
root@debian-11-server:~# systemctl restart dnsmasq

Сброс аренды IP-адресов

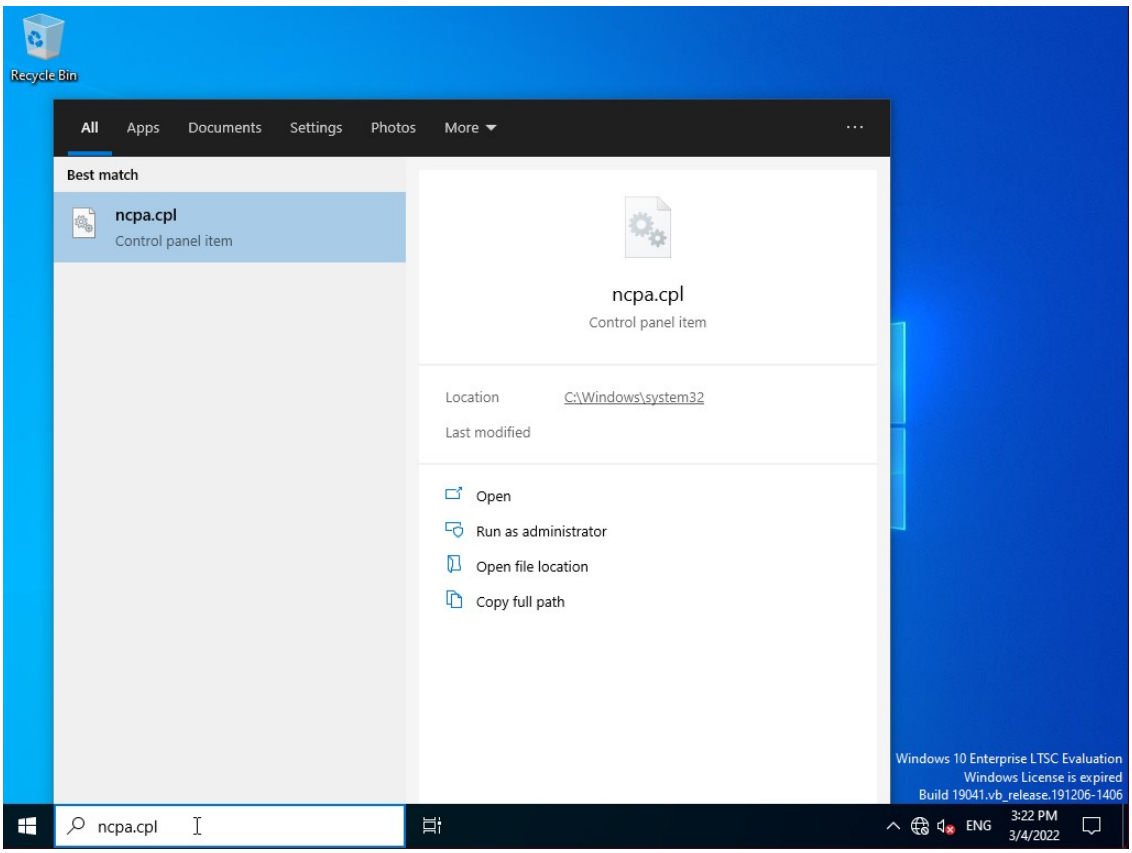
Служба dnsmasq хранит информацию о аренде IP-адресов в /var/lib/misc/dnsmasq.leases. Если вы хотите удалить аренду преждевременно, выключите службу dnsmasq, удалите аренду из файла dnsmasq.leases и снова запустите службу:

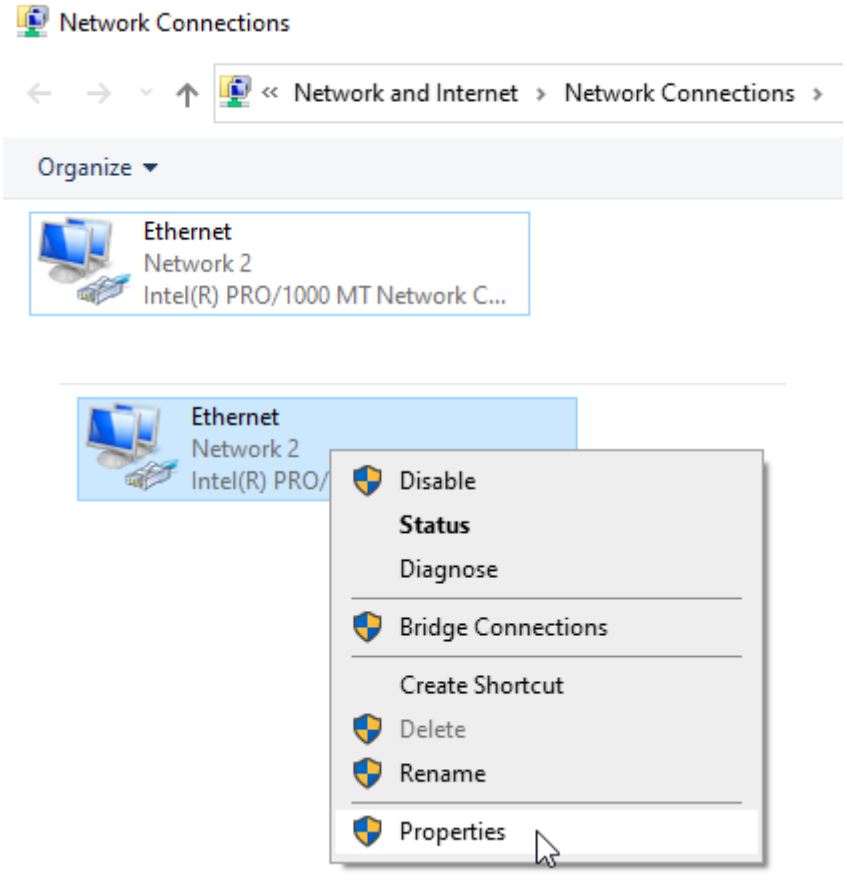
 # systemctl stop dnsmasq
 # nano -w /var/lib/misc/dnsmasq.leases (Почистите там всю информацию)
 # systemctl start dnsmasq

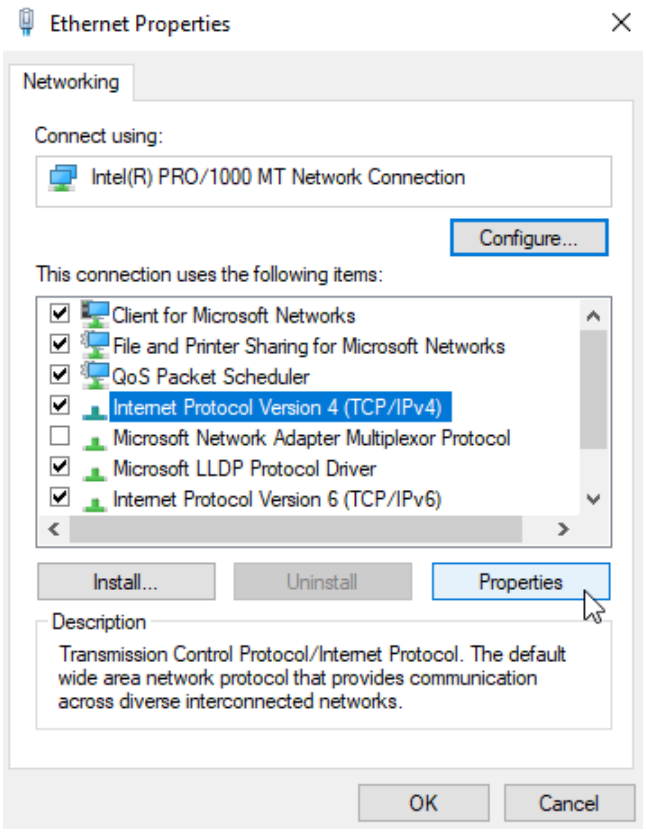
Входим на виртуальную машину Windows 10 – Host.

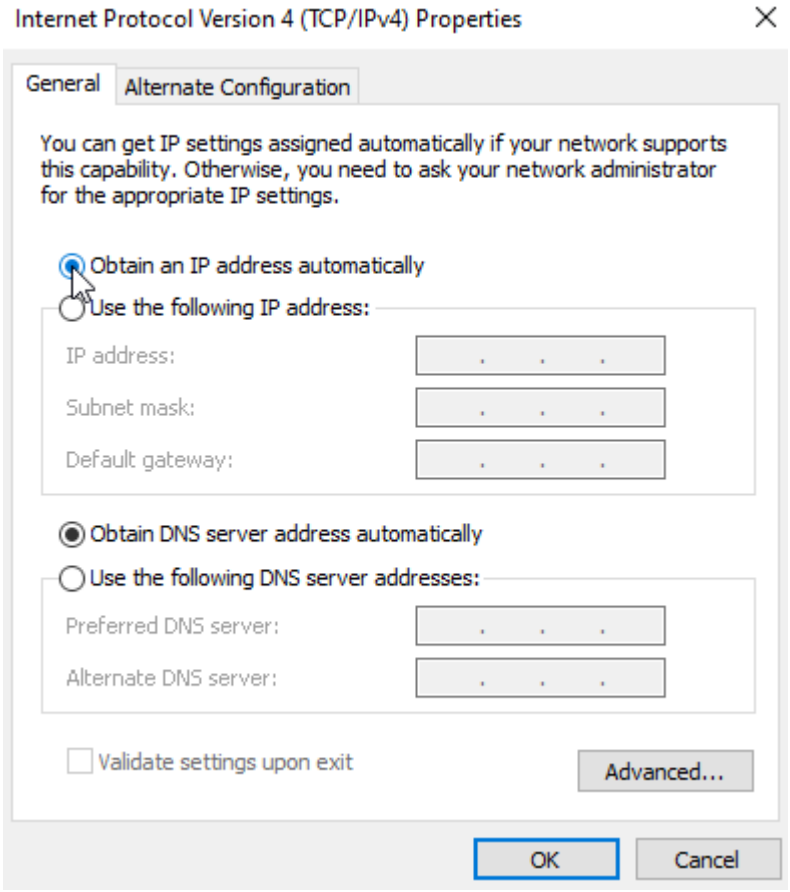
Настройка сетевого адаптера на работу по DHCP

Убедитесь, что ваш сетевой адаптер настроен на получение IP-адреса по DHCP и что DHCP-сервер выдал IP-адрес вашей машине.


Recycle Bin
All
Apps
Documents
Settings
Photos
More
Best match
ncpa.cpl
Control panel item
ncpa.cpl
Control panel item
Location
C:\Windows\system32
Last modified
Open
Run as administrator
Open file location
Copy full path
Windows 10 Enterprise LTSC Evaluation
Windows License is expired
Build 19041.vb_release.191206-1406
ncpa.cpl
ENG
3:22 PM
3/4/2022

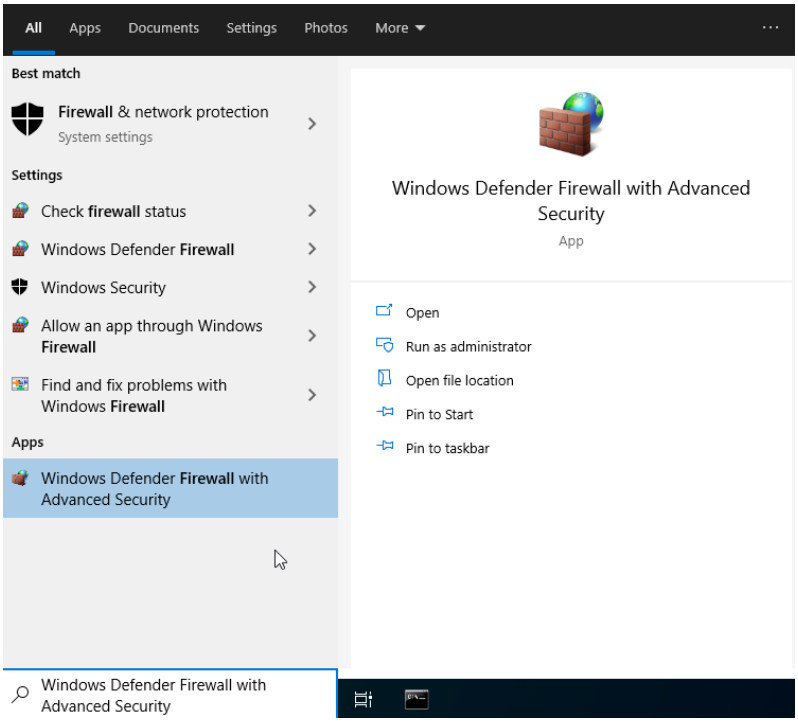

Network Connections
Network and Internet
Network Connections
Organize
Ethernet
Network 2
Intel(R) PRO/1000 MT Network C...
Ethernet
Network 2
Intel(R) PRO/
Disable
Status
Diagnose
Bridge Connections
Create Shortcut
Delete
Rename
Properties

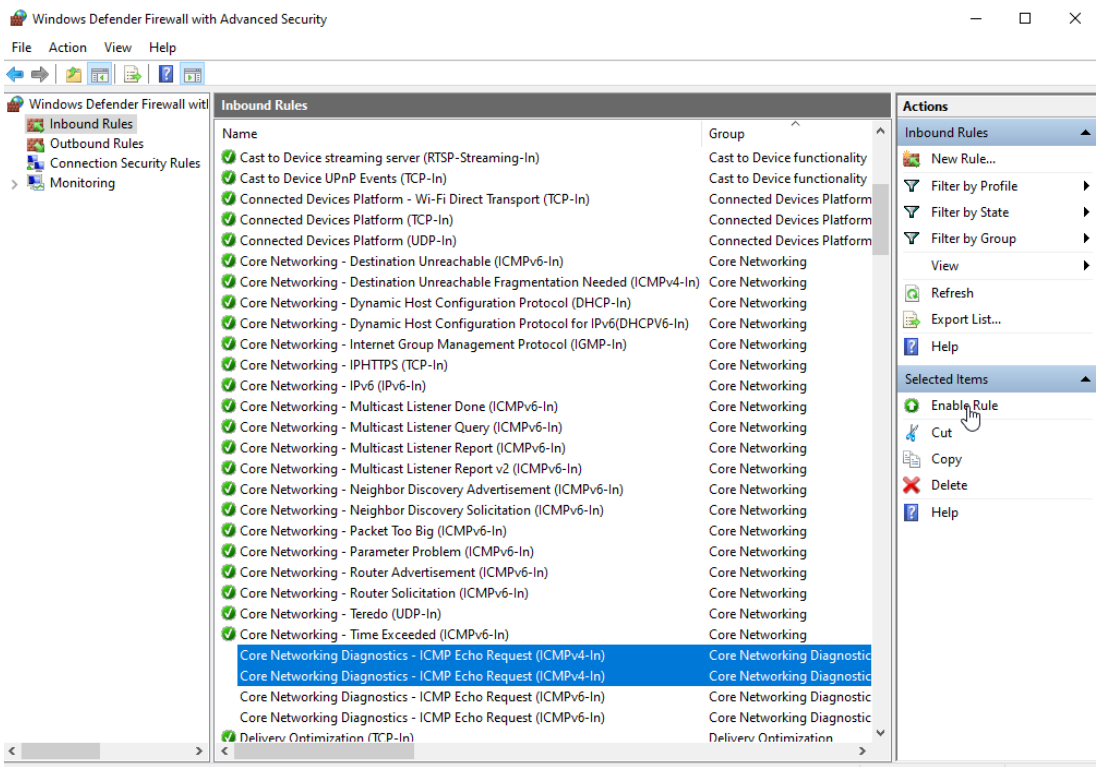

Ethernet Properties
Networking
Connect using:
Intel(R) PRO/1000 MT Network Connection
Configure...
This connection uses the following items:
Client for Microsoft Networks
File and Printer Sharing for Microsoft Networks
QoS Packet Scheduler
Internet Protocol Version 4 (TCP/IPv4)
Microsoft Network Adapter Multiplexor Protocol
Microsoft LLDP Protocol Driver
Internet Protocol Version 6 (TCP/IPv6)
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
OK
Cancel


Internet Protocol Version 4 (TCP/IPv4) Properties
General
Alternate Configuration
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address:
Subnet mask:
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel

Разрешение ICMP-трафика в Брандмауэре Windows

Для того, чтобы ping работал в Windows 10, необходимо настроить определенные правила в Брандмауэре Windows, которые разрешают входящий ICMP-трафик.





Проверка сетевых интерфейсов и доступа к Интернету

Выполнить команду ping можно в командной строке Windows (cmd):

C:\Users\Win10>ipconfig
C:\Users\Win10>ping (Введите IP-адрес, который вам выдал DHCP-сервер)
C:\Users\Win10>ping 192.168.1.1
C:\Users\Win10>ping 172.16.1.1

C:\Users\Win10>ping 172.16.1.254
C:\Users\Win10>ping ya.ru

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №13

Тема: Развертывание сервера DHCP на основе isc-dhcp-server

Цель работы: Сконфигурировать виртуальную машину в качестве сервера DHCP на основе isc-dhcp-server:

– Изменение имени хоста (Hostname);

– Настройка сетевых интерфейсов и установка статических IP-адресов ;

– Изменение DNS-сервера по умолчанию;

– Проверка сетевых интерфейсов и доступа к Интернету;

– Обновление локального индекса пакетов;

– Установка пакета с программой isc-dhcp-server и проверка его на работоспособность;

– Настройка isc-dhcp-server как DHCP-сервера;

– Логирование DHCP-сервера в отдельный файл;

– Настройка сетевого адаптера на работу по DHCP;

– Разрешение ICMP-трафика в Брандмауэре Windows;

– Проверка сетевых интерфейсов и доступа к Интернету.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Обратите внимание, что для выполнения этого задания необходимо выполнить следующие задания «Обзор схемы стенда сети предприятия» и «02.2 Настройка шлюза сети предприятия». Также нужно выполнить часть задания, относящегося к «03.1 Развертывание сервера DHCP»: изменение имени хоста (Hostname), настройка сетевых интерфейсов и установка статических IP-адресов, изменение DNS-сервера по умолчанию и проверка сетевых интерфейсов и доступа к Интернету.

Для выполнения этого задания и исключения конфликтов между программами, необходимо предварительно удалить предыдущий DHCP-сервер «dnsmasq». Сделать это можно следующим образом, выполнив команду:

root@localhost:~# apt purge dnsmasq

Программа dhcpd (аббревиатура от «DHCP-демон») – это программа DHCP-сервера, которая работает как демон на сервере для предоставления в сети службы протокола динамической конфигурации хоста (DHCP). Эта реализация, также известная

как ISC DHCP, является одной из первых и наиболее известных, но в настоящее время существует ряд других доступных реализаций программного обеспечения DHCP-сервера.

> Обратите внимание, что ISC объявила об окончании срока службы ISC DHCP в конце 2022 года. ISC продолжит предоставлять услуги профессиональной поддержки существующим подписчикам, но не намерена выпускать какие-либо дополнительные версии обслуживания.

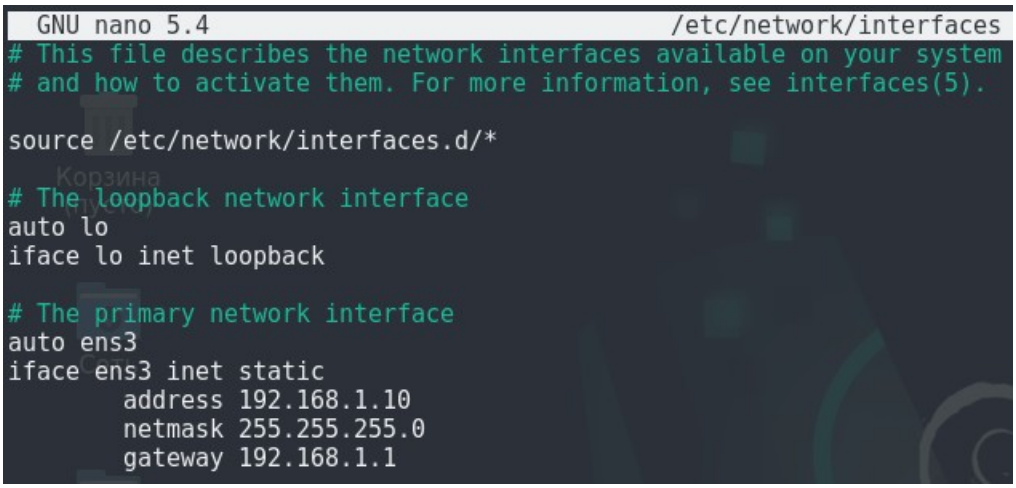
Входим на виртуальную машину Debian 11 – Server.

Изменение имени хоста (Hostname)

guest@localhost:~$ su -
Пароль: guest
root@localhost:~# hostnamectl set-hostname debian-11-server

Настройка сетевых интерфейсов и установка статических IP-адресов

root@debian-11-server:~# ip address
root@debian-11-server:~# nano /etc/network/interfaces



root@debian-11-server:~# systemctl disable avahi-daemon.service
root@debian-11-server:~# systemctl restart networking.service
root@debian-11-server:~# systemctl reboot

Также можно выключить службу connman, чтобы настройка статических IP-адресов применялась сразу после перезагрузки операционной системы:

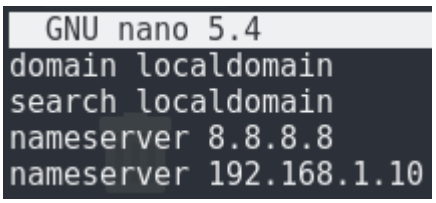
root@debian-11-server:~# systemctl disable connman.service
root@debian-11-server:~# systemctl stop connman.service

Если после перезагрузки виртуальной машины конфигурация не применилась, то снова перезагрузите службу networking.service с помощью команды:

root@debian-11-server:~# systemctl restart networking.service

Изменение DNS-сервера по умолчанию

root@debian-11-server:~# nano /etc/resolv.conf



Проверка сетевых интерфейсов и доступа к Интернету

root@debian-11-server:~# ip address
root@debian-11-server:~# ping 192.168.1.10
[ctrl+c]
root@debian-11-server:~# ping 192.168.1.1
[ctrl+c]
root@debian-11-server:~# ping 172.16.1.1
[ctrl+c]
root@debian-11-server:~# ping 172.16.1.254
[ctrl+c]
root@debian-11-server:~# ping ya.ru
[ctrl+c]

Обновление локального индекса пакетов

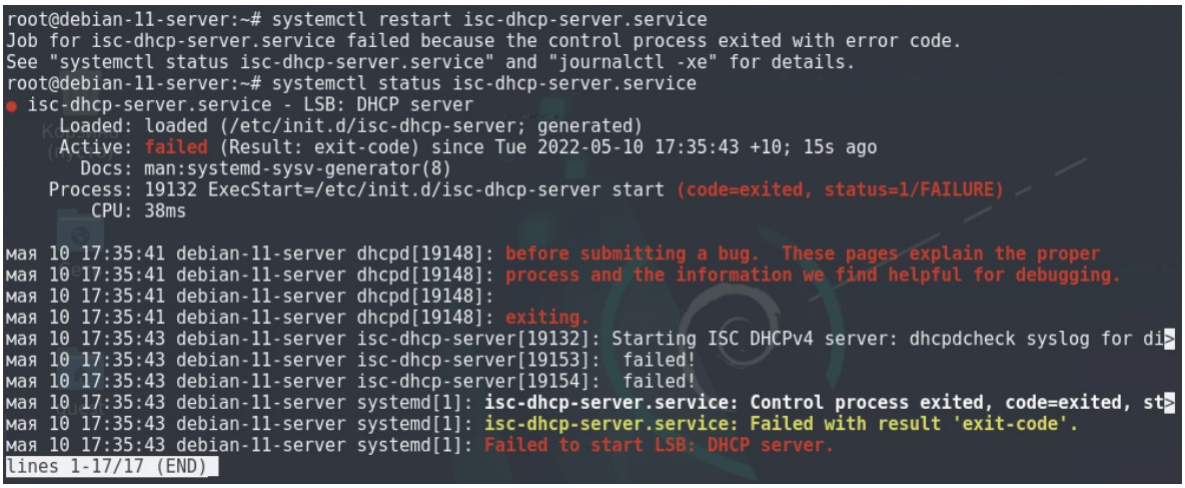
guest@debian-11-server:~$ su –
Пароль: guest
root@debian-11-server:~# apt update

Установка пакета с программой isc-dhcp-server и проверка его на работоспособность

root@debian-11-server:~# apt install isc-dhcp-server
root@debian-11-server:~# systemctl status isc-dhcp-server.service

Настройка isc-dhcp-server как DHCP-сервера

Сразу после установки, DHCP-сервер не сможет запуститься. Для запуска сервера, нужно произвести некоторые настройки.



Сначала нужно зайти в файл /etc/default/isc-dhcp-server, и внести в него некоторые изменения. Раскомментируем две строчки:

root@debian-11-server:~# nano -c /etc/default/isc-dhcp-server

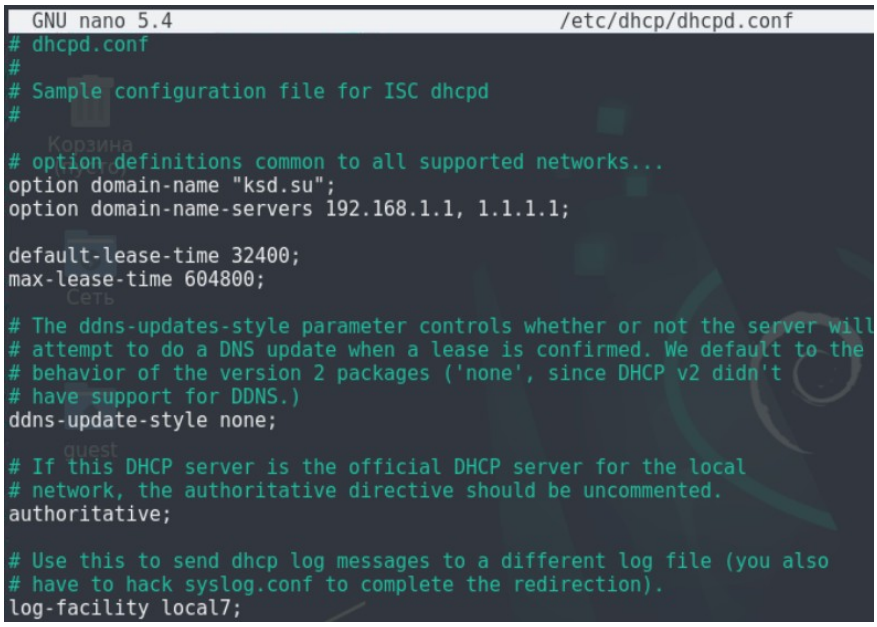
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf (Строка 4)
…
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf (Строка 8)

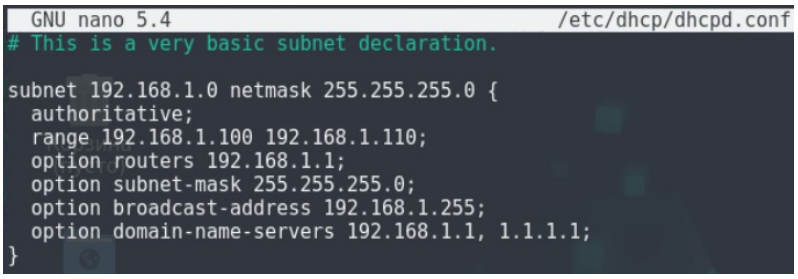
Далее находим строчку INTERFACESv4="" и внутри кавычек, прописываем название сетевого интерфейса, на котором будут обслуживаться запросы DHCP:

INTERFACESv4="ens3" (Строка 17)

После произведенных действий, нужно произвести базовую настройку DHCP-сервера. Находим файл /etc/dhcp/dhcpd.conf, и вносим конфигурацию:

root@debian-11-server:~# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.old (Сохранение конфигурационного файла по умолчанию, чтобы в случае ошибок откатиться обратно к нему)
root@debian-11-server:~# nano -c /etc/dhcp/dhcpd.conf





Расшифровка строчек конфигурационного файла:

– option domain-name – доменное имя;

– option domain-name-servers – DNS-сервер;

– default-lease-time – время по умолчанию (в секундах) аренды ip-адреса для клиентов;

– max-lease-time – максимальное время (в секундах) аренды ip-адреса для клиентов;

– log-facility local7 – логирование в отдельный файл;

– subnet, netmask – подсеть, и маска подсети;

– authoritative – назначение главным DHCP-сервером в этой сети;

– range – диапазон IP-адресов для раздачи клиентам;

– option routers – основной шлюз по умолчанию;

– option subnet-mask – маска подсети;

– option broadcast-address – широковещательный IP-адрес.

Неправильная конфигурация isc-dhcp-server не позволит запустить службу. Вы можете проверить ее в любой момент, запустив dhcpd -t. Если вы не увидели никаких предупреждений, значит, проверка синтаксиса пройдена, и служба сможет запуститься.

root@debian-11-server:~# dhcpd -t

```
root@debian-11-server:~# dhcpd -t
Internet Systems Consortium DHCP Server 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
root@debian-11-server:~#
```

Когда вы закончите настройку isc-dhcp-server, перезапустите службу DHCP-сервера, чтобы убедиться, что он работает и добавьте его в автозапуск:

root@debian-11-server:~# systemctl enable isc-dhcp-server
root@debian-11-server:~# systemctl restart isc-dhcp-server

Если с запуском сервера возникнут проблемы, то возможно, при установке был запущен PID-файл, и он мешает старту вашего DHCP-сервера. Чтобы исправить эту проблему, нужно принудительно удалить /var/run/dhcpd.pid.

Логирование DHCP-сервера в отдельный файл

Для настройки логирования в отдельный файл, нужно произвести настройку rsyslog. Найдите файл /etc/rsyslog.conf, и в конце добавьте строчку:

root@debian-11-server:~# nano -c /etc/rsyslog.conf

```
local7.* /var/log/dhcpd.log (Строка 94)
```

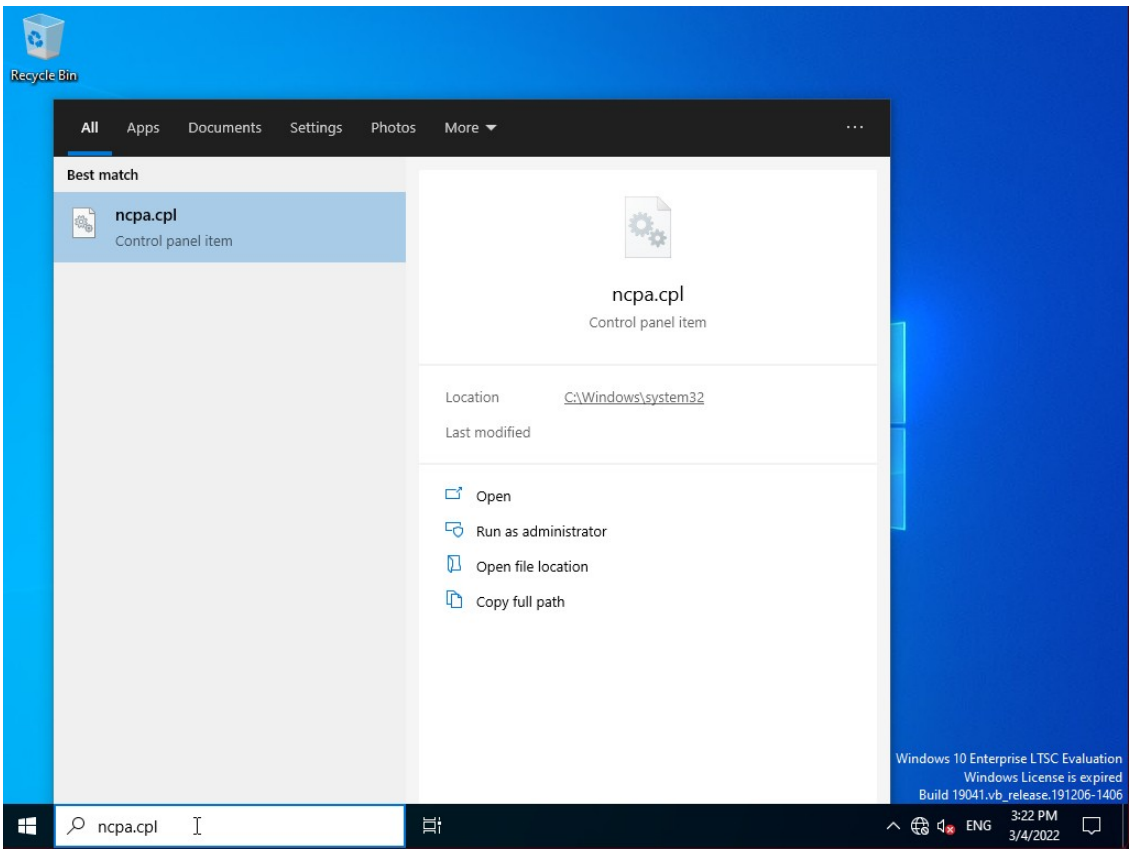
root@debian-11-server:~# systemctl restart rsyslog

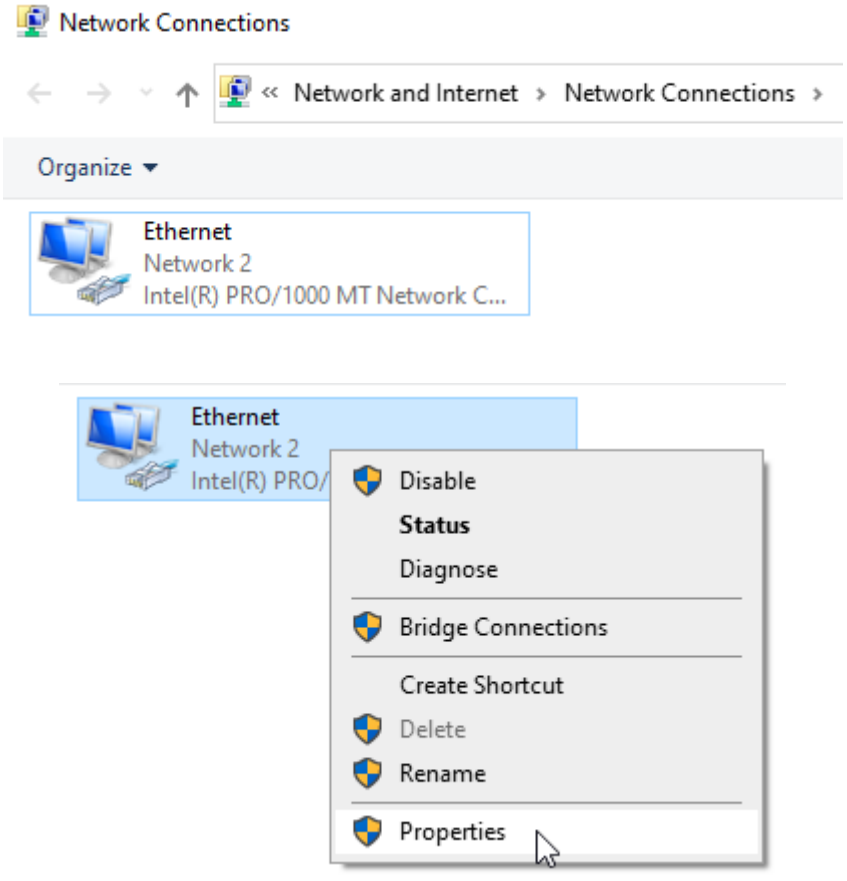
Теперь логирование DHCP-сервера будет производится в файл /var/log/dhcpd.log. Проверьте этот файл после того, как вы настроите Windows 10 клиент на получение IP-адреса по DHCP.

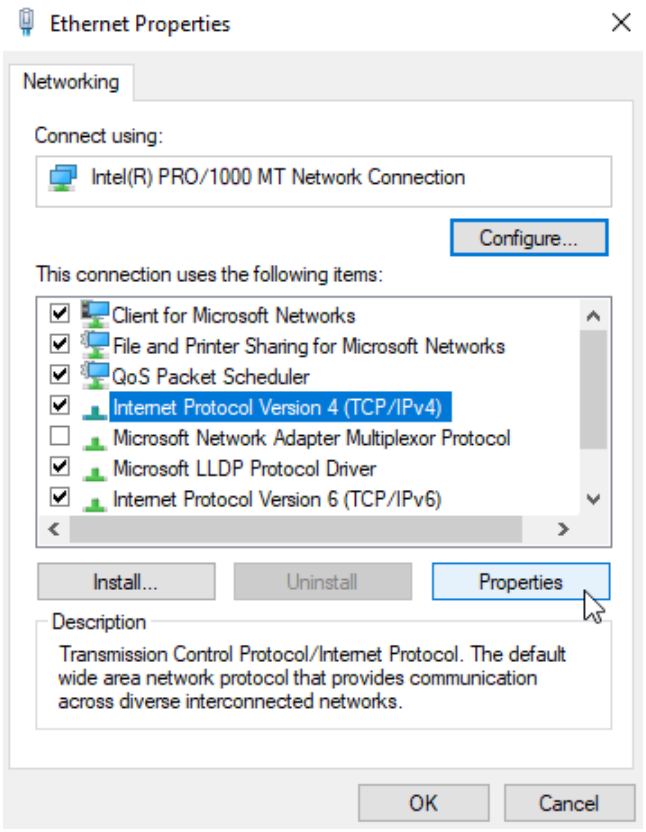
Входим на виртуальную машину Windows 10 – Host.

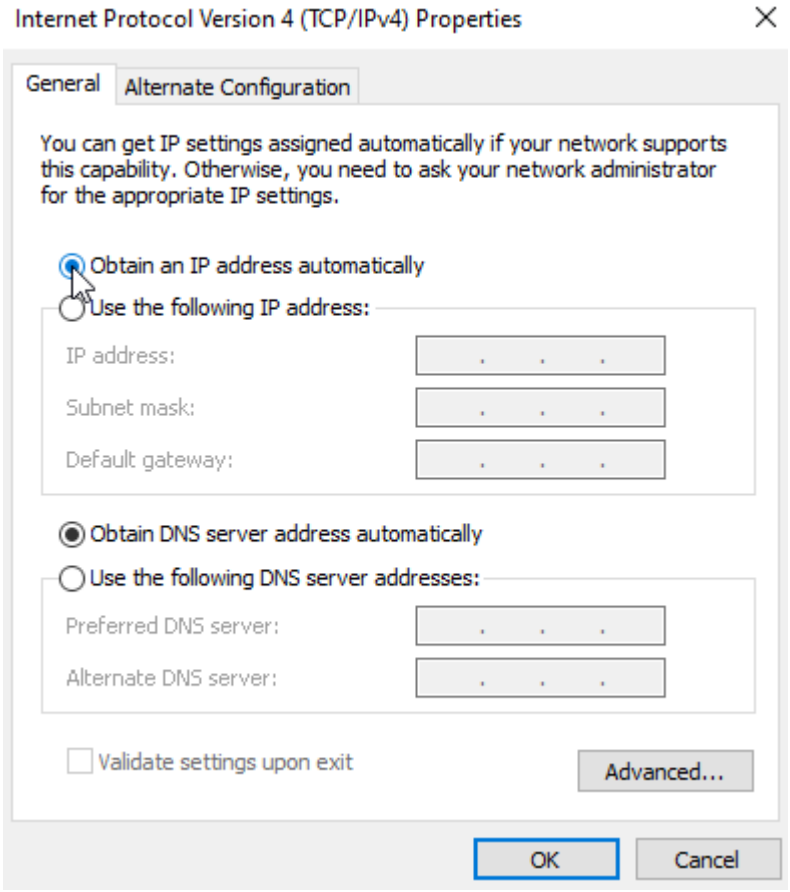
Настройка сетевого адаптера на работу по DHCP

Убедитесь, что ваш сетевой адаптер настроен на получение IP-адреса по DHCP и что DHCP-сервер выдал IP-адрес вашей машине.


Recycle Bin
All
Apps
Documents
Settings
Photos
More
Best match
ncpa.cpl
Control panel item
ncpa.cpl
Control panel item
Location
C:\Windows\system32
Last modified
Open
Run as administrator
Open file location
Copy full path
Windows 10 Enterprise LTSC Evaluation
Windows License is expired
Build 19041.vb_release.191206-1406
ncpa.cpl
ENG
3:22 PM
3/4/2022

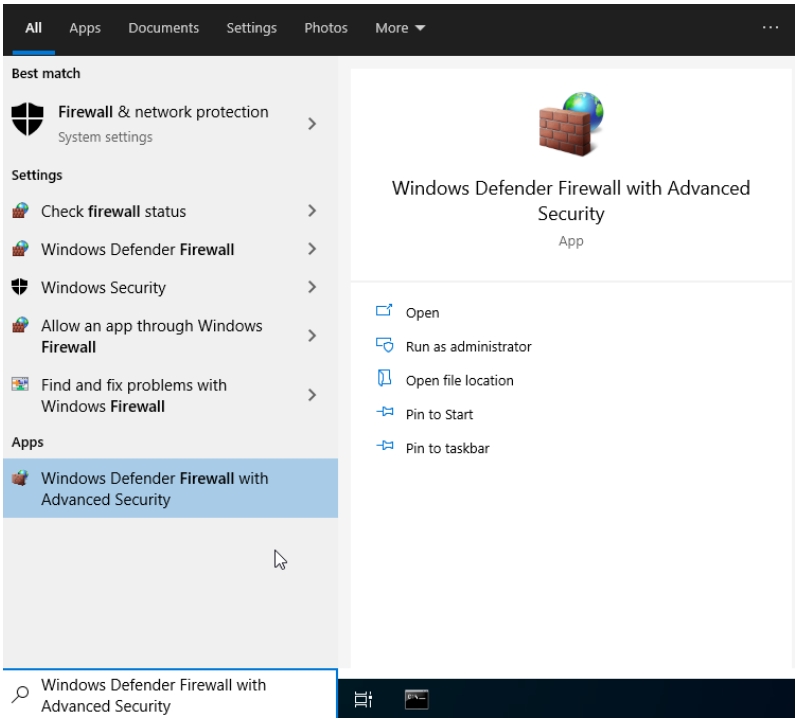

Network Connections
Network and Internet
Network Connections
Organize
Ethernet
Network 2
Intel(R) PRO/1000 MT Network C...
Ethernet
Network 2
Intel(R) PRO/
Disable
Status
Diagnose
Bridge Connections
Create Shortcut
Delete
Rename
Properties

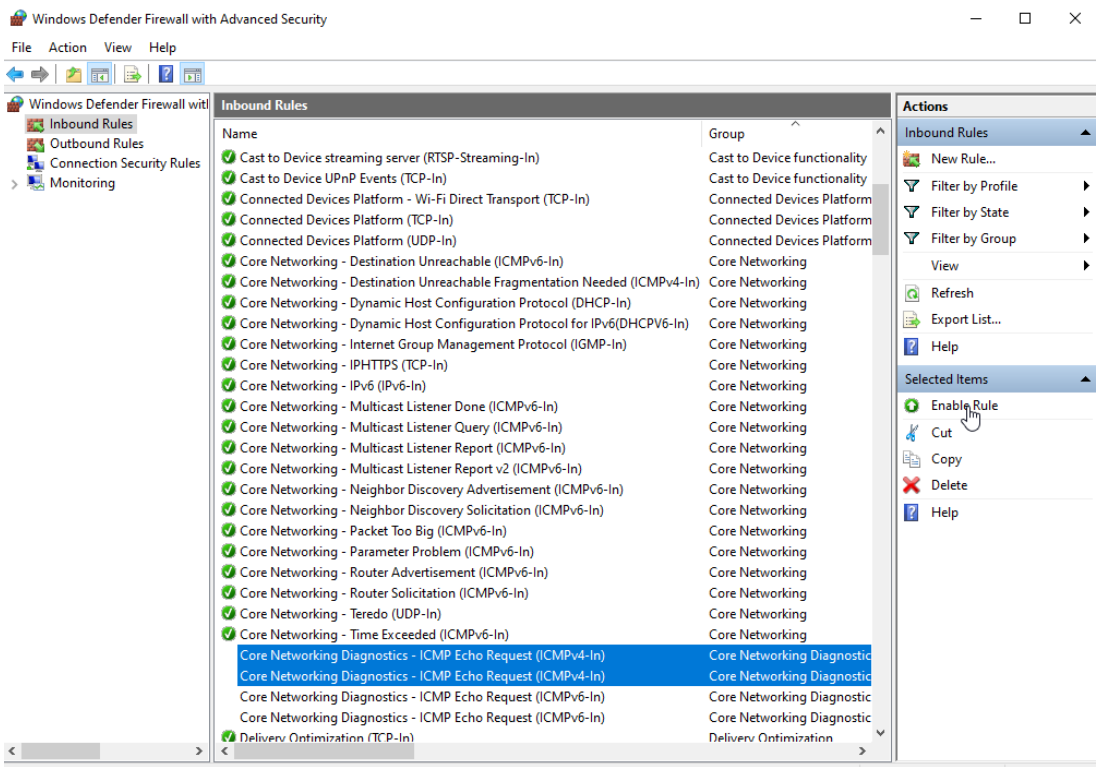

Ethernet Properties
Networking
Connect using:
Intel(R) PRO/1000 MT Network Connection
Configure...
This connection uses the following items:
Client for Microsoft Networks
File and Printer Sharing for Microsoft Networks
QoS Packet Scheduler
Internet Protocol Version 4 (TCP/IPv4)
Microsoft Network Adapter Multiplexor Protocol
Microsoft LLDP Protocol Driver
Internet Protocol Version 6 (TCP/IPv6)
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
OK
Cancel


Internet Protocol Version 4 (TCP/IPv4) Properties
General
Alternate Configuration
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address:
Subnet mask:
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel

Разрешение ICMP-трафика в Брандмауэре Windows

Для того, чтобы ping работал в Windows 10, необходимо настроить определенные правила в Брандмауэре Windows, которые разрешают входящий ICMP-трафик.





Проверка сетевых интерфейсов и доступа к Интернету

Выполнить команду ping можно в командной строке Windows (cmd):

C:\Users\Win10>ipconfig
C:\Users\Win10>ping (Введите IP-адрес, который вам выдал DHCP-сервер)
C:\Users\Win10>ping 192.168.1.1
C:\Users\Win10>ping 172.16.1.1
C:\Users\Win10>ping 172.16.1.254
C:\Users\Win10>ping ya.ru

Входим на виртуальную машину Debian 11 – Server.

Убедитесь, что информация об аренде IP-адреса записалась в файл /var/log/dhcpd.log:

root@debian-11-server:~# less /var/log/dhcpd.log

[ctrl+z]

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №14

Тема: Развертывание сервера DNS

Цель работы: Сконфигурировать виртуальную машину в качестве сервера DNS на основе dnsmasq:

– Настройка dnsmasq как DNS-сервера;

– Проверка совместной работы DNS- и DHCP-сервера;

– Изменение DNS-сервера по умолчанию;

– Проверка сетевых интерфейсов и доступа к Интернету.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

DNS (Domain Name System – система доменных имен) – это протокол прикладного уровня и компьютерная распределительная система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты и/или обслуживающих узлах для протоколов в домене (SRV-запись).

Распределенная база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определенному протоколу. DNS-сервер – это

программа, предназначенная для ответов на DNS-запросы по соответствующему протоколу. Также DNS-сервером могут называть хост, на котором запущено соответствующее приложение.

Основой DNS является представление об иерархической структуре имени и зонах. Каждый сервер, отвечающий за имя, может передать ответственность за дальнейшую часть домена другому серверу (с административной точки зрения – другой организации или человеку), что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

Входим на виртуальную машину Debian 11 – Server.

Настройка dnsmasq как DNS-сервера

Настройка конфигурационного файла DNSmasq выполняется следующим образом: root@debian-11-server:~# nano -c /etc/dnsmasq.conf

Укажите прослушиваемый порт DNS-сервера:

Port=53 (Строка 10)

Раскомментируйте следующую строку, чтобы DNS-сервер использовал только полное доменное имя (FQDN):

domain-needed (Строка 19)

Чтобы запретить пересылать адреса в немаршрутизированные адресные пространства, раскомментируйте следующую строку в файле конфигурации:

bogus-priv (Строка 21)

Запрет в чтении файла /etc/resolv.conf или другого файла, выполняющего его функцию:

no-resolv (Строка 58)

Если DNS-сервер не сможет ответить на запрос из своего кэша или зон, он запрашивает у других серверов ответ. Чтобы определить этот DNS-сервер верхнего уровня (корневого сервера), нужно раскомментировать следующую строку:

server=8.8.8.8 (Строка 66)

Указываем на DNS-сервер для нашего домена:

server=/ksd.su/192.168.1.10 (Строка 67)

Указываем IP-адрес для обратного запроса DNS (Reverse DNS Lookup). Для определения имени узла по его IPv4-адресу с помощью PTR-записи:

server=/ksd.su/1.168.192.in-addr.arpa/192.168.1.10 (Строка 71)

Раскомментируйте expand-hosts, чтобы добавить доменные имена в DNS-сервер из файла hosts:

```
expand-hosts (Строка 136)
```

Указываем IP-адрес для обратного запроса DNS (Reverse DNS Lookup). Для определения имени узла по его IPv4-адресу с помощью PTR-записи:

```
domain=ksd.su (Строка 145)
```

Настройка DHCP-опции списка IP-адресов серверов DNS по умолчанию:

```
dhcp-option=6,192.168.1.10 (Строка 333)
```

Настройка DHCP-опции автоматической установки NTP-сервера по умолчанию:

```
dhcp-option=option:ntp-server,192.168.1.10 (Строка 334)
```

Когда вы закончите настройку dnsmasq, проверьте его конфигурацию, чтобы убедиться, что ваши изменения действительны:

root@debian-11-server:~# dnsmasq --test

Чтобы добавить DNS-записи, необходимо добавить их в файл /etc/hosts. Затем dnsmasq будет отвечать на запросы клиентов.

root@debian-11-server:~# nano /etc/hosts

```
# add records
192.168.1.10  debian-11-server.ksd.su
192.168.1.5           debian-11-host.ksd.su
192.168.1.1           debian-11-gateway.ksd.su
172.16.1.1            debian-11-gateway.ksd.su
```

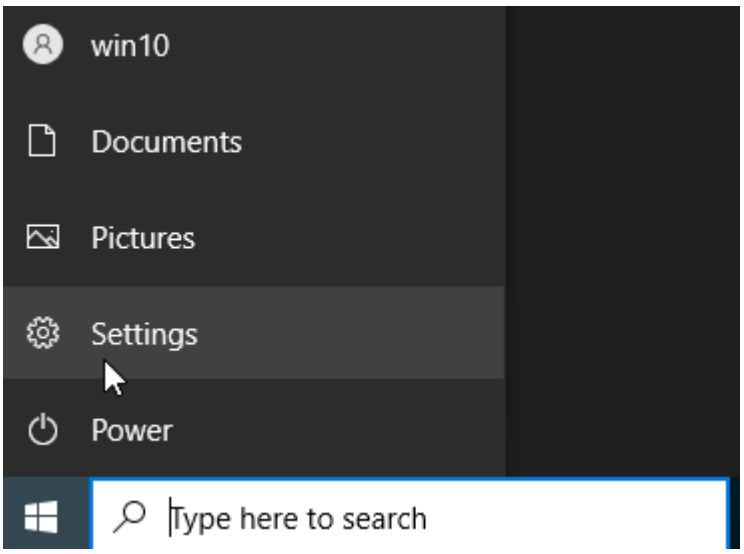
Перезагрузим службу DNSmasq:

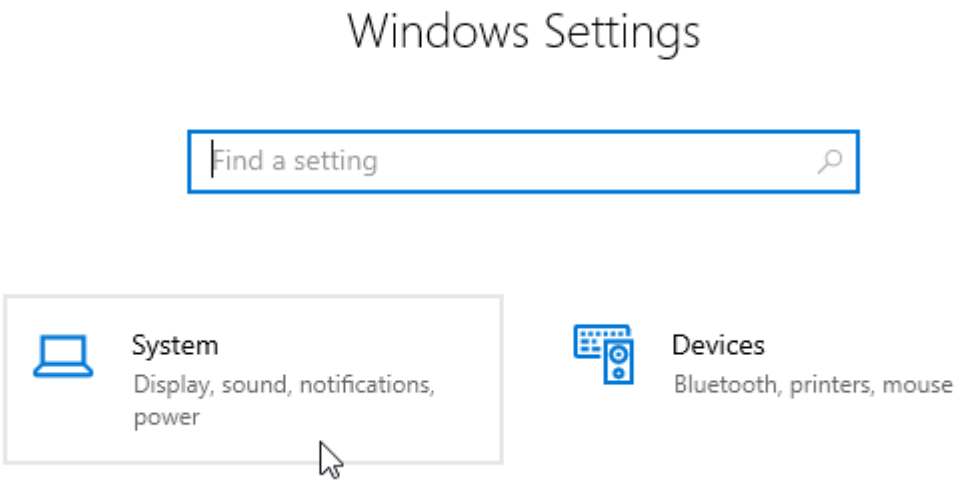
root@debian-11-server:~# systemctl restart dnsmasq

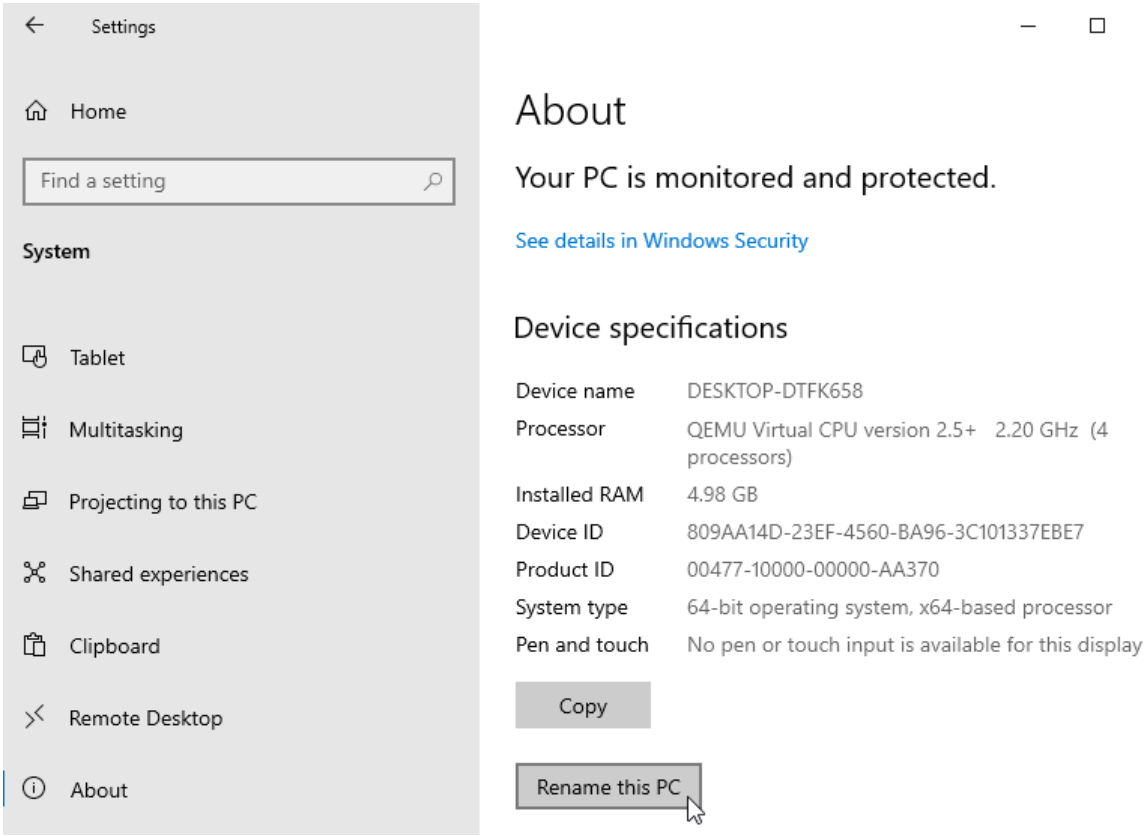
Проверка совместной работы DNS- и DHCP-сервера

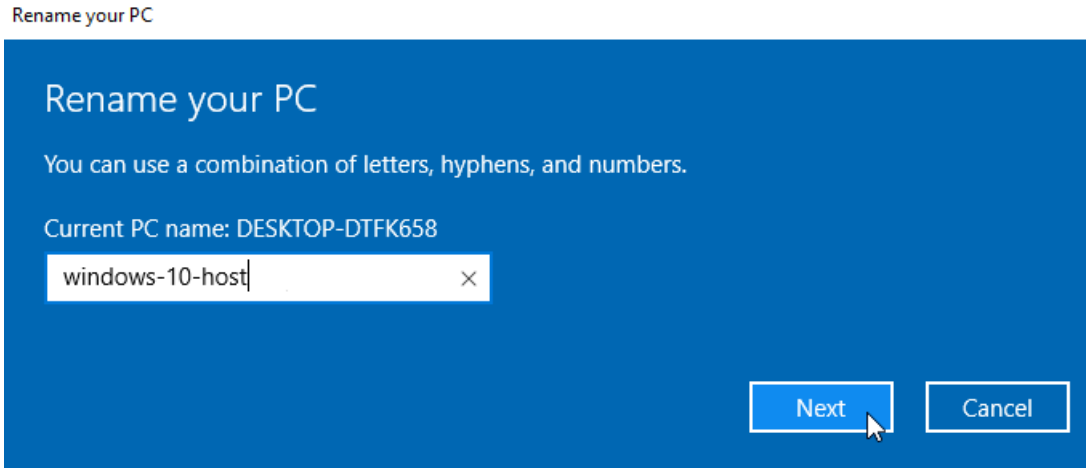
Входим на виртуальную машину Windows 10 – Host.

Переименуйте имя компьютера в Windows 10 и получите новый IP-адрес от DHCP-сервера.


win10
Documents
Pictures
Settings
Power
Type here to search


Windows Settings
Find a setting
System
Display, sound, notifications, power
Devices
Bluetooth, printers, mouse





Перезагрузите компьютер и, затем, введите следующий текст в командной строке Windows (cmd), чтобы получить новый IP-адрес:

```
C:\Users\Win10>ipconfig /release
C:\Users\Win10>ipconfig /renew
```

Убедитесь, что ваш сетевой адаптер настроен на получение IP-адреса по DHCP и что DHCP-сервер выдал IP-адрес вашей машине.

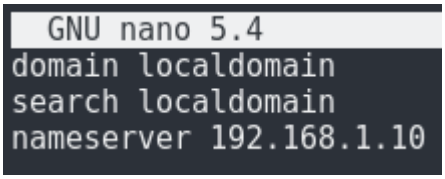
Выполнить команду ping можно в командной строке Windows (cmd):

```
C:\Users\Win10>ipconfig
C:\Users\Win10>ping debian-11-server.ksd.su
C:\Users\Win10>ping debian-11-gateway.ksd.su
C:\Users\Win10>ping ya.ru
```

Входим на виртуальную машину Debian 11 – Gateway.

Изменение DNS-сервера по умолчанию

root@debian-11-gateway:~# nano /etc/resolv.conf



root@debian-11-gateway:~# systemctl restart networking.service
root@debian-11-gateway:~# systemctl restart default-route.service

Проверка сетевых интерфейсов и доступа к Интернету

root@debian-11-server:~# ip address
root@debian-11-server:~# ping debian-11-server.ksd.su
[ctrl+c]
root@debian-11-server:~# ping windows-10-host.ksd.su
[ctrl+c]
root@debian-11-server:~# ping ya.ru
[ctrl+c]

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №15

Тема: Развертывание сервера DNS на основе bind9

Цель работы: Сконфигурировать виртуальную машину в качестве сервера DNS на основе bind9:

– Обновление локального индекса пакетов;

– Установка пакета с программой bind9 и проверка его на работоспособность;

– Генерация безопасного RNDC-ключа для обновлений;

– Настройка bind9 как главного DNS-сервера;

– Настройка файла параметров главного DNS-сервера;

– Настройка локального файла главного DNS-сервера;

– Проверка конфигурации bind9 главного DNS-сервера;

– Создание файла для зоны прямого просмотра главного DNS-сервера;

– Создание файла для зоны обратного просмотра главного DNS-сервера;

– Проверка зон bind9 главного DNS-сервера;

– Настройка isc-dhcp-server для совместной работы с динамическим DNS сервера bind9;

– Сброс кеша аренды DHCP-сервера;

– Изменение DNS-сервера по умолчанию ;

– Проверка работы главного DNS-сервера с помощью утилиты nslookup;

– Проверка совместной работы DNS- и DHCP-сервера и проверка выдачи зарезервированных IP-адресов;

– Динамическое обновление зон DHCP-сервером;

– Настройка AppArmor;

– Проверка выдачи незарезервированных IP-адресов;

– Настройка реплицирующего дополнительного DNS-сервера;

– Обновление локального индекса пакетов;

– Установка пакета с программой bind9 и проверка его на работоспособность;

– Изменение DNS-сервера по умолчанию ;

– Проверка работы дополнительного DNS-сервера с помощью программ nslookup и traceroute;

– Динамическое обновление зон DHCP-сервером;

– Настройка AppArmor;

– Проверка выдачи незарезервированных IP-адресов;

– Настройка реплицирующего дополнительного DNS-сервера;

– Обновление локального индекса пакетов;

– Установка пакета с программой bind9 и проверка его на работоспособность;

– Изменение DNS-сервера по умолчанию ;

– Проверка работы дополнительного DNS-сервера с помощью программ nslookup и traceroute.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Обратите внимание, что для выполнения этого задания необходимо выполнить следующие задания «Обзор схемы стенда сети предприятия», «Настройка шлюза сети предприятия» и «Развертывание сервера DHCP на основе isc-dhcp-server».

Для выполнения этого задания и исключения конфликтов между программами, необходимо предварительно удалить предыдущий DNS-сервер «dnsmasq». Сделать это можно следующим образом, выполнив команду:

root@localhost:~# apt purge dnsmasq

BIND (Berkeley Internet Name Domain, до этого назывался: Berkeley Internet Name Daemon) – это открытая и наиболее распространенная реализация DNS-сервера, обеспечивающая выполнение преобразования DNS-имени в IP-адрес и наоборот. Исполняемый файл-демон сервера BIND называется named. BIND поддерживается

организацией Internet Systems Consortium. В мире 10 из 13 корневых серверов DNS работают на BIND, оставшиеся 3 работают на NSD.

BIND был создан студентами в начале 1980-х на грант, выданный DARPA и впервые был выпущен в BSD 4.3. Версия 9 была переписана заново компанией Nominum, релиз был выпущен в сентябре 2000 года, а версия 10 содержит большое количество кода на Python.

BIND является де-факто стандартом для UNIX-подобных операционных систем, поэтому попробуем реализовать на нем достаточно непростую задачу: настройку главного и дополнительного DNS-сервера, работающего совместно с isc-dhcp-server, который будет реализовывать динамическую регистрацию и резервацию IP-адресов хостов.

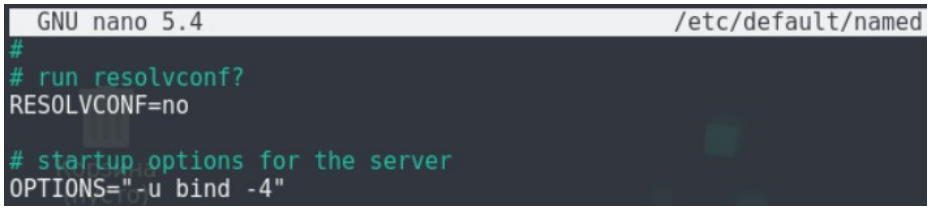
Входим на виртуальную машину Debian 11 – Server.

Обновление локального индекса пакетов

guest@debian-11-server:~$ su –
Пароль: guest
root@debian-11-server:~# apt update

Установка пакета с программой bind9 и проверка его на работоспособность

root@debian-11-server:~# apt install bind9 bind9utils bind9-doc
root@debian-11-server:~# systemctl status bind9 или systemctl status named.service
root@debian-11-server:~# nano /etc/default/named



Генерация безопасного RNDC-ключа для обновлений

RNDC[10] управляет работой сервера доменных имен. RNDC использует TCP-соединение для связи с сервером bind9 для отправки команд, аутентифицированных с помощью цифровых подписей. Чтобы обновления происходили безопасно, нам нужно сгенерировать случайный ключ для использования как bind9, так и isc-dhcp-server.

Затем скопируйте верхний раздел (закомментируйте все строки пункта «options») в новый файл с именем /etc/bind/rndc.conf. Вам нужно будет включить этот файл в конфигурацию DHCP и DNS.

Сделать оба этих действия можно одной командой:

root@debian-11-server:~# rndc-confgen > /etc/bind/rndc.conf

[10] rndc (Remote Name Daemon Control) – это утилита, позволяющая управлять дистанционно DNS-сервером bind9.

root@debian-11-server:~# nano -c /etc/bind/rndc.conf

```
key "rndc-key" {
        algorithm hmac-sha256;
        secret "H+XwdALIgPS/6A7Xt0N3GL4UNxEhKVtgBGM49q9wHG4=";
};

#options {
#       default-key "rndc-key";
#       default-server 127.0.0.1;
#       default-port 953;

controls {
        inet 127.0.0.1 port 953
                allow { 127.0.0.1; } keys { "rndc-key"; };
};
```

Теперь измените права доступа к файлу, чтобы его могли прочитать только root и bind, и создайте ссылку для DHCP:

root@debian-11-server:~# chmod 660 /etc/bind/rndc.conf
root@debian-11-server:~# chmod 660 /etc/bind/zones/ksd.wsr
root@debian-11-server:~# chmod 660 /etc/bind/zones/168.192.in-addr.arpa
root@debian-11-server:~# chown -R bind /etc/bind (Убедитесь, чтобы эта команда была выполнена. Без нее isc-dhcp-server не сможет вписывать динамические изменения в зоны DNS)
root@debian-11-server:~# chown root:bind /etc/bind/rndc.conf
root@debian-11-server:~# ln -s /etc/bind/rndc.conf /etc/dhcp/rndc.conf

Скопируйте файл rndc.conf в ту же папку под названием rndc.key:

root@debian-11-server:~# cp /etc/bind/rndc.conf /etc/bind/rndc.key
root@debian-11-server:~# nano -c /etc/bind/rndc.key

```
key "rndc-key" {
        algorithm hmac-sha256;
        secret "WFOF2zdwot8xRd7KcmxxqfQEmgEBEo4Y/FiYv49Tw7E=";
};
```

root@debian-11-server:~# chmod 660 /etc/bind/rndc.key
root@debian-11-server:~# chown root:bind /etc/bind/rndc.key
root@debian-11-server:~# ln -s /etc/bind/rndc.key /etc/dhcp/rndc.key

Настройка bind9 как главного DNS-сервера

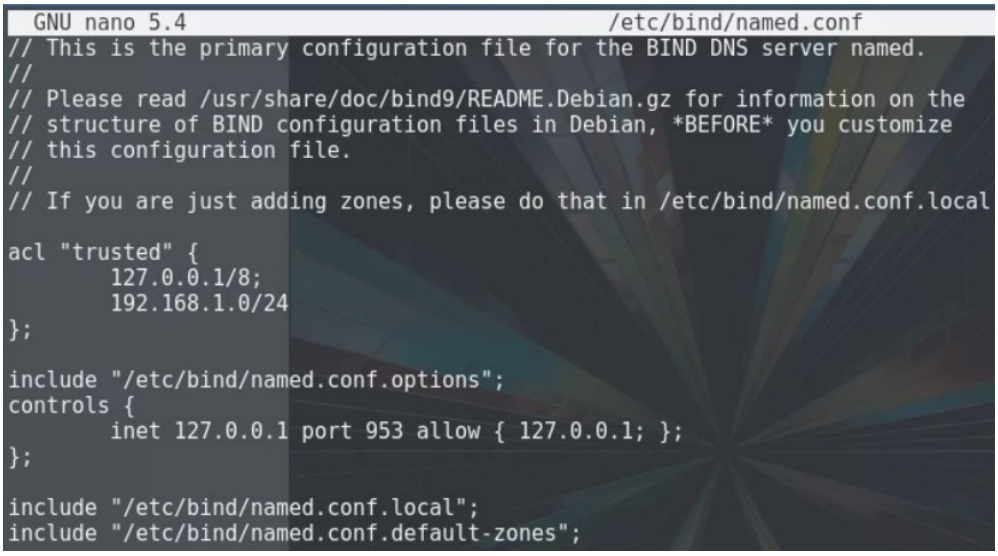
Конфигурация bind9 состоит из множества файлов, которые включены в основной файл конфигурации, named.conf. Эти имена файлов начинаются с named, потому что это имя процесса, который запускает bind9 (сокращение от «domain name daemon»). Мы начнем с настройки основного файла, а затем файла параметров.

Настройка основного файла конфигурации главного DNS-сервера

Создайте *новый* блок ACL (список контроля доступа) под названием «trusted». Именно здесь мы добавим подсеть с клиентскими компьютерами, для которых будем разрешать рекурсивные DNS-запросы (т.е. запросы от серверов, находящихся в том же

локальной сети, что и Debian 11 – Server). На сервере Debian 11 – Server откройте файл named.conf для редактирования:

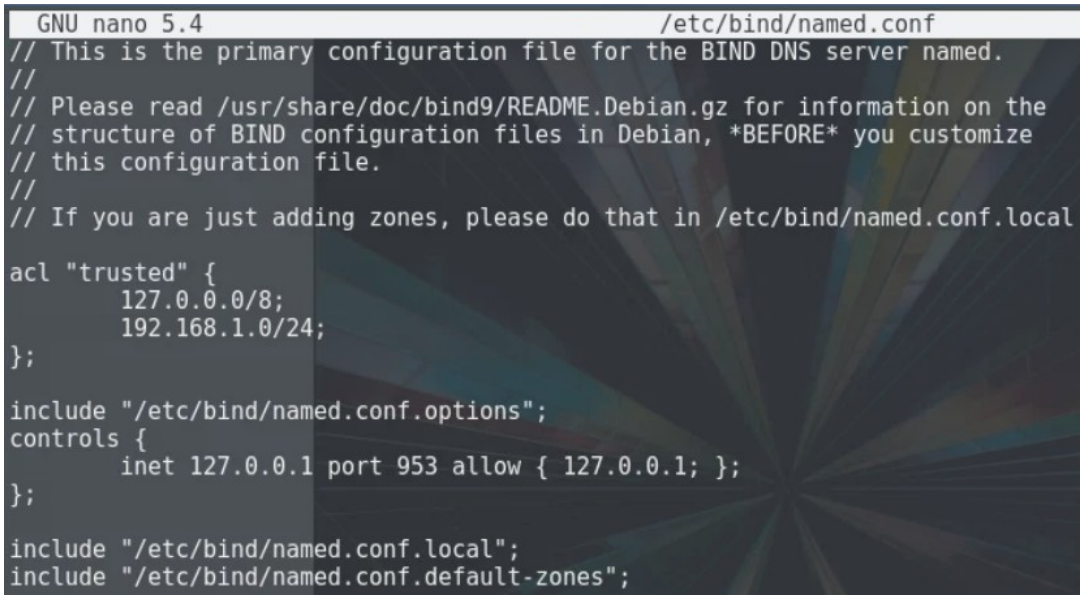
root@debian-11-server:~# cp /etc/bind/named.conf /etc/bind/named.conf.old
root@debian-11-server:~# nano -c /etc/bind/named.conf



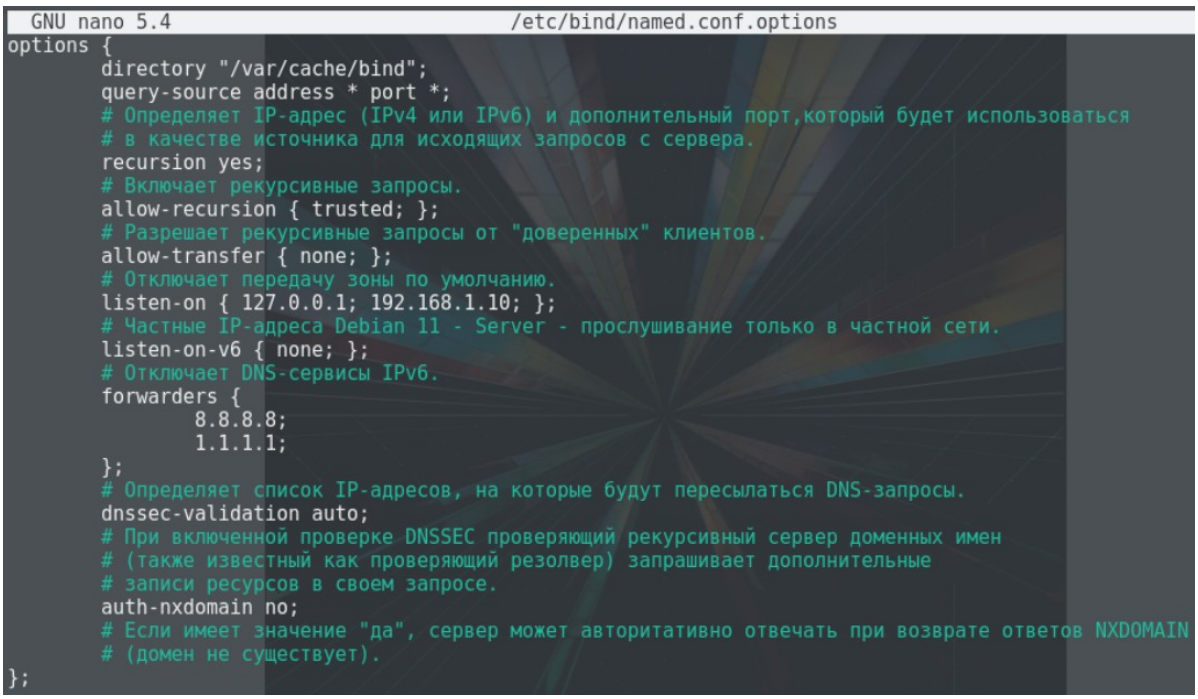
Настройка файла параметров главного DNS-сервера

Теперь, когда мы добавили подсеть с доверенными клиентскими компьютерами в DNS, нам нужно отредактировать блок options. В данный момент начало блока выглядит следующим образом:

root@debian-11-server:~# cp /etc/bind/named.conf.options /etc/bind/named.conf.options.old
root@debian-11-server:~# nano -c /etc/bind/named.conf.options



Под директивой directory добавьте строки конфигурации, изображенные на рисунке ниже , чтобы результат выглядел примерно следующим образом.



После завершения редактирования сохраните и закройте файл named.conf.options. Согласно конфигурация выше, только ваши собственные серверы (т.е. доверенные) смогут запрашивать у вашего DNS-сервера внешние домены.

Далее мы настроим локальный файл, чтобы задать ваши DNS-зоны.

Настройка локального файла главного DNS-сервера

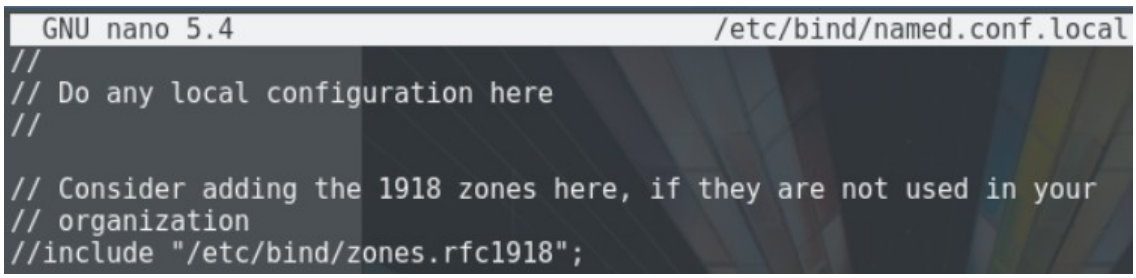
Локальный файл должен содержать только несколько комментариев. Здесь мы зададим наши зоны. DNS-зоны определяют конкретную область для управления и определения записей DNS.

В качестве разнообразия в этой теме вместо домена «ksd.su», мы выберем и будем настраивать домен «ksd.wsr».

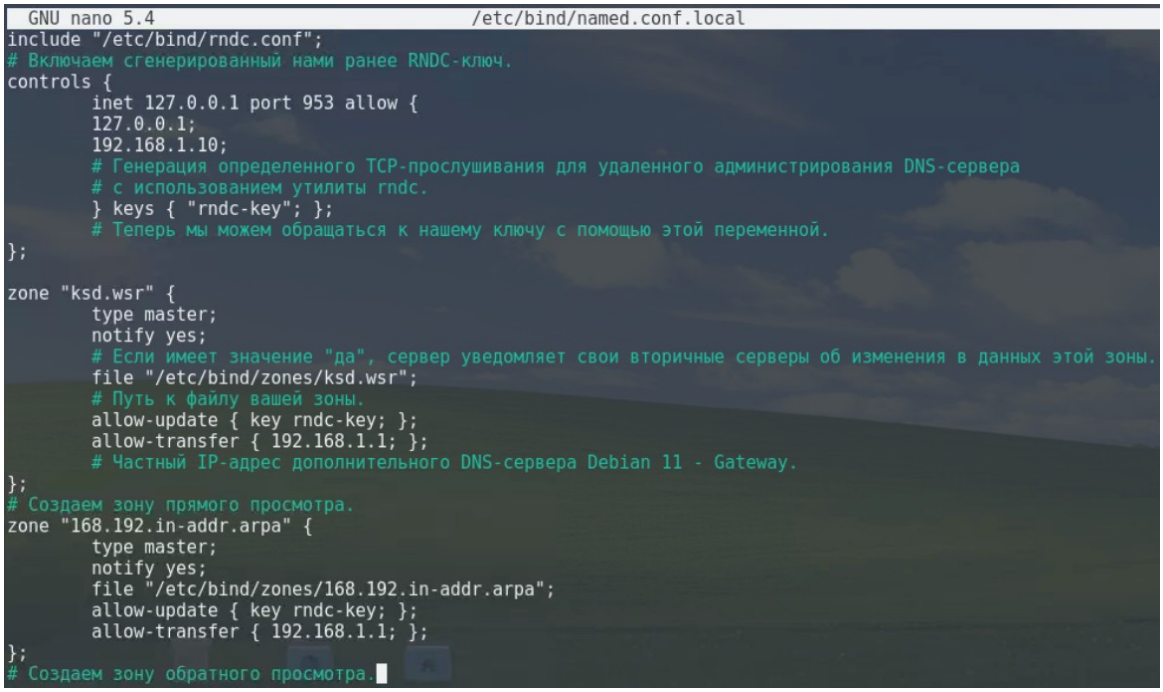
Поскольку наши домены будут находиться в домене «ksd.wsr», мы будем использовать его в качестве зоны прямого просмотра. Поскольку частные IP-адреса нашего сервера находятся в пространстве IP-адресов 192.168.1.0/24, мы создадим зону обратного просмотра, чтобы мы могли определять обратный просмотр в этом диапазоне.

На сервере Debian 11 – Server откройте файл named.conf.local для редактирования. В данный момент локальный конфигурационный файл выглядит следующим образом:

root@debian-11-server:~# cp /etc/bind/named.conf.local /etc/bind/named.conf.local.old
root@debian-11-server:~# nano -c /etc/bind/named.conf.local



Добавьте зону прямого просмотра и закрытый IP-адрес дополнительного DNS-сервера в директиве allow-transfer согласно изображению ниже. Затем добавьте зону обратного просмотра с помощью следующих строк (обратите внимание, что имя зоны обратного просмотра начинается с «168.192», что представляет собой битное преобразование «192.168»).



Если ваши серверы охватывают несколько частных подсетей, но находятся в одной общей локальной вычислительной сети, обязательно указывайте дополнительную зону и файл зоны для каждой отдельной подсети.

После добавления всех необходимых зон сохраните и закройте файл named.conf.local.

Проверка конфигурации bind9 главного DNS-сервера

Чтобы проверить конфигурацию, выполните следующую команду:

root@debian-11-server:~# named-checkconf

Если вывод консоли ничего не пишет, то это значит, что конфигурация в порядке и мы можем двигаться дальше. Если нет, вам нужно погуглить вывод программы и проверить ваши конфигурационные файлы на опечатки. После каждого изменения файлов конфигурации повторно запускайте проверку, пока не убедитесь, что ошибок больше нет

и сервер bind9 точно запустится. Если вы получаете предупреждения об отсутствующих файлах зон, вы можете их пока спокойно игнорировать – мы создадим их уже дальше.

Создание файла для зоны прямого просмотра главного DNS-сервера

Теперь, когда наши зоны указаны и конфигурационные файлы проверены в bind9, нам нужно создать соответствующие файлы для зоны прямого и обратного просмотра.

Файл зоны прямого просмотра – это место, где мы будем определять DNS-записи для прямого просмотра DNS, т.е., когда DNS получает запрос имени, например, «debian-11-server.ksd.wsr», будет выполняться поиск в файле зоны прямого просмотра для получения, соответствующего частного IP-адреса для Debian 11 – Server.

Давайте создадим директорию, в которой будут находиться наши файлы зоны. Согласно конфигурации named.conf.local, это должна быть директория /etc/bind/zones:

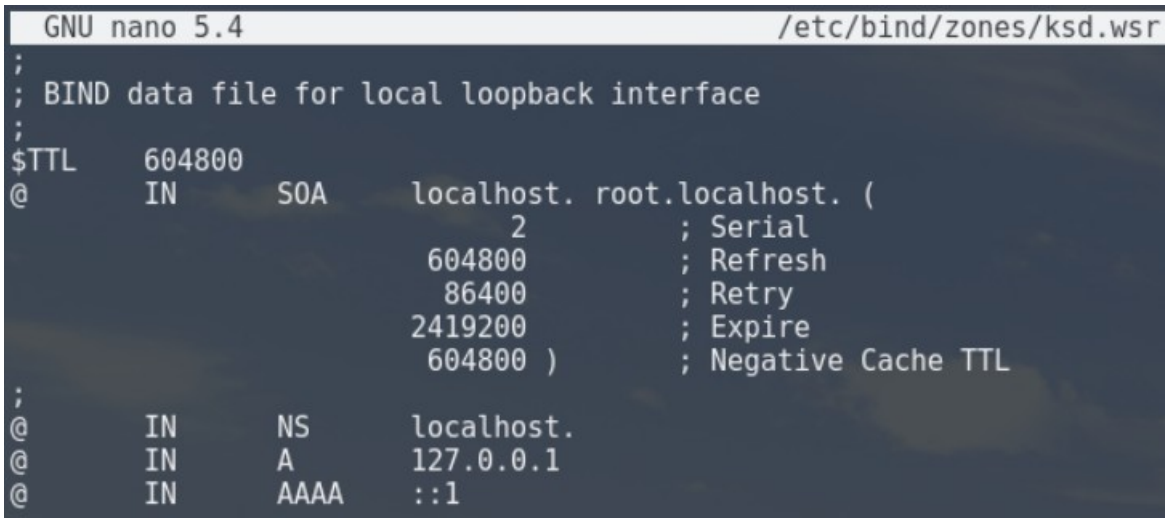
root@debian-11-server:~# mkdir /etc/bind/zones

При создании нашего файла зоны для прямого просмотра мы будем опираться в качестве примера на файл зоны db.local. Скопируйте его в надлежащее место с помощью следующих команд:

root@debian-11-server:~# cp /etc/bind/db.local /etc/bind/zones/ksd.wsr

Теперь необходимо отредактировать наш файл зоны для прямого просмотра. Первоначально он будет выглядеть примерно следующим образом:

root@debian-11-server:~# nano /etc/bind/zones/ksd.wsr



Во-первых, вам стоит отредактировать запись SOA[11]. Замените первую запись localhost на полное доменное имя (FQDN) debian-11-server, а затем замените root.localhost на debian-11-server.ksd.wsr. При каждом изменении файла зоны вам нужно будет

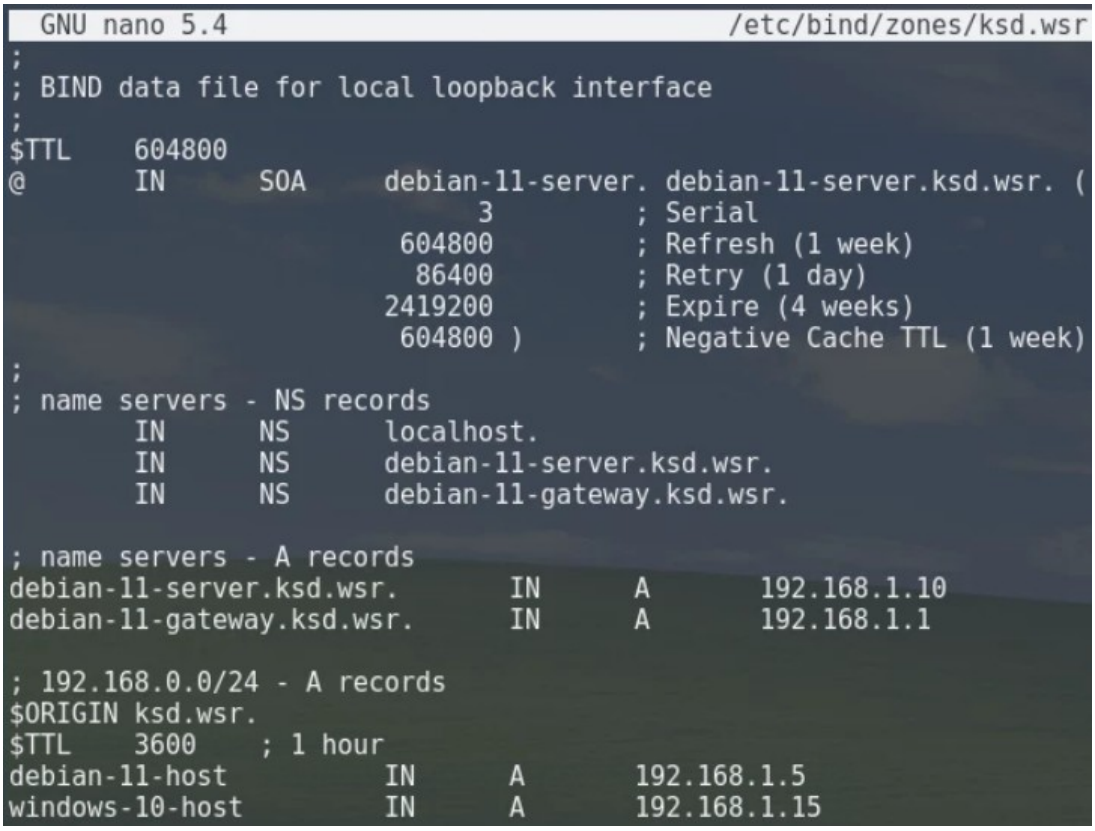
[11] SOA-запись (Start of Authority) – это начальная запись зоны, которая указывает местоположение эталонной записи о домене. Она содержит в себе контактную информацию лица, ответственного за данную зону, время кэширования информации на серверах и данные о взаимодействии DNS. SOA-запись создается автоматически. Удалять ее нельзя.

увеличивать значение serial, прежде чем перезапускать процесс named. Мы увеличим значение до 3. Это необходимо для отслеживания изменений другими DNS-серверами.

Далее удалите три записи в конце файла (после записи SOA). Вместо них добавьте в конце файла записи для серверов имен нашей локальной сети (Debian 11 – Server и Debian 11 – Gateway). Обратите внимание, что во втором столбце указывается, что это записи NS.

Потом добавьте записи A для ваших хостов, которые принадлежат к этой зоне. Это может быть любой сервер, имя которого будет заканчиваться на «.ksd.wsr.» (замените имена и частные IP-адреса).

Полученный нами в итоге пример файла зоны для прямого просмотра должен выглядеть следующим образом.



Мы можем добавлять в этот конфигурационный файл и другие типы записей, такие как CNAME, TXT или даже записи AAAA (только для IPv6). Это также позволит нам реорганизовывать нашу сеть так, как мы того хотим.

Давайте определим, что означают некоторые параметры в нашем конфигурационном файле:

– IN (Internet) – это класс записи (для Интернета), предназначенный для общих записей DNS, включающих Интернет имена хостов, серверов или IP-адресов;

– Директива $ORIGIN – это стандартная директива DNS, которая определяет базовое имя, из которого производится замена «неполных» имен (без завершающей

точки) при обработке файла зоны. Файлы зон, которые не содержат директивы $ORIGIN, хотя и являются вполне допустимыми, также могут сильно сбивать с толку. В общем, всегда явно определяйте директиву $ORIGIN, если нет очень веской причины не делать этого (лень не является такой причиной);

– Директива $TTL – это директива, которая в контексте DNS определяет продолжительность в секундах, в течение которой запись может кэшироваться любым преобразователем. Нуль указывает, что запись не должна кэшироваться.

Убедитесь, что ввели конфигурацию прямой зоны верно, сохраните и закройте файл.

Создание файла для зоны обратного просмотра главного DNS-сервера

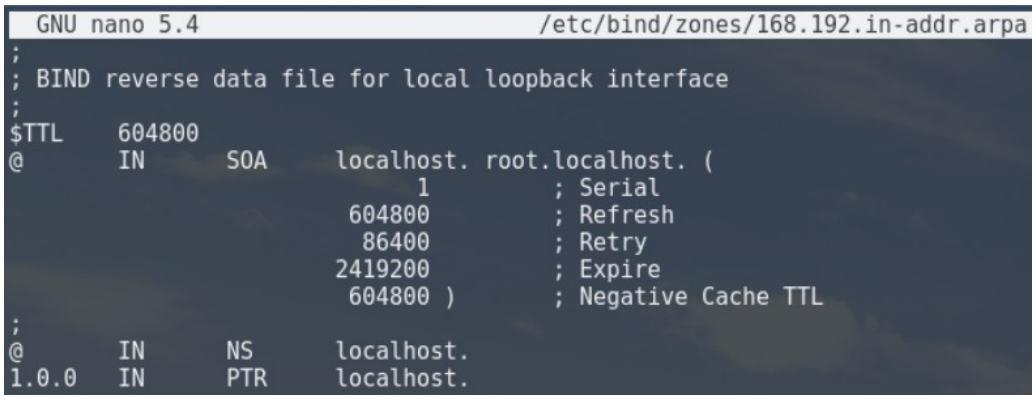
Файлы зоны для обратного просмотра служат местом, где мы будем определять PTR записей DNS для обратного просмотра DNS, т.е., когда DNS получает запрос для IP-адреса, например, «192.168.1.10», она будет выполнять поиск по файлу (файлам) зоны для обратного просмотра, чтобы получить соответствующее полное доменное имя, в нашем случае это «debian-11-server.ksd.wsr».

В debian-11-server.ksd.wsr для каждой зоны обратного просмотра, заданной в файле named.conf.local, необходимо создать файл зоны для обратного просмотра. При создании нашего файла (или файлов) зоны для обратного просмотра мы будем опираться в качестве примера на файл зоны db.local. Скопируйте его в надлежащее место с помощью следующих команд (замените имя файла назначения, чтобы оно соответствовало определению вашей зоны для обратного просмотра):

root@debian-11-server:~# cp /etc/bind/db.127 /etc/bind/zones/168.192.in-addr.arpa

Теперь необходимо отредактировать наш файл зоны для обратного просмотра. Первоначально он будет выглядеть примерно следующим образом:

root@debian-11-server:~# nano -c /etc/bind/zones/168.192.in-addr.arpa



Как и в случае с файлом зоны для прямого просмотра, вам нужно изменить запись SOA и увеличить значение serial.

Далее удалите три записи в конце файла (после записи SOA). Вместо них добавьте в конце файла записи для серверов имен нашей локальной сети (Debian 11 – Server и Debian 11 – Gateway). Обратите внимание, что во втором столбце указывается, что это записи NS.

Затем добавьте записи PTR для всех ваших серверов, чей IP-адрес соответствует подсети файла зоны, который вы редактируете. В нашем случае это будут все наши хосты, поскольку все они находятся в подсети «192.168.1.0/24». Обратите внимание, что первый столбец включает два последних байта частных IP-адресов ваших серверов в обратном порядке. Обязательно замените имена и частные IP-адреса согласно данным ваших серверов.

Записи PTR – это записи-указатели для вашего DNS-сервера. В идеале они должны совпадать с вашими записями зоны прямого просмотра, но в обратном направлении.

Вы также можете использовать этот файл, если ваша маска подсети больше «/24». Например, маска подсети «/23» будет иметь адреса от «192.168.1.1» до «192.168.1.254» включительно, для 510 возможных адресов/имен.

Полученный нами в итоге пример файла зоны для обратного просмотра должен выглядеть следующим образом.

```
 GNU nano 5.4                          /etc/bind/zones/168.192.in-addr.arpa
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ksd.wsr. debian-11-server.ksd.wsr. (
                              3         ; Serial
                         604800         ; Refresh (1 week)
                          86400         ; Retry (1 day)
                        2419200         ; Expire (4 weeks)
                         604800 )       ; Negative Cache TTL (1 week)
        NS              debian-11-server.
        A               192.168.1.10
;
; name servers - NS records
        IN      NS      localhost.
        IN      NS      debian-11-server.ksd.wsr.
        IN      NS      debian-11-gateway.ksd.wsr.

; PTR records
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 3600       ; 1 hour
10      PTR     debian-11-server.ksd.wsr.
1       PTR     debian-11-gateway.ksd.wsr.
5       PTR     debian-11-host.ksd.wsr.
15      PTR     windows-10-host.ksd.wsr.
```

Проверка зон bind9 главного DNS-сервера

Команда named-checkzone может использоваться для проверки корректности ваших файлов зоны. Первый аргумент команды указывает имя зоны, а второй аргумент определяет соответствующий файл зоны, оба из которых определены в named.conf.local.

Чтобы проверить конфигурацию зоны для прямого просмотра «ksd.wsr», запустите следующую команду:

root@debian-11-server:~# named-checkzone ksd.wsr /etc/bind/zones/ksd.wsr

А чтобы проверить конфигурацию зоны для обратного просмотра «168.192.in-addr.arpa», запустите следующую команду:

root@debian-11-server:~# named-checkzone 168.192.in-addr.arpa /etc/bind/zones/168.192.in-addr.arpa

```
root@debian-11-server:~# named-checkzone ksd.wsr /etc/bind/zones/ksd.wsr
zone ksd.wsr/IN: loaded serial 3
OK
root@debian-11-server:~# named-checkzone 168.192.in-addr.arpa /etc/bind/zones/168.192.in-addr.arpa
zone 168.192.in-addr.arpa/IN: loaded serial 3
OK
```

Когда все файлы конфигурации и зоны не будут иметь ошибок, вы должны будете перезапустить службу bind9:

root@debian-11-server:~# systemctl restart named.service

Настройка isc-dhcp-server для совместной работы с динамическим DNS сервера bind9

Теперь сконфигурируем наш DHCP-сервер. В предыдущей теме мы его уже настроили, поэтому наш нужно будет добавить несколько строк, чтобы обеспечить его совместную работу с DNS-сервером. Сначала сделаем резервную копию нашего текущего файла конфигурации:

root@debian-11-server:~# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.old-2

Теперь можно приступить к настройке DHCP-сервера. Заходим в файл /etc/dhcp/dhcpd.conf, и вносим конфигурацию:

root@debian-11-server:~# nano -c /etc/dhcp/dhcpd.conf

```
default-lease-time 11400;
max-lease-time 18000;
one-lease-per-client true;
authoritative;
log-facility local7;
```

n

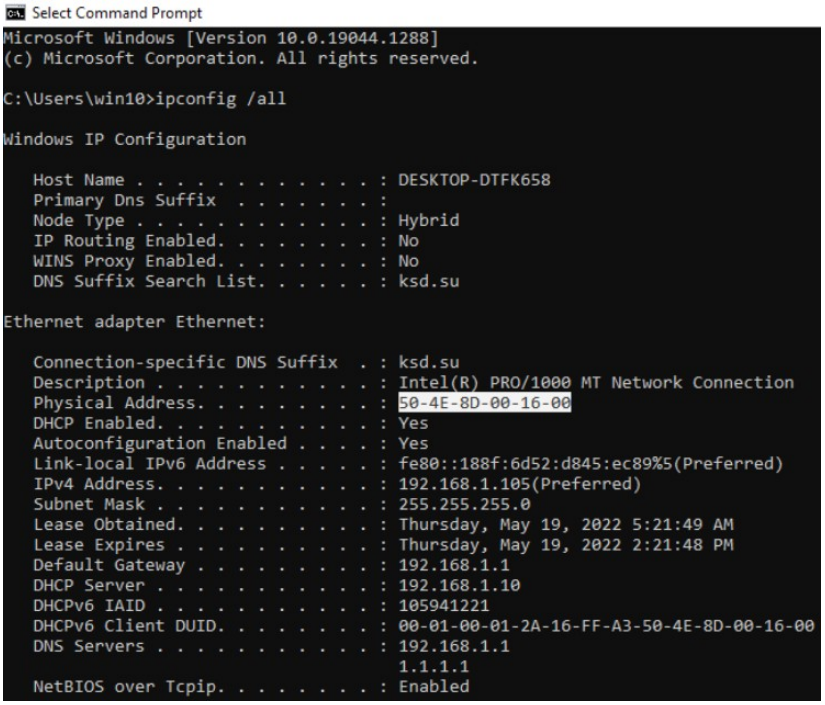
```
 GNU nano 5.4                                          /etc/dhc
ddns-domainname "ksd.wsr.";
ddns-rev-domainname "in-addr.arpa.";
ddns-update-style interim;
ignore client-updates;
update-static-leases on;
use-host-decl-names on;
option domain-name "ksd.wsr.";
include "/etc/dhcp/rndc.key";
update-optimization off;
update-conflict-detection off;

zone ksd.wsr. {
        primary 192.168.1.10;
        key rndc-key;
}

zone 168.192.in-addr.arpa. {
        primary 192.168.1.10;
        key rndc-key;
}

subnet 192.168.1.0 netmask 255.255.255.0 {
  authoritative;
  range 192.168.1.100 192.168.1.110;
  option routers 192.168.1.1;
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.1.255;
  option domain-name-servers 192.168.1.10, 192.168.1.1;
}
```

```
GNU nano 5.4                                  /etc/dhcp/dhcpd.conf

        host debian-11-host {
          hardware ethernet 50:db:8e:00:17:00;
          # Здесь вы должны написать MAC-адрес вашей виртуальной машины, которой вы хотите выставить
          # IP-адрес. У вас будет свой идентификатор.
          # Чтобы его узнать в ОС Linux, введите в терминале на требуемой машине
          # хоста команду "ip a" и скопируйте значение поля "link/ether".
          # Что его узнать в ОС , введите в командной строке на требуемой машине
          # хоста команду "ipconfig /all" и скопируйте значение поля "Physical Address".
          fixed-address 192.168.1.15;
        }
        host windows-10-host {
          hardware ethernet 50:4e:8d:00:16:00;
          fixed-address 192.168.1.5;
        }
```

Добавьте любые дополнительные устройства со статическими IP-адресами в соответствии с указанным выше форматом, которые есть в вашей сети. При добавлении MAC-адресов можно использовать символы нижнего регистра.

MAC-адреса в операционных системах Linux и Windows можно узнать следующими способами, приведенными на скриншотах ниже.

```
guest@localhost: ~
Файл  Действия  Правка  Вид  Справка
guest@localhost: ~ ×
root@localhost:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 50:db:8e:00:17:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
```



MAC-адреса в операционных системах Linux и Windows можно узнать следующими способами, приведенными на скриншотах ниже.

Обратите внимание, что MAC-адреса в программе PNETLab постоянно меняются после перезагрузки виртуальных машин, поэтому не выключайте их до того момента, пока не проверите работу главного DNS-сервера.

Неправильная конфигурация dhcpd не позволит запустить службу. Вы можете проверить ее в любой момент, запустив dhcpd -t. Если вы не увидели никаких предупреждений, значит, проверка синтаксиса пройдена, и служба сможет запуститься.

root@debian-11-server:~# dhcpd -t

```
root@debian-11-server:~# dhcpd -t
Internet Systems Consortium DHCP Server 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
```

Когда вы закончите настройку isc-dhcp-server, перезапустите службу DHCP-сервера, чтобы он заработал:

root@debian-11-server:~# systemctl restart isc-dhcp-server

Сброс кеша аренды DHCP-сервера

Сначала удалите временный файл dhcpd.leases~:

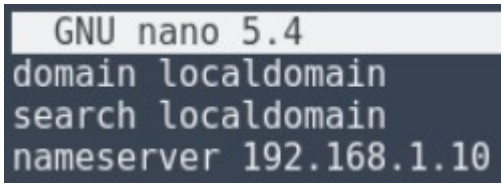
root@debian-11-server:~# rm /var/lib/dhcp/dhcpd.leases~

Потом очистите кеш аренды dhcpd.leases:

root@debian-11-server:~# echo "" > /var/lib/dhcp/dhcpd.leases

Изменение DNS-сервера по умолчанию

root@debian-11-server:~# nano /etc/resolv.conf

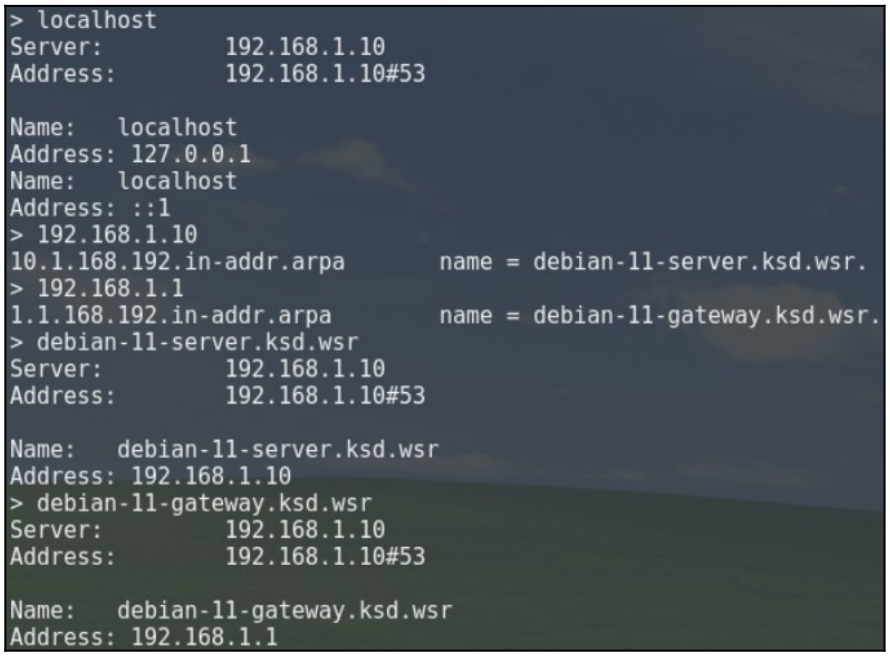


Проверка работы главного DNS-сервера с помощью утилиты nslookup

Программа nslookup (name server lookup – поиск на сервере имен) – это утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS (проще говоря, DNS-клиент). Позволяет задавать различные типы запросов и опрашивать произвольно указываемые сервера. Ее аналогом являются утилиты host и dig. Разработана в составе пакета BIND (для UNIX-систем).

Давайте с помощью этой программы проверим, что наш DNS-сервер работает исправно:

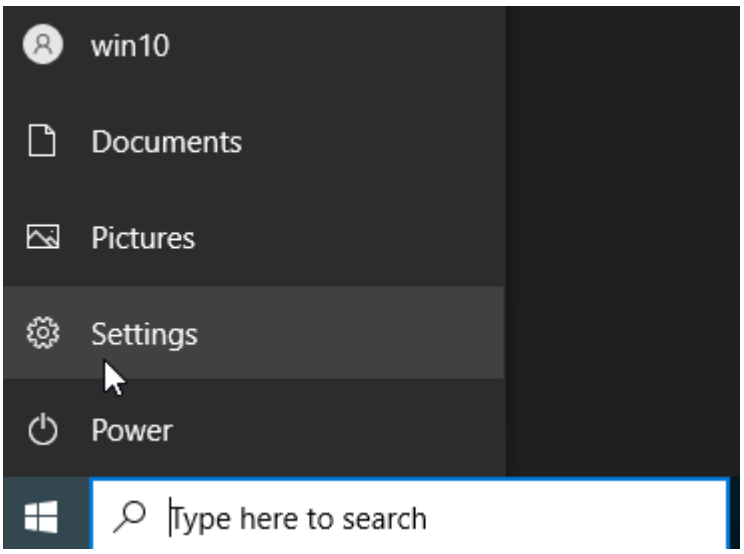
root@debian-11-server:~# nslookup

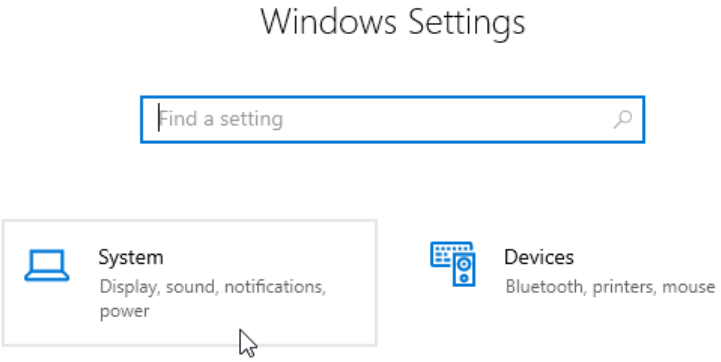


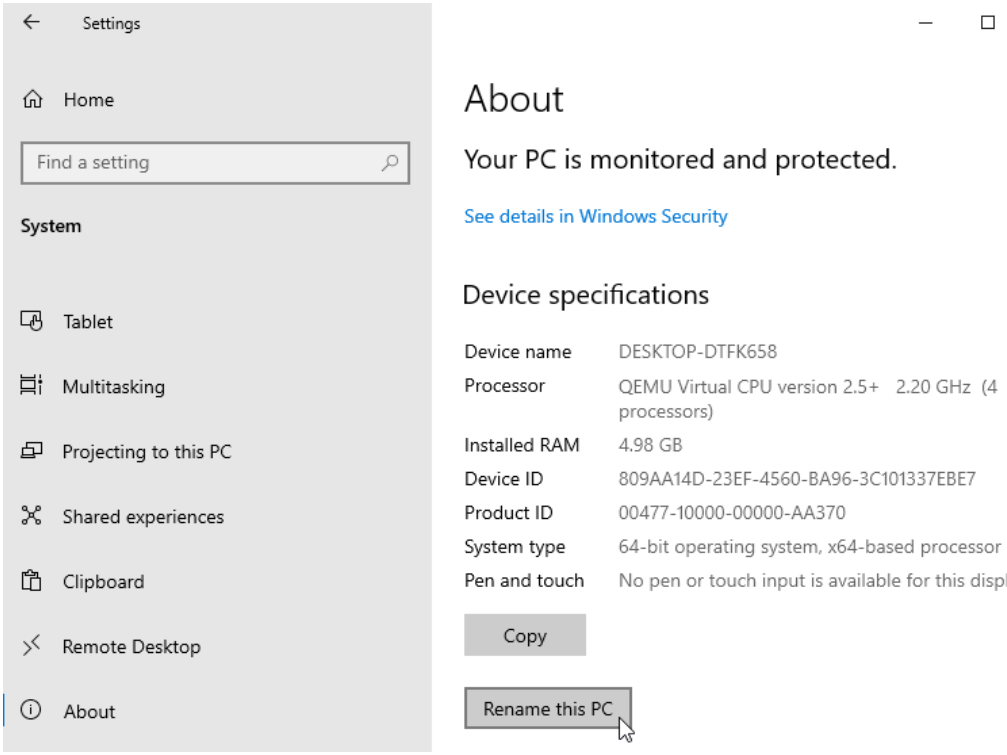
Входим на виртуальную машину Windows 10 – Host.

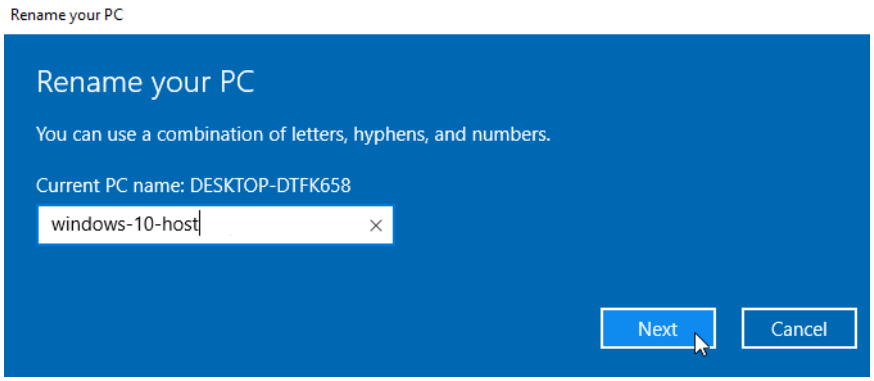
Проверка совместной работы DNS- и DHCP-сервера и проверка выдачи зарезервированных IP-адресов

Переименуйте имя компьютера в Windows 10 и получите новый IP-адрес от DHCP-сервера.


win10
Documents
Pictures
Settings
Power
Type here to search


Windows Settings
Find a setting
System
Display, sound, notifications, power
Devices
Bluetooth, printers, mouse


Settings
Home
Find a setting
System
Tablet
Multitasking
Projecting to this PC
Shared experiences
Clipboard
Remote Desktop
About
About
Your PC is monitored and protected.
See details in Windows Security
Device specifications
Device name DESKTOP-DTFK658
Processor QEMU Virtual CPU version 2.5+ 2.20 GHz (4 processors)
Installed RAM 4.98 GB
Device ID 809AA14D-23EF-4560-BA96-3C101337EBE7
Product ID 00477-10000-00000-AA370
System type 64-bit operating system, x64-based processor
Pen and touch No pen or touch input is available for this display
Copy
Rename this PC



Перезагрузите компьютер и, затем, введите следующий текст в командной строке Windows (cmd), чтобы получить новый IP-адрес:

C:\Users\Win10>ipconfig /release
C:\Users\Win10>ipconfig /renew

Убедитесь, что ваш сетевой адаптер настроен на получение IP-адреса по DHCP и что DHCP-сервер выдал IP-адрес вашей машине.

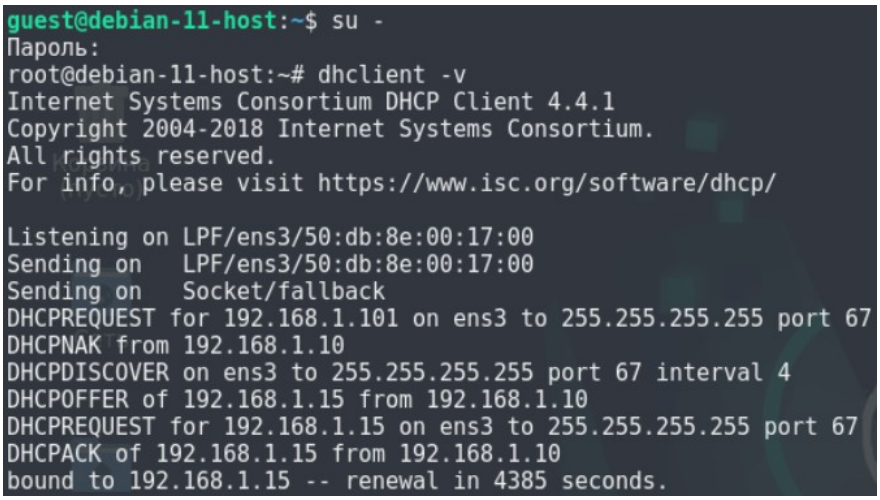
Выполнить команду ping можно в командной строке Windows (cmd):

C:\Users\Win10>ipconfig
C:\Users\Win10>ping debian-11-server.ksd.wsr
C:\Users\Win10>ping debian-11-gateway.ksd.wsr
C:\Users\Win10>ping ya.ru

Входим на виртуальную машину Debian 11 – Host.

Переименуйте имя компьютера в Debian 11 и получите новый IP-адрес от DHCP-сервера.

guest@localhost:~$ su -
Пароль: guest
root@localhost:~# hostnamectl set-hostname debian-11-host
root@debian-11-host:~# dhclient -r (Освобождение текущей аренды IP-адреса)
root@debian-11-host:~# dhclient -v (Получение нового IP-адреса с подробным выводом действий DHCP-клиента)



Убедитесь, что ваша виртуальная машина может пропинговать другие машины в локальной сети по доменному имени:

```
root@debian-11-host:~# ip a
root@debian-11-host:~# ping debian-11-server.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping debian-11-gateway.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping debian-11-host.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping windows-10-host.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping ya.ru
[ctrl+c]
```

Как вы можете заметить виртуальные машины автоматически получают зарезервированные IP-адреса, которые мы им явно выделили. DNS- и DHCP-сервер работают вместе успешно. Теперь стоит настроить динамическое обновление зон DHCP-сервером.

Входим на виртуальную машину Debian 11 – Server.

Динамическое обновление зон DHCP-сервером

Прежде чем выполнять какие-либо действия, перезапустите DHCP- и DNS-сервера:

```
root@debian-11-server:~# systemctl restart isc-dhcp-server.service
root@debian-11-server:~# systemctl restart bind9.service
```

Каждый раз, когда мы вносим изменения в конфигурацию bind9, необходимо увеличивать серийный номер как минимум на 1. «Стандартный» метод после его запуска заключается в том, чтобы добавить дату к вашему серийному номеру, например, ГГГГММДДxx. Если этого не делать, то дополнительные DNS-сервера не станут реплицировать данные с основного сервера.

Если вы хотите внести изменения после запуска DNS-сервера, вам необходимо временно заморозить свою зону:

```
root@debian-11-server:~# rndc freeze ksd.wsr
root@debian-11-server:~# rndc freeze 168.192.in-addr.arpa
```

Если вы внесли изменения, то увеличьте свой серийный номер на 1, затем перезагрузите и разморозьте зоны, чтобы снова разрешить динамические обновления:

```
root@debian-11-server:~# nano -c /etc/bind/zones/ksd.wsr
```

```
…
4       ; serial
…
```

```
root@debian-11-server:~# nano -c /etc/bind/zones/168.192.in-addr.arpa
```

```
…
4       ; serial
```

…

root@debian-11-server:~# rndc reload ksd.wsr
root@debian-11-server:~# rndc thaw ksd.wsr
root@debian-11-server:~# rndc reload 168.192.in-addr.arpa
root@debian-11-server:~# rndc thaw 168.192.in-addr.arpa

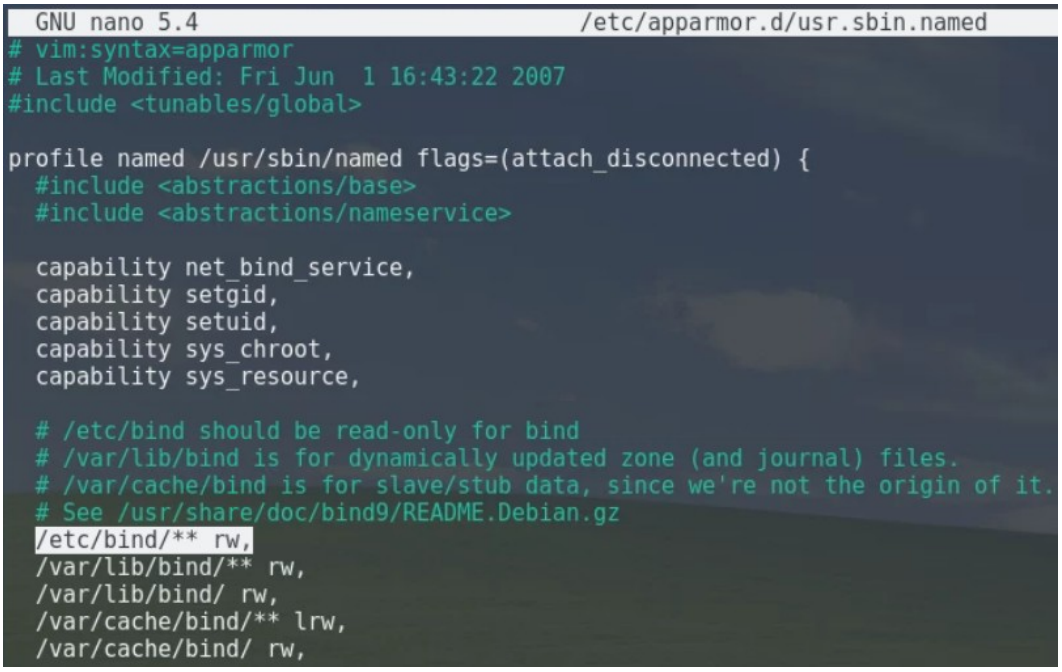
Настройка AppArmor

AppArmor — программный инструмент упреждающей защиты, основанный на политиках безопасности (известных также как профили), которые определяют, к каким системным ресурсам и с какими привилегиями может получить доступ то или иное приложение. В AppArmor включен набор стандартных профилей, а также инструменты статического анализа и инструменты, основанные на обучении, позволяющие ускорить и упростить построение новых профилей

В нашей системе используется apparmor, поэтому нам необходимо перенастроить его. Из-за работы этой программы по умолчанию пользователь от которого работает DNS сервер bind9 не имеет права записи в каталог с файлами конфигурации /etc/bind, а удаленное обновление зон как раз требует возможности записи в этот каталог от пользователя bind. Исправить это можно, отредактировав конфигурационный файл apparmor для ограничения bind, найти его можно в директории /etc/apparmor.d/, имя файла usr.sbin.named.

Отредактируйте файл как показано ниже и исправьте значение по умолчанию: /etc/bind/** r, на /etc/bind/** rw,:

root@debian-11-server:~# nano -c /etc/apparmor.d



После этого перезапустите apparmor и bind9:

root@debian-11-server:~# systemctl restart apparmor.service
root@debian-11-server:~# systemctl restart bind9.service

Также вы можете просмотреть логи, чтобы выявить возможные ошибки. Для это введите следующую команду:

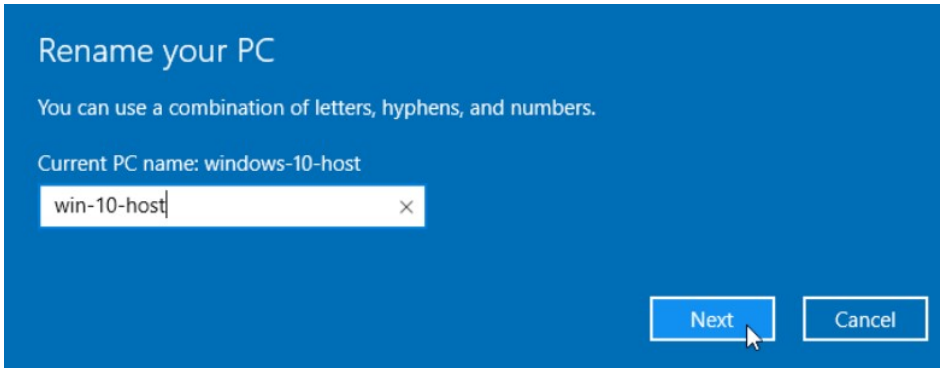
root@debian-11-server:~# systemctl grep -E \(dhcpd\|named\) /var/log/syslog

Теперь сервер готов к удаленному обновлению зон, перейдем к конечной настройке клиента.

Входим на виртуальную машину Windows 10 – Host.

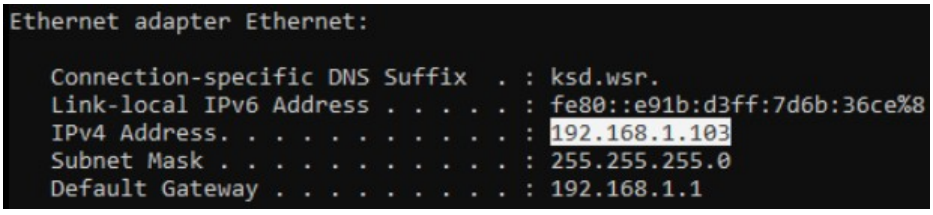
Проверка выдачи незарезервированных IP-адресов

Теперь давайте снова проверим работу DHCP-сервера в выдаче IP-адресов, которые не были зарезервированы в конфигурационном файле. Повторите действия по переименованию компьютера в Windows 10 и снова получите новый IP-адрес от DHCP-сервера.



Перезагрузите компьютер и, затем, введите следующий текст в командной строке Windows (cmd), чтобы получить новый IP-адрес:

C:\Users\Win10>ipconfig /release
C:\Users\Win10>ipconfig /renew



Убедитесь, что ваш сетевой адаптер настроен на получение IP-адреса по DHCP и что DHCP-сервер выдал IP-адрес вашей машине.

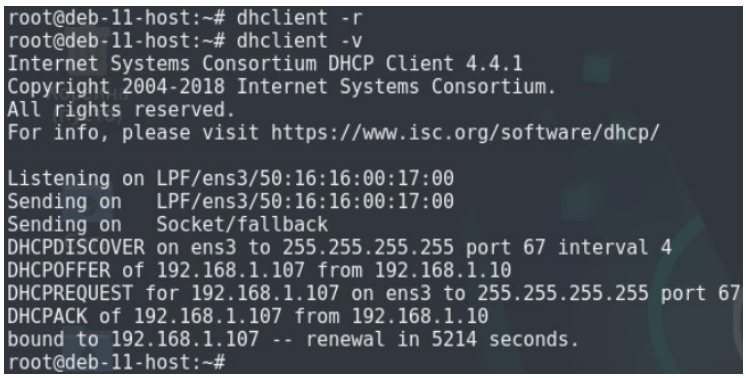
Выполнить команду ping можно в командной строке Windows (cmd):

C:\Users\Win10>ipconfig
C:\Users\Win10>ping debian-11-server.ksd.wsr
C:\Users\Win10>ping debian-11-gateway.ksd.wsr
C:\Users\Win10>ping win-10-host.ksd.wsr
C:\Users\Win10>ping ya.ru

Входим на виртуальную машину Debian 11 – Host.

Повторите действия по переименованию компьютера в Debian 11 и снова получите новый IP-адрес от DHCP-сервера.

guest@localhost:~$ su -
Пароль: guest
root@localhost:~# hostnamectl set-hostname deb-11-host
root@deb-11-host:~# dhclient -r
root@deb-11-host:~# dhclient -v



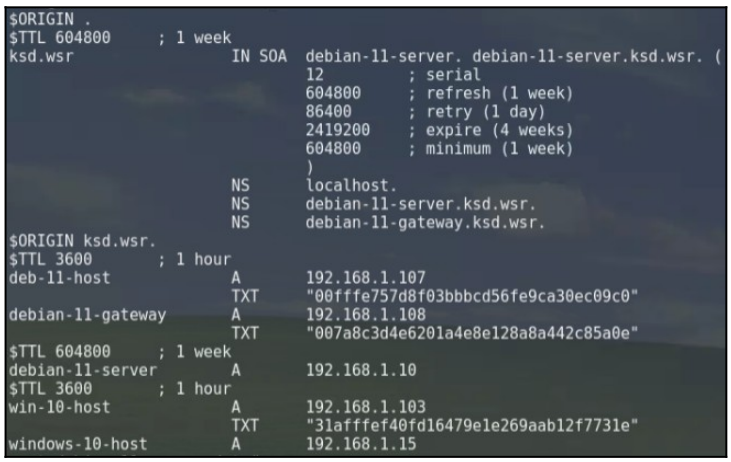
Убедитесь, что ваша виртуальная машина может пропинговать другие машины в локальной сети по доменному имени:

root@debian-11-host:~# ip a
root@debian-11-host:~# ping debian-11-server.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping debian-11-gateway.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping deb-11-host.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping win-10-host.ksd.wsr
[ctrl+c]
root@debian-11-host:~# ping ya.ru
[ctrl+c]

Входим на виртуальную машину Debian 11 – Server.

Можно наблюдать, что из-за того, что у виртуальных машин теперь имя хоста отличается от тех, что мы зарезервировали в конфигурационных файлах DNS-сервера, то теперь он выдает IP-адреса из другого диапазона от 192.168.1.100 до 192.168.1.110. Помимо этого наши имена хостов виртуальных машин стали DNS-суффиксами доменных имен и они автоматически записались в DNS-зоны ksd.wsr и 168.192.in-addr.arpa:

root@debian-11-host:~# cat /etc/bind/zones/ksd.wsr



```
$ORIGIN .
$TTL 604800     ; 1 week
ksd.wsr                 IN SOA  debian-11-server. debian-11-server.ksd.wsr. (
                                12         ; serial
                                604800     ; refresh (1 week)
                                86400      ; retry (1 day)
                                2419200    ; expire (4 weeks)
                                604800     ; minimum (1 week)
                                )
                        NS      localhost.
                        NS      debian-11-server.ksd.wsr.
                        NS      debian-11-gateway.ksd.wsr.
$ORIGIN ksd.wsr.
$TTL 3600       ; 1 hour
deb-11-host             A       192.168.1.107
                        TXT     "00fffe757d8f03bbbcd56fe9ca30ec09c0"
debian-11-gateway       A       192.168.1.108
                        TXT     "007a8c3d4e6201a4e8e128a8a442c85a0e"
$TTL 604800     ; 1 week
debian-11-server        A       192.168.1.10
$TTL 3600       ; 1 hour
win-10-host             A       192.168.1.103
                        TXT     "31afffef40fd16479e1e269aab12f7731e"
windows-10-host         A       192.168.1.15
```

Теперь можно приступать к настройке дополнительного DNS-сервера.

Входим на виртуальную машину Debian 11 – Gateway.

Настройка реплицирующего дополнительного DNS-сервера

В большинстве сред правильным решением будет создание дополнительного DNS-сервера, который будет отвечать на запросы, если основной сервер окажется недоступным. К счастью, настройка дополнительного DNS-сервера выполняется намного проще.

Обновление локального индекса пакетов

guest@debian-11-gateway:~$ su –
Пароль: guest
root@debian-11-gateway:~# apt update

Установка пакета с программой bind9 и проверка его на работоспособность

root@debian-11-gateway:~# apt install bind9 bind9utils bind9-doc
root@debian-11-gateway:~# systemctl status named.service

На сервере Debian-11-Gateway отредактируйте файл named.conf.options:

guest@debian-11-gateway:~$ nano /etc/bind/named.conf.options

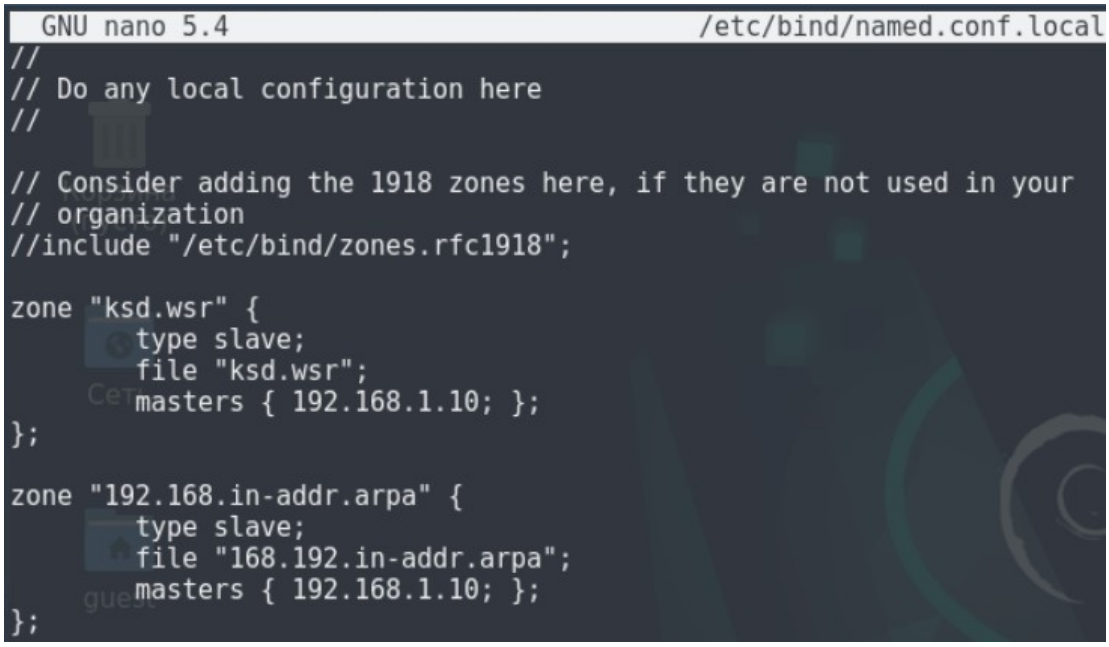
Сначала в верхней части файла добавьте ACL с частными IP-адресами всех ваших доверенных серверов, а затем под директивой directory добавьте следующие строки.

Сохраните и закройте файл named.conf.options. Этот файл должен выглядеть так же, как файл named.conf.options главного DNS-сервера, за исключением того, что его необходимо настроить на прослушивание частного IP-адреса Debian-11-Gateway.

Теперь необходимо отредактировать файл named.conf.local:

root@debian-11-gateway:~# nano /etc/bind/named.conf.local

Определите slave-зоны, соответствующие master-зонам основного DNS-сервера. Обратите внимание, что в качестве типа используется slave, в файле отсутствует путь, и существует директива masters, которая должна быть настроена на частный IP-адрес основного DNS-сервера.



Сохраните и закройте файл named.conf.local. После запустите следующую команду для проверки валидности ваших файлов конфигурации:

root@debian-11-gateway:~# named-checkconf

Если проверка файлов конфигурации была выполнена успешно, то перезапустите bind9:

root@debian-11-gateway:~# systemctl restart bind9

Теперь у вас есть основной и дополнительный DNS-серверы для имени частной сети и преобразования IP-адреса. Теперь вам нужно настроить ваши клиентские серверы, чтобы они могли использовать ваши частные DNS-серверы.

Изменение DNS-сервера по умолчанию

root@debian-11-server:~# nano /etc/resolv.conf

```
domain ksd.wsr.
search ksd.wsr.
nameserver 192.168.1.1
nameserver 1.1.1.1
```

Входим на виртуальную машину Debian 11 – Host.

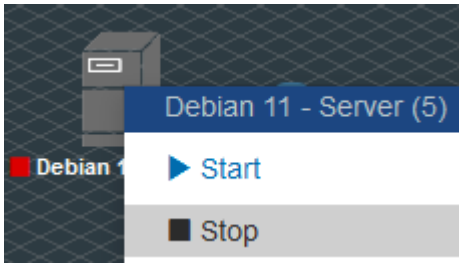
Проверка работы дополнительного DNS-сервера с помощью программ nslookup и traceroute

В первую очередь попробуем вызвать эту команду, чтобы узнать видит ли клиентский компьютер второй DNS-сервер:

root@debian-11-host:~# nslookup -type=ns ksd.wsr

```
root@deb-11-host:~# nslookup -type=ns ksd.wsr
Server:         192.168.1.10
Address:        192.168.1.10#53

ksd.wsr nameserver = debian-11-server.ksd.wsr.
ksd.wsr nameserver = debian-11-gateway.ksd.wsr.
ksd.wsr nameserver = localhost.
```

Далее необходимо выключить наш основной DNS-сервер, чтобы проверить обратиться ли клиентский компьютер к дополнительному DNS-серверу. И затем снова введите команду nslookup -type=ns ksd.wsr.



root@debian-11-host:~# nslookup -type=ns ksd.wsr

```
Server:         192.168.1.1
Address:        192.168.1.1#53

ksd.wsr nameserver = localhost.
ksd.wsr nameserver = debian-11-gateway.ksd.wsr.
ksd.wsr nameserver = debian-11-server.ksd.wsr.
```

Входим на виртуальную машину Windows 10 – Host.

Программа traceroute – это утилита, которая позволяет проследить маршрут следования данных до удаленного адресата в сетях TCP/IP. В Linux используется команда

traceroute, а в Windows – tracert. При помощи этих команд можно увидеть путь пакета данных от вашего компьютера до целевого сервера или сайта.

Давайте с помощью этой программы проверим, что наш DNS-сервер работает исправно:

```
root@debian-11-gateway:~# traceroute win-10-host.ksd.wsr
root@debian-11-gateway:~# traceroute debian-11-gateway.ksd.wsr
root@debian-11-gateway:~# traceroute ya.ru
```

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №16

Тема: Развертывание прокси-сервера HTTP – Squid

Цель работы: Сконфигурировать виртуальную машину в качестве сервера HTTP на основе Squid:

– Обновление локального индекса пакетов;

– Установка пакета с программой Squid и его базовая настройка;

– Авторизация по логину и паролю;

– Блокировка сайтов на прокси-сервере;

– Разрешение доступа на определенные сайты по паролю;

– Блокировка определенных ключевых слов в URL-запросах с помощью Squid;

– Проверка работы Squid-сервера.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Squid – это кэширующий и пересылающий HTTP веб-прокси. Он имеет широкий спектр применений, включая ускорение работы веб-сервера за счет кэширования повторяющихся запросов, кэширование поиска в Интернете, DNS и других компьютерных сетях для группы людей, совместно использующих сетевые ресурсы, а также обеспечение безопасности путем фильтрации трафика. Хотя Squid в основном используется для HTTP и FTP, он включает ограниченную поддержку нескольких других протоколов, включая SSL, TLS и HTTPS.

Нет смысла устанавливать прокси на своей домашней машине, так как функции кэширования выполняет браузер. Прокси-сервер стоит применять лишь в том случае, если в вашей сети три-четыре компьютера, которым нужен выход в Интернет. В этом случае запрос от браузера к прокси-серверу обрабатывается быстрее, чем от браузера к ресурсам Интернет, и таким образом увеличивается производительность. При этом можно смело установить размер кэша в браузерах клиентов равным нулю.

Squid – это нечто большее, чем просто прокси-сервер. Это своеобразный стандарт кэширования информации в сети Интернет. В силу его повсеместной распространенности, мы в этом гайде будем настраивать именно его.

Входим на виртуальную машину Debian 11 – Gateway.

Обновление локального индекса пакетов

root@debian-11-gateway:~# su –
Пароль: guest
root@debian-11-gateway:~# apt update

Установка пакета с программой Squid и его базовая настройка

root@debian-11-gateway:~# apt install squid
nano -c /etc/squid/squid.conf

Схема управления доступом веб-прокси-сервера Squid состоит из двух разных компонентов:

– Элементы Access Control List (ACL) – это строки директив, которые начинаются со слова «acl» и представляют типы тестов, которые выполняются для любой транзакции запроса;

– Правила списка доступа состоят из действия allow (разрешения) или deny (запрета), за которым следует ряд элементов ACL, они используются для указания того, какое действие или ограничение необходимо применить для данного запроса. Они проверяются по порядку, и поиск по списку прекращается, как только одно из правил совпадает. Если правило имеет несколько элементов ACL, оно реализуется как логическая операция И (все элементы ACL правила должны выполняться, чтобы правило считалось совпавшим).

Синтаксис acl выглядит следующим образом:

 # acl имя тип определение1 определение2 определение3 …

Виды и примеры использования http_access:

 # http_access allow localnet
 # http_access allow localhost
 # http_access deny !Safe_ports
 # http_access deny all
 # http_access allow auth_users
 # http_access allow all

Вы также можете использовать списки определений, которые хранятся в файлах на вашем жестком диске. Предположим, у вас есть список URL-адресов поисковых систем, которые вы хотите разрешить:

cat /etc/squid/search-engines-urls.txt:

```
.google.com
.bing.com
```

```
.yandex.ru
.duckduckgo.com
.yahoo.com
```

В таком случае ACL для этого файла будет выглядеть так:

acl accessess_to_search_engines dstdomain "/etc/squid/search-engines-urls.txt"

Кавычки здесь необходимы чтобы сообщить Squid, что ему нужно искать определения в этом файле.

Сами по себе элементы acl ничего не меняют в поведении прокси-сервера. Это только списки для дальнейшего использования с правилами списка доступа. Сами по себе эти тесты ничего не делают, например, слово «воскресенье» соответствует дню недели, но не указывает, в какой день недели вы это читаете.

Наши подсети клиентских компьютеров отличаются от стандартных (192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8), поэтому необходимо их добавить в acl, например:

```
# TAG: acl
…
acl localnet src 172.16.1.0/24# RFC 1918 local private network (LAN) (Строка ~1192)
acl localnet src 192.168.1.0/24         # RFC 1918 local private network (LAN) (Строка ~1193)
```

Если вы записали только списки управления доступом, то фактически ничего не блокируется – это всего лишь определения. ACL можно использовать в различных местах вашего squid.conf. Самая полезная функция, с которой они могут использоваться в паре, это инструкция http_access. Она работает аналогично тому, как брандмауэр обрабатывает правила. Для каждого запроса, который получает Squid, он будет просматривать все операторы http_access по порядку, пока не найдет соответствующую строку. Затем он либо принимает, либо отклоняет запрос в зависимости от ваших настроек. Остальные правила, идущие после сработавшего, игнорируются.

Общий синтаксис http_access выглядит так:

http_access (allow|deny) acl1 acl2 acl3 ...

Разрешаем доступ для локальных сетей, которые заданы опцией acl localnet:

```
# TAG: http_access
…
http_access allow localnet (Строка ~1407)
```

Настраиваем директорию для кэша[12]:

```
# TAG: cache_dir
```

[12] Кэш – это область для размещения данных или сами данные, к которым, с высокой вероятностью, может понадобиться повторный доступ или идет частое обращение. Процесс помещения данных в кэш называют кэшированием. Нужен для ускорения работы системы и приложений.

```
…
cache_dir ufs /var/spool/squid 4096 32 256 (Строка ~3635)
```

** где ufs – это файловая система (ufs для Squid является самой подходящей); /var/spool/squid – это директория хранения кэша; 4096 – объем пространства в мегабайтах, которое будет выделено под кэш; 32 – количество каталогов первого уровня, которое будет создано для размещение кэша; 256 – количество каталогов второго уровня, которое будет создано для размещение кэша.*

Добавим имя хоста в конфигурационный файл:

```
# TAG: visible_hostname
…
visible_hostname debian-11-gateway.ksd.su (Строка ~6028)
```

Эту директиву обычно используют, когда Squid не может самостоятельно определить полное доменное имя или же если вы хотите выставить в Интернет к какое-нибудь специальное внешнее имя.

Останавливаем Squid:

```
root@debian-11-gateway:~# systemctl stop squid
```

Создаем структуру папок под кеш следующей командой:

```
root@debian-11-gateway:~# squid -z
```

Запускаем Squid и разрешаем его автозапуск:

```
root@debian-11-gateway:~# systemctl start squid
root@debian-11-gateway:~# systemctl enable squid
```

Авторизация по логину и паролю

Открываем конфигурационный файл:

```
root@debian-11-gateway:~# nano -c /etc/squid/squid.conf
```

Вставляем следующее:

```
# TAG: auth_param
…
auth_param basic program /usr/lib/squid/basic_ncsa_auth /etc/squid/auth_users (Строка ~583)
auth_param basic children 25 (Строка ~584)
auth_param basic realm SQUID PROXY (Строка ~585)
auth_param basic credentialsttl 3 hours (Строка ~586)
```

** где /usr/lib/squid/basic_ncsa_auth – расположение ncsa_auth (в зависимости от системы может находиться в другом каталоге); /etc/squid/auth_users – файл с логинами и паролями; children 25 разрешает 25 одновременных подключений; SQUID PROXY – произвольная фраза для приветствия; credentialsttl 3 hours будет держать сессию 3 часа, после потребуется повторный ввод логина и пароля.*

Создаем новый acl для пользователей, которые прошли регистрацию. Сделаем регистрацию обязательной:

```
#  TAG: acl
…
```

```
acl auth_users proxy_auth REQUIRED (Строка ~1188)
```

Находим опцию:

```
http_access deny !Safe_ports (Строка ~1386)
```

И после нее добавляем:

```
http_access allow auth_users (Строка ~1387)
```

Устанавливаем утилиту apache2-utils:

```
root@debian-11-gateway:~# apt install apache2-utils
```

Создаем файл с пользователями и создаем первую пару логина и пароля:

```
root@debian-11-gateway:~# htpasswd -c /etc/squid/auth_users user1
New password: guest
Re-type password: guest
```

Перечитываем конфигурацию и перезапускаем Squid:

```
root@debian-11-gateway:~# squid -k reconfigure
root@debian-11-gateway:~# systemctl restart squid
```

Блокировка сайтов на прокси-сервере

Чтобы заблокировать доступ к нежелательным веб-сайтам, сначала создайте файл с именем «blacklisted_sites.acl», в котором будут храниться сайты из черного списка:

```
root@debian-11-gateway:~# nano -c /etc/squid/blacklisted_sites.acl
```

Теперь добавьте веб-сайты, доступ к которым вы хотите заблокировать, например:

```
.vk.com
.youtube.com
```

Начальная точка сообщает squid о необходимости заблокировать все адреса этих сайтов, включая под-домены www.vk.com, m.vk.com и так далее.

Теперь откройте файл конфигурации Squid:

```
root@debian-11-gateway:~# nano -c /etc/squid/squid.conf
```

Добавьте в него следующие строки:

```
acl bad_urls dstdomain "/etc/squid/blacklisted_sites.acl" (Строка ~1381)
http_access deny bad_urls
```

Очень важен порядок http_access – если в вашем конфигурационном файле есть описанная выше директива для включения аутентификации на прокси-сервере по паролю («http_access allow auth_users»), то блокировка доменов должна идти перед ней, иначе ничего не будет работать:

Теперь сохраните файл и перезапустите Squid:

```
root@debian-11-gateway:~# squid -k reconfigure
root@debian-11-gateway:~# systemctl restart squid
```

Эта блокировка работает для сайтов, использующих протокол HTTP, а также для сайтов, использующих протокол HTTPS. Но в зависимости от используемого протокола сообщение ошибки будет разным. Для сайтов на HTTP будет четкое сообщение:

ОШИБКА Запрошенный URL не может быть получен

Также для сайтов на HTTP можно настроить свои собственные сообщения об ошибках, в том числе для каждого сайта в отдельности.

Для сайтов на HTTPS будет ошибка «Прокси-сервер отказывается принимать соединения». Различие из-за того, что протокол HTTP позволяет вмешиваться в передаваемый трафик и полностью менять страницу сайта, в данном случае показывая ошибку. Что касается HTTPS, то прокси-сервер не может как-либо модифицировать передаваемые данные, но все еще может заблокировать запрос от пользователя на этапе разрешения имен (DNS-запрос) или на этапе SSL/TLS-рукопожатия, когда запрашиваемый домен передается в открытом виде. Поэтому других мер, кроме как прервать подключение, у прокси нет.

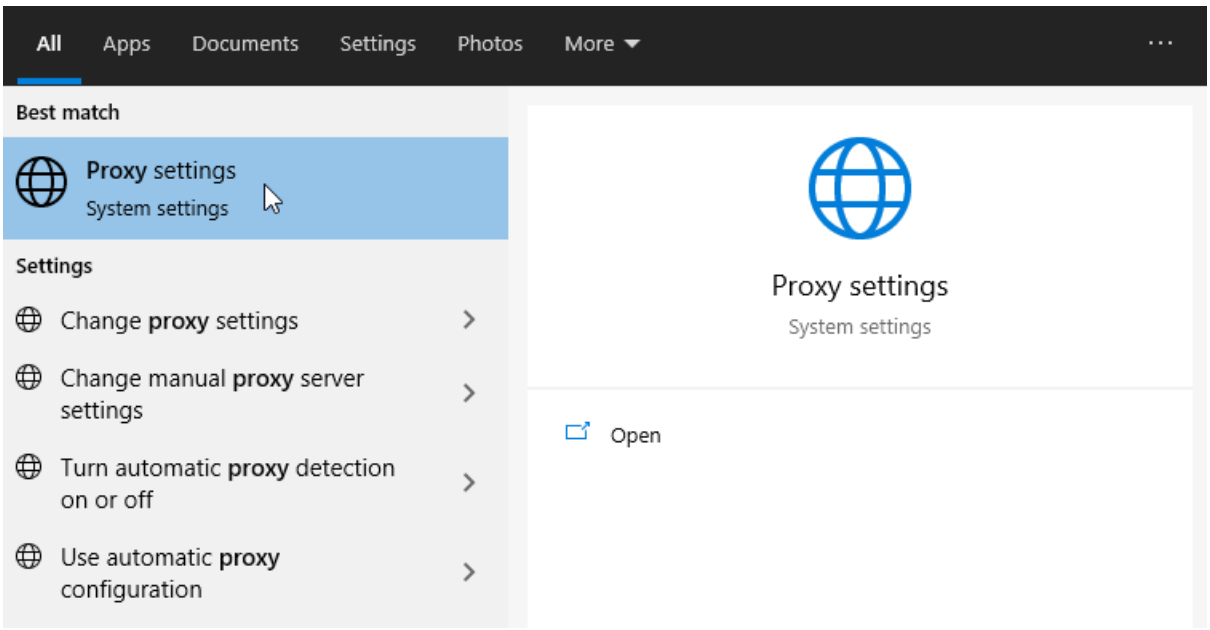
Кроме dstdomain, также имеются такие типы ACL как dstdom_regex (домены по регулярным выражениям[13]), url_regex (поиск совпадение по регулярном выражениям в полном URL), urlpath_regex (поиск совпадение по регулярном выражениям в пути URL) и некоторые другие.

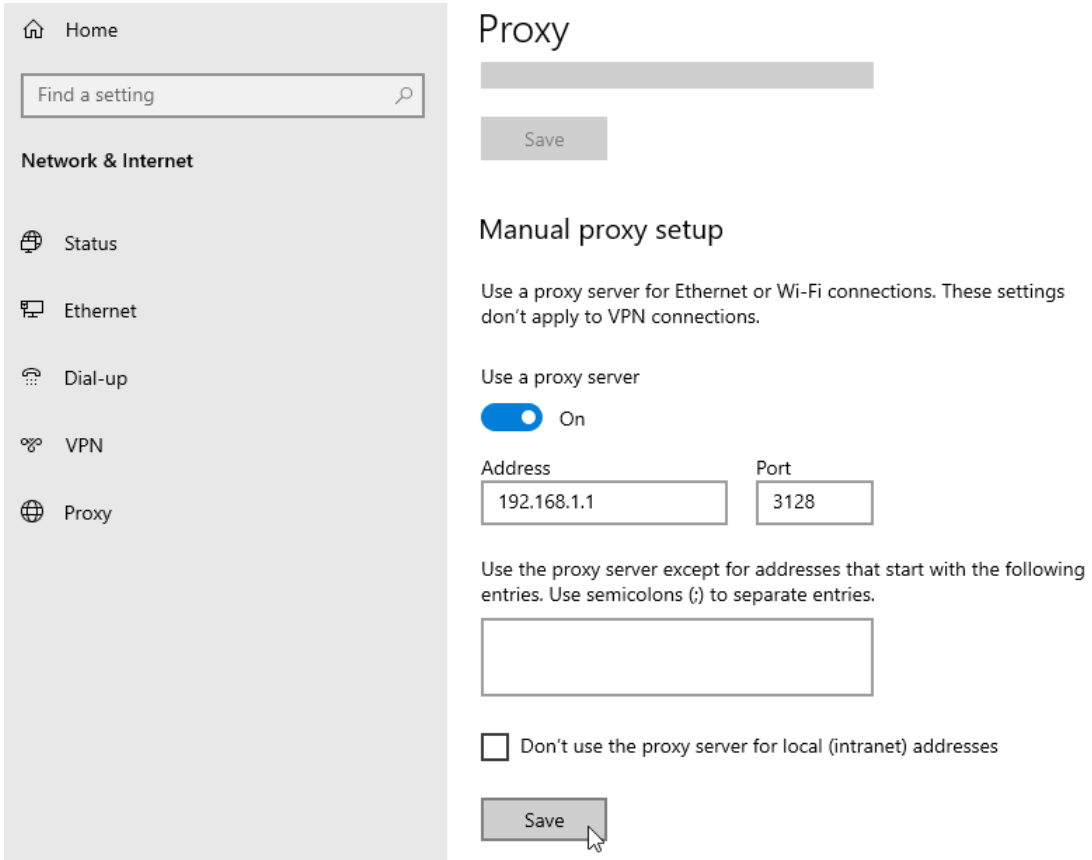
Входим на виртуальную машину Windows 10 – Host.

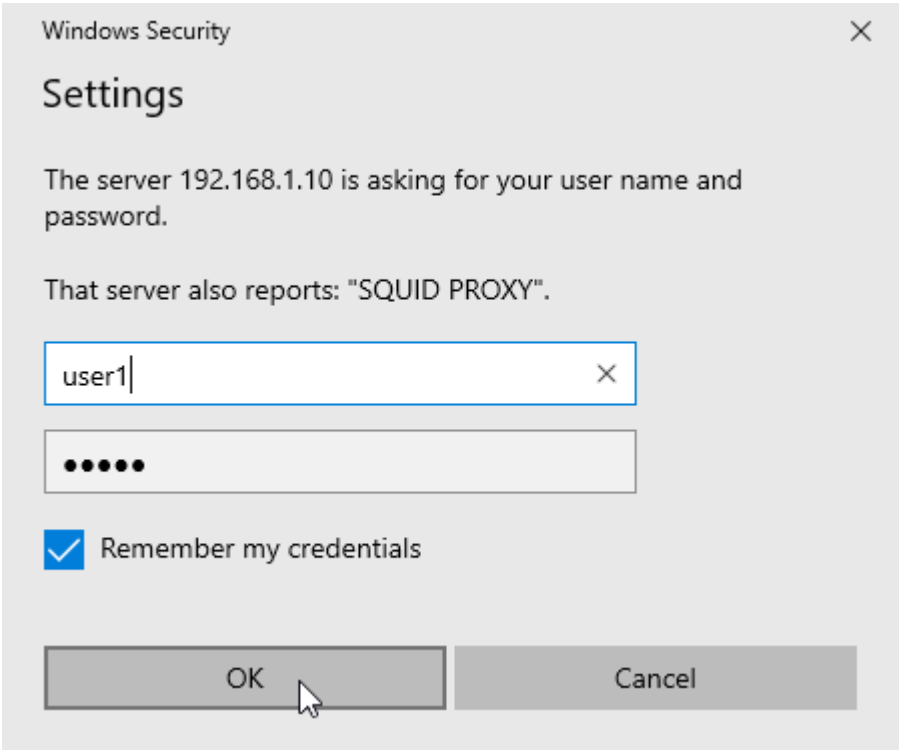
Проверка работы Squid-сервера

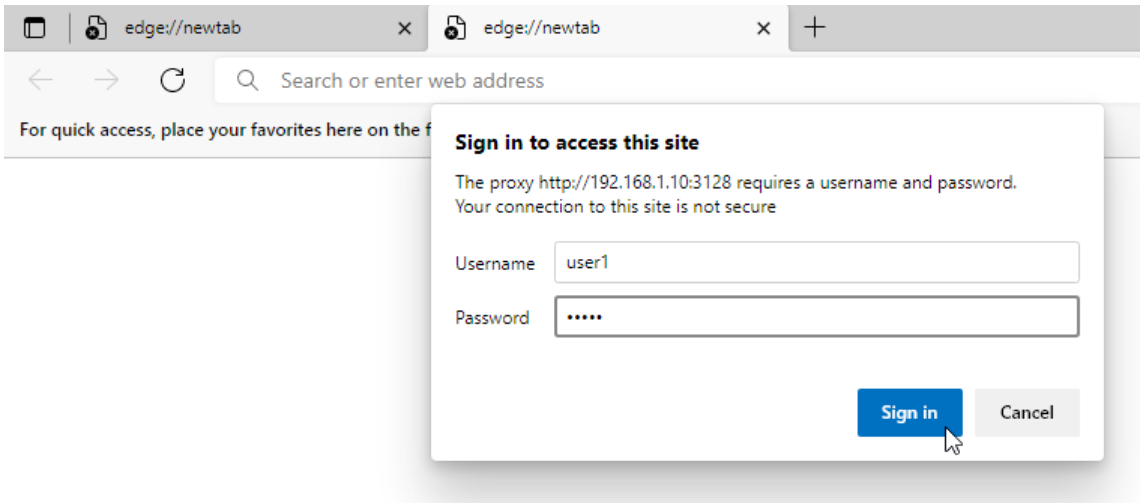
Заходим в настройки браузера и настраиваем использование прокси-сервера. Например, в Microsoft Edge настройки нужно выставить такими.

[13] Регулярные выражения (Regular Expressions) – это шаблоны, используемые для сопоставления последовательностей символов. Они являются особенностью многих программ и почти всех существующих языков программирования. Регулярные выражения чаще всего используются для поиска конкретных элементов в большом наборе текста, замены одних символов на другие, проверки ввода и т.д.


All
Apps
Documents
Settings
Photos
More
Best match
Proxy settings
System settings
Settings
Change proxy settings
Change manual proxy server settings
Turn automatic proxy detection on or off
Use automatic proxy configuration
Proxy settings
System settings
Open

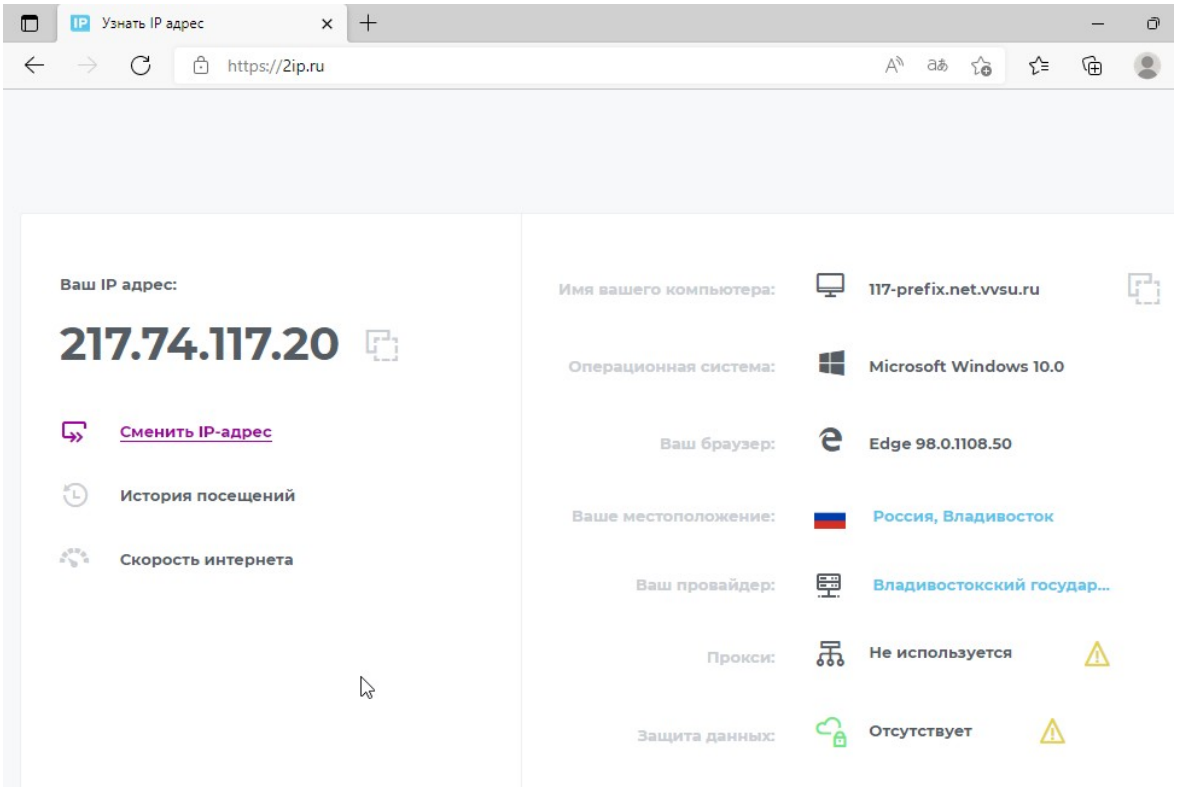

Home
Find a setting
Network & Internet
Status
Ethernet
Dial-up
VPN
Proxy
Proxy
Save
Manual proxy setup
Use a proxy server for Ethernet or Wi-Fi connections. These settings don't apply to VPN connections.
Use a proxy server
On
Address
192.168.1.1
Port
3128
Use the proxy server except for addresses that start with the following entries. Use semicolons (;) to separate entries.
Don't use the proxy server for local (intranet) addresses
Save



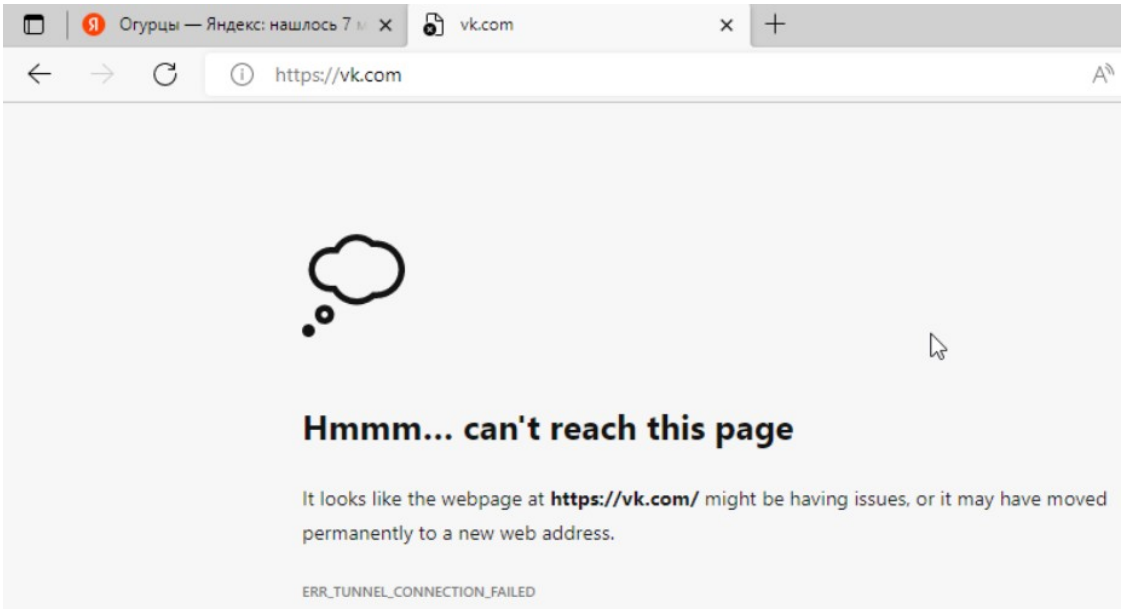


** Примечание! В вашем случае вместо IP-адреса 192.168.1.10 будет 192.168.1.1.*

Убедимся, что прокси-сервер работает. Заходим в браузер Microsoft Edge и открываем сайт 2ip.ru.



Теперь проверьте, что вы не можете войти на те сайты, которые добавили в черный список.



Входим на виртуальную машину Debian 11 – Gateway.

Разрешение доступа на определенные сайты по паролю

Рассмотрим, как настроить разрешение доступа на определенные сайты только аутентифицированным пользователям. Допустим, мы хотим использовать прокси-сервер для ограничения доступа на некоторые сайты – но не просто для блокировки, а с возможностью ввести пароль и получить доступ к сайту.

В качестве пароля будем использовать аутентификацию на прокси-сервере.

Список сайтов, доступ к которым доступен только по паролю, расположен в файле /etc/squid/restricted_sites.acl, тогда конфигурация, следующая:

root@debian-11-gateway:~# nano -c /etc/squid/restricted_sites.acl

```
.myspace.com
```

Теперь откройте файл конфигурации Squid:

root@debian-11-gateway:~# nano -c /etc/squid/squid.conf

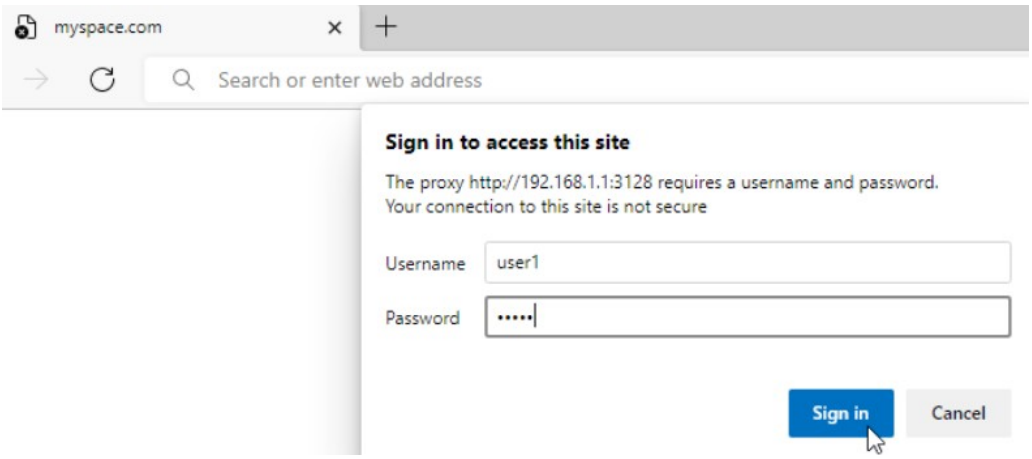
```
acl restricted_urls dstdomain "/etc/squid/restricted_sites.acl" (Строка ~1384)
# Добавляем правило со списком ограниченных к доступу сайтов.
http_access allow restricted_urls auth_users
# Если пользователь пытается открыть сайты из правила restricted_urls, то
# появляется окно аутентификации на сервере. Пока не будет введен пароль от
# прокси, невозможно открыть ограниченные сайты.
```

Теперь сохраните файл и перезапустите Squid:

```
root@debian-11-gateway:~# squid -k reconfigure
root@debian-11-gateway:~# systemctl restart squid
```

Проверка работы Squid-сервера с приведенными изменениями (1)

Убедимся, что настройки на наш прокси-сервер применились. Заходим в браузер Microsoft Edge и пытаемся открыть сайт myspace.com.



Блокировка определенных ключевых слов в URL-запросах с помощью Squid

Чтобы заблокировать список ключевых слов, которые могут встречаться в URL, сначала создайте файл с именем «block_keywords.acl», в котором будут храниться ключевые слова из черного списка.

root@debian-11-gateway:~# nano /etc/squid/block_keywords.acl

Теперь добавьте ключевые слова, к которым вы хотите заблокировать доступ, например:

```
duck
gmail
```

Теперь откройте файл конфигурации Squid и добавьте следующее правило:

root@debian-11-gateway:~# nano -c /etc/squid/squid.conf

```
acl block_keywords url_regex "/etc/squid/block_keywords.acl" (Строка ~1384)
# Добавляем правило со списком запрещенных ключевых слов.
http_access deny block_keywords
# Если пользователь пытается открыть сайты, в которых есть запрещенные
# ключевые слова, то прокси-сервер запрещает к ним доступ.
```

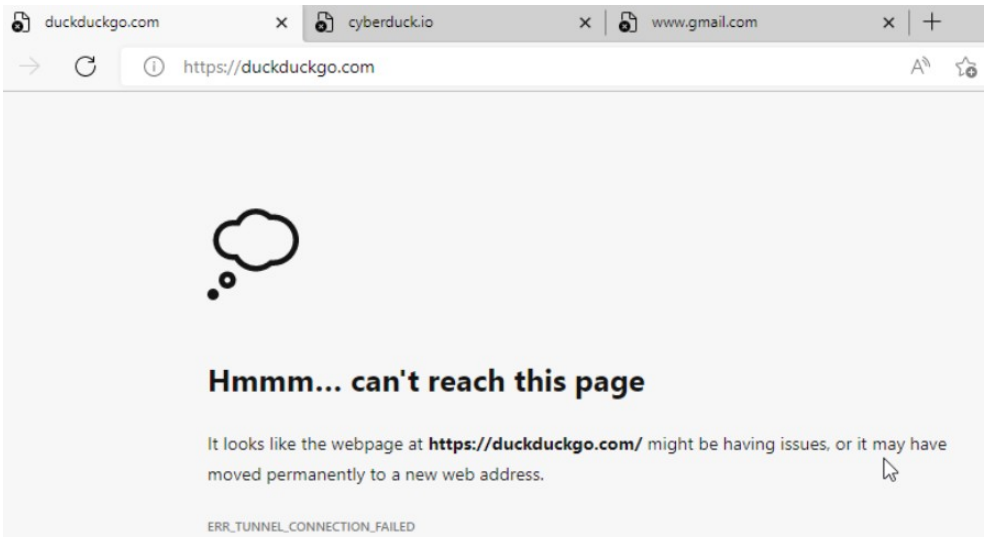
Теперь сохраните файл и перезапустите Squid:

```
root@debian-11-gateway:~# squid -k reconfigure
root@debian-11-gateway:~# systemctl restart squid
```

Обратите внимание, опять имеется разница в обработке сайтов на HTTPS и HTTP. Если поиск выполняется по части доменного имени, то данные правила будут работать в любом случае, но ошибки вновь будут отображаться по-разному. Поиск по пути URL за пределами доменного имени работает только для сайтов на HTTP. При использовании HTTPS весь трафик, в том числе адреса запрашиваемых страниц, передается в зашифрованном виде, поэтому возможность фильтровать отсутствует в принципе.

Проверка работы Squid-сервера с приведенными изменениями (2)

Убедимся, что настройки на наш прокси-сервер применились. Заходим в браузер Microsoft Edge и пытаемся открыть сайты duckduckgo.com, cyberduck.io и gmail.com.



ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №17

Тема: Развертывание веб-сервера – Apache

Цель работы: Сконфигурировать виртуальную машину в качестве веб-сервера на основе Apache:

– Обновление локального индекса пакетов;

– Установка пакета с программой Apache и его базовая настройка;

– Проверка веб-сервера;

– Изменение HTML-кода страницы;

– Изменение имени хоста (Hostname);

– Настройка сетевых интерфейсов и установка статических IP-адресов ;

– Изменение DNS-сервера по умолчанию;

– Проверка сетевых интерфейсов и доступа к Интернету;

– Проверка работы веб-сервера.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Apache HTTP Server – это бесплатное кроссплатформенное ПО и веб-сервер с открытым исходным кодом, выпущенный в соответствии с условиями Apache License 2.0. Apache разрабатывается и поддерживается открытым сообществом разработчиков под эгидой Apache Software Foundation.

Подавляющее большинство экземпляров HTTP-сервера Apache работают в дистрибутиве Linux, но текущие версии также работают в Microsoft Windows, OpenVMS и множестве UNIX-подобных систем.

Первоначально основанный на HTTPd-сервере NCSA, разработка Apache началась в начале 1995 года после того, как работа над кодом NCSA застопорилась. Apache сыграл ключевую роль в начальном росте World Wide Web, быстро обогнав NCSA HTTPd в качестве доминирующего HTTP-сервера. В 2009 году он стал первым веб-сервером, который обслуживал более 100 миллионов веб-сайтов.

По оценкам Netcraft, по состоянию на январь 2021 года Apache обслуживал 24,63% миллионов самых загруженных веб-сайтов, в то время как Nginx обслуживал 23,21%. В свою очередь веб-сервер Microsoft находился на третьем месте с 6,85%.

Входим на виртуальную машину Debian 11 – Server.

Обновление локального индекса пакетов

```
root@debian-11-server:~# su –
Пароль: guest
root@debian-11-server:~# apt update
```

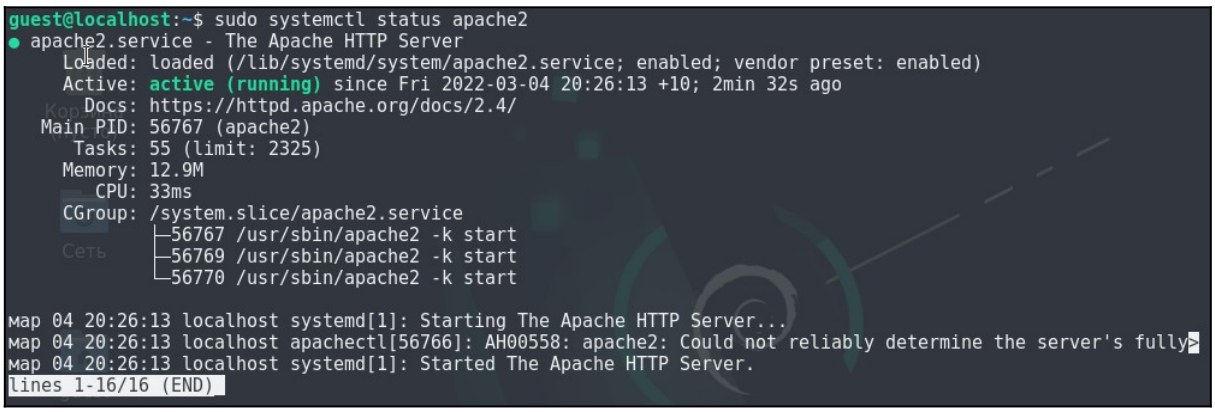
Установка пакета с программой Apache и его базовая настройка

```
root@debian-11-server:~# apt install apache2
```

Проверка веб-сервера

Используйте команду systemctl чтобы проверить работу службы:

```
root@debian-11-server:~# systemctl status apache2
```



Изменение HTML-кода страницы

Для ввода текста будем использовать текстовый редактор nano:

root@debian-11-server:~# rm /var/www/html/index.html
root@debian-11-server:~# touch /var/www/html/index.html
root@debian-11-server:~# nano /var/www/html/index.html

Чтобы изменить страницу по умолчанию необходимо зайти в директорию /var/www/html и вписать в файл index.html следующий текст:

```
<!DOCTYPE html>
<html>
	<head>
		<title>Привет, Мир!</title>
		<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
	</head>
	<body>
		<h1>Это заголовок!</h1>
		<p>Это параграф!</p>
	</body>
</html>
```

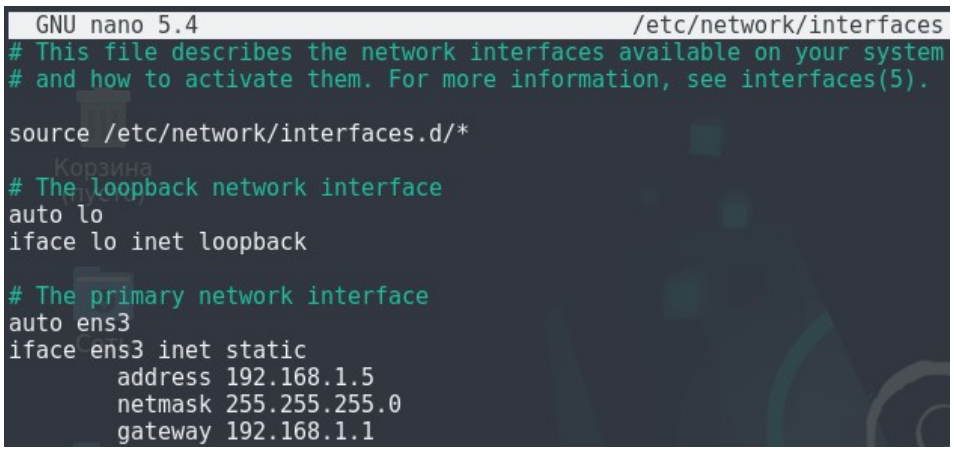
Входим на виртуальную машину Debian 11 – Host.

Изменение имени хоста (Hostname)

guest@localhost:~$ su -
Пароль: guest
root@localhost:~# hostnamectl set-hostname debian-11-host

Настройка сетевых интерфейсов и установка статических IP-адресов

root@debian-11-host:~# ip address
root@debian-11-host:~# nano /etc/network/interfaces



root@debian-11-host:~# systemctl disable avahi-daemon.service
root@debian-11-host:~# systemctl restart networking.service
root@debian-11-host:~# systemctl reboot

Также можно выключить службу connman, чтобы настройка статических IP-адресов применялась сразу после перезагрузки операционной системы:

root@debian-11-host:~# systemctl disable connman.service
root@debian-11-host:~# systemctl stop connman.service

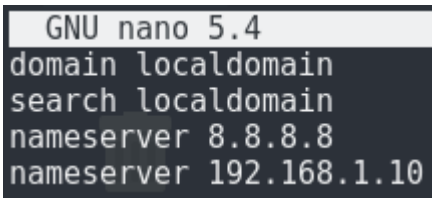
Если после перезагрузки виртуальной машины конфигурация не применилась, то снова перезагрузите службу networking.service с помощью команды:

root@debian-11-host:~# systemctl restart networking.service

Изменение DNS-сервера по умолчанию

Убедитесь, что в конфигурационном файле выставлены именно те настройки, что на рисунке ниже.

root@debian-11-host:~# nano /etc/resolv.conf

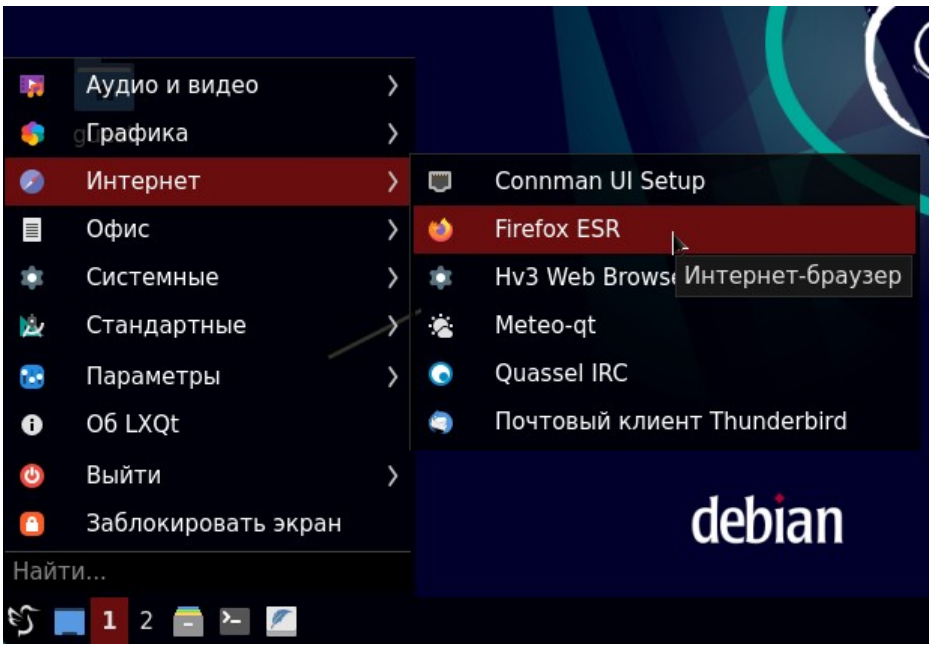


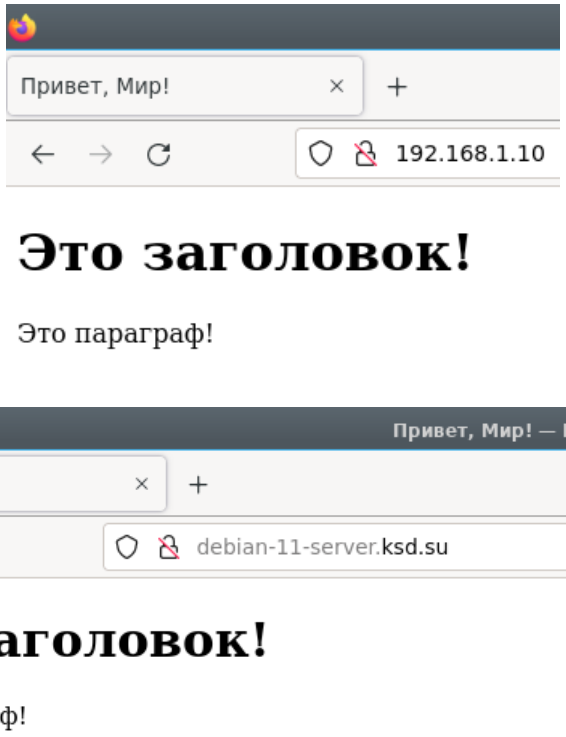
Проверка сетевых интерфейсов и доступа к Интернету

root@debian-11-server:~# ip address
root@debian-11-server:~# apt install sl
root@debian-11-server:~# login guest
Пароль: guest
guest@debian-11-server:~$ sl
guest@debian-11-server:~$ exit

Проверка работы веб-сервера

Необходимо зайти на веб-страницу нашего веб-сервера. Делается это следующим образом.





ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №18

Тема: Развертывание NTP-сервера

Цель работы: Сконфигурировать виртуальную машину в качестве NTP-сервера на основе Chrony:

– Обновление локального индекса пакетов;

– Установка пакета с программой Chrony и его базовая настройка;

– Проверка NTP-сервера и установка точного времени на Windows-10-Host.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

NTP обозначает Network Time Protocol и является стандартом для синхронизации времени между двумя устройствами по сети. Точное время имеет решающее значение для вычислений. Это гарантирует, что ваши журналы имеют правильную метку времени, что упрощает поиск и диагностику проблем, возникших в определенное время. Он также используется для многих методов безопасности, таких как двухфакторная аутентификация, когда токен действителен только в течение определенного времени. Использование NTP считается одним из лучших способов поддерживать точность системного времени. Единственным недостатком является то, что вам нужно поддерживать сетевое соединение. Он также стал важным элементом для устройств.

Входим на виртуальную машину Debian 11 – Server.

Обновление локального индекса пакетов

root@debian-11-server:~# su -
Пароль: guest
root@debian-11-server:~# apt update

Установка пакета с программой Chrony и его базовая настройка

root@debian-11-server:~# apt install chrony
root@debian-11-server:~# cat /etc/chrony/chrony.conf

Убедитесь, что работают представленные ниже команды по мониторингу работы Chrony.

root@debian-11-server:~# chronyc activity (Эта команда предоставит вам статус клиента NTP, работающего на вашем устройстве Linux)

root@debian-11-server:~# chronyc sources -v (Эта команда покажет статистику для источников, которые в настоящее время используются демоном chrony в качестве источника времени)

root@debian-11-server:~# chronyc tracking (Эта команда выведет на экран источники времен, которые Chrony использует в данный момент)

root@debian-11-server:~# chronyc clients (Эта команда покажет какие клиенты используют Chrony)

Убедитесь, что работают следующие Зайдите в конфигурационный файл Chrony и добавьте там следующие строки.

root@debian-11-server:~# nano /etc/chrony/chrony.conf

allow 192.168.1.0/24 (Эта строка разрешает устройствам с определенными IP-адресами или подсетями иметь доступ к NTP-серверу)

root@debian-11-server:~# systemctl restart chrony

Проверка NTP-сервера и установка точного времени на Windows-10-Host

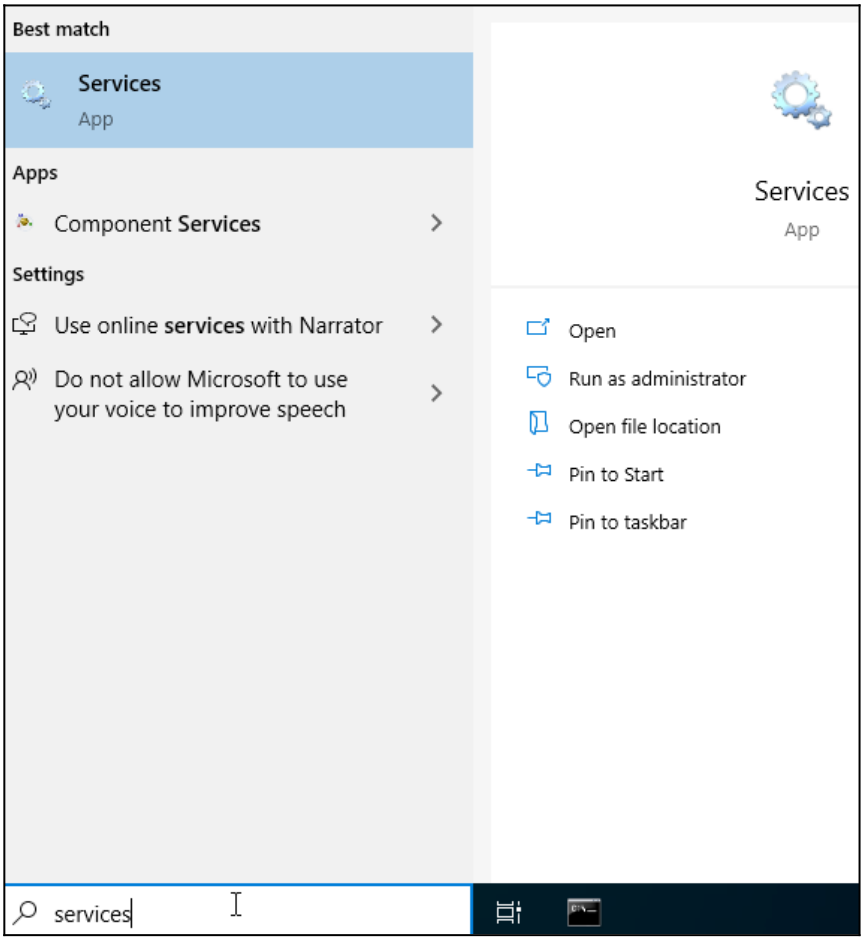
Входим на виртуальную машину Windows 10 – Host.

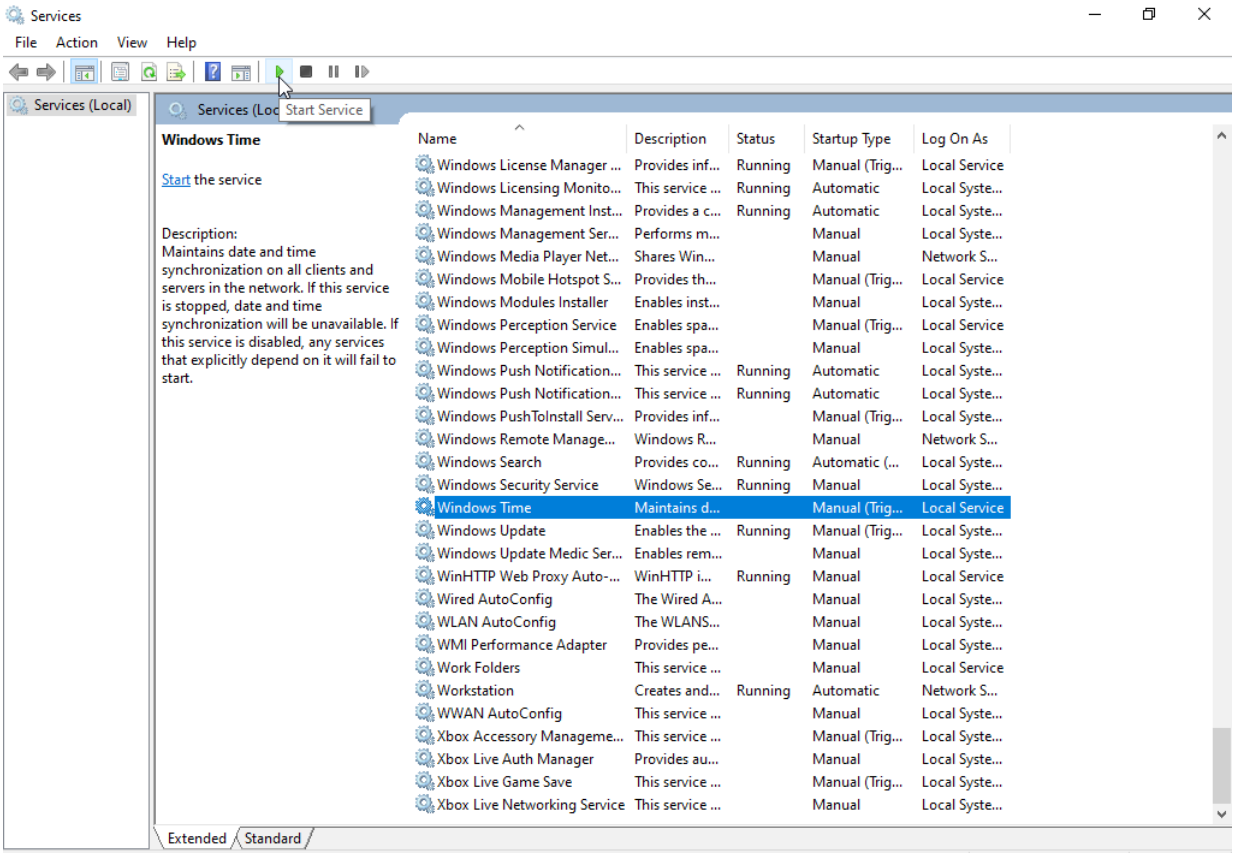
Утилита командной строки w32tm предназначена для удаления или установки службы времени Windows W32Time и управления ею на локальном или удаленном компьютере.

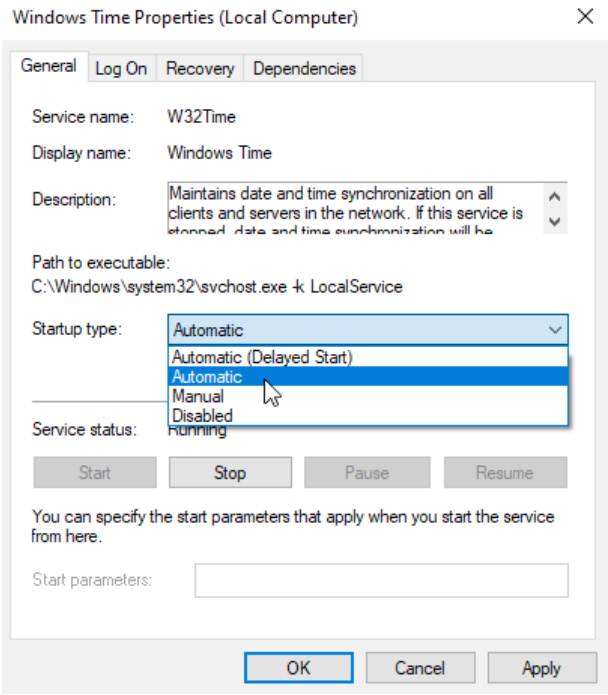
Выполните в командной строке Windows (cmd):

C:\Windows\system32>w32tm /query /status

Если в командной строке будет написать, что служба не запущена, то необходимо ее запустить.







Снова выполните в командной строке Windows (cmd):

C:\Windows\system32>w32tm /query /status
C:\Windows\system32>w32tm /config /manualpeerlist:"192.168.1.10" /syncfromflags:manual /update

Перезагрузите виртуальную машину и в последний раз выполните в командной строке Windows (cmd):

C:\Windows\system32>w32tm /query /status

Входим на виртуальную машину Debian 11 – Server.

Убедитесь, что Chrony добавил вашу виртуальную машину Windows-10-Host в список клиентов:

root@debian-11-server:~# chronyc clients (Эта команда покажет какие клиенты используют Chrony)

```
root@debian-11-server:~# chronyc clients
Hostname                      NTP   Drop Int IntL Last     Cmd   Drop Int  Last
===============================================================================
192.168.1.15                    3      0   8   -   191       0      0   -     -
```

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №19

Тема: Развертывание FTP-сервера

Цель работы: Сконфигурировать виртуальную машину в качестве FTP-сервера на основе vsFTPd:

– Обновление локального индекса пакетов;

– Установка пакета с программой vsFTPd и его базовая настройка;

– Обновление локального индекса пакетов;

– Установка пакета с FTP-клиентом FileZilla и проверка подключения к FTP-серверу с настройками по умолчанию;

– Создание пользователя anonymous;

– Создание и настройка входа с анонимным профилем, и конфигурация защищенного FTP-сервера;

– Тестирование входа с анонимным профилем;

– Добавление пользователю guest прав на запись и создание папок.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

FTP или File Transfer Protocol – это достаточно старый, но в то же время надежный и проверенный временем протокол выгрузки файлов на удаленный сервер или их скачивания. Также иногда этот протокол применяется веб-мастерами для управления файлами или организации хранилища данных.

Входим на виртуальную машину Debian 11 – Server.

Обновление локального индекса пакетов

root@debian-11-server:~# su -
Пароль: guest
root@debian-11-server:~# apt update

Установка пакета с программой vsFTPd и его базовая настройка

root@debian-11-server:~# apt install vsftpd

Когда установка будет завершена, вам необходимо включить сервис vsftpd, поскольку он не будет запущен по умолчанию, а также добавить службу в автозагрузку:

root@debian-11-server:~# systemctl start vsftpd
root@debian-11-server:~# systemctl enable vsftpd

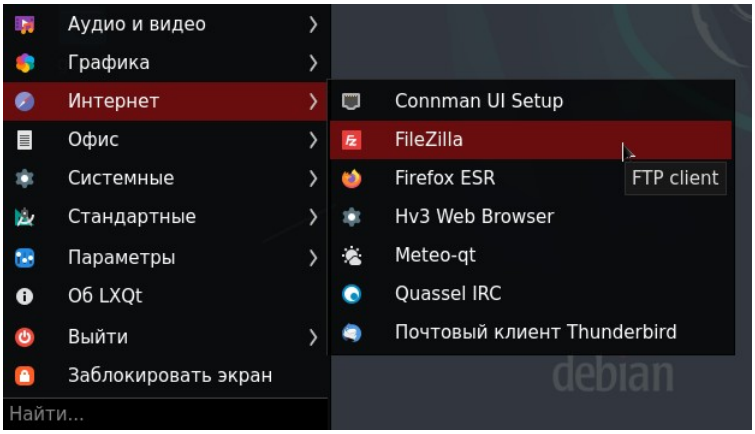
Входим на виртуальную машину Debian 11 – Host.

Обновление локального индекса пакетов

root@debian-11-host:~# su -
Пароль: guest
root@debian-11-host:~# apt update

Установка пакета с FTP-клиентом FileZilla и проверка подключения к FTP-серверу с настройками по умолчанию

root@debian-11-host:~# apt install filezilla



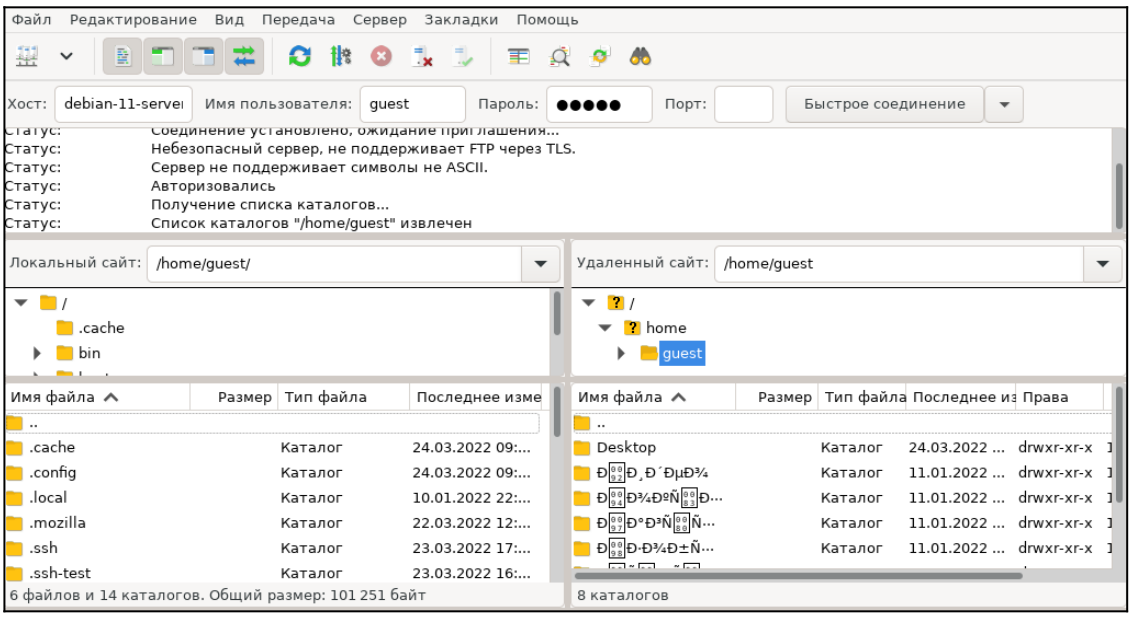
По умолчанию доступно подключение только через аутентифицированного пользователя. Чтобы войти введите следующие значения в поля ниже:

– Хост: Debian-11-Server;

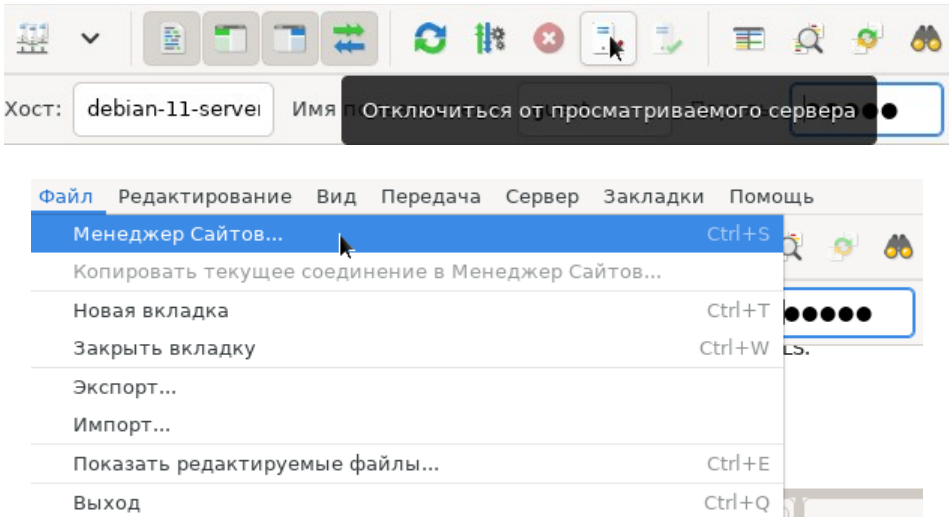
– Имя пользователя: guest;

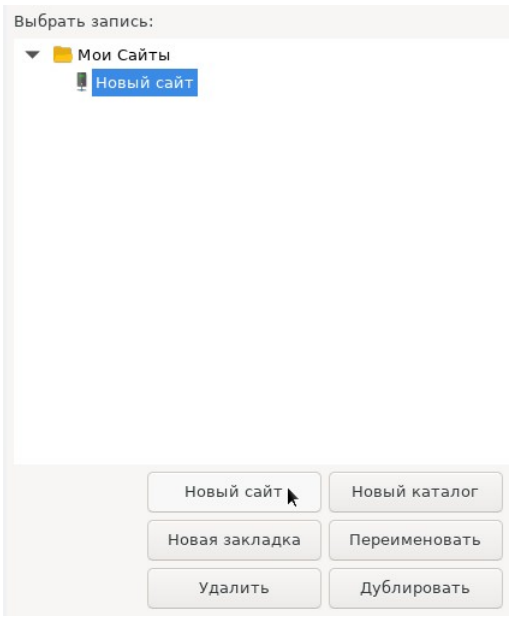
– Пароль: guest;

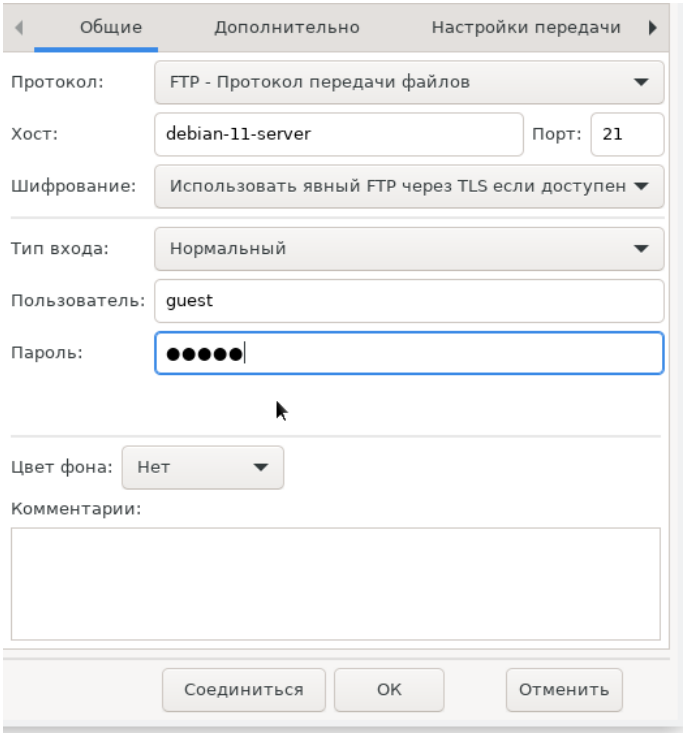
– Порт: 21 (По умолчанию).

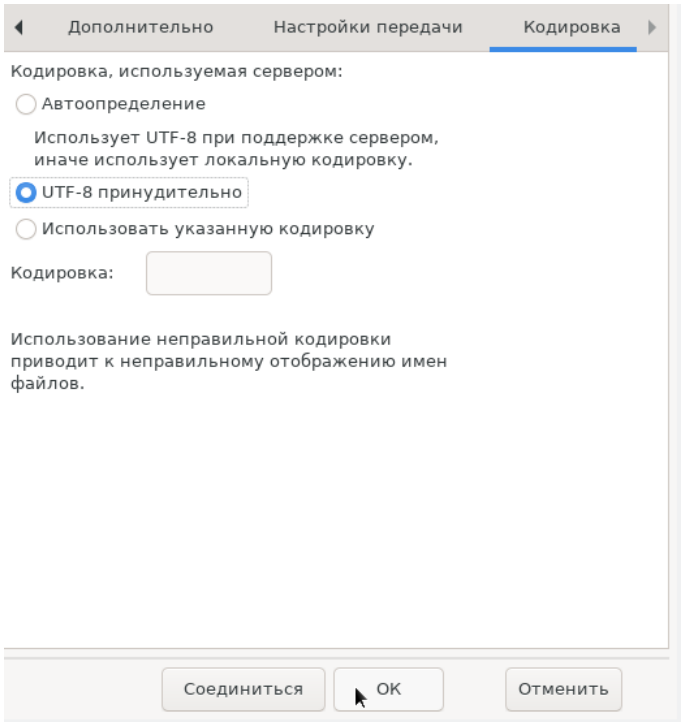


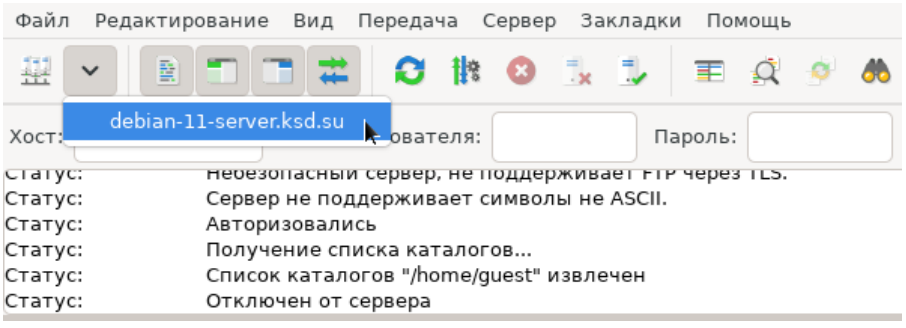
Мы видим, что FTP-сервер работает, но у нас возникла ошибка с кодировкой папок и файлов. Чтобы это исправить, необходимо выполнить следующие действия:

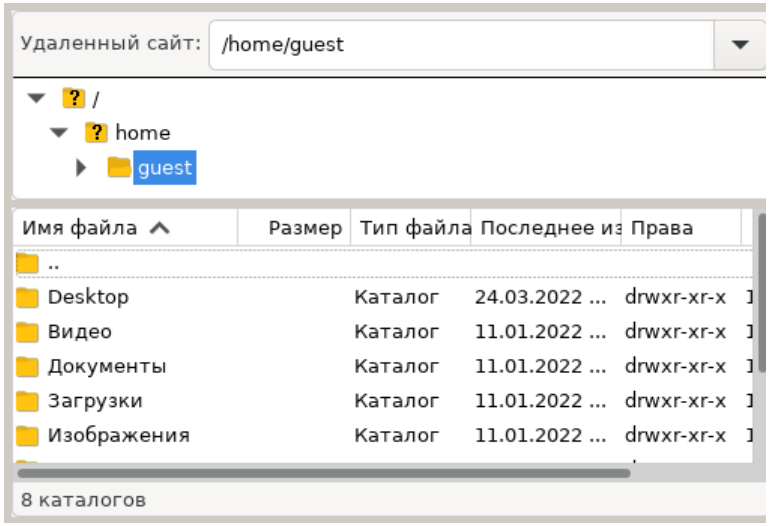





Общие
Дополнительно
Настройки передачи
Протокол:
FTP - Протокол передачи файлов
Хост:
debian-11-server
Порт:
21
Шифрование:
Использовать явный FTP через TLS если доступен
Тип входа:
Нормальный
Пользователь:
guest
Пароль:
Цвет фона:
Нет
Комментарии:
Соединиться
OK
Отменить


Дополнительно
Настройки передачи
Кодировка
Кодировка, используемая сервером:
Автоопределение
Использует UTF-8 при поддержке сервером, иначе использует локальную кодировку.
UTF-8 принудительно
Использовать указанную кодировку
Кодировка:
Использование неправильной кодировки приводит к неправильному отображению имен файлов.
Соединиться
OK
Отменить


Файл
Редактирование
Вид
Передача
Сервер
Закладки
Помощь
debian-11-server.ksd.su
Хост:
Пароль:
Статус:
Небезопасный сервер, не поддерживает FTP через TLS.
Статус:
Сервер не поддерживает символы не ASCII.
Статус:
Авторизовались
Статус:
Получение списка каталогов...
Статус:
Список каталогов "/home/guest" извлечен
Статус:
Отключен от сервера


Удаленный сайт: /home/guest
/
home
guest
Имя файла
Размер
Тип файла
Последнее из
Права
..
Desktop Каталог 24.03.2022 ... drwxr-xr-x
Видео Каталог 11.01.2022 ... drwxr-xr-x
Документы Каталог 11.01.2022 ... drwxr-xr-x
Загрузки Каталог 11.01.2022 ... drwxr-xr-x
Изображения Каталог 11.01.2022 ... drwxr-xr-x
8 каталогов

Входим на виртуальную машину Debian 11 – Server.

Создание пользователя anonymous

Так как для работы FTP-сервера с анонимным пользователя требуется собственно анонимный пользователь, то имеет смысл его создать. Делается это следующим образом:

root@debian-11-server:~# adduser anonymous

Также не забудем создать каталоги, которые мы написали в конфигурационный файл в качестве директорий для анонимных пользователей и всех остальных по умолчанию:

root@debian-11-server:~# mkdir /home/guest/share
root@debian-11-server:~# touch /home/guest/share/test.txt (Создать файл для дальнейшей проверки работы FTP-сервера)
root@debian-11-server:~# echo "Share" > /home/guest/share/test.txt
root@debian-11-server:~# mkdir /home/anonymous/anonymous-share
root@debian-11-server:~# touch /home/anonymous/anonymous-share/test.txt
root@debian-11-server:~# echo "Anonymous-Share" > /home/anonymous/anonymous-share/test.txt

Создание и настройка входа с анонимным профилем, и конфигурация защищенного FTP-сервера

Теперь перейдем к настройке FTP-сервера. Нам нужно поменять несколько параметров, чтобы полностью защитить ваш FTP-сервер и добавить возможность подключения с анонимным профилем. Сначала необходимо скопировать оригинальный файл настроек, чтобы в случае проблем вернуть все как было:

root@debian-11-server:~# cp /etc/vsftpd.conf /etc/vsftpd.conf.old

Затем откройте файл в редакторе:

root@debian-11-host:~# nano -c /etc/vsftpd.conf

Затем добавьте такие настройки. Вам нужно будет найти и изменить значения указанных строк, добавлять новые, если они уже есть, не стоит. Сначала указываем, что нужно ожидать входящих соединений:

listen=YES (Строка 14)

Затем выключаем прослушивание IPv6-протокола за ненадобностью:

listen_ipv6=NO (Строка 22)

Затем разрешаем анонимный вход:

anonymous_enable=YES (Строка 25)

Разрешает или запрещает вход анонимных пользователей. Если разрешено, пользователи с именами ftp и anonymous распознаются как анонимные пользователи.

Разрешаем использовать имена локальных пользователей для входа:

local_enable=YES (Строка 28)

Для авторизованных пользователей разрешаем команды, позволяющие изменять файловую систему:

write_enable=YES (Строка 31)

Установим значение umask[14] для новых файлов, создаваемых по FTP:

local_umask=022 (Строка 35)

Также нужно использовать порт 20 для передачи данных вместо случайного. Это необходимо для нормальной работы UFW, который мы будем настраивать в дальнейшем:

connect_from_port_20=YES (Строка 60)

Отключаем строку chroot_local_user, чтобы локальные пользователи по умолчанию не переносились в chroot[15] в их домашнем каталоге после входа:

chroot_local_user=NO (Строка 122)

Включаем строку chroot_list_enable, чтобы использовать список локальных пользователей помещаемых в chroot() тюрьму в их домашнем каталоге после входа. Если используется совместно с включенным chroot_local_user означает список пользователей, которые не помещаются в chroot тюрьму. По умолчанию список содержится в файле /etc/vsftpd.chroot_list:

chroot_list_enable=YES (Строка 123)

Укажем явно путь к файлу, содержащему список локальных пользователей, которые будут перемещены в chroot тюрьму их домашние каталоги при входе:

chroot_list_file=/etc/vsftpd.chroot_list (Строка 125)

Имейте в виду, что эта опция уместна только при разрешенной chroot_list_enable. Если опция chroot_local_user включена, в таком случае, файл становится списком пользователе, которые не помещаются в chroot.

Убедитесь, что включено использование PAM-библиотек:

pam_service_name=vsftpd (Строка 145)

Раскомментируем строку utf8_filesystem=YES, чтобы явно указать, что используется в файловой системе используется кодировка UTF-8:

utf8_filesystem=YES (Строка 155)

Разрешим аутентификацию только пользователей, перечисленных в файле userlist:

userlist_enable=YES

Указываем файл с нашими виртуальными пользователями:

---

[14] umask – это функция среды POSIX, изменяющая права доступа, которые присваиваются новым файлам и каталогам по умолчанию.

[15] chroot – это операция изменения корневого каталога в UNIX-подобных операционных системах. Программа, запущенная с измененным корневым каталогом, будет иметь доступ только к файлам, содержащимся в данном каталоге.

```
userlist_file=/etc/vsftpd.userlist
```

По умолчанию таким пользователям запрещен вход в систему, но мы хотим совсем обратное, поэтому добавьте такую строчку:

```
userlist_deny=NO
```

Необходимо указать каталог, в который vsftpd будет переводить всех пользователей после входа по умолчанию:

```
user_sub_token=$USER ($USER добавляет переменную $USER, в которой содержится имя пользователя)
local_root=/home/guest/share
```

В свою очередь нужно указать каталог, в который будут переводить анонимных пользователей после входа:

```
anon_root=/home/anonymous/anonymous-share
```

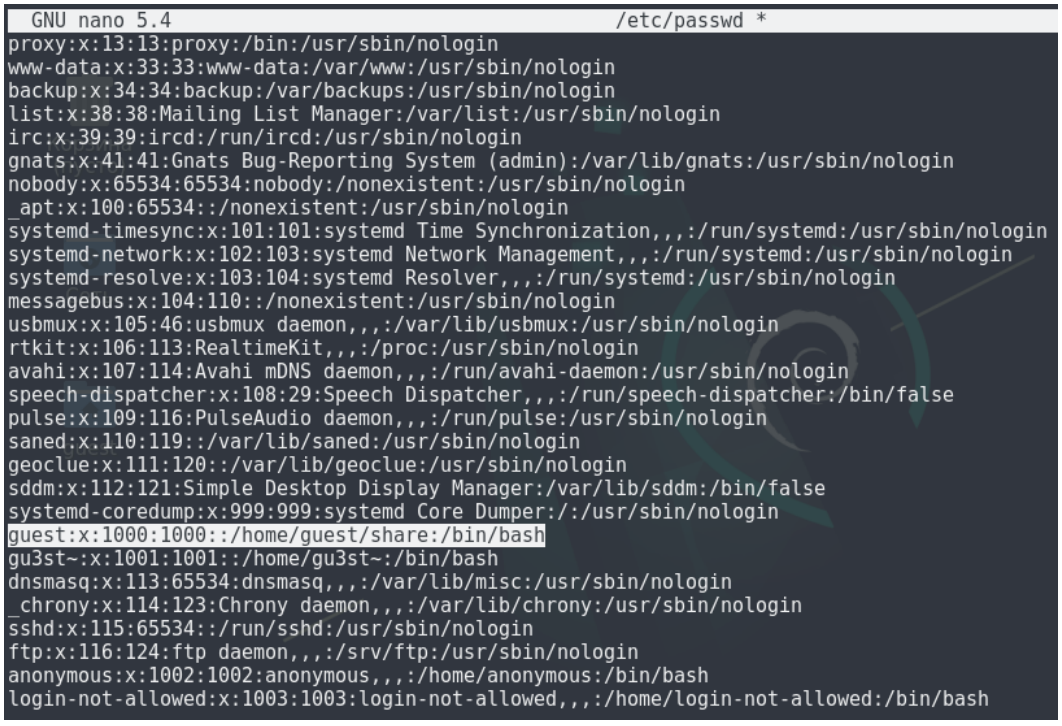
Чтобы vsftpd не спрашивал пароль у анонимных пользователей, позволяя им подключаться сразу, необходимо раскомментировать следующую строку:

```
no_anon_password=YES
```

Чтобы при входе на FTP-сервер с гостевого пользователя (guest) попадать сразу в расшаренную папку, необходимо прописать в конфигурационном файле /etc/passwd путь /home/guest/share как домашнюю папку, вместо /home/guest:

root@debian-11-server:~# nano -c /etc/passwd

```
guest:x:1000:1000::/home/guest/share:/bin/bash
```



Убедитесь, что гостевой аккаунт (guest) сменил домашнюю папку по умолчанию. Если вы увидите, что по умолчанию попали в директорию /home/guest/share и слева от значка доллара ($) находиться знак тильда (~), обозначающий домашний каталог текущего пользователя, то вы все сделали правильно.

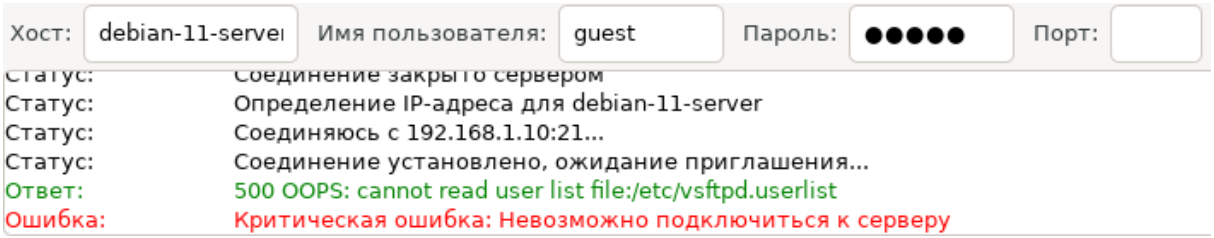
Настройка FTP-сервера почти завершена, сохраните изменения в конфигурационном файле и перезапустите vsftpd:

root@debian-11-server:~# systemctl restart vsftpd

Входим на виртуальную машину Debian 11 – Host.

Тестирование входа с анонимным профилем

Сначала войдите в программу FileZilla и убедитесь, что вы не можете авторизироваться на FTP-сервере используя учетные записи пользователя guest и анонимного пользователя, так как этих пользователей нет в конфигурационном файле vsftpd.userlist.



Входим на виртуальную машину Debian 11 – Server.

Поскольку мы хотим подключаться от имени пользователей guest и anonymous к FTP-серверу, то нам нужно добавить их в vsftpd.userlist:

root@debian-11-server:~# nano -c /etc/vsftpd.userlist

```
guest
anonymous
```

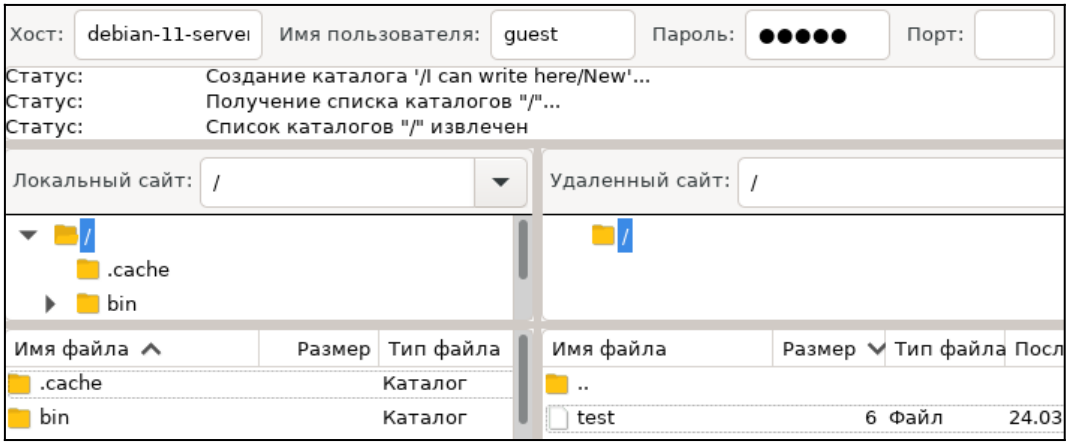
Также добавьте пользователей guest и anonymous в список локальных пользователей, которые будут перемещены в chroot() vsftpd.chroot_list:

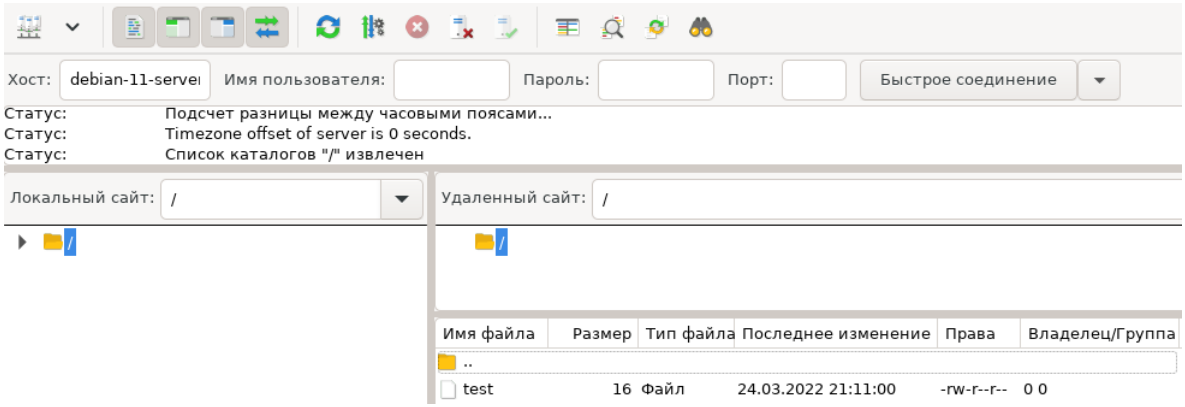
root@debian-11-server:~# nano -c /etc/vsftpd.chroot_list

guest anonymous

Входим на виртуальную машину Debian 11 – Host.

Снова войдите в программу FileZilla и попробуйте авторизироваться на FTP-сервере используя учетные записи пользователя guest и анонимного пользователя. Обратите внимание, что из-за использования chroot тюрьмы, /home/guest/share и /home/anonymous/anonymous-share в FTP-сервере являются корнями FTP-соединений.





Входим на виртуальную машину Debian 11 – Server.

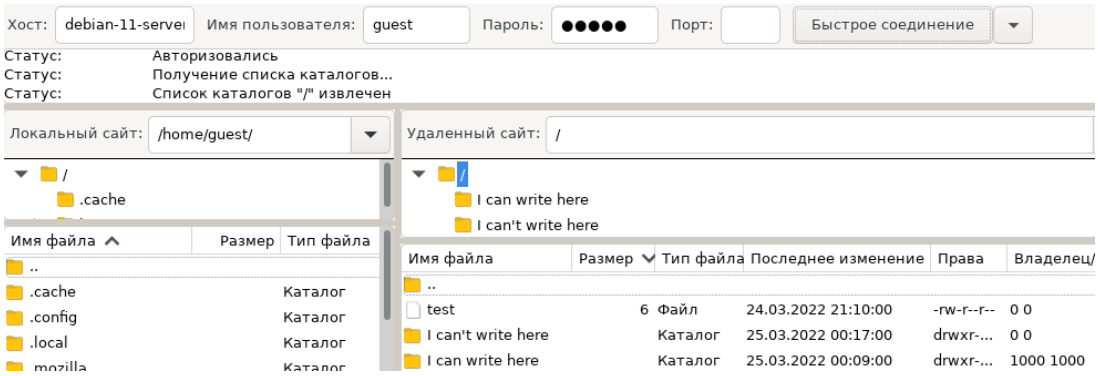
Добавление пользователю guest прав на запись и создание папок

Создадим каталоги для хранения файлов и передадим пользователю guest права собственности на них:

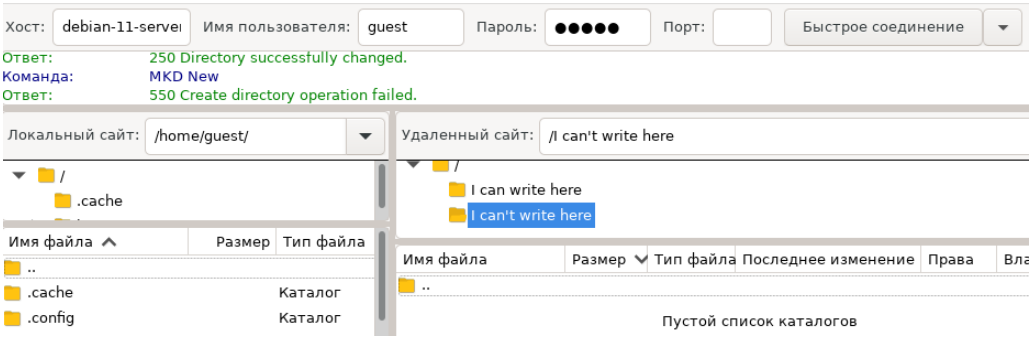
```
root@debian-11-server:~# mkdir /home/guest/share/"I can write here"
root@debian-11-server:~# mkdir /home/guest/share/"I can't write here"
root@debian-11-server:~# chown guest:guest /home/guest/share/"I can write here"
root@debian-11-server:~# chown root:root /home/guest/share/"I can't write here"
root@debian-11-server:~# ls -la /home/guest/share
```

Входим на виртуальную машину Debian 11 – Host.

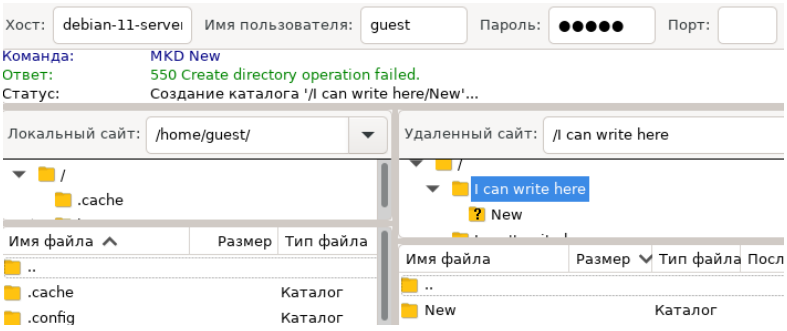
Еще раз войдите в программу FileZilla и попробуйте авторизироваться на FTP-сервере используя учетные записи пользователя guest:



Убедитесь, что вы не можете создать свой каталог, находясь в каталоге «I can’t write here»:



А теперь создайте свой каталог, находясь в каталоге «I can write here»:



ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №20

Тема: Развертывание Samba в режиме файлового сервера

Цель работы: Сконфигурировать виртуальную машину в качестве файлового сервера на основе Samba:

– Обновление локального индекса пакетов;

– Установка пакета программ Samba и его базовая настройка;

– Создание пользователя share;

– Подготовка пользовательской директории;

– Создание пользователя private;

– Управление пользователями и настройка доступа к директориям;

– Настройка конфигурационного файла smb.conf;

– Проверка работы Samba-сервера через аутентифицированных пользователей;

– Создание и под готовка гостевой сетевой директории;

– Проверка работы гостевой сетевой директории Samba-сервера.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Samba – это файловый сервер TCP/IP, при помощи которого компьютеры могут обмениваться файлами или управлять заданиями на печать. Он поддерживает любые реализации клиентов SMB/CIFS. Огромным плюсом Samba-сервера является то, что он позволяет использовать смешанные сети из Windows и Linux компьютеров вместе, не требуя при этом отдельного сервера Windows.

Входим на виртуальную машину Debian 11 – Server.

Обновление локального индекса пакетов

root@debian-11-server:~# su -
Пароль: guest
root@debian-11-server:~# apt update

Установка пакета программ Samba и его базовая настройка

root@debian-11-server:~# apt install samba

Создание пользователя share

Добавляем нового пользователя с отключенным shell, одноименной группой и без домашней директории:

root@debian-11-server:~# useradd smb -U -s /sbin/nologin (-U Создать одноименную группу; -s Указывает на отключенный shell)
root@debian-11-server:~# smbpasswd -a smb (Задать пользователю пароль для Samba)

Подготовка пользовательской директории

root@debian-11-server:~# mkdir -p -m 770 /mnt/data/smb/{share,private} (-p Создать родительские каталоги, если необходимо; -m Установить режим разрешения (как в chmod)
root@debian-11-server:~# chgrp[16] -R[17] smb /mnt/data/smb/ (Иначе не будут работать права на директории для разных пользователей)

[16] chgrp – это утилита UNIX, которая может использоваться непривилегированными пользователями для изменения группы файлов. В отличие от команды chown, chgrp позволяет рядовым пользователям изменять группы, но только те, членами которых они являются.

[17] Ключ -R в данном случае значит рекурсивно, включая подкаталоги.

root@debian-11-server:~# chmod 775 /mnt/data/smb/

Создание пользователя private

Добавляем еще одного нового пользователя:

root@debian-11-server:~# useradd private -U -s /sbin/nologin
root@debian-11-server:~# chgrp private /mnt/data/smb/private/
root@debian-11-server:~# smbpasswd -a private
root@debian-11-server:~# id private (Проверить в какие группы входит пользователь private)

Теперь, пользователи smb и private будут иметь доступ только к своим директориям «share» и «private».

Управление пользователями и настройка доступа к директориям

Существуют следующие дополнительные команды по управления пользователями (Команды даны для примера, вводить их не нужно):

root@debian-11-server:~# smbpasswd -e %username% (Включить пользователя)
root@debian-11-server:~# smbpasswd -d %username% (Отключить пользователя)
root@debian-11-server:~# smbpasswd -x %username% (Удалить пользователя)

Выполним следующие настройки, чтобы пользователь «private» также имел доступ к директории «share»:

root@debian-11-server:~# usermod -a -G smb private (Добавить пользователя private в группу smb)
root@debian-11-server:~# chgrp smb /mnt/data/smb/private
root@debian-11-server:~# id private (Проверить в какие группы входит пользователь private)
root@debian-11-server:~# gpasswd[18] -d private smb (-d Удалить пользователя private из группы smb)
root@debian-11-server:~# systemctl restart smbd
root@debian-11-server:~# systemctl restart nmbd
root@debian-11-server:~# systemctl status smbd
root@debian-11-server:~# systemctl status nmbd
root@debian-11-server:~# systemctl reboot

[18] Программа gpasswd редактирует пароли групп пользователей. В каждой группе могут быть определены администраторы, члены и пароль. Пароли групп имеют врожденную проблему с безопасностью, так как пароль знает более одного человека. Однако, группы являются полезным инструментом совместной работы различных пользователей.

Настройка конфигурационного файла smb.conf

root@debian-11-server:~# cp /etc/samba/smb.conf /etc/samba/smb.conf.old
root@debian-11-server:~# nano -c /etc/samba/smb.conf

```
read only = no (Строка 175)
```

```
[global] (Строка 24)
   server string = Samba-Server
   workgroup = WORKGROUP
   min protocol = SMB2
   netbios name = smb
   netbios aliases = smb-share smb-private

# Securtity (Строка 32)
security = user
passdb backend = tdbsam
create mask = 0660
directory mask = 0770

# Logging (Строка 38)
log file = /var/log/samba/log.%m
max log size = 1000
logging = file
log level = 1

#### Networking #### (Строка 44)
;   interfaces = ens3 (Строка 49)
;   bind interfaces only = yes (Строка 57)
hosts allow = 192.168.1.0/24, 172.16.1.0/24
```

```
[share] (Строка 250)
   path = /mnt/data/smb/share
   valid users = smb
   guest ok = no
   force user = smb
   browseable = yes
   read only = no
   create mask = 0777 (Права пользователей в этой директории)

[private]
   path = /mnt/data/smb/private
   valid users = private
   guest ok = no
   force user = private
   browseable = yes
   read only = no
   create mask = 0777
```

Можно разрешить как в секции «[global]», так и конкретно на конкретный сетевой ресурс, например, «[private]». Права на сетевой ресурс переопределяют глобальные права:

root@debian-11-server:~# nano -c /etc/samba/smb.conf

acl allow execute always = yes (Строка 58)

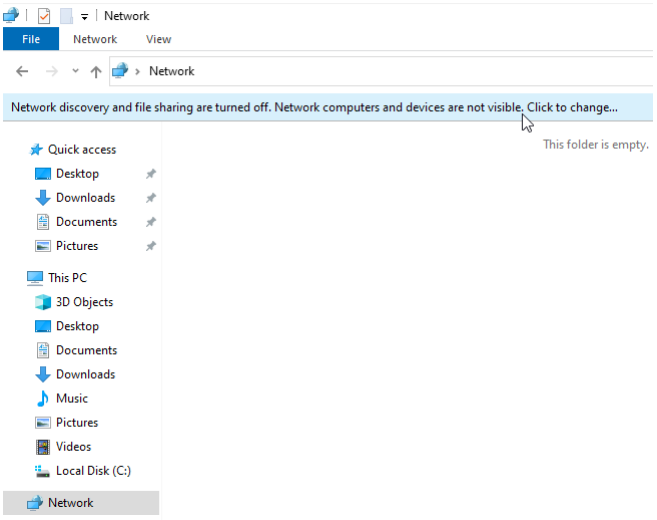
root@debian-11-server:~# testparm (Проверка работоспособности конфигурационного файла)

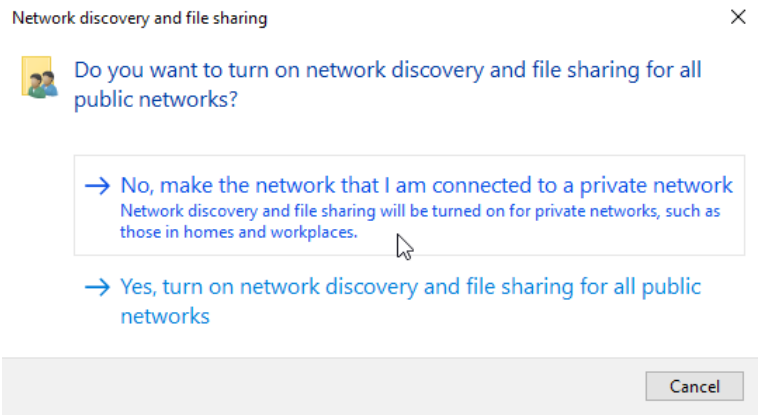
root@debian-11-server:~# systemctl restart smbd
root@debian-11-server:~# systemctl restart nmbd
root@debian-11-server:~# systemctl status smbd
[ctrl+c]
root@debian-11-server:~# systemctl status smbd
[ctrl+c]
root@debian-11-server:~# watch smbstatus

Входим на виртуальную машину Windows 10 – Host.

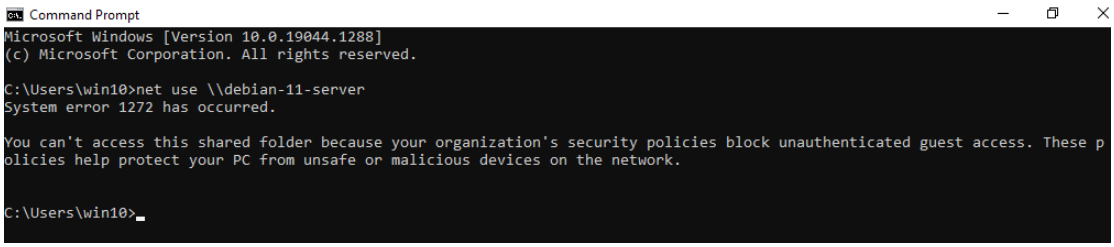
Проверка работы Samba-сервера через аутентифицированных пользователей

Включите обнаружение сети в Windows 10 в проводнике:



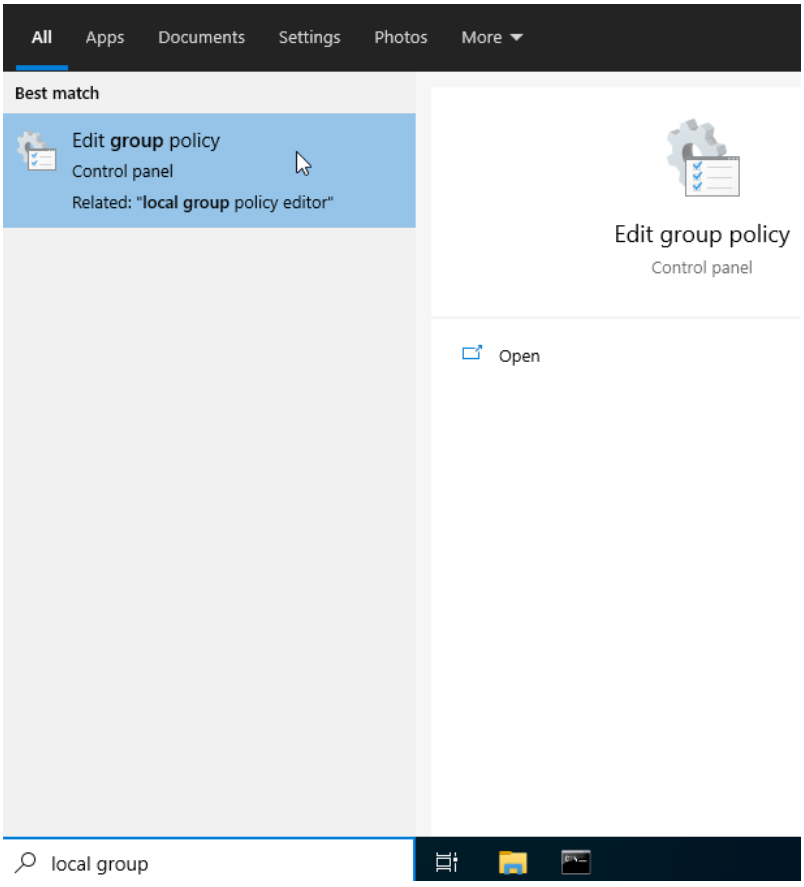


Попробуйте подключиться к Samba-серверу, используя консольную утилиту net use:



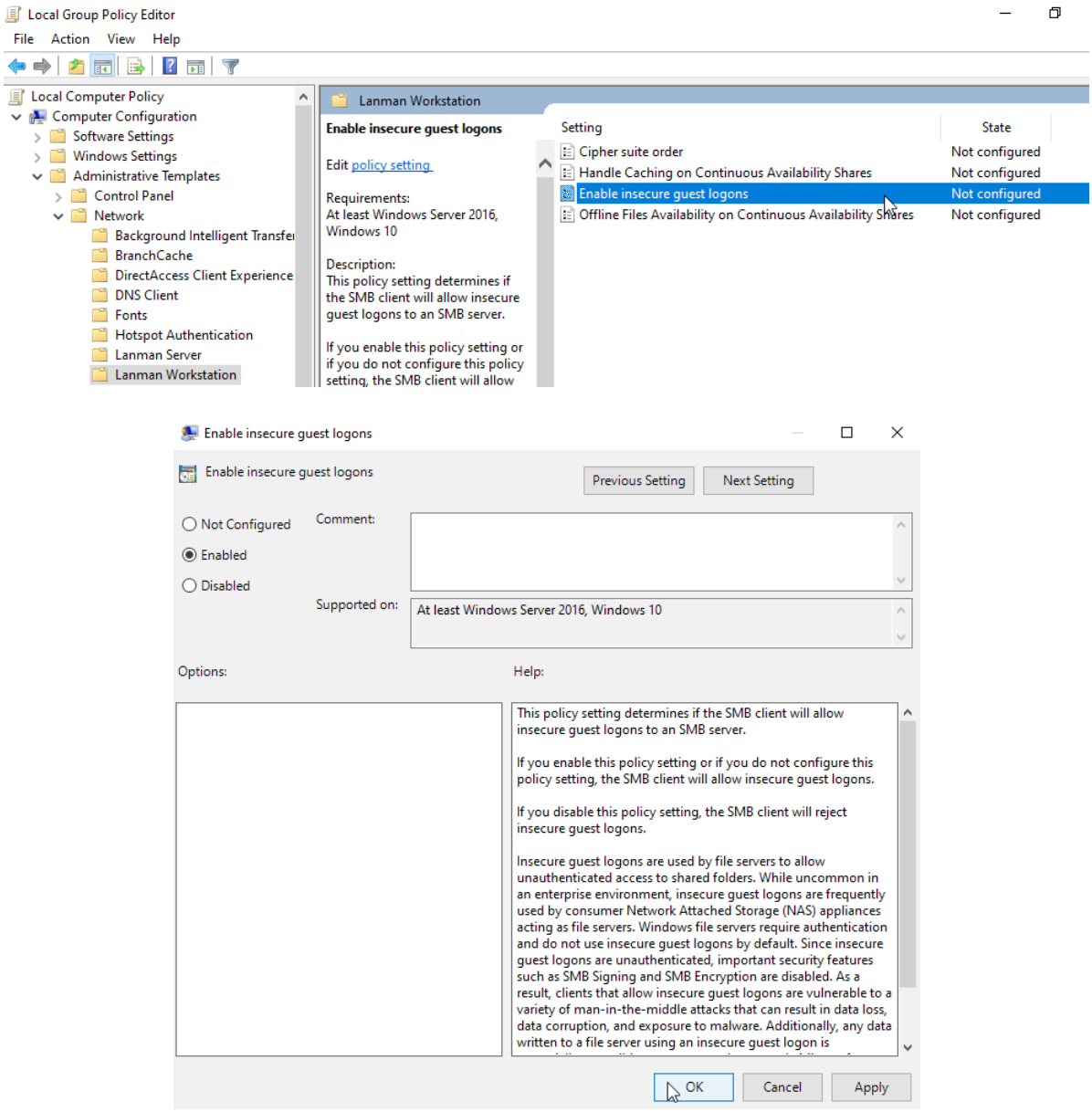
К сожалению, подключиться не удалось – произошла системная ошибка 1272. Эта ошибка связана с ограничениями локальной групповой политики Windows, не разрешающими доступ к «небезопасным гостевым входам». Похоже, Microsoft решила отключить доступ по умолчанию в новых обновлениях Windows из-за проблем с безопасностью.

Чтобы исправить эту ошибку, необходимо выполнить некоторые настройки в реестре Windows:



После открытия редактора локальной групповой политики перейдите по этому пути:

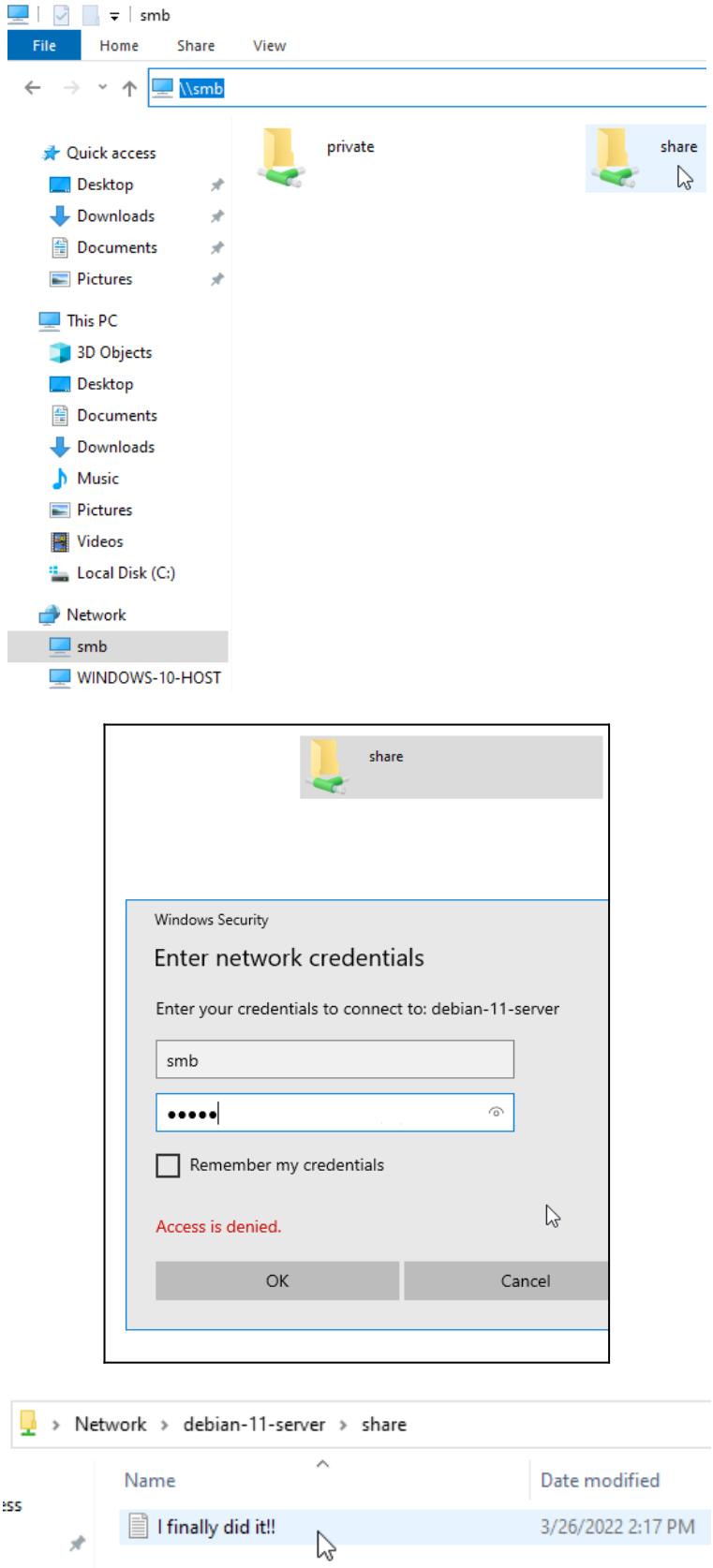
Computer configuration > Administrative Templates > Network > Lanman Workstation



Удалите все предыдущие соединения к Samba-серверу, используя консольную утилиту net use, так как они могут помешать его работе:

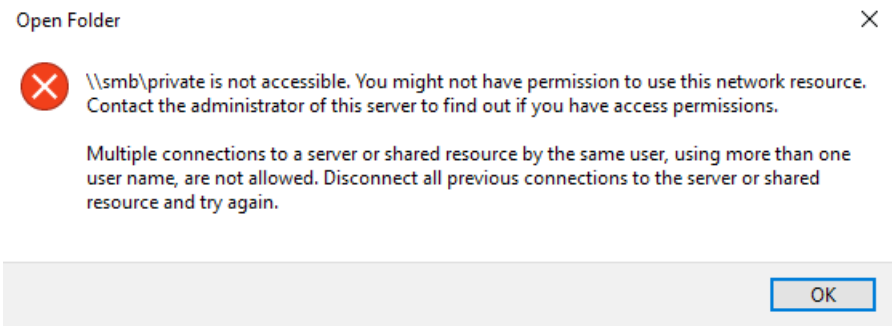
C:\Users\win10>net use * /delete

Далее зайдите в проводник, попробуйте войти в папку share, авторизуйтесь и создайте тестовый файл, чтобы проверить, что файловый сервер работает:

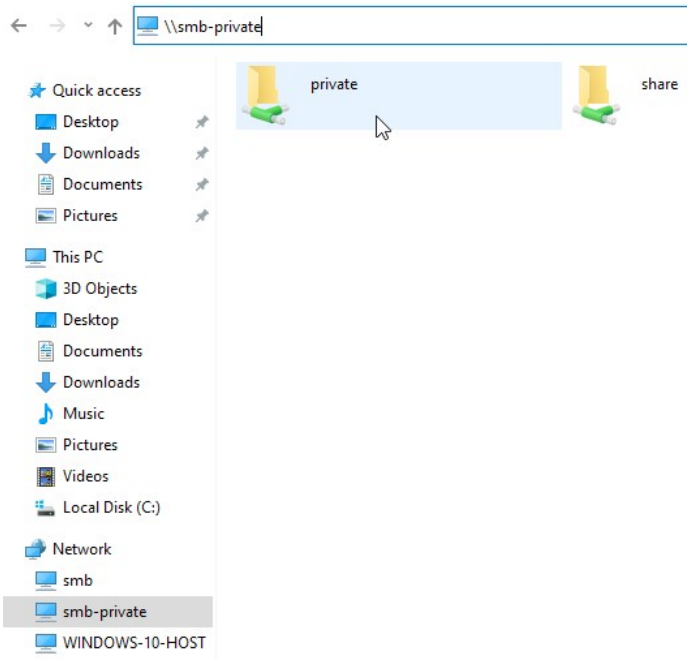


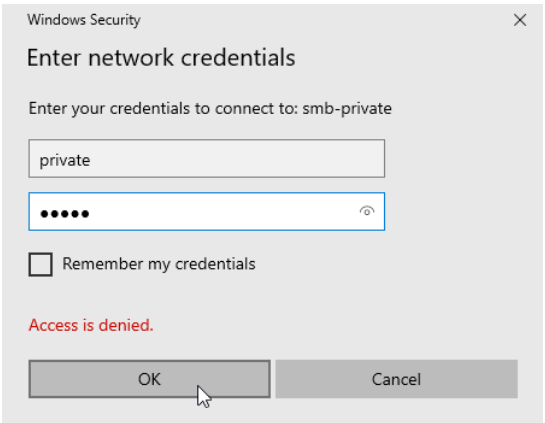
Обратите внимание, что в Windows 10 множественные подключения к серверу или общему ресурсу одним и тем же пользователем с использованием более одного имени пользователя не допускаются. Отключите все предыдущие подключения к серверу или общему ресурсу и повторите попытку.

Если у вас возникают проблемы с доступом в сетевые ресурсы, то проверьте, что ваши пользователи являются владельцами своих сетевых папок (Команда chgrp smb *<Путь_До_Вашей_Папки>*).



Попробуйте подключиться к сетевой папке private, используя учетные данные уже другого пользователя. Сделать это можно либо предварительно стерев все предыдущие подключения, либо используя другие псевдонимы (alias) файлового сервера, которые мы предварительно настроили в конфигурационном файле Samba-сервера.





Входим на виртуальную машину Debian 11 – Server.

Создание и под готовка гостевой сетевой директории

root@debian-11-server:~# mkdir -p -m 777 /mnt/data/smb/free-to-use
root@debian-11-server:~# nano -c /etc/samba/smb.conf

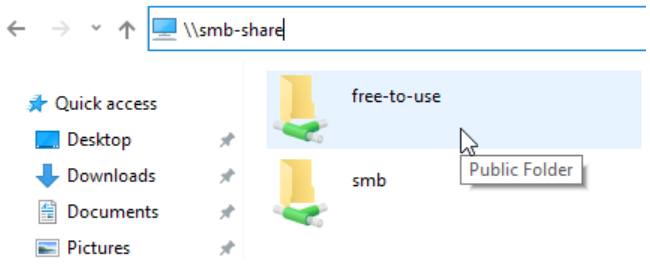
```
[global] (Строка 24)
map to guest = Bad Password
…
[free-to-use] (Строка 272)
   comment = Public Folder (Комментарий для удобства использования)
   path = /mnt/data/smb/free-to-use (Путь на сервере, где будут храниться данные)
   public = yes (Для общего доступа. Установите в yes, если хотите, чтобы все могли работать с ресурсом)
   writable = yes (Разрешает запись на сетевой ресурс)
   read only = no (Только для чтения. Установите no, если у пользователей должна быть возможность создавать папки и файлы)
   guest ok = yes (Разрешает доступ к папке гостевой учетной записи)
   create mask = 0777 (При создании новой папки или файла назначаются полные права)
   directory mask = 0777
   force create mode = 0777
   force directory mode = 0777
```

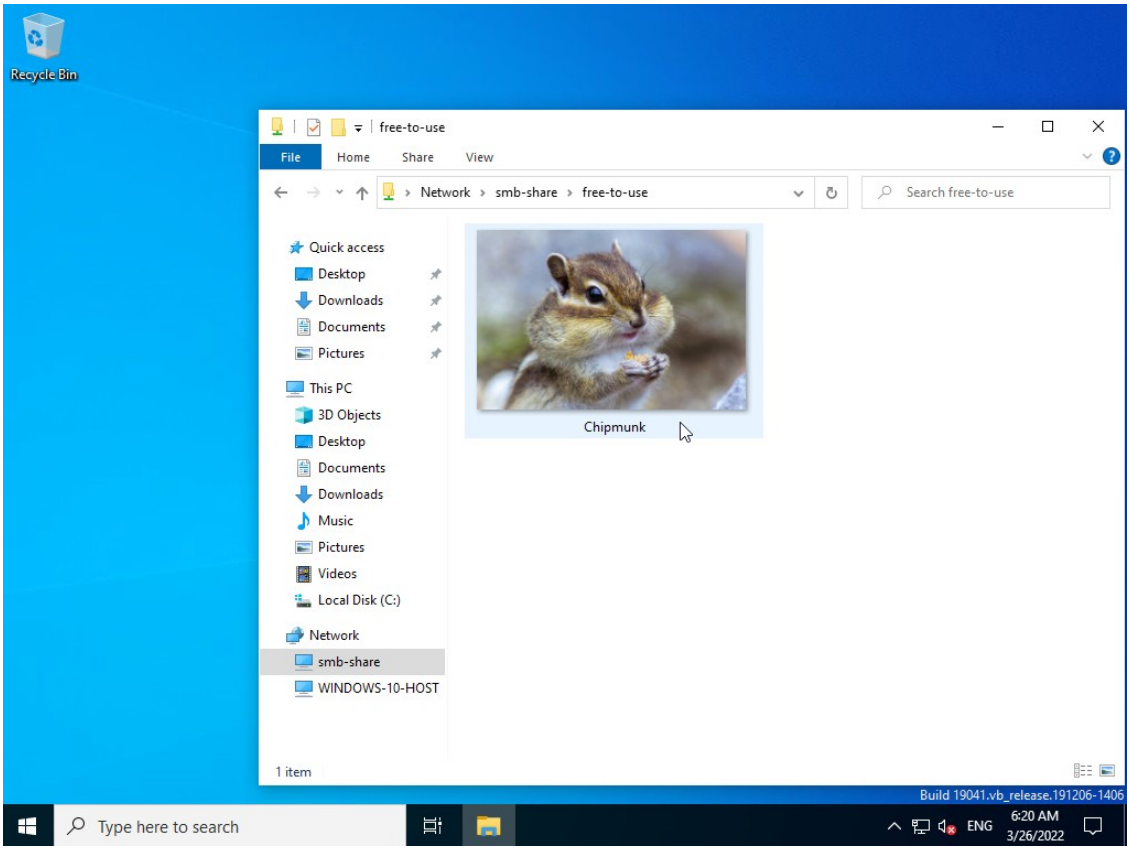
root@debian-11-server:~# testparm (Проверка работоспособности конфигурационного файла)
root@debian-11-server:~# systemctl restart smbd
root@debian-11-server:~# systemctl restart nmbd
root@debian-11-server:~# systemctl status smbd
[ctrl+c]
root@debian-11-server:~# systemctl status smbd
[ctrl+c]
root@debian-11-server:~# watch smbstatus

Входим на виртуальную машину Windows 10 – Host.

Проверка работы гостевой сетевой директории Samba-сервера

Зайдите в проводник, попробуйте войти в папку free-to-use и создайте тестовый файл, чтобы проверить, что гостевая сетевая папка работает:





ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №21

Тема: Развертывание почтового сервера с использованием Postfix и Dovecot

Цель работы: Сконфигурировать виртуальную машину в качестве почтового сервера на основе Postfix и Dovecot:

– Обновление локального индекса пакетов;

– Установка пакета с программой Postfix и его базовая настройка;

– Установка пакета с программой Dovecot и его базовая настройка;

– Создание пользователя postman;

– Настройка почтового клиента Thunderbird и проверка почтового сервера;

– Проверка работы почтового сервера и отправка электронной почты на почтовый ящик в Интернете.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

В процессе передачи электронного письма участвуют почтовый сервер и почтовый клиент. Почтовый сервер – это программа, которая передает сообщение от одного

компьютера к другому. Например, популярными почтовыми серверами являются Gmail, Яндекс, Mail.ru. Почтовый клиент – это программа, в которой вы работаете с почтой: пишете, читаете и храните письма. Например, почтовые клиенты: Microsoft Outlook, Thunderbird, The Bat и т.д.

Процесс электронной передачи письма можно сравнить с работой традиционной почты. Сервер электронной почты – это почтовое отделение. В него попадают письма, сортируются и отправляются в почтовые отделения получателя. А почтовый клиент – это конверт, в который мы вкладываем письмо, а также указываем на нем адрес, имя отправителя и получателя.

Существуют несколько типов протоколов для работы с электронной почтой:

– SMTP (Simple Mail Transfer Protocol) – это протокол, отвечающий исключительно за отправку писем. Работает на портах 110 (без шифрования) и 465 (порт SSL/TLS – SMTPS);

– POP3 – это протокол для получения электронных писем. По протоколу Post Office Protocol 3 можно связаться с удаленным сервером и загрузить сообщение на локальный почтовый клиент. При этом информация с сервера удалится. Обычно почтовые клиенты предлагают выбор – оставлять копии сообщений на сервере или нет. Работает на портах 25 (без шифрования) и 995 (порт SSL/TLS – POP3S);

– IMAP это протокол, который похож на POP3, но отличается тем, что Internet Message Access Protocol работает с почтой непосредственно на сервере. Все чаще провайдеры советуют использовать именно его. Работает на портах 143 (без шифрования) и 993 (порт SSL/TLS – IMAPS).

Входим на виртуальную машину Debian 11 – Server.

Обновление локального индекса пакетов

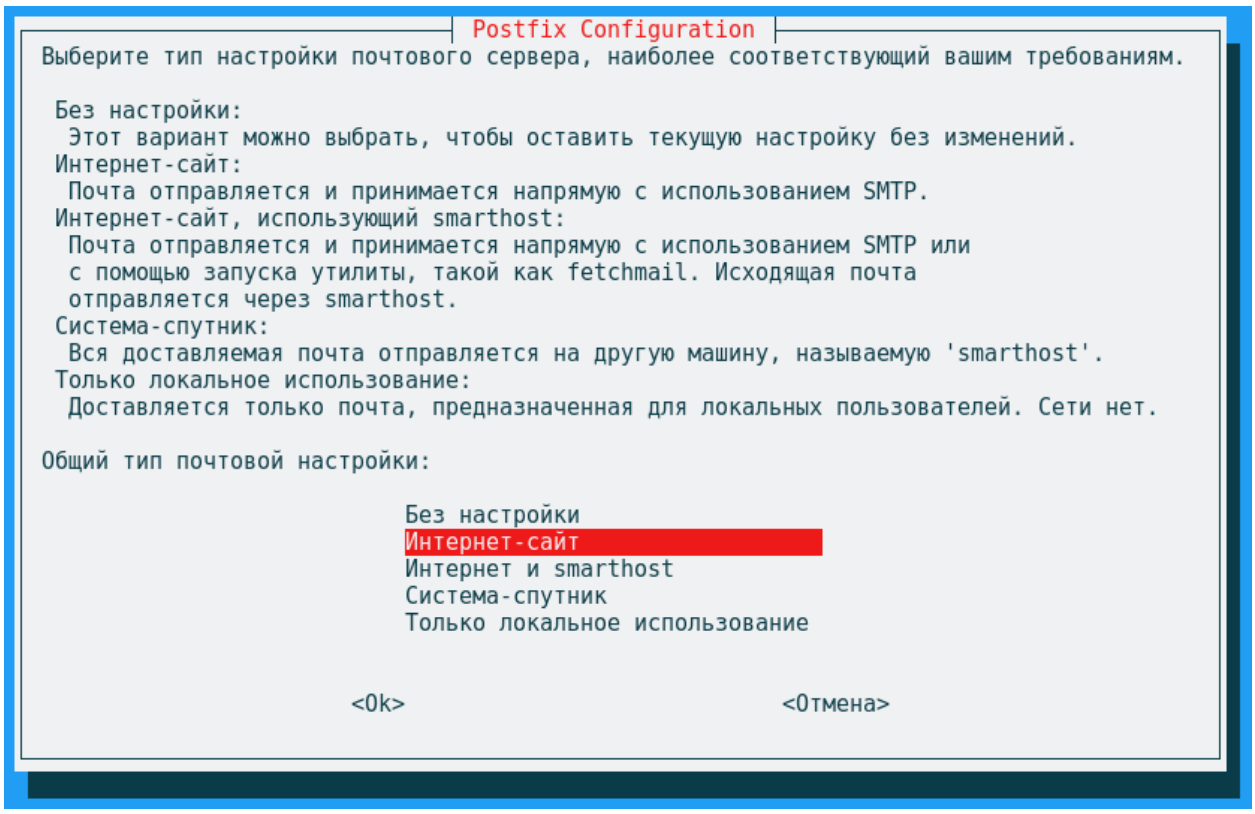
```
root@debian-11-server:~# su -
Пароль: guest
root@debian-11-server:~# apt update
```

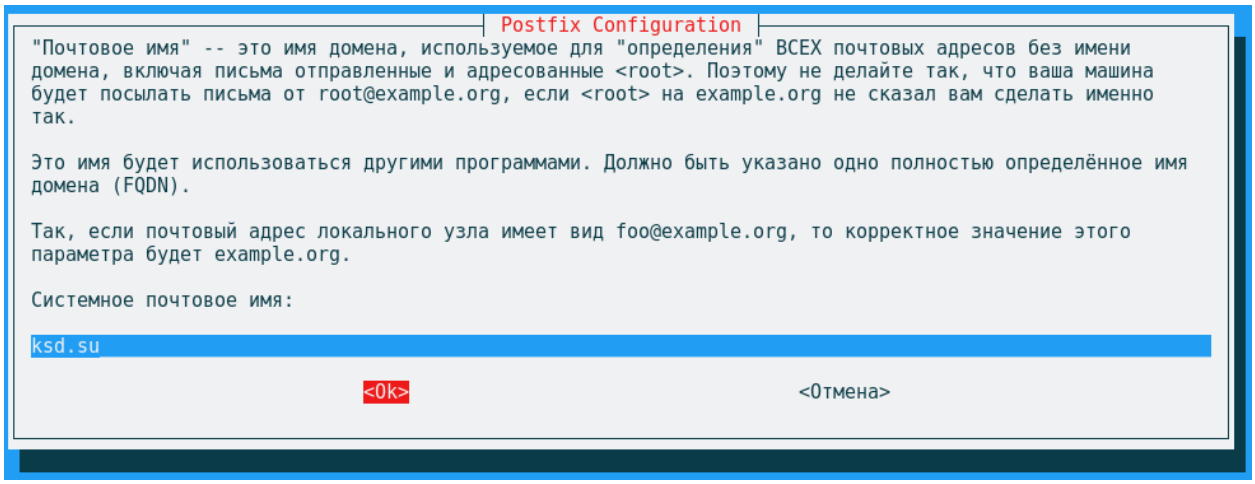
Установка пакета с программой Postfix и его базовая настройка

Postfix – это агент передачи почты (Mail Transfer Agent, MTA), т.е. приложение для отправки и приема электронной почты по протоколу SMTP.

```
root@debian-11-server:~# apt install postfix
```

Во время установки программы в консоли появится псевдографический интерфейс с конфигурацией и настройкой пакета Postfix. Выберите вопросы, изображенные на скриншотах ниже.





Убедитесь, что служба Postfix работает:

root@debian-11-server:~# systemctl status postfix

Нам необходимо проверить открытые порты в Linux с соответствующим сервисом-процессом. Для этого можно или воспользоваться утилитой lsof[19], или установить net-tools[20], чтобы использовать оттуда устаревшую утилиту netstat[21]:

root@debian-11-server:~# apt install net-tools

[19] Утилита lsof в основном отображает список открытых файлов. Тем не менее с некоторыми настройками параметров (-i) мы в состоянии проверить открытые порты в Linux.

[20] net-tools – набор сетевых онлайн инструментов и утилит для оптимизации и диагностики сетевых ресурсов, круглосуточный мониторинг сетевых ресурсов.

[21] В основной форме команда netstat выводит на экран или печатает информацию о сетевых подключениях и таблицу маршрутизации и т.д. Однако та же команда вместе с параметром -ntulp может быть использована для проверки открытых портов в Linux.

root@debian-11-server:~# lsof -i | grep master

```
root@debian-11-server:~# lsof -i | grep master
master  18822     root   13u  IPv4  42082      0t0  TCP *:smtp (LISTEN)
master  18822     root   14u  IPv6  42083      0t0  TCP *:smtp (LISTEN)
```

root@debian-11-server:~# netstat -ntulp | grep master

```
root@debian-11-server:~# netstat -ntulp | grep master
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      18822/master
tcp6       0      0 :::25                   :::*                    LISTEN      18822/master
```

Для работы современной почты только этих портов недостаточно. Добавим порты smtps и submission. Откроем конфигурацию, раскомментируем нижеперечисленные строки, а затем изменим и добавим еще строк.

root@debian-11-server:~# nano -c /etc/postfix/master.cf

```
submission inet n       -       y       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_sasl_type=dovecot
  -o smptd_sasl_path=private/auth
  -o smptd_sasl_security_options=noanonymous
  -o smptd_sasl_local_domain=debian-11-server
  -o smtpd_tls_auth_only=yes
  -o smtpd_reject_unlisted_recipient=no
  -o smtpd_client_restrictions=$mua_client_restrictions
  -o smtpd_helo_restrictions=$mua_helo_restrictions
  -o smtpd_sender_restrictions=$mua_sender_restrictions
  -o smtpd_recipient_restrictions=
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_reject_unlisted_recipient=no
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o smtpd_helo_restrictions=$mua_helo_restrictions
  -o smtpd_sender_login_maps=hash:/etc/postfix/virtual
  -o smtpd_sender_restrictions=reject_sender_login_mismatch
  -o smtpd_recipient_restrictions=reject_non_fqdn_recipient,reject_recipient_domain,permit_sasl_authentica
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
```

** Обратите внимание, что строка -o smtpd_recipient_restrictions полностью в один скриншот не влезла.*

root@debian-11-server:~# systemctl restart postfix
root@debian-11-server:~# lsof -i | grep master

```
root@debian-11-server:~# lsof -i | grep master
master  113360     root   13u  IPv4 245237      0t0  TCP *:smtp (LISTEN)
master  113360     root   14u  IPv6 245238      0t0  TCP *:smtp (LISTEN)
master  113360     root   18u  IPv4 245243      0t0  TCP *:submission (LISTEN)
master  113360     root   19u  IPv6 245244      0t0  TCP *:submission (LISTEN)
master  113360     root   22u  IPv4 245249      0t0  TCP *:submissions (LISTEN)
master  113360     root   23u  IPv6 245250      0t0  TCP *:submissions (LISTEN)
```

Теперь нам необходимо включить авторизацию postfix dovecot. Открываем файл конфигурации и добавляем следующие строки в конце:

root@debian-11-server:~# nano -c /etc/postfix/main.cf

```
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
```

```
smtpd_sasl_auth_enable = yes
smtpd_relay_restrictions = permit_mynetworks, permit_sasl_authenticated
```

root@debian-11-server:~# systemctl restart postfix

Установка пакета с программой Dovecot и его базовая настройка

Dovecot – это «бэкенд» для доступа к почтовым ящикам пользователей по протоколу POP3 или IMAP.

root@debian-11-server:~# apt install dovecot-core dovecot-imapd
root@debian-11-server:~# systemctl status dovecot
root@debian-11-server:~# lsof -i | grep dovecot

```
root@debian-11-server:~# lsof -i | grep dovecot
dovecot 123929     root   33u  IPv4 263303      0t0  TCP *:imap2 (LISTEN)
dovecot 123929     root   34u  IPv6 263304      0t0  TCP *:imap2 (LISTEN)
dovecot 123929     root   35u  IPv4 263305      0t0  TCP *:imaps (LISTEN)
dovecot 123929     root   36u  IPv6 263306      0t0  TCP *:imaps (LISTEN)
```

Начиная с версии 2.3, Postfix поддерживает SMTP-аутентификацию через Dovecot SASL[22], представленный в серии Dovecot 1.0. Если вы используете Postfix, полученный из двоичного файла (например, файла .rpm или .deb), вы можете проверить, был ли Postfix скомпилирован с поддержкой Dovecot SASL, выполнив команду:

root@debian-11-server:~# postconf -a

После того, как мы убедились, что установка Postfix поддерживает Dovecot SASL, можно начать ее настраивать:

root@debian-11-server:~# nano -c /etc/dovecot/conf.d/10-master.conf

```
service auth { (Строка 86)
…
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    # Assuming the default Postfix user and group
    user = postfix
    group = postfix
  }
  …
}

# Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0666
}
```

```
# Outlook and Windows Mail works only with LOGIN mechanism, not the standard PLAIN:
auth_mechanisms = plain login (Строка 132)
```

[22] SASL – это инфраструктура аутентификации для протоколов, ориентированных на соединения.

```
service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Users that have
  # full permissions to this socket are able to get a list of all usernames and
  # get the results of everyone's userdb lookups.
  #
  # The default 0666 mode allows anyone to connect to the socket, but the
  # userdb lookups will succeed only if the userdb returns an "uid" field that
  # matches the caller process's UID. Also if caller's uid or gid matches the
  # socket's uid or gid the lookup succeeds. Anything else causes a failure.
  #
  # To give the caller full permissions to lookup all users, set the mode to
  # something else than 0666 and Dovecot lets the kernel enforce the
  # permissions (e.g. 0777 allows everyone full permissions).
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }

  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0666
  }
```

root@debian-11-server:~# systemctl restart dovecot
root@debian-11-server:~# systemctl status dovecot
root@debian-11-server:~# systemctl reboot (Перезагрузим сервер во избежание ошибок в работе двух служб почтового сервера)

Создание пользователя postman

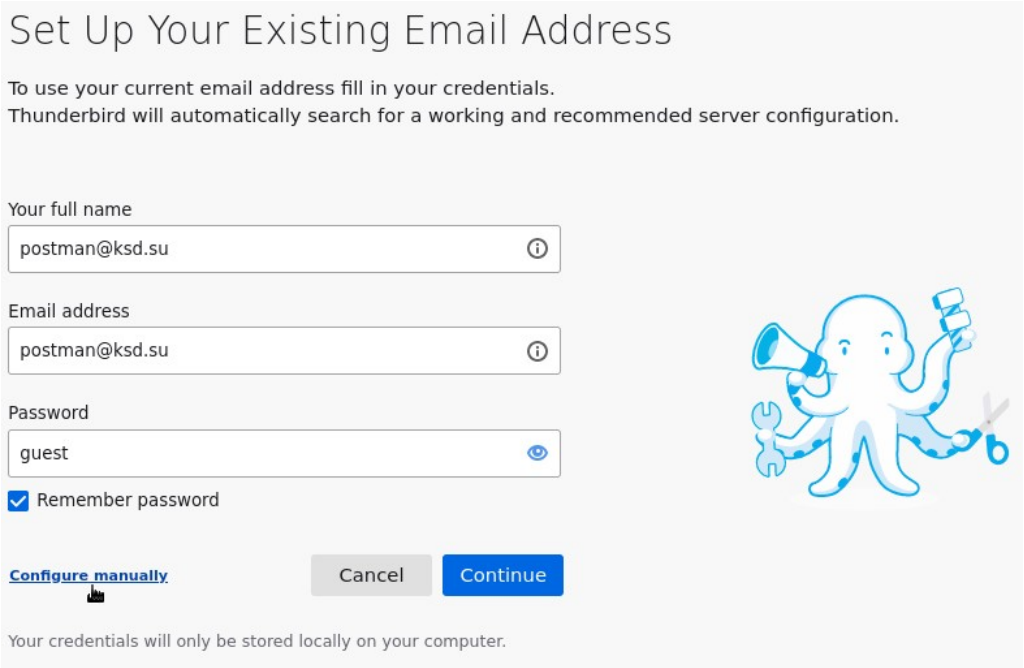
Создадим тестовых пользователей для проверки работы почтового сервера под именами postman и postwoman:

root@debian-11-server:~# adduser postman
root@debian-11-server:~# adduser postwoman

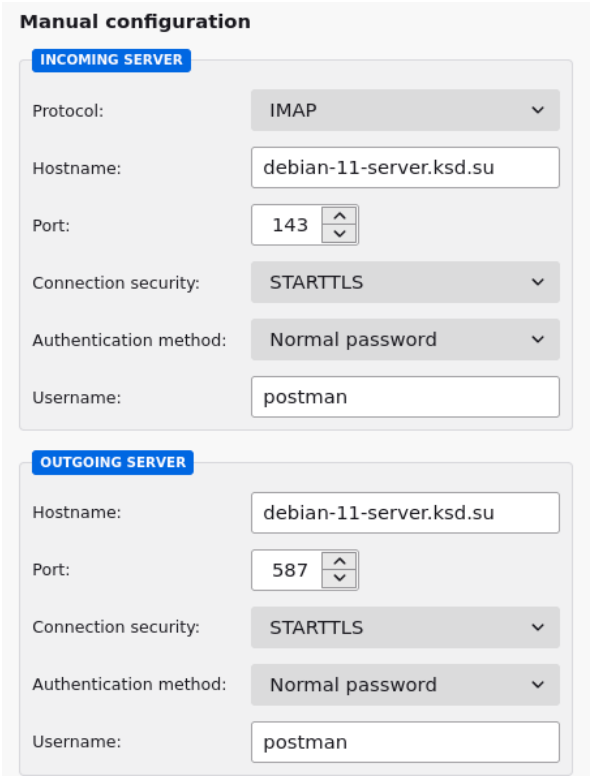
Входим на виртуальную машину Debian 11 – Host.

Настройка почтового клиента Thunderbird и проверка почтового сервера

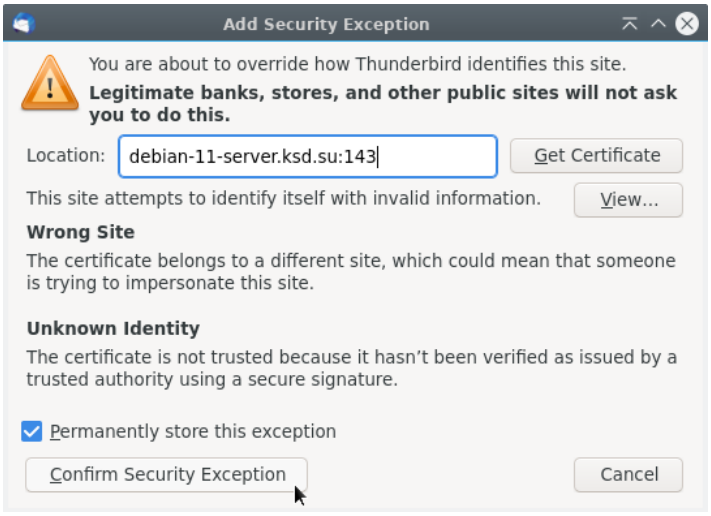
Создадим учетную запись почты и введем учетные данные для входа:



Нажмите на кнопку «Configure manually» и введите те данные, что указаны на скриншоте ниже:



Так как сертификат является «самоподписанным» то подтверждаем исключение безопасности. Такой же вопрос будет при отправке сообщения, но мы на это не будем обращать внимание.



Входим на виртуальную машину Debian 11 – Gateway.

Повторите действия по созданию учетной запись почты, но только для пользователя postwoman.

Входим на виртуальную машину Debian 11 – Host.

Проверка работы почтового сервера и отправка электронной почты на почтовый ящик в Интернете

К сожалению, почтовый сервер полноценно настроен только на функционирование в локальной сети. Для работы почтового сервера в Интернете необходимо приобрести у провайдера белый IP-адрес и зарегистрировать собственное доменное имя, чего в рамках нашего задания сделать не выйдет.

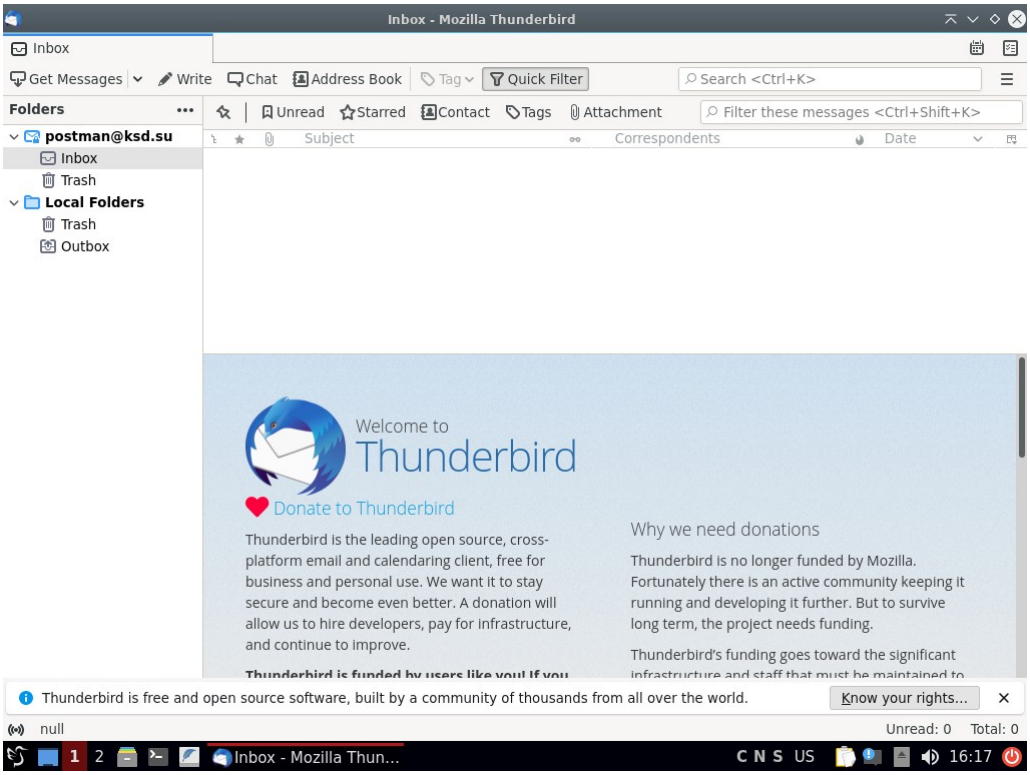
Также почтовый сервер должен соответствовать определенным требованиям, иначе ваши письма будут уходить в спам, так как почтовые серверы настороженно относятся к ресурсам с неизвестным доменом и без PTR-записи:

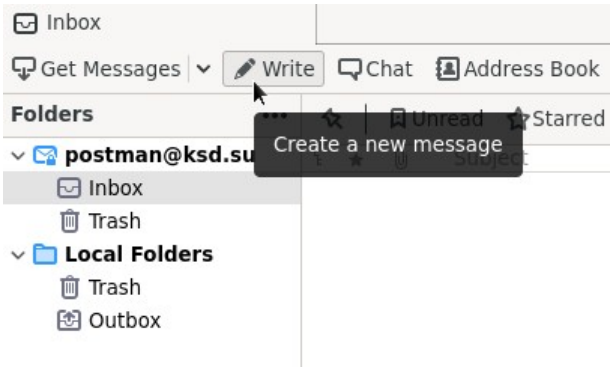
– IP-адрес почтового сервера должен быть публичным и статическим;

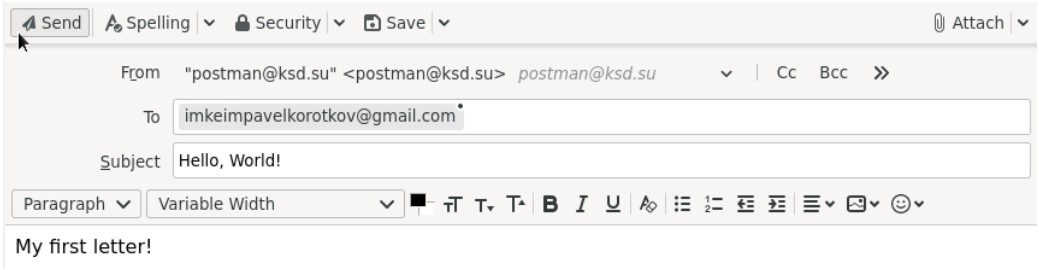
– DNS-запись публичного доменного имени почтового сервера должен указывать на этот IP-адрес;

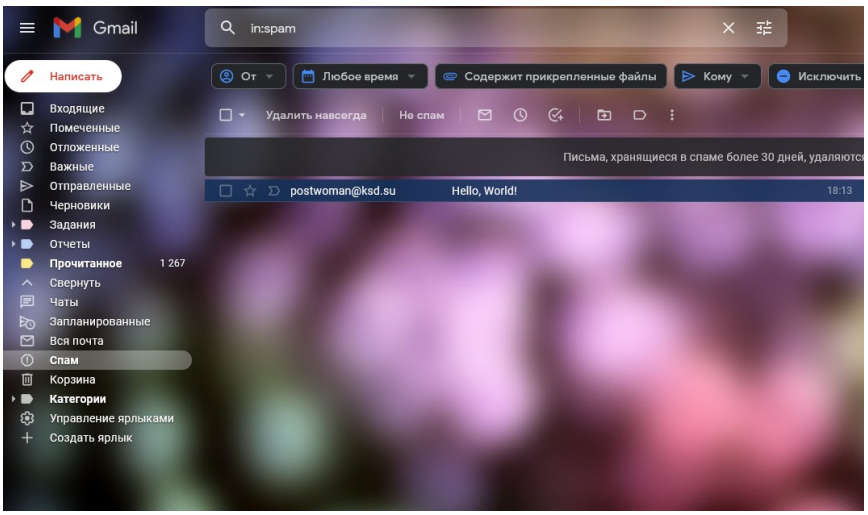
– Обратная DNS-запись (PTR-запись) должна указывать на доменное имя почтового сервера по IP-адресу, в противном случае.

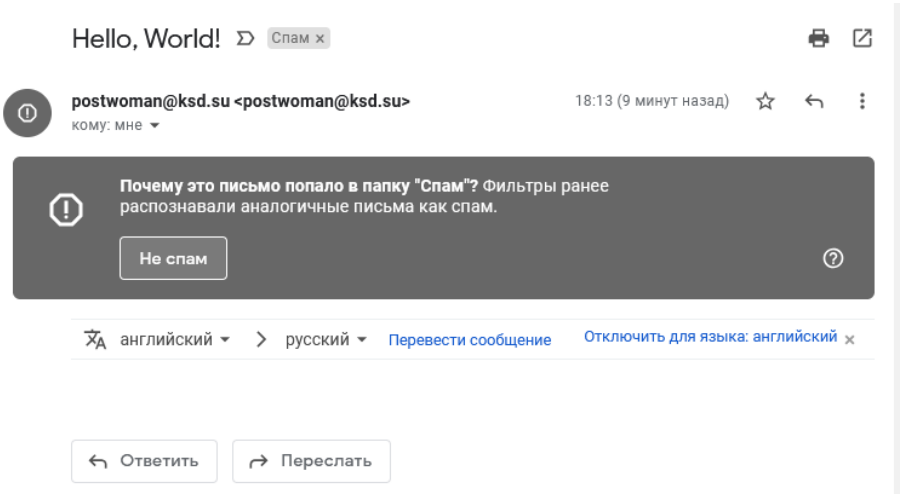
Войдите в почтовый клиент и напишите письмо на вашу электронную почту (Замените почту «imkeimpavelkorotkov@gmail.com», указанную на скриншоте, на личную).


Inbox - Mozilla Thunderbird
Inbox
Get Messages
Write
Chat
Address Book
Tag
Quick Filter
Search <Ctrl+K>
Folders
Unread
Starred
Contact
Tags
Attachment
Filter these messages <Ctrl+Shift+K>
Subject
Correspondents
Date
postman@ksd.su
Inbox
Trash
Local Folders
Trash
Outbox
Welcome to
Thunderbird
Donate to Thunderbird
Thunderbird is the leading open source, cross-platform email and calendaring client, free for business and personal use. We want it to stay secure and become even better. A donation will allow us to hire developers, pay for infrastructure, and continue to improve.
Why we need donations
Thunderbird is no longer funded by Mozilla. Fortunately there is an active community keeping it running and developing it further. But to survive long term, the project needs funding.
Thunderbird's funding goes toward the significant
Thunderbird is free and open source software, built by a community of thousands from all over the world.
Know your rights...
null
Unread: 0
Total: 0
1
2
Inbox - Mozilla Thun...
C N S US
16:17


Inbox
Get Messages
Write
Chat
Address Book
Folders
Unread
Starred
Create a new message
postman@ksd.su
Inbox
Trash
Local Folders
Trash
Outbox


Send
Spelling
Security
Save
Attach
From
"postman@ksd.su" <postman@ksd.su>
postman@ksd.su
Cc
Bcc
To
imkeimpavelkorotkov@gmail.com
Subject
Hello, World!
Paragraph
Variable Width
My first letter!





После этого попробуйте обменяться электронными письмами между почтовыми ящиками «postman@ksd.su» и «postwoman@ksd.su».

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №22

Тема: Развертывание SSH-сервера

Цель работы: Сконфигурировать виртуальную машину в качестве SSH-сервера на основе OpenSSH-Server:

– Обновление локального индекса пакетов;

– Установка пакета с программой OpenSSH-Server и его базовая настройка;

– Настройка локального пользователя guest в группу sudoers;

– Проверка подключения к Debian 11 по SSH без авторизации по публичному ключу;

– Установка программы OpenSSH-Server и его базовая настройка;

– Проверка подключения к Windows 10 по SSH без авторизации по публичному ключу;

– Загрузка ключа на сервер Debian-11-Server и Windows-10-Host;

– Настройка безопасного SSH-сервера;

– Проверка подключения к Windows 10 и Debian 11 по SSH с публичным ключом.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

SSH или Secure Shell – это протокол безопасного доступа из одного компьютера к другому по сети. У протокола SSH очень много возможностей. Вы можете создавать защищенные соединения между компьютерами, открывать командную строку на удаленном компьютере, запускать графические программы, передавать файлы и организовывать частные сети.

За поддержку протокола SSH в Linux отвечает набор программного обеспечения OpenSSH. Это открытая реализация этого протокола, которая предоставляет все необходимые возможности. В состав пакета OpenSSH входят утилиты для установки соединения, передачи файлов, а также сам SSH-сервер.

Входим на виртуальную машину Debian 11 – Host.

Обновление локального индекса пакетов

```
root@debian-11-host:~# su -
Пароль: guest
root@debian-11-host:~# apt update
```

Установка пакета с программой OpenSSH-Server и его базовая настройка

```
root@debian-11-host:~# apt install openssh-server
```

Если вы хотите, чтобы служба запускалась автоматически нужно добавить его в автозагрузку. Поэтому чтобы включить службу SSH выполните:

```
root@debian-11-host:~# systemctl enable ssh
```

Что касается SSH-клиента, то он уже установлен в системе по умолчанию. Сейчас вы можете попробовать подключиться к локальному SSH-серверу просто набрав:

```
root@debian-11-host:~# ssh guest@debian-11-host.ksd.su
```

Настройка локального пользователя guest в группу sudoers

sudo – это утилита командной строки, которая позволяет доверенным пользователям запускать команды от имени другого пользователя, по умолчанию root.

Самый быстрый и простой способ предоставить пользователю привилегии sudo – добавить пользователя в группу «sudo». Члены этой группы могут выполнять любую команду от имени пользователя root через sudo, и им будет предложено пройти аутентификацию с помощью своего пароля при использовании sudo.

Запустите приведенную ниже команду от имени пользователя root, чтобы добавить пользователя в группу sudo:

```
root@debian-11-host:~# su -
Пароль: guest
root@debian-11-host:~# usermod -aG sudo guest
```

Чтобы убедиться, что пользователь добавлен в группу, введите:

```
root@debian-11-host:~# login guest
Пароль: guest
guest@debian-11-host:~# sudo whoami
```

Вас попросят ввести пароль. Если у пользователя есть доступ к sudo, команда напечатает «root». В противном случае вы получите сообщение об ошибке «пользователь отсутствует в файле sudoers».

Проверка подключения к Debian 11 по SSH без авторизации по публичному ключу

Входим на виртуальную машину Debian 11 – Server.

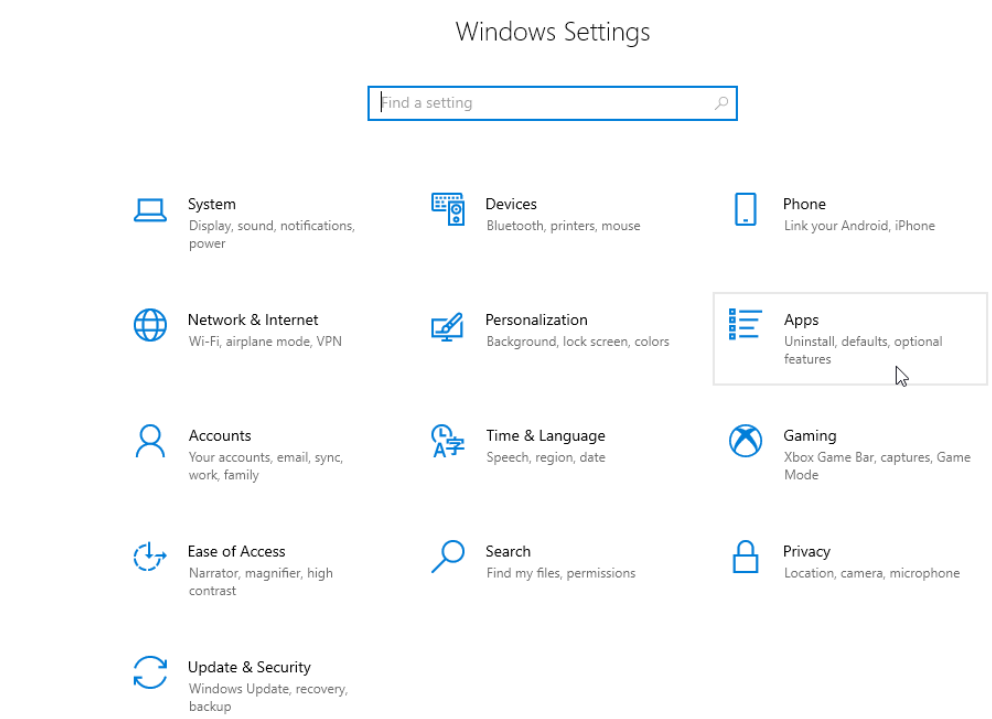
Попробуйте подключиться к Windows-10-Host по SSH:

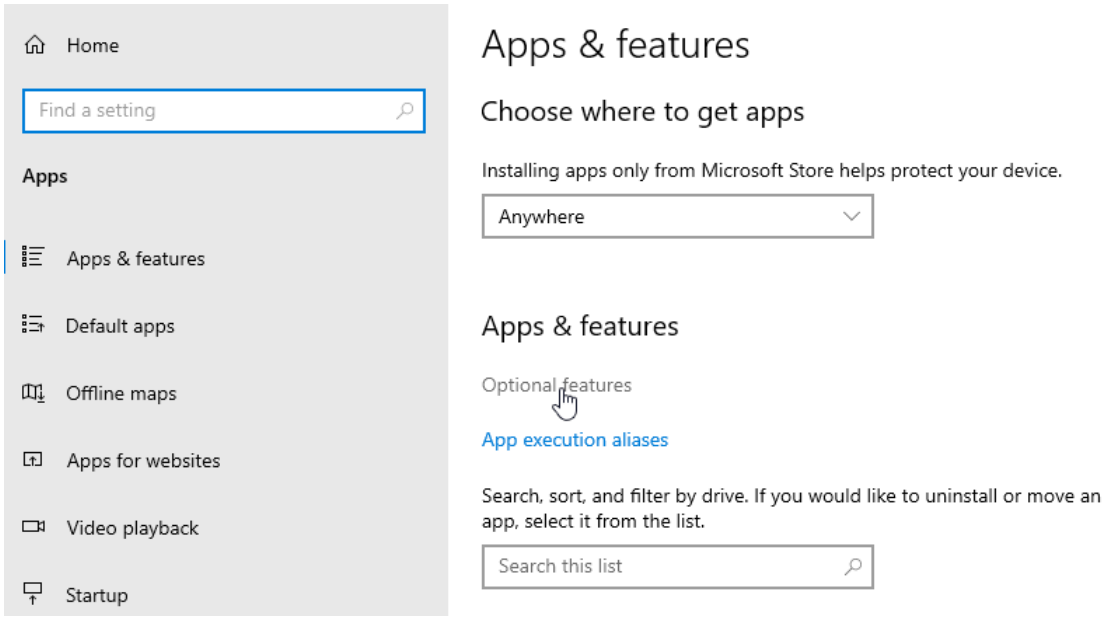
```
guest@debian-11-server:~# ssh guest@debian-11-host.ksd.su
guest@debian-11-host:~# hostname
guest@debian-11-host:~# exit
```

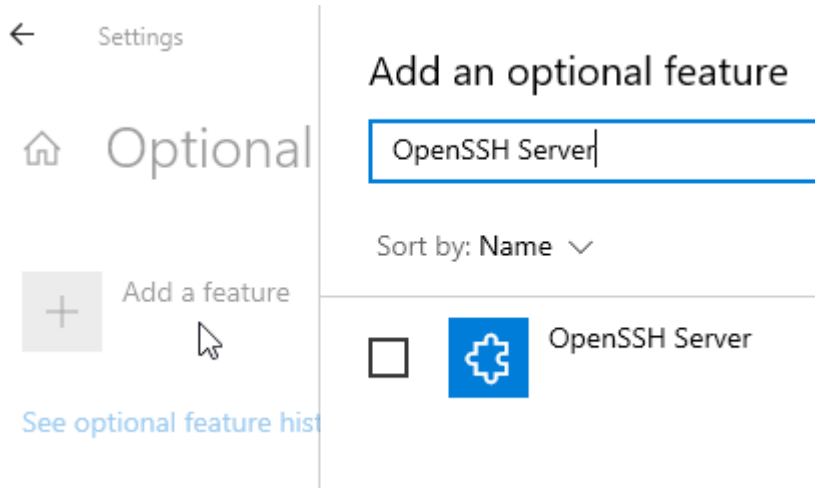
Входим на виртуальную машину Windows 10 – Host.

Установка программы OpenSSH-Server и его базовая настройка

В первую очередь необходимо установить компонент OpenSSH Server. В Windows 10 этот компонент можно установить через панель Параметры (Приложения → Управление дополнительными компонентами → Добавить компонент). Найдите в списке OpenSSH-Server и нажмите кнопку «Install»).


Windows Settings
Find a setting
System
Display, sound, notifications, power
Devices
Bluetooth, printers, mouse
Phone
Link your Android, iPhone
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Apps
Uninstall, defaults, optional features
Accounts
Your accounts, email, sync, work, family
Time & Language
Speech, region, date
Gaming
Xbox Game Bar, captures, Game Mode
Ease of Access
Narrator, magnifier, high contrast
Search
Find my files, permissions
Privacy
Location, camera, microphone
Update & Security
Windows Update, recovery, backup


Home
Find a setting
Apps
Apps & features
Default apps
Offline maps
Apps for websites
Video playback
Startup
Apps & features
Choose where to get apps
Installing apps only from Microsoft Store helps protect your device.
Anywhere
Apps & features
Optional features
App execution aliases
Search, sort, and filter by drive. If you would like to uninstall or move an app, select it from the list.
Search this list


Settings
Optional
Add a feature
See optional feature hist
Add an optional feature
OpenSSH Server
Sort by: Name
OpenSSH Server

После установки сервера OpenSSH в Windows вам нужно изменить тип запуска службы sshd на автоматический и запустить службу с помощью PowerShell:

```
PS C:\Windows\system32> Set-Service -Name sshd -StartupType 'Automatic'
PS C:\Windows\system32> Start-Service sshd
WARNING: Waiting for service 'OpenSSH SSH Server (sshd)' to start...
```

netstat – это утилита командной строки выводящая на дисплей состояние TCP-соединений, таблицы маршрутизации, число сетевых интерфейсов и сетевую статистику по протоколам. С помощью netstat убедитесь, что теперь в системе запущен SSH-сервер и ждет подключений на 22 порту:

```
C:\Windows\system32>netstat -na| find "22"
  TCP    0.0.0.0:22             0.0.0.0:0              LISTENING
  TCP    [::]:22                [::]:0                 LISTENING
```

Проверьте, что включено правило брандмауэра (Windows Defender Firewall), разрешающее входящие подключения к Windows по порту TCP – 22:

```
PS C:\Windows\system32> Get-NetFirewallRule -Name *OpenSSH-Server* |select Name, DisplayName, Description, Enabled

Name                  DisplayName              Description                           Enabled
----                  -----------              -----------                           -------
OpenSSH-Server-In-TCP OpenSSH SSH Server (sshd) Inbound rule for OpenSSH SSH Server (sshd)    True
```

Если правило отключено (состоянии Enabled=False) или отсутствует, вы можете создать новое входящее правило командой New-NetFirewallRule:

PS C:\Windows\system32> New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)' -Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22

По умолчанию важным компоненты OpenSSH хранятся в следующих каталогах:

– Исполняемые файлы OpenSSH Server: C:\Windows\System32\OpenSSH\;

– Конфигурационный файл sshd_config (создается после первого запуска службы): C:\ProgramData\ssh;

– Журнал OpenSSH: C:\windows\system32\OpenSSH\logs\sshd.log;

– Файл authorized_keys и ключи: %USERPROFILE%\.ssh\.

При установке OpenSSH сервера в системе создается новый локальный пользователь sshd.

Проверка подключения к Windows 10 по SSH без авторизации по публичному ключу

Входим на виртуальную машину Debian 11 – Host.

Попробуйте подключиться к Windows-10-Host по SSH:

guest@debian-11-host:~# ssh win10@windows-10-host.ksd.su
win10@WINDOWS-10-HOST C:\Users\win10>mkdir MyFolder
win10@WINDOWS-10-HOST C:\Users\win10>cd MyFolder

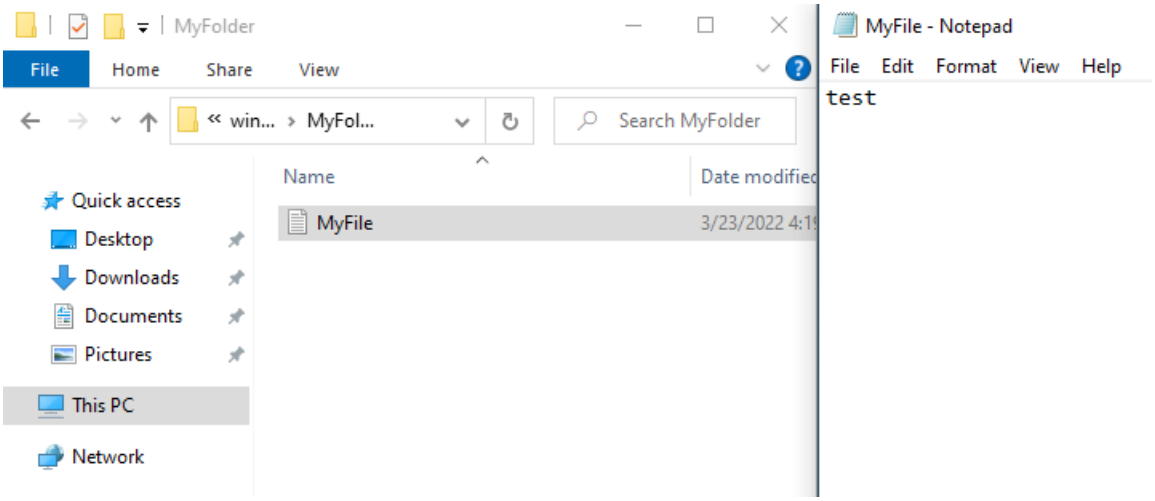
win10@WINDOWS-10-HOST C:\Users\win10\MyFolder>copy con MyFile.txt

test

[ctrl+c]
win10@WINDOWS-10-HOST C:\Users\win10\MyFolder\>exit

Входим на виртуальную машину Windows 10 – Host.

Проверьте на виртуальной машине Windows, что ваши команды применились:



Входим на виртуальную машину Debian 11 – Server.

Загрузка ключа на сервер Debian-11-Server и Windows-10-Host

root@debian-11-server:~# su -
Пароль: guest
root@debian-11-server:~# apt update
root@debian-11-server:~# apt install openssh-server

В SSH существует несколько способов авторизации. Вы можете каждый раз вводить пароль пользователя или использовать более безопасный и надежный способ – ключи SSH. Что самое интересное, он более удобен для применения, вам даже не нужно будет вводить пароль. Генерация ключа SSH между Linux-устройствами настраивается с помощью утилиты ssh-keygen:

guest@debian-11-server:~# ssh-keygen

Утилита предложит вам выбрать расположение ключей. По умолчанию ключи располагаются в папке ~/.ssh/. Лучше ничего не менять, чтобы все работало по умолчанию и ключи автоматически подхватывались. Секретный ключ будет называться id_rsa, а публичный id_rsa.pub.

```
guest@debian-11-server:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/guest/.ssh/id_rsa):
/home/guest/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/guest/.ssh/id_rsa
Your public key has been saved in /home/guest/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jrjIVRTbFZZAhritHVhMNhmLl2uaMKsCBZDqKhkkZYw guest@debian-11-server
The key's randomart image is:
+---[RSA 3072]----+
|o+   +*=+.+o     |
|E + .o=0 o.      |
|.+  .=* .        |
|o.. ooo.         |
|+.o  o+.S        |
|o. +.*.o         |
|oo. = . .        |
|=o o .           |
|+ o .            |
+----[SHA256]-----+
```

Далее необходимо передать ключ на сервер, используя команду ssh-copy-id:

guest@debian-11-host:~# ssh-copy-id guest@Debian-11-Host.ksd.su

Входим на виртуальную машину Windows 10 – Host.

Чтобы скачать и перенести публичный ключ на виртуальную машину Windows, утилита ssh-copy-id не подойдет. Зато вместо нее можно использовать утилиту scp.

Выполняем те же действия, что выполняли при генерации ключей для Debian-11-Server:

C:\Users\win10>ssh-keygen

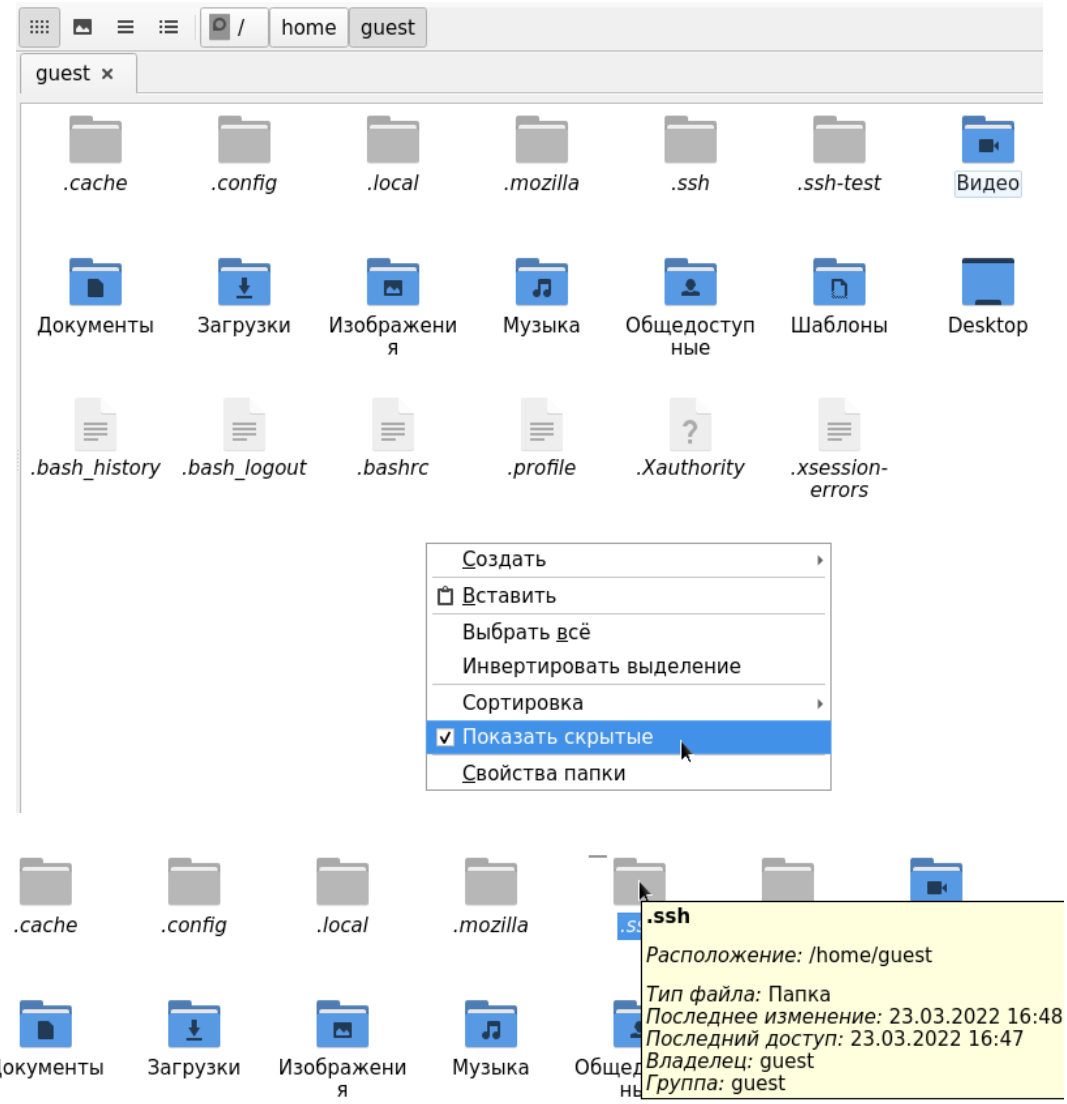
```
C:\Users\win10>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\win10/.ssh/id_rsa):
Created directory 'C:\Users\win10/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\win10/.ssh/id_rsa.
Your public key has been saved in C:\Users\win10/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:YWjbRa5BS7v3siaEE04w9XRAEg1MxlR+rvvLlw+Rg+M win10@windows-10-host
The key's randomart image is:
+---[RSA 3072]----+
|     *O*B.o      |
|    o.oO.*       |
|     oo O +      |
|     .o+ O . .   |
|     o.oS = +    |
|      + .+ o o   |
|       o. E o.   |
|         .o.oo.  |
|         .+=o .. |
+----[SHA256]-----+
```

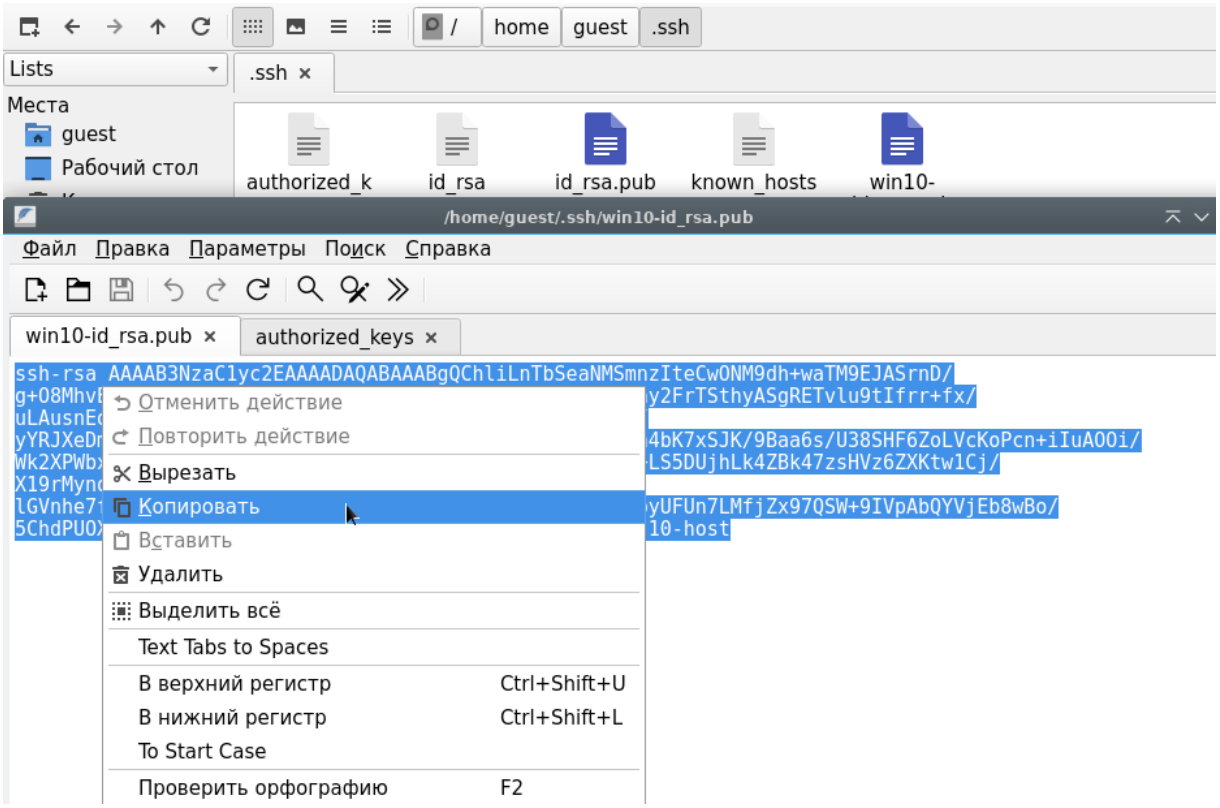
C:\Users\win10>scp C:\Users\win10\.ssh\id_rsa.pub
guest@Debian-11-Host.ksd.su:/home/guest/.ssh/win10-id_rsa.pub

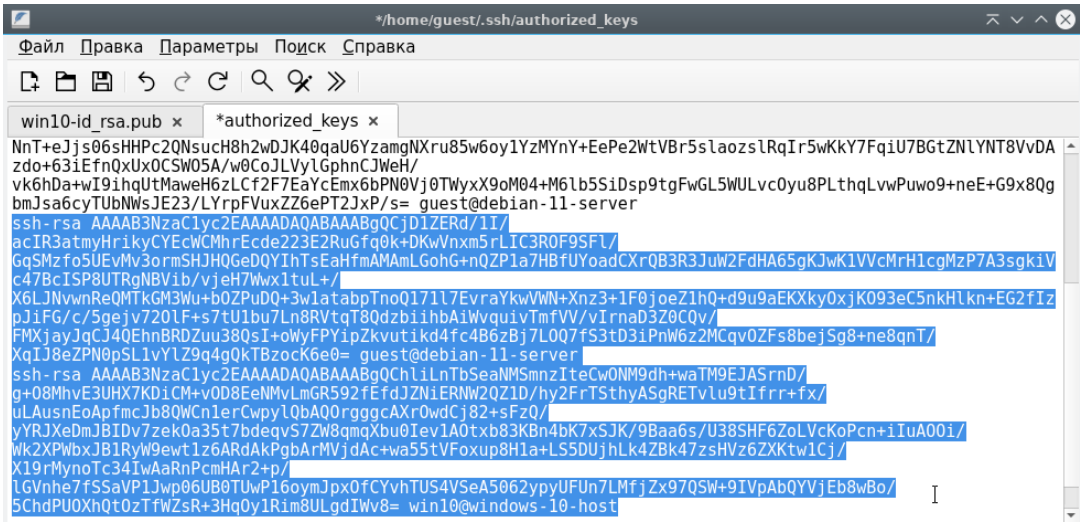
```
C:\Users\win10>scp C:\Users\win10\.ssh\id_rsa.pub guest@Debian-11-Host:/home/guest/.ssh/win10-id_rsa.pub
guest@debian-11-host's password:
id_rsa.pub                    100%  576    36.0KB/s   00:00
```

Входим на виртуальную машину Debian 11 – Host.

Зайдите в файловый менеджер, найдите там публичный ключ и скопируйте его из файла win10-id_rsa.pub в authorized_keys:







Настройка безопасного SSH-сервера

С параметрами по умолчанию сервер SSH не очень безопасен поэтому перед тем, как программа будет готова к полноценному использованию ее нужно немного настроить. Все настройки сервера SSH хранятся в конфигурационном файле sshd_config, который находится в папке /etc/ssh.

Перед тем как вносить изменения в этот конфигурационный файл рекомендуется сделать его резервную копию, для этого можете использовать такую команду:

guest@debian-11-host:~# sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old

Дальше вы можете перейти к настройке конфигурационного файла:

guest@debian-11-host:~# sudo nano -c /etc/ssh/sshd_config

Первым делом желательно сменить порт, на котором работает ssh, чтобы возможный злоумышленник не знал включен ли у вас этот сервис. Найдите в конфигурационном файле строчку Port и замените ее значение на любое число, например, Port 2222:

Port 2222 (Строка 15)

По умолчанию вход от имени суперпользователя включен, рекомендуется отключить такую возможность. Для этого найдите строчку PermitRootLogin и замените ее значение на no:

PermitRootLogin no (Строка 34)

Чтобы разрешить аутентификацию по ключу, а не по паролю найдите строку PubkeyAuthentication и убедитесь, что ее значение yes.

PubkeyAuthentication yes (Строка 39)

Если пароль больше не будет использоваться, то для увеличения безопасности системы лучше его вовсе отключить. Но убедитесь, что ключ надежно сохранен и вы его не потеряете, потому что по паролю вы больше не войдете. Авторизуйтесь на сервере, затем откройте конфигурационный файл /etc/ssh/sshd_config и найдите там директиву PasswordAuthentication. Нужно установить ее значение в no:

PasswordAuthentication no (Строка 58)

После того как все настройки будут завершены, сохраните изменения нажав :w и перезапустите службу SSH:

guest@debian-11-host:~# sudo systemctl restart ssh

Проверка подключения к Windows 10 и Debian 11 по SSH с публичным ключом

Входим на виртуальную машину Debian 11 – Server.

Попробуйте подключиться к Windows-10-Host по SSH:

guest@debian-11-server:~# ssh -p 2222 guest@debian-11-host.ksd.su
guest@debian-11-host:~# hostname
guest@debian-11-host:~# exit

Входим на виртуальную машину Debian 11 – Gateway.

Убедитесь в том, что вы не можем получить доступ к серверу без публичного ключа. Попробуйте подключиться к Debian-11-Host по SSH:

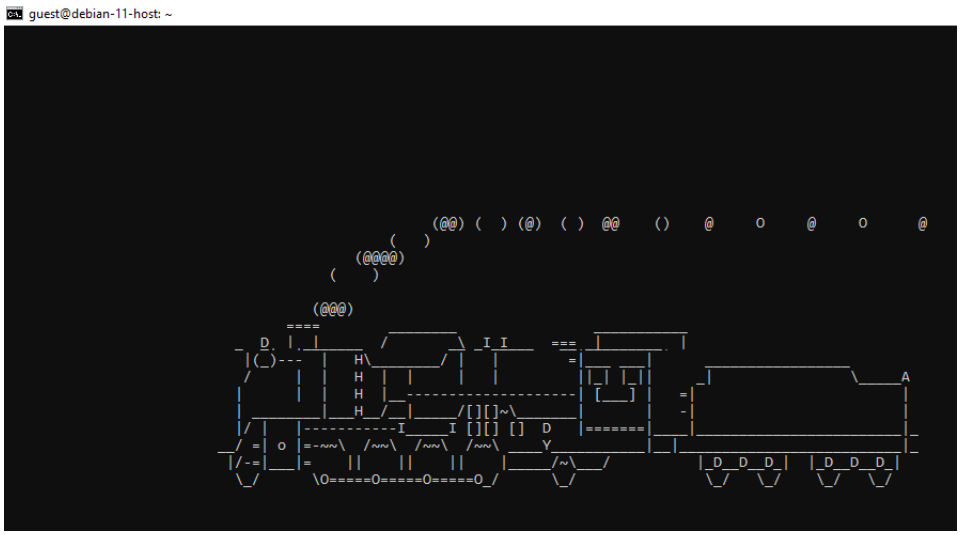
guest@debian-11-gateway:~# ssh -p 2222 guest@debian-11-host.ksd.su

Входим на виртуальную машину Windows 10 – Host.

Попробуйте подключиться к Windows-10-Host по SSH:

C:\Users\win10>ssh -p 2222 guest@debian-11-host.ksd.su
guest@debian-11-host:~# sl
guest@debian-11-host:~# hostname
guest@debian-11-host:~# exit

```
C:\Users\win10>ssh -p 2222 guest@Debian-11-Host.ksd.su
Linux debian-11-host 5.10.0-10-amd64 #1 SMP Debian 5.10.84-1 (2021-12-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar 23 16:51:36 2022 from 192.168.1.10
guest@debian-11-host:~$ sl
```



ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №23

Тема: Настройка фильтрации пакетов с помощью UFW

Цель работы: Сконфигурировать виртуальную машину в роли межсетевого экрана с использованием UFW:

– Обновление локального индекса пакетов;

– Установка пакета с программой UFW;

– Проверка статуса UFW;

– Политики UFW по умолчанию;

– Просмотр профилей приложений;

– Разрешение SSH-соединений;

– Включение и проверка работы межсетевого экрана UFW;

– Разрешение диапазона портов;

– Разрешение определенных IP-адресов) на определенном порту;

– Разрешение подсетей;

– Запрет соединений.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Правильно настроенный брандмауэр является одним из наиболее важных аспектов общей безопасности системы.

UFW или Uncomplicated Firewall (Несложным межсетевой экран) – это удобный интерфейс для управления правилами брандмауэра Netfilter (iptables). Его основная цель – сделать управление iptables проще или, как следует из названия, несложным.

Входим на виртуальную машину Debian 11 – Host.

Обновление локального индекса пакетов

root@debian-11-server:~# su -
Пароль: guest
root@debian-11-server:~# apt update

Установка пакета с программой UFW

root@debian-11-server:~# apt install ufw

Проверка статуса UFW

Установка не активирует брандмауэр автоматически, чтобы избежать блокировки сервера. Вы можете проверить статус UFW, набрав:

root@debian-11-server:~# ufw status verbose

Вывод будет выглядеть следующим образом:

```
root@debian-11-server:~# ufw status verbose
Status: inactive
```

Политики UFW по умолчанию

По умолчанию UFW блокирует все входящие подключения и разрешает все исходящие. Это означает, что любой, кто пытается получить доступ к вашему серверу, не сможет подключиться, если вы специально не откроете порт. Приложения и службы, работающие на сервере, смогут получить доступ к внешнему миру.

Политики по умолчанию определены в файле /etc/default/ufw и могут быть изменены с помощью команды ufw default <*политика*> <*цепочка*>.

Политики брандмауэра являются основой для создания более подробных и определяемых пользователем правил. Как правило, исходные политики UFW по умолчанию являются хорошей отправной точкой.

Просмотр профилей приложений

Большинство приложений поставляются с профилем приложения, который описывает службу и содержит настройки UFW. Профиль автоматически создается в каталоге /etc/ufw/applications.d во время установки пакета.

Чтобы просмотреть все профили приложений, доступные в вашей системе, введите: root@debian-11-server:~# ufw app list

```
root@debian-11-server:~# ufw app list
Available applications:
  AIM
  Bonjour
  CIFS
  DNS
  Deluge
  Dovecot IMAP
  Dovecot Secure IMAP
  IMAP
  IMAPS
  IPP
  KTorrent
  Kerberos Admin
  Kerberos Full
  Kerberos KDC
  Kerberos Password
  LDAP
  LDAPS
  LPD
  MSN
  MSN SSL
  Mail submission
  NFS
  OpenSSH
  POP3
  POP3S
  PeopleNearby
  Postfix
  Postfix SMTPS
  Postfix Submission
  SMTP
  SSH
  Samba
  Socks
```

Чтобы найти дополнительную информацию о конкретном профиле и включенных правилах, используйте следующую команду:

root@debian-11-server:~# ufw app info 'OpenSSH'

```
root@debian-11-server:~# ufw app info 'OpenSSH'
Profile: OpenSSH
Title: Secure shell server, an rshd replacement
Description: OpenSSH is a free implementation of the Secure Shell protocol.

Port:
  22/tcp
root@debian-11-server:~# ufw app info 'OpenSSH'
Profile: OpenSSH
Title: Secure shell server, an rshd replacement
Description: OpenSSH is a free implementation of the Secure Shell protocol.

Port:
  22/tcp
```

Как видно из вывода выше, профиль «OpenSSH» открывает порт 22.

Разрешение SSH-соединений

Перед включением брандмауэра UFW нам нужно добавить правило, разрешающее входящие SSH-соединения. Если вы подключаетесь к своему серверу из удаленного места, что почти всегда имеет место, и вы включаете брандмауэр UFW, прежде чем явно разрешить входящие SSH-соединения, вы больше не сможете подключаться к своему серверу Debian.

Чтобы настроить брандмауэр UFW для разрешения входящих SSH-подключений, введите следующую команду:

root@debian-11-server:~# ufw allow ssh

```
root@debian-11-server:~# ufw allow ssh
Rules updated
Rules updated (v6)
```

Если вы изменили порт SSH на пользовательский порт вместо порта 22, вам нужно будет открыть этот порт.

Например, если ваш демон ssh прослушивает порт 2222, вы можете использовать следующую команду, чтобы разрешить соединения на этом порту:

root@debian-11-server:~# ufw allow 2222/tcp

Разрешим на брандмауэре также порт 53, отвечающий за работу DNS, иначе мы не сможем использовать доменные имя при подключении к Debian-11-Server:

root@debian-11-server:~# ufw app info 'DNS'
root@debian-11-server:~# ufw allow dns
root@debian-11-server:~# ufw status

Включение и проверка работы межсетевого экрана UFW

Если вы изменили порт SSH на пользовательский порт вместо порта 22, вам нужно будет открыть этот порт.

Теперь, когда ваш брандмауэр UFW настроен на разрешение входящих SSH-соединений, мы можем включить его, набрав:

root@debian-11-server:~# ufw enable

Вы будете предупреждены, что включение брандмауэра может нарушить существующие соединения ssh, просто введите y и нажмите Enter.

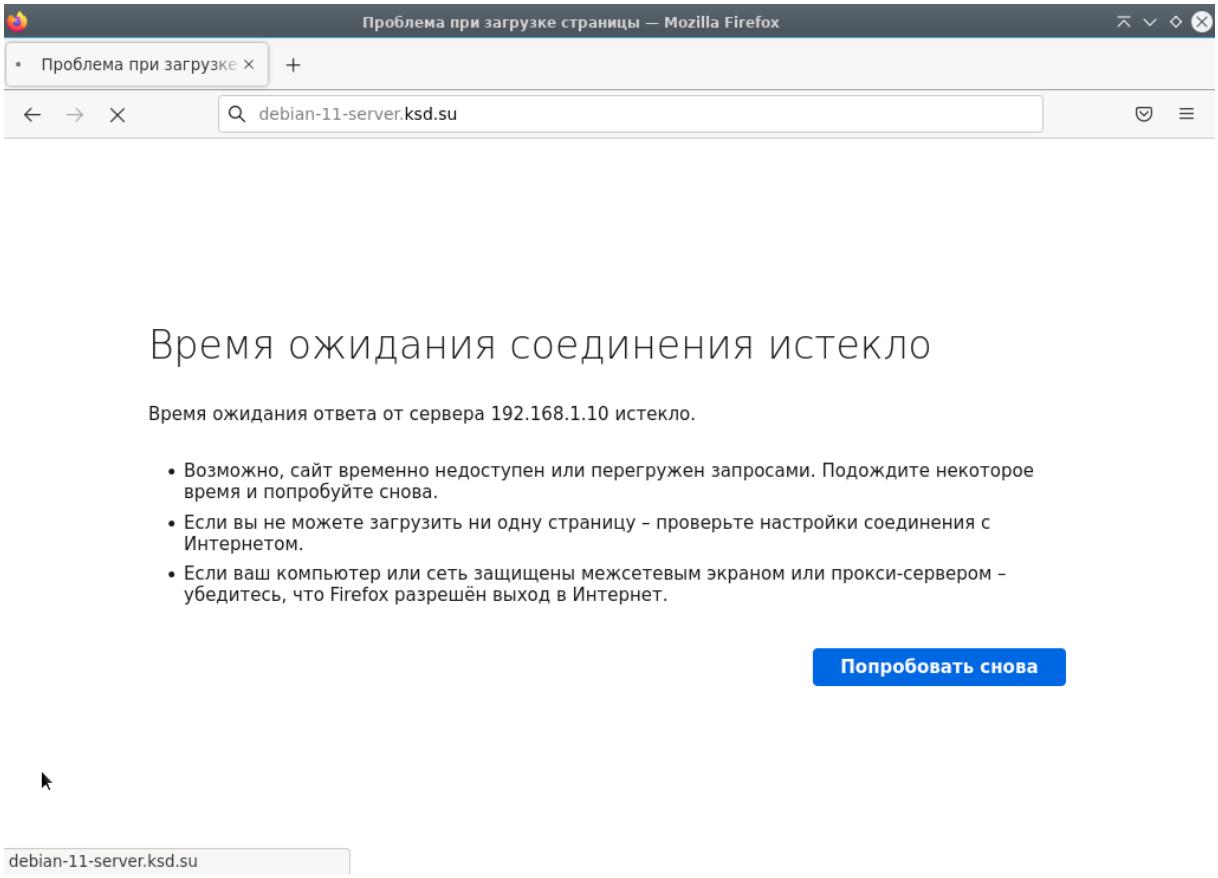
Входим на виртуальную машину Debian 11 – Host.

Попробуйте подключиться к Debian-11-Server по SSH:

root@debian-11-host:~# ssh guest@debian-11-server.ksd.su
[ctrl+c]

```
guest@debian-11-host:~$ ssh guest@debian-11-server.ksd.su
guest@debian-11-server.ksd.su's password:
Linux debian-11-server 5.10.0-10-amd64 #1 SMP Debian 5.10.84-1 (2021-12-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 28 11:05:40 2022 from 192.168.1.5
```

Теперь проверьте то, что брандмауэр действительно запрещает все неразрешенные соединения. Попробуйте подключиться к нашему веб-серверу и убедитесь, что мы на данный момент не можем это сделать:



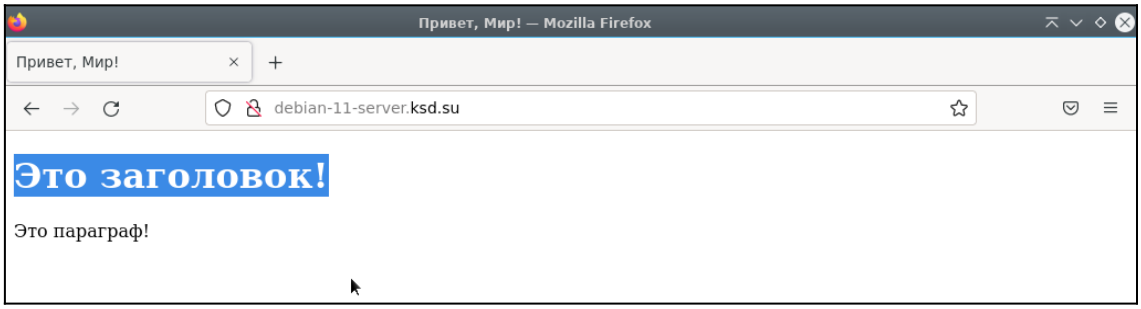
Входим на виртуальную машину Debian 11 – Server.

Разрешим на брандмауэре порты HTTP (80) и HTTPS (443), чтобы брандмауэр разрешал подключения на веб-сервер:

```
root@debian-11-server:~# ufw app info 'HTTP'
root@debian-11-server:~# ufw allow 80/tcp && ufw allow 443/tcp
root@debian-11-server:~# ufw status
```

Снова входим на виртуальную машину Debian 11 – Host.

Теперь мы можем подключиться к веб-серверу и увидеть нашу веб-страницу.



Входим на виртуальную машину Debian 11 – Server.

Разрешение диапазона портов

Вместо предоставления доступа к отдельным портам UFW позволяет нам разрешить доступ к диапазонам портов. При разрешении диапазонов портов с помощью UFW необходимо указать протокол: tcp или udp. Например, если вы хотите разрешить порты с 7100 по 7200 как для tcp, так и для udp, выполните следующую команду:

root@debian-11-server:~# ufw allow 7100:7200/tcp && ufw allow 7100:7200/udp
root@debian-11-server:~# ufw status

Вывод команды показал, что ваше правило для брандмауэра успешно применилось.

Разрешение определенных IP-адресов) на определенном порту

Чтобы разрешить доступ к определенному порту, скажем, порту 22 с вашей рабочей машины с IP-адресом 192.168.1.5, используйте любой порт, за которым следует номер порта:

root@debian-11-server:~# ufw status numbered (Показать правила брандмауэра)
root@debian-11-server:~# ufw delete allow ssh (Удалить правило брандмауэра. Можно использовать для удаления номер строки вместо полного названия правила)
root@debian-11-server:~# ufw delete allow 2222/tcp
root@debian-11-server:~# ufw allow from 192.168.1.5 to any port 22
root@debian-11-server:~# ufw status

Убедитесь, что теперь вы можете подключиться по ssh к Debian-11-Server только через Debian-11-Host.

Разрешение подсетей

Команда для разрешения подключения к подсети IP-адресов такая же, как и при использовании одного IP-адреса, с той лишь разницей, что вам нужно указать сетевую маску:

root@debian-11-server:~# ufw allow from 192.168.1.0/24
root@debian-11-server:~# ufw status

Убедитесь, что теперь вы можете подключиться к вашим почтовым ящикам или к FTP-серверу.

Запрет соединений

Политика по умолчанию для всех входящих соединений настроена на отказ, и если вы ее не изменили, UFW будет блокировать все входящие соединения, если вы специально не откроете соединение.

Допустим, вы открыли порт 22 и заметили, что компьютер Debian-11-Gateway пытается перебрать пароли (забрутфорсить) ваш SSH-сервер. Чтобы запретить все соединения с этой IP-адреса, вы можете использовать следующую команду:

```
root@debian-11-server:~# ufw deny from 192.168.1.1 to any app 'ssh'
root@debian-11-server:~# ufw delete allow from 192.168.1.0/24
root@debian-11-server:~# ufw allow from 192.168.1.0/24
root@debian-11-server:~# ufw status
```

Необходимо удалить и заново установить правило delete allow from 192.168.1.0/24 для брандмауэра, так как для него важно в каком порядке написаны эти правила. Если бы мы этого не сделали, то сначала бы применилось разрешающее правило и в таком случае запрещающее правило deny from 192.168.1.1 to any app 'ssh' не применилось.

Проверьте, что теперь у вас не получается подключиться по ssh к Debian-11-Server через Debian-11-Gateway.

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №24

Тема: Фильтрации пакетов и настройка NAT с использованием nftables

Цель работы: Сконфигурировать виртуальную машину в роли межсетевого экрана на работу с NAT с использованием nftables:

– Обновление локального индекса пакетов и установка дополнительных пакетов;

– Принцип работы nftables;

– Настройка nftables;

– Трансляция правил из iptables;

– Настройка NAT Overload;

– Сохранение текущего набора правил;

– Установка пакета с программой conntrack и проверка работы NAT Overload с помощью программы conntrack.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Фреймворк nftables – это подсистема ядра Linux, обеспечивающая фильтрацию и классификацию сетевых пакетов/датаграмм/кадров. Представляет из себя проект по замене фреймворков iptables, ip6tables, arptables, ebtables в межсетевом экране Netfilter. За счет объединения функциональности фреймворков, в nftables присутствует меньшее дублирование кода при построении правил для Netfilter и низкоуровневая оптимизация. В пространстве пользователя nftables настраивается при помощи утилиты nft.

Входим на виртуальную машину Debian 11 – Gateway.

Обновление локального индекса пакетов и установка дополнительных пакетов

root@debian-11-gateway:~# su -
Пароль: guest
root@debian-11-gateway:~# apt update

По умолчанию, nftables установлен в дистрибутиве Debian, так что устанавливать его не нужно, но дополнительно стоит поставить утилиту iptables-nftables-compat, которая позволит нам транслировать правила iptables в nftables:

root@debian-11-gateway:~# install iptables (Пакет iptables-nftables-compat находится в его зависимостях и будет установлен автоматически)

Запускаем nftables в работу:

Теперь можно включить службу nftables:

root@debian-11-gateway:~# systemctl enable nftables
root@debian-11-gateway:~# systemctl restart nftables

Принцип работы nftables

С помощью описывания правил мы можем управлять таблицами, цепочками, правилами и инструкциями для обработки пакетов:

– Таблицы определяют семейство протоколов, с которым ведётся работа. Таблицы содержат цепочки. В отличие от iptables, в nftables отсутствуют встроенные таблицы. Количество таблиц и их имена определяется пользователем. Тем не менее, каждая таблица имеет только одно семейство адресации и применяется к пакетам только этого семейства. Каждая таблица может обозначить только одно семейство (есть допущение для inet), т.е. мы не можем в рамках одной таблицы работать с ip, и arp, например. Для понимания – раньше при работе с ip4 мы так же использовали iptables, а для arp – arptables отдельно. Предусмотрено пять семейств, которые можно использовать в таблицах – ip, ip6, inet (объединяет ip и ip6), arp, bridge;

– Цепочки состоят из правил, в рамках которых обрабатываются пакеты. Доступно три типа цепочек – filter (для фильтрации), route (для маршрутизации) и nat (для NAT соответственно). Здесь могут быть использованы *хуки* – prerouting, input, forward, output, postrouting;

– Правила – это непосредственно то, с помощью чего мы описываем как будут обрабатываться пакеты;

– Инструкции (Statements) определяют, что будет сделано с пакетом, который попал под определённое условие или правило. Здесь доступны accept, drop, reject, queue, return, jump, goto, continue.

Настройка nftables

Рассмотрим базовый пример с открытием и закрытием порта с помощью nftables. Допустим, у нас есть сервис, который работает на 80 (http) порте. Мы сперва ограничим доступ к нему, а затем откроем его вновь.

Таблица семейства ip (т.е. IPv4) – это семейство по умолчанию; используется, если семейство не было указано. Семейство inet объединяет протоколы IPv4 и IPv6, что позволяет унифицировать семейства ip и ip6 и упростить создание правил. В командах ниже параметр семейство будет указываться не всегда; если его нет, то подразумевается семейство ip:

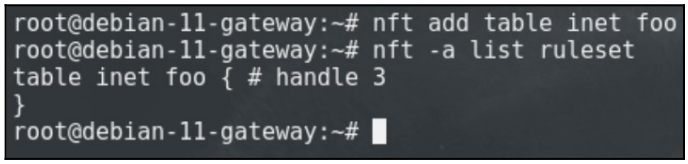
 # nft add table семейство таблица

Для начала, создадим таблицу:

root@debian-11-gateway:~# nft add table inet foo

Проверим что она появилась у нас:

root@debian-11-gateway:~# nft -a list ruleset



Чтобы добавить базовую *цепочку*, укажите *хук* и значение *приоритета*:

 # nft 'add chain семейство таблица цепочка { type тип hook хук priority приоритет ; }'

Параметр *тип* может принимать значения filter, route или nat. Для семейств адресации ipv4/ipv6/inet *хук* может принимать значения prerouting, input, forward, output или postrouting. Параметр *приоритет* задается названием приоритета или его численным значением . Цепочки с более низкими значениями обрабатываются раньше. Значение приоритета может быть отрицательным.

Теперь добавим цепочку:

root@debian-11-gateway:~# nft 'add chain inet filter input { type filter hook input priority 0; }'

```
root@debian-11-gateway:~# nft 'add chain inet foo input { type filter hook input priority 0 ; }'
root@debian-11-gateway:~# nft -a list ruleset
table inet foo { # handle 3
        chain input { # handle 1
                type filter hook input priority filter; policy accept;
        }
}
root@debian-11-gateway:~#
```

Обратите внимание, что nft использует специальные символы, такие как фигурные скобки и точку с запятой. Если вы запускаете эти команды из оболочки, такой как bash, все специальные символы должны быть заключены в кавычки. Кроме того, вы можете использовать команду
 # nft -i
и запустить nft в интерактивном режиме.

Чтобы добавить правило в цепочку, укажите следующие значения:

 # nft add rule семейство таблица цепочка handle маркер оператор

Правило будет прикреплено после *маркер*, который можно не указывать. Если маркер (handle) не задать, то правило добавится в конец цепочки.

Чтобы добавить правило перед определённой позицией, выполните:

 # nft insert rule семейство таблица цепочка handle маркер оператор

Если маркер не указан, то правило добавится в начало цепочки.

Добавим правило, которым ограничим доступ к 80 порту:

root@debian-11-gateway:~# nft add rule inet foo input tcp dport 80 drop

```
root@debian-11-gateway:~# nft add rule inet foo input tcp dport 80 drop
root@debian-11-gateway:~# nft -a list ruleset
table inet foo { # handle 3
        chain input { # handle 1
                type filter hook input priority filter; policy accept;
                tcp dport 80 drop # handle 2
        }
}
root@debian-11-gateway:~#
```

Логично предположить, что для открытия порта нам нужно выполнить:

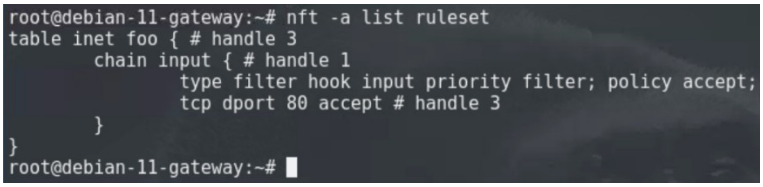
root@debian-11-gateway:~# nft add rule inet foo input tcp dport 80 accept

Но получить доступ к 80 порту после этого нам не удастся, и, если мы заглянем в список правил, будет понятно почему.

```
root@debian-11-gateway:~# nft add rule inet foo input tcp dport 80 accept
root@debian-11-gateway:~# nft -a list ruleset
table inet foo { # handle 3
        chain input { # handle 1
                type filter hook input priority filter; policy accept;
                tcp dport 80 drop # handle 2
                tcp dport 80 accept # handle 3
        }
}
root@debian-11-gateway:~#
```

Нам нужно удалить правило с drop, для этого мы обращаем внимание на отметку handle 2 в выводе, и с её помощью удаляем запрещающее правило:

root@debian-11-gateway:~# nft delete rule inet foo input handle 2



Теперь мы без проблем можем получить доступ к 80 порту нашего сервера.

Если мы хотим полной очистки правил в таблице, то выполняем следующую команду:

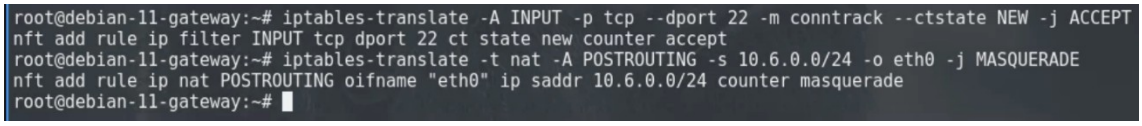
root@debian-11-gateway:~# nft delete table inet foo

После чего, вывод команды «nft -a list ruleset» будет пустым.

Трансляция правил из iptables

Для того, чтобы быстро перевести имеющееся iptables правило в nft формат, можно воспользоваться утилитой iptables-translate. Работает она очень просто:

root@debian-11-gateway:~# iptables-translate -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
root@debian-11-gateway:~# iptables-translate -t nat -A POSTROUTING -s 10.6.0.0/24 -o eth0 -j MASQUERADE



Мы просто вводим правило, и получаем его аналог, который можем использовать в nftables.

Настройка NAT Overload

Network Address Translation (NAT) – это механизм в сетях TCP/IP, позволяющий изменять IP адрес в заголовке пакета, проходящего через устройство маршрутизации трафика. Принимая пакет от локального компьютера, маршрутизатор смотрит на IP-адрес назначения. Если это локальный адрес, то пакет пересылается другому локальному компьютеру. Если нет, то пакет надо переслать наружу в интернет.

Маршрутизатор подменяет обратный IP-адрес пакета на свой внешний (видимый из интернета) IP-адрес и меняет номер порта (чтобы различать ответные пакеты, адресованные разным локальным компьютерам). Комбинацию, нужную для обратной подстановки, маршрутизатор сохраняет у себя во временной таблице. Через некоторое время после того, как клиент и сервер закончат обмениваться пакетами, маршрутизатор сотрет у себя в таблице запись об *n*-ом порте за сроком давности.

Основная функция NAT – это сохранение публичных адресов, однако дополнительной функцией является конфиденциальность сети, путем скрытия внутренних IPv4 адресов от внешней.

Существует множество типов технологии NAT, однако основными принято считать: Статический NAT (Static Network Address Translation), Динамический NAT (Dynamic Network Address Translation) и Перегруженный NAT (Network Address Translation Overload). Мы будем конфигурировать последний тип NAT.

NAT Overload, Many-to-One, PAT (Port Address Translation) и IP Masquerading, однако в большинстве источников указывается как NAT Overload. Перегруженный NAT – это форма динамического NAT, который отображает несколько незарегистрированных адресов в единственный зарегистрированный IP-адрес, используя различные порты. При перегрузке каждый компьютер в частной сети транслируется в тот же самый адрес, но с различным номером порта.

Сначала добавьте новую таблицу:

```
root@debian-11-gateway:~# nft add table inet nat
```

Добавьте цепочки prerouting и postrouting в вашу таблицу:

```
root@debian-11-gateway:~# nft -- add chain inet nat prerouting { type nat hook prerouting priority -100 \; }
root@debian-11-gateway:~# nft add chain inet nat postrouting { type nat hook postrouting priority 100 \; }
```

Даже если вы не добавите ни одного правила в цепочку предварительной маршрутизации (prerouting), инфраструктура системы nftables требует, чтобы эта цепочка соответствовала входящим ответам на пакеты.

Выведите на экран правила созданной вами таблицы nat семейства inet:

```
root@debian-11-gateway:~# nft list table inet nat
```

Выведите на экран все правила в цепочке:

```
root@debian-11-gateway:~#  nft list chain inet nat prerouting
root@debian-11-gateway:~#  nft list chain inet nat postrouting
```

Обратите внимание, что вы должны передать параметр -- команде nft, чтобы оболочка не интерпретировала отрицательное значение приоритета как параметр команды nft.

Чтобы добавить правило в цепочку, нужно выполнить:

 # nft add rule семейство таблица цепочка handle маркер оператор

Добавьте правило в цепочку postrouting, которое соответствует исходящим пакетам на интерфейсе ens4:

```
root@debian-11-gateway:~# nft add rule inet nat postrouting oifname[23] "ens4" masquerade
```

После чего установим пакет netfilter-persistent для автоматической загрузки правил nftables:

```
root@debian-11-gateway:~# apt install netfilter-persistent
```

[23] Интерфейс-отправитель

netfilter-persistent – это программа, которая позволяет сохранять и автоматически загружать правила брандмауэра, не оглядываясь на используемую систему инициализации сети. Она позволяет удобно управлять брандмауэром, так если вы внесли изменения в файл с правилами и хотите их применить, то выполните:

root@debian-11-gateway:~# netfilter-persistent reload

Для сохранения текущей конфигурации отдайте команду:

root@debian-11-gateway:~# netfilter-persistent save

А если вам надо полностью очистить конфигурацию брандмауэра, то можно набрать следующую команду:

 # netfilter-persistent flush (Не применяйте эту команду)

Сохранение текущего набора правил

Выходные данные команды nft list ruleset также являются допустимым входным файлом для нее. Текущий набор правил можно сохранить в файл, а затем загрузить обратно:

root@debian-11-gateway:~# nft -s list ruleset | tee /etc/nftables.conf

Чтобы сбросить текущий набор правил примените следующую строку:

root@debian-11-gateway:~# nft flush ruleset (Введите эту команду только после того как выполните и проверите работу целиком)

Установка пакета с программой conntrack и проверка работы NAT Overload с помощью программы conntrack

Утилиты iptables или nftables не выполняют NAT: это делает Netfilter. Утилиты iptables и nftables используют хуки в Netfilter, чтобы отдавать некоторые «приказы» для создания новых состояний NAT. После этого все обрабатывается непосредственно Netfilter. В связи с этой причиной, чтобы проверить состояние NAT нужен Netfilter и его подсистема conntrack.

root@debian-11-gateway:~# apt install conntrack
root@debian-11-gateway:~# conntrack -L -n

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №25

Тема: Настройка fail2ban

Цель работы: Сконфигурировать виртуальную машину в работу с fail2ban:

– Обновление локального индекса пакетов;

– Установка пакета с программой fail2ban;

– Базовая настройка fail2ban;

– Увеличение времени бана;

– Проверка работы fail2ban;

– Блокировка IP-адреса навсегда;

– Примеры использования fail2ban-client;

– Белый список IP-адресов;

– Настройка количества попыток авторизации по умолчанию;

– Дополнительные рекомендации по настройке fail2ban;

– Настройка уведомлений fail2ban в Telegram.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

fail2ban – это простой в использовании локальный сервис, который отслеживает log-файлы запущенных программ, и на основании различных условий блокирует по IP-адресу найденных нарушителей.

Программа умеет бороться с различными атаками на все популярные *NIX-сервисы, такие как Apache, Nginx, ProFTPD, vsftpd, Exim, Postfix, named, и т.д.

В первую очередь Fail2ban известен благодаря готовности «из коробки» к защите SSH–сервера от атак типа «брутфорс», то есть к защите SSH от перебора паролей.

Входим на виртуальную машину Debian 11 – Host.

Обновление локального индекса пакетов

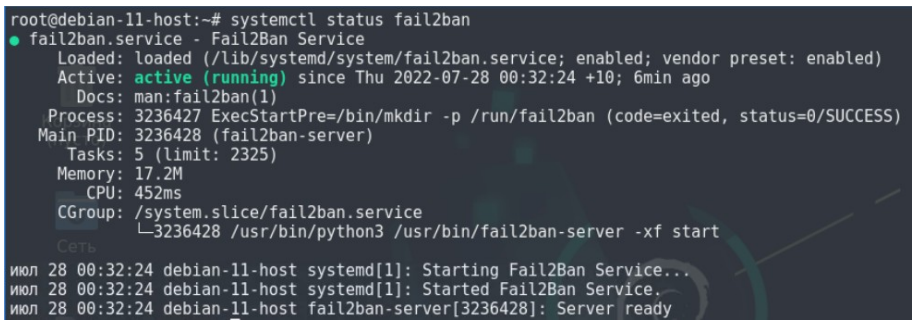
```
root@debian-11-host:~# su -
Пароль: guest
root@debian-11-host:~# apt update
```

Установка пакета с программой fail2ban

```
root@debian-11-host:~# apt install fail2ban curl
```

По умолчанию fail2ban после установки должен быть активен и включен. Чтобы это проверить, используйте следующую команду systemctl:

```
root@debian-11-host:~# systemctl status fail2ban
```



Затем, чтобы fail2ban автоматически запускался при загрузке системы, используйте следующее:

root@debian-11-host:~# systemctl enable fail2ban

На крайний случай, вы можете проверить версию fail2ban:

root@debian-11-host:~# fail2ban-client –-version

Таким образом вы можете убедиться, что ваша версия программы – это одна из последних стабильных версий.

Базовая настройка fail2ban

После завершения установки нам теперь нужно выполнить некоторые приготовления и базовую настройку. fail2ban поставляется с двумя конфигурационными файлами, которые расположены в /etc/fail2ban/jail.conf и /etc/fail2ban/jail.d/defaults-debian.conf. Не нужно изменять эти конфигурационные файлы! Они являются вашими оригиналами и будут заменены при любом обновлении fail2ban в будущем.

В таком случае можно задаться вопросом, как нужно настраивать fail2ban, если мы его обновили и потеряли свои настройки. Просто мы будем записывать изменения в копии файлов, оканчивающиеся на .local вместо .conf, поскольку fail2ban всегда сначала читает файлы .local перед загрузкой .conf, если он смог их найти.

Чтобы сделать это, используйте следующую команду:

root@debian-11-host:~# cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

Теперь откройте файл конфигурации, чтобы продолжить настройку:

root@debian-11-host:~# nano -c /etc/fail2ban/jail.local

На данном этапе fail2ban уже готов к работе, базовая защита SSH сервера от перебора паролей будет включена по умолчанию. Но лучше все же следовать некоторым рекомендациям ниже.

У программы два основных файла конфигурации:

– /etc/fail2ban/fail2ban.conf – отвечает за настройки запуска процесса fail2ban;

– /etc/fail2ban/jail.conf – содержит настройки защиты конкретных сервисов, в том числе sshd.

Файл jail.conf поделён на секции, так называемые «изоляторы» (jails), каждая секция отвечает за определённый сервис и тип атаки:

```
[DEFAULT]
ignoreip = 127.0.0.1/8
bantime  = 600
maxretry = 3

[ssh]
enabled  = true
port     = ssh
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 3
```

* Не вписывайте эти данные в конфигурационный файл.

Параметры из секции [DEFAULT] применяются ко всем остальным секциям, если не будут переопределены.

Секция [ssh] отвечает за защиту SSH от повторяющихся неудачных попыток авторизации на SSH–сервере, проще говоря, «брутфорса».

Подробнее по каждому из основных параметров файла jail.conf:

– ignoreip – IP–адреса, которые не должны быть заблокированы. Можно задать список IP-адресов, разделённых пробелами, маску подсети, или имя DNS-сервера;

– bantime – время бана в секундах, по истечении которого IP–адрес удаляется из списка заблокированных;

– maxretry – количество подозрительных совпадений, после которых применяется правило. В контексте [ssh] – это число неудавшихся попыток логина, после которых происходит блокировка;

– enabled – значение true указывает что данный «изолятор» активен, false выключает действие изолятора;

– port – указывает на каком порту или портах запущен целевой сервис. Стандартный порт SSH-сервера – 22, или его буквенное наименование – ssh.

– filter – имя фильтра с регулярными выражениями, по которым идёт поиск «подозрительных совпадений» в журналах сервиса. Фильтру sshd соответствует файл «/etc/fail2ban/filter.d/sshd.conf»;

– logpath – путь к файлу журнала, который программа fail2ban будет обрабатывать с помощью заданного ранее фильтра. Вся история удачных и неудачных входов в систему, в том числе и по SSH, по умолчанию записывается в log-файл «/var/log/auth.log».

Увеличение времени бана

Первая настройка, с которой вы столкнетесь, – это увеличение времени бана. По умолчанию в fail2ban оно составляет 10 минут. Время блокировки – это промежуток времени (в секундах), в течение которого IP-адрес блокируется после определенного количества неудачных попыток аутентификации. Предпочтительно установить это время, достаточное для прерывания злонамеренных действий пользователя. Однако не должно быть слишком много времени, чтобы законный пользователь был ошибочно забанен за неудачные попытки аутентификации. Обратите внимание, что когда законный пользователь забанен, вы также можете вручную разблокировать его, вместо того, чтобы ждать, пока истечет время блокировки.

Время бана можно изменить, настроив параметр bantime в файле конфигурации fail2ban. Добавьте значение параметра bantime на желаемое (строка 279). Например, чтобы заблокировать IP-адреса, скажем, на 20 секунд , вам нужно будет изменить существующее значение bantime на 20. Также не забудьте сменить порт, на котором работает fail2ban, так как мы его изменили в конфигурационном файле OpenSSH-Server.

Затем сохраните и выйдите из файла jail.local.

Перезапустите службу fail2ban следующим образом:

```
root@debian-11-host:~# systemctl restart fail2ban
root@debian-11-host:~# systemctl status fail2ban
```

Проверка работы fail2ban

Перед проверкой работы fail2ban, не забудьте временно прописать в конфигурационном файле SSH-сервера «/etc/ssh/sshd_config» строки «PubkeyAuthentication no» и «PasswordAuthentication yes», иначе вы не сможете проверить работу этой сетевой службы.

Входим на виртуальную машину Windows 10 – Host.

Попробуйте 5 раз ошибиться при авторизации на виртуальную машину Debian 11 – Host, чтобы проверить, что fail2ban по умолчанию работает. Это можно сделать в командной строке Windows (cmd):

```
C:\Users\Win10>ssh -p 2222 guest@debian-11-host.ksd.su
```

Входим на виртуальную машину Debian 11 – Host.

После выполнения этих действий, тот IP-адрес, с которого вы сделали определенное количество неудачных попыток подключения, будет заблокирован на 20 секунд. Вы также можете подтвердить это, посмотрев логи:

root@debian-11-host:~# cat /var/log/fail2ban.log

```
2022-08-01 18:25:35,744 fail2ban.jail           [2370210]: INFO    Creating new jail 'sshd'
2022-08-01 18:25:35,757 fail2ban.jail           [2370210]: INFO    Jail 'sshd' uses pyinotify {}
2022-08-01 18:25:35,764 fail2ban.jail           [2370210]: INFO    Initiated 'pyinotify' backend
2022-08-01 18:25:35,766 fail2ban.filter         [2370210]: INFO      maxLines: 1
2022-08-01 18:25:35,784 fail2ban.filter         [2370210]: INFO      maxRetry: 5
2022-08-01 18:25:35,784 fail2ban.filter         [2370210]: INFO      findtime: 600
2022-08-01 18:25:35,784 fail2ban.actions        [2370210]: INFO      banTime: 20
2022-08-01 18:25:35,784 fail2ban.filter         [2370210]: INFO      encoding: UTF-8
2022-08-01 18:25:35,785 fail2ban.filter         [2370210]: INFO    Added logfile: '/var/log/auth.log' (pos = 1
8960, hash = e3c9130fb19cdb0d2e7ad6cbd8fa443db79cdc14)
2022-08-01 18:25:35,791 fail2ban.jail           [2370210]: INFO    Jail 'sshd' started
2022-08-01 18:26:11,030 fail2ban.filter         [2370210]: INFO    [sshd] Found 192.168.1.18 - 2022-08-01 18:2
6:10
2022-08-01 18:26:15,271 fail2ban.filter         [2370210]: INFO    [sshd] Found 192.168.1.18 - 2022-08-01 18:2
6:15
2022-08-01 18:26:21,630 fail2ban.filter         [2370210]: INFO    [sshd] Found 192.168.1.18 - 2022-08-01 18:2
6:21
2022-08-01 18:27:24,371 fail2ban.filter         [2370210]: INFO    [sshd] Found 192.168.1.18 - 2022-08-01 18:2
7:24
2022-08-01 18:27:27,823 fail2ban.filter         [2370210]: INFO    [sshd] Found 192.168.1.18 - 2022-08-01 18:2
7:27
2022-08-01 18:27:27,963 fail2ban.actions        [2370210]: NOTICE  [sshd] Ban 192.168.1.18
```

Блокировка IP-адреса навсегда

Вы также можете навсегда блокировать исходящие IP-адреса в fail2ban. Чтобы это сделать, вам нужно будет изменить значение параметра bantime на -1. Для этого сначала отредактируйте файл конфигурации jail.local следующим образом:

root@debian-11-host:~# nano -c /etc/fail2ban/jail.local

```
[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
port    = 2222
logpath = %(sshd_log)s
backend = %(sshd_backend)s
bantime = -1
```

root@debian-11-host:~# systemctl restart fail2ban
root@debian-11-host:~# systemctl status fail2ban

Входим на виртуальную машину Windows 10 – Host.

Попробуйте снова выполнить пять неудачных попыток входа на виртуальную машину Windows 10 – Host, чтобы проверить, что fail2ban заблокирует ваш IP-адрес:

C:\Users\Win10>ssh -p 2222 guest@debian-11-host.ksd.su

```
C:\Users\gu3st~>ssh -p 2222 guest@192.168.1.15
ssh: connect to host 192.168.1.15 port 2222: Connection timed out
```

Входим на виртуальную машину Debian 11 – Host.

Затем посмотрите на log-файлы на SSH-сервере:

root@debian-11-host:~# cat /var/log/fail2ban.log

Примеры использования fail2ban-client

Теперь, когда мы поработали с fail2ban, нам требуется узнать некоторые основные рабочие команды fail2ban-client.

Чтобы заблокировать IP-адрес, выполните следующую команду в терминале:

root@debian-11-host:~# fail2ban-client set sshd banip windows-10-host.ksd.su

Чтобы разблокировать IP-адрес, выполните команду:

root@debian-11-host:~# fail2ban-client set sshd unbanip windows-10-host.ksd.su

```
2022-08-03 03:03:47,433 fail2ban.jail           [311422]: INFO    Jail 'sshd' started
2022-08-03 03:03:47,445 fail2ban.actions        [311422]: NOTICE  [sshd] Restore Ban 192.168.1.10
2022-08-03 03:05:20,427 fail2ban.actions        [311422]: NOTICE  [sshd] Unban 192.168.1.10
```

После ввода вышенаписанных команд попробуйте проверить их работоспособность на виртуальной машине Windows-10-Host.

Команда для вызова справки, если вам нужно найти дополнительные настройки или получить помощь по конкретной команде:

root@debian-11-host:~# fail2ban-client -h

Белый список IP-адресов

Далее по списку мы сталкиваемся с опциями белого списка (Whitelist). Давайте их раскомментируем и впишем подсеть с IP-адресами, которые мы хотим добавить в белый список. Если вы добавляете сразу несколько IP-адресов, то не забудьте поставить пробел или запятую между ними. Вы также можете внести в белый список диапазон IP-адресов:

root@debian-11-host:~# nano -c /etc/fail2ban/jail.local

```
# "ignoreip" can be a list of IP addresses, CIDR masks or DNS hosts. Fail2ban
# will not ban a host which matches an address in this list. Several addresses
# can be defined using space (and/or comma) separator.
ignoreip = 127.0.0.1/8 ::1 192.168.1.0/24
```

root@debian-11-host:~# systemctl restart fail2ban
root@debian-11-host:~# systemctl status fail2ban

Входим на виртуальную машину Windows 10 – Host.

Попробуйте еще раз выполнить пять неудачных попыток входа на виртуальную машину Windows 10 – Host, чтобы проверить, что fail2ban вас не заблокирует, так как ваш IP-адрес находится в белом списке:

C:\Users\Win10>ssh -p 2222 guest@debian-11-host.ksd.su

Входим на виртуальную машину Debian 11 – Host.

Затем посмотрите на log-файлы на SSH-сервере:

root@debian-11-host:~# cat /var/log/fail2ban.log

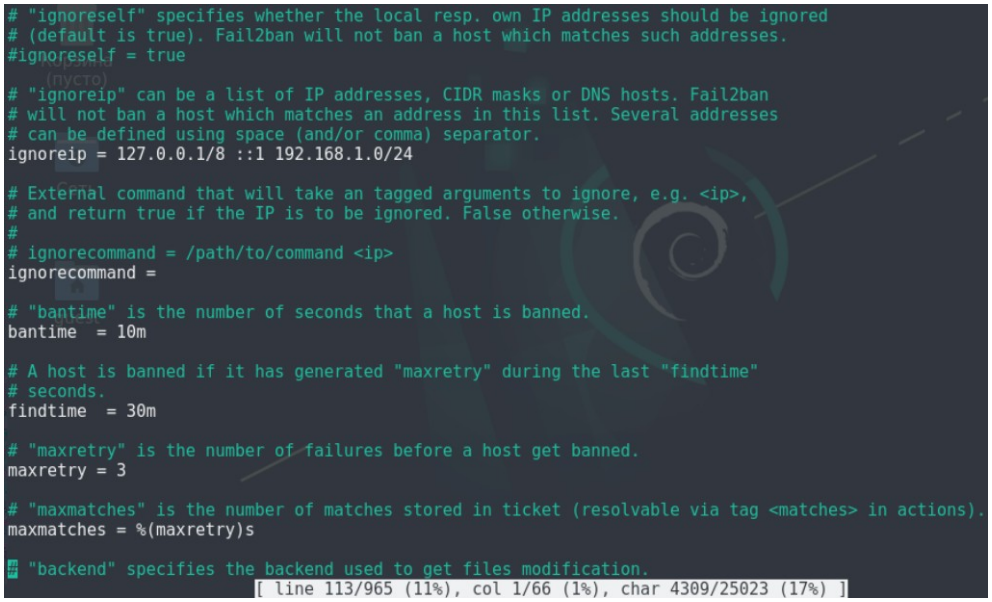
```
2022-08-01 23:33:20,198 fail2ban.jail           [113549]: INFO    Creating new jail 'sshd'
2022-08-01 23:33:20,215 fail2ban.jail           [113549]: INFO    Jail 'sshd' uses pyinotify {}
2022-08-01 23:33:20,220 fail2ban.jail           [113549]: INFO    Initiated 'pyinotify' backend
2022-08-01 23:33:20,223 fail2ban.filter         [113549]: INFO      maxLines: 1
2022-08-01 23:33:20,240 fail2ban.filter         [113549]: INFO      maxRetry: 5
2022-08-01 23:33:20,240 fail2ban.filter         [113549]: INFO      findtime: 600
2022-08-01 23:33:20,240 fail2ban.actions        [113549]: INFO      banTime: -1
2022-08-01 23:33:20,241 fail2ban.filter         [113549]: INFO      encoding: UTF-8
2022-08-01 23:33:20,241 fail2ban.filter         [113549]: INFO    Added logfile: '/var/log/auth.log' (pos = 33653, hash = e3
c9130fb19cdb0d2e7ad6cbd8fa443db79cdc14)
2022-08-01 23:33:20,244 fail2ban.jail           [113549]: INFO    Jail 'sshd' started
2022-08-01 23:47:59,511 fail2ban.filter         [113549]: INFO    [sshd] Ignore 192.168.1.1 by ip
2022-08-01 23:48:02,746 fail2ban.filter         [113549]: INFO    [sshd] Ignore 192.168.1.1 by ip
2022-08-01 23:48:06,967 fail2ban.filter         [113549]: INFO    [sshd] Ignore 192.168.1.1 by ip
2022-08-01 23:48:12,501 fail2ban.filter         [113549]: INFO    [sshd] Ignore 192.168.1.1 by ip
2022-08-01 23:48:18,412_fail2ban.filter         [113549]: INFO    [sshd] Ignore 192.168.1.1 by ip
```

Настройка количества попыток авторизации по умолчанию

По умолчанию время бана составляет 10 минут, если 5 повторных попыток входа было совершенно в течение 10 минут. Объяснение этому заключается в том, что «изолятор» fail2ban с фильтрацией заблокирует злоумышленника на 10 минут после того, как он повторит ту же атаку через 10 минут (время поиска) 5 раз (повторные попытки). Здесь вы можете установить некоторые настройки запрета по умолчанию. Однако, когда вы попадаете в «изолятор», рекомендуется установить разное время бана, так как некоторые блокировки должны автоматически быть длиннее других, включая повторные попытки, которые должны быть меньше или больше.

Выполните настройки, изображенные на рисунке ниже. Не забудьте временно закомментировать ранее добавленную строку с белым списком IP-адресов:

root@debian-11-host:~# nano -c /etc/fail2ban/jail.local



root@debian-11-host:~# systemctl restart fail2ban
root@debian-11-host:~# systemctl status fail2ban

Входим на виртуальную машину Windows 10 – Host.

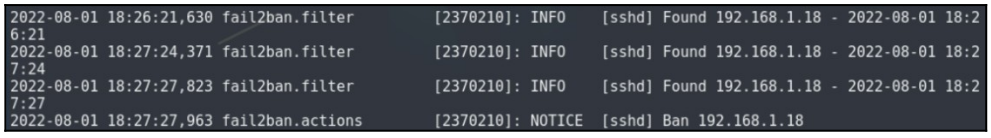
Попробуйте выполнить три неудачные попытки входа на виртуальную машину Debian 11 – Host, чтобы проверить, что ваша конфигурация fail2ban изменилась:

C:\Users\Win10>ssh -p 2222 guest@debian-11-host.ksd.su

Входим на виртуальную машину Debian 11 – Host.

Затем посмотрите на log-файлы на SSH-сервере:

root@debian-11-host:~# cat /var/log/fail2ban.log

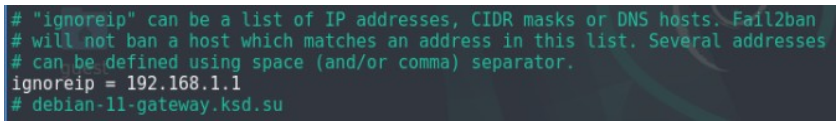


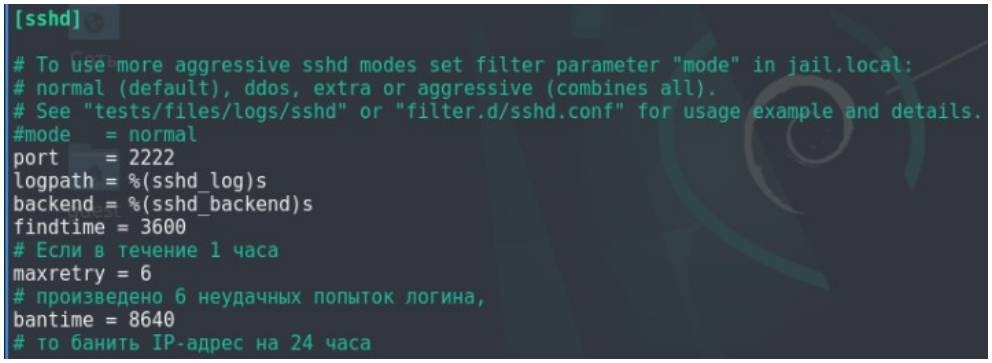
Дополнительные рекомендации по настройке fail2ban

Не рекомендуется оставлять параметр ignoreip со значением по умолчанию 127.0.0.1/8, это создаёт очевидную угрозу в многопользовательских системах – если злоумышленник получил доступ хотя-бы к одному shell–аккаунту, то он имеет возможность беспрепятственно запустить брутфорс-программу для атаки на root или других пользователей прямо с этого-же сервера.

Новая опция findtime – определяет длительность интервала в секундах, за которое событие должно повториться определённое количество раз, после чего санкции вступят в силу. Если специально не определить этот параметр, то будет установлено значение по умолчанию равное 600 (10 минут). Проблема в том, что ботнеты, участвующие в «медленном брутфорсе», умеют обманывать стандартное значение. Иначе говоря, при maxretry равным 6, атакующий может проверить 5 паролей, затем выждать 10 минут, проверить ещё 5 паролей, повторять это снова и снова, и его IP-адрес забанен не будет. В целом, это не угроза, но всё же лучше блокировать таких ботов.

root@debian-11-host:~# nano -c /etc/fail2ban/jail.local





root@debian-11-host:~# systemctl restart fail2ban
root@debian-11-host:~# systemctl status fail2ban

Входим на виртуальную машину Debian 11 – Server.

Попробуйте выполнить шесть неудачных попыток входа на Debian 11 – Host, чтобы проверить, что ваша конфигурация fail2ban применилась:

root@debian-11-server:~# ssh -p 2222 guest@debian-11-host.ksd.su

Входим на виртуальную машину Debian 11 – Host.

Затем посмотрите на log-файлы на SSH-сервере:

root@debian-11-host:~# cat /var/log/fail2ban.log

```
2022-08-03 14:30:50,225 fail2ban.server         [470534]: INFO    Starting Fail2ban v0.11.2
2022-08-03 14:30:50,226 fail2ban.observer       [470534]: INFO    Observer start...
2022-08-03 14:30:50,234 fail2ban.database       [470534]: INFO    Connected to fail2ban persistent database '/
var/lib/fail2ban/fail2ban.sqlite3'
2022-08-03 14:30:50,234 fail2ban.jail           [470534]: INFO    Creating new jail 'sshd'
2022-08-03 14:30:50,245 fail2ban.jail           [470534]: INFO    Jail 'sshd' uses pyinotify {}
2022-08-03 14:30:50,252 fail2ban.jail           [470534]: INFO    Initiated 'pyinotify' backend
2022-08-03 14:30:50,254 fail2ban.filter         [470534]: INFO      maxLines: 1
2022-08-03 14:30:50,271 fail2ban.filter         [470534]: INFO      maxRetry: 6
2022-08-03 14:30:50,272 fail2ban.filter         [470534]: INFO      findtime: 3600
2022-08-03 14:30:50,272 fail2ban.actions        [470534]: INFO      banTime: 8640
2022-08-03 14:30:50,272 fail2ban.filter         [470534]: INFO      encoding: UTF-8
2022-08-03 14:30:50,272 fail2ban.filter         [470534]: INFO    Added logfile: '/var/log/auth.log' (pos = 74
291, hash = e3c9130fb19cdb0d2e7ad6cbd8fa443db79cdc14)
2022-08-03 14:30:50,275 fail2ban.jail           [470534]: INFO    Jail 'sshd' started
2022-08-03 14:31:09,865 fail2ban.filter         [470534]: INFO    [sshd] Found 192.168.1.10 - 2022-08-03 14:31
:09
2022-08-03 14:31:15,006 fail2ban.filter         [470534]: INFO    [sshd] Found 192.168.1.10 - 2022-08-03 14:31
:15
2022-08-03 14:31:18,913 fail2ban.filter         [470534]: INFO    [sshd] Found 192.168.1.10 - 2022-08-03 14:31
:18
2022-08-03 14:31:25,072 fail2ban.filter         [470534]: INFO    [sshd] Found 192.168.1.10 - 2022-08-03 14:31
:25
2022-08-03 14:31:30,217 fail2ban.filter         [470534]: INFO    [sshd] Found 192.168.1.10 - 2022-08-03 14:31
:30
2022-08-03 14:31:36,402 fail2ban.filter         [470534]: INFO    [sshd] Found 192.168.1.10 - 2022-08-03 14:31
:36
2022-08-03 14:31:36,543 fail2ban.actions        [470534]: NOTICE  [sshd] Ban 192.168.1.10
```

В этом задании затронуты только базовые возможности fail2ban, применимые для защиты SSH на типовом Linux-сервере.

Настройка уведомлений fail2ban в Telegram

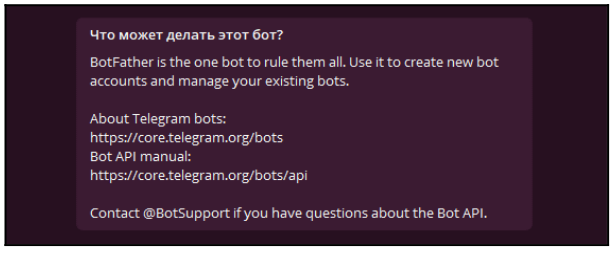
fail2ban довольно мощная программа для блокировки попыток взлома вашего сервера. Давайте теперь добавим к нему возможность получать уведомления в Telegram.

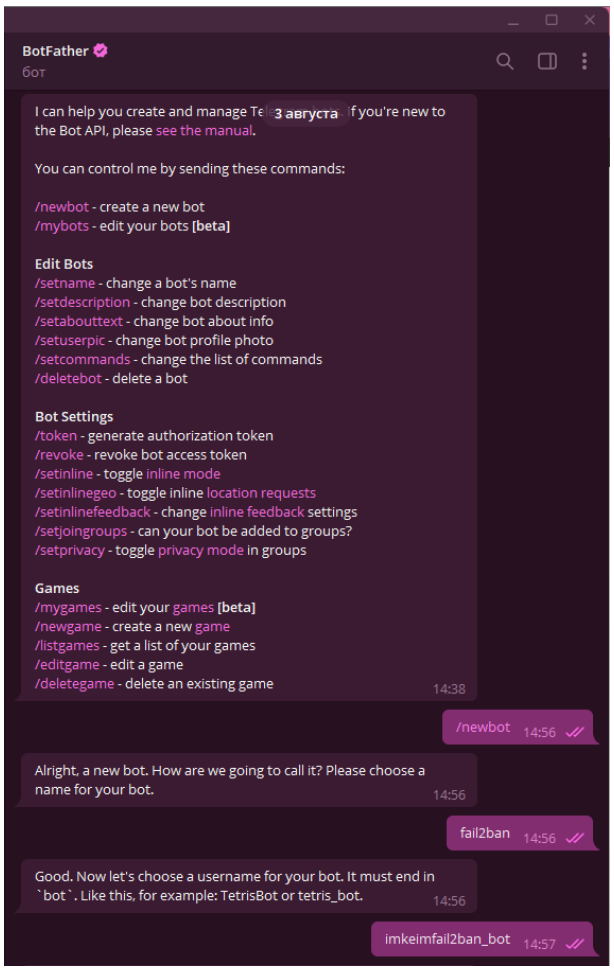
Конечно в стандартной комплектации уже имеются скрипты для отправки уведомлений по электронной почте, но намного удобнее читать такие сообщения в мессенджере.

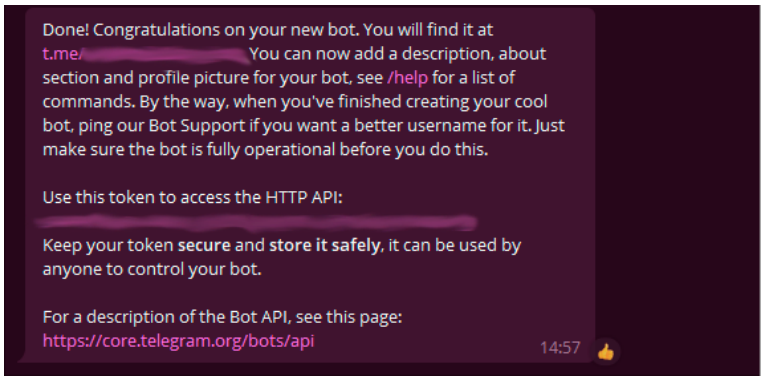
Регистрация бота Telegram. Первым делом необходимо зарегистрировать бота, именно он и будет отправлять вам сообщения с информацией о блокировках. Ниже находятся скриншоты с инструкцией как зарегистрировать бота в Telegram:

– Найти в Telegram бота @botfather;

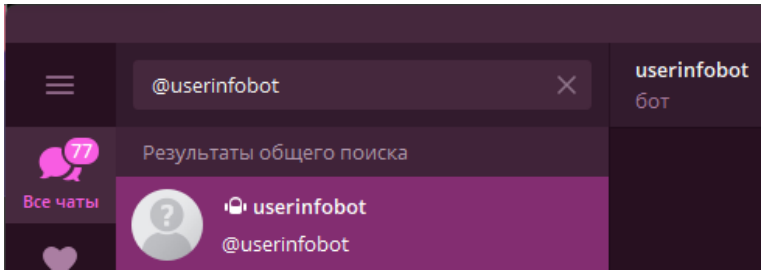
– Следуя простым инструкциям этого бота создать своего бота и получить «Access Token». Этот токен нам понадобится в скрипте отправки сообщений.

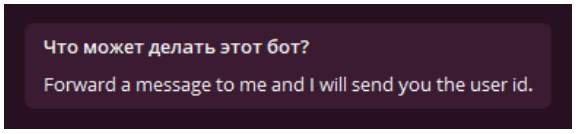






После того как мы создадим бота необходимо получить ID вашей учетной записи в Telegram. Сделать это можно например с помощью бота @userinfobot. Он покажет вам ваш ID или ID любой другой учетной записи.

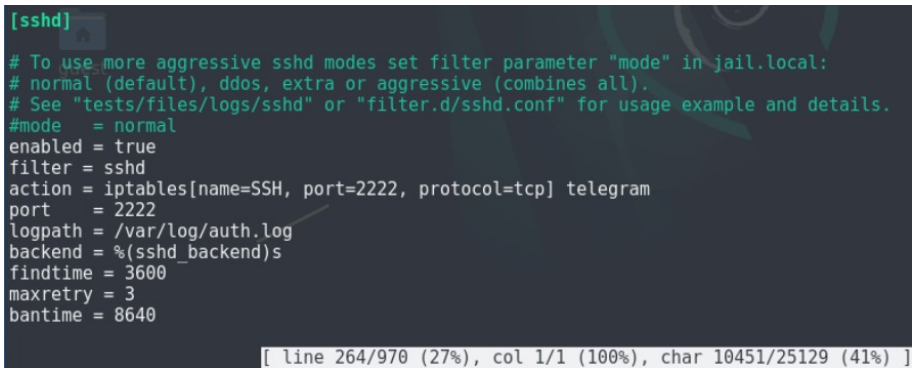




Настройка fail2ban на отправку уведомлений через Telegram.

Первым делом давайте снова обновим конфигурацию нашего «изолятора»:

root@debian-11-host:~# nano -c /etc/fail2ban/jail.local



После создайте директорию /etc/fail2ban/scripts и добавьте туда скрипт отправки сообщений в Telegram fail2ban-telegram.sh:

root@debian-11-host:~# mkdir /etc/fail2ban/scripts
root@debian-11-host:~# touch /etc/fail2ban/scripts/fail2ban-telegram.sh
root@debian-11-host:~# chmod ugo+x /etc/fail2ban/scripts/fail2ban-telegram.sh
root@debian-11-host:~# nano -c /etc/fail2ban/scripts/fail2ban-telegram.sh

```
#!/bin/bash

# Sends text messages using Telegram
# to alert webmaster of banning.

# Require one argument, one of the following
# start
# stop
# ban
# unban
# Optional second argument: Ip for ban/unband


# Display usage information
function show_usage {
  echo "Usage: $0 action <ip>"
  echo "Where action start, stop, ban, unban"
  echo "and IP is optional passed to ban, unban"
  exit
}


# Send notification
function send_msg {
  apiToken=Вставьте_сюда_ваш_API_Token
```

```
  chatId=Вставьте_сюда_ваш_ID_чата
  url="https://api.telegram.org/bot$apiToken/sendMessage"

  curl -s -X POST $url -d chat_id=$chatId -d text="$1"
  exit
}


# Check for script arguments
if [ $# -lt 1 ]
then
  show_usage
fi


# Take action depending on argument
if [ "$1" = 'start' ]
then
  msg='Fail2ban+just+started.'
  send_msg $msg
elif [ "$1" = 'stop' ]
then
  msg='Fail2ban+just+stoped.'
  send_msg $msg
elif [ "$1" = 'ban' ]
then
  msg=$([ "$2" != '' ] && echo "Fail2ban+just+banned+$2" || echo
'Fail2ban+just+banned+an+ip.' )
  send_msg $msg
elif [ "$1" = 'unban' ]
then
  msg=$([ "$2" != '' ] && echo "Fail2ban+just+unbanned+$2" || echo
"Fail2ban+just+unbanned+an+ip." )
  send_msg $msg
else
  show_usage
fi
```

Скрипт принимает текст сообщения со стандартного ввода и отправляет его указанному в скрипте ID. Следующим шагом требуется создать и поместить файл конфигурации telegram.conf в директорию /etc/fail2ban/action.d/:

root@debian-11-host:~# nano -c /etc/fail2ban/action.d/telegram.conf

```
# Fail2Ban configuration file
#
# Author: imKeim
#
#

[Definition]
```

```
# Option:  actionstart
# Notes.:  command executed once at the start of Fail2Ban.
# Values:  CMD
#
actionstart = /etc/fail2ban/scripts/fail2ban-telegram.sh start

# Option:  actionstop
# Notes.:  command executed once at the end of Fail2Ban
# Values:  CMD
#
actionstop = /etc/fail2ban/scripts/fail2ban-telegram.sh stop

# Option:  actioncheck
# Notes.:  command executed once before each actionban command
# Values:  CMD
#
actioncheck =

# Option:  actionban
# Notes.:  command executed when banning an IP. Take care that the
#          command is executed with Fail2Ban user rights.
# Tags:    See jail.conf(5) man page
# Values:  CMD
#
actionban = /etc/fail2ban/scripts/fail2ban-telegram.sh ban <ip>

# Option:  actionunban
# Notes.:  command executed when unbanning an IP. Take care that the
#          command is executed with Fail2Ban user rights.
# Tags:    See jail.conf(5) man page
# Values:  CMD
#
actionunban = /etc/fail2ban/scripts/fail2ban-telegram.sh unban <ip>

[Init]

init = 123
```

Именно к этому файлу будет обращаться fail2ban перед тем, как отправить сообщение. После всех этих манипуляций не забудьте перезагрузить демона fail2ban:

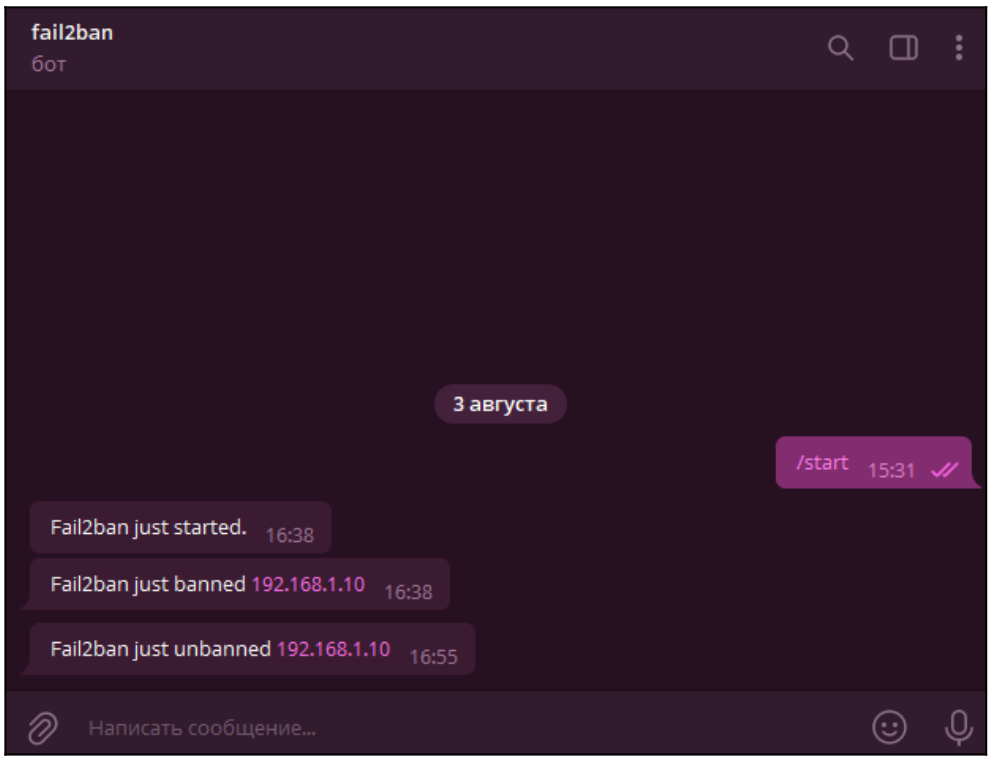
```
root@debian-11-host:~# systemctl restart fail2ban
root@debian-11-host:~# systemctl status fail2ban
```

Входим на виртуальную машину Debian 11 – Server.

Попробуйте выполнить три неудачные попытки авторизации на Debian 11 – Host, чтобы проверить, что fail2ban отправит вам сообщение об этом событии:

```
root@debian-11-server:~# ssh -p 2222 guest@debian-11-host.ksd.su
```

Наконец, зайдите в Telegram бота, которого вы до этого создали и убедитесь, что получили сообщение.



ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №26

Тема: Настройка Port Knocking (knockd)

Цель работы: Сконфигурировать виртуальную машину в работу с knockd:

– Предварительная настройка брандмауэра;

– Сохранение текущего набора правил;

– Установка knockd;

– Настройка knockd;

– Дополнительные возможности Port Knocking.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

Безопасность – это всегда компромисс между многими показателями, включающими, в том числе, удобство пользования. Современные реалии предусматривают широкое применение удаленного доступа, но при этом не все служебные сервисы администраторы хотят открывать наружу. Одним из вариантов решения проблемы является VPN, но это применимо не всегда и не везде. Хорошей альтернативой в этом случае может являться Port Knocking – специальная технология, позволяющая буквально «постучать» в порты особым образом, после чего вы сможете получить доступ к системе. Стучите и вам откроют.

В переводе с английского Port Knocking буквально означает стук в порт, по аналогии стука в дверь. В детстве многие из нас использовали подобный «секретный стук» приходя в гости к другу, домофонов тогда еще не было, и мы просто особым образом стучали или звонили в дверной звонок. Друг сразу понимал, что пришли к нему и бежал открывать.

Подобный принцип используется и здесь – нужно за определенное время в нужном порядке обратиться к определенным портам системы, после чего вам на некоторое время будет открыт доступ к служебному порту, чтобы вы могли подключиться. Для всех остальных порт будет закрыт.

Входим на виртуальную машину Debian 11 – Host.

Предварительная настройка брандмауэра

Перед тем как настраивать Port Knocking, выключите fail2ban, так как он будет нам мешать настраивать брандмауэр. После выполнения всех настроек вы можете его снова включить:

```
root@debian-11-host:~#  systemctl disable fail2ban
root@debian-11-host:~#  systemctl stop fail2ban
```

Добавим в автозапуск и включим службу nftables:

```
root@debian-11-host:~# systemctl enable nftables
root@debian-11-host:~# systemctl restart nftables
```

Чтобы Port Knocking работал в системе должен быть правильно настроен брандмауэр. В частности, это касается цепочки INPUT, которая отвечает за фильтрацию входящих подключений. Она обязательно должна начинаться с правила, разрешающего уже установленные и связанные соединения, а завершаться правилом, запрещающим любые входящие подключения, между ними должны располагаться правила, разрешающие доступ к отдельным сервисам.

Первым делом стоит установить iptables, чтобы у нас появилась возможность переводить команды iptables в nftables:

```
root@debian-11-host:~#  apt install iptables (Если он не установлен в системе)
```

Затем создадим таблицу под названием filter:

```
root@debian-11-host:~# nft add table ip filter
```

После добавляем в эту таблицу цепочку под названием INPUT:

```
root@debian-11-host:~# nft 'add chain ip filter INPUT { type filter hook input priority 1 ; policy drop ; }'
```

Цепочки с более низкими значениями обрабатываются раньше. Эта цепочка обязана проверяться последней.

В любом случае в минимальной конфигурации должны присутствовать следующие два правила, где ens3 – имя внешнего интерфейса:

Разрешаем установленные и связанные соединения, остальные соединения сбрасываем (policy drop):

root@debian-11-host:~# iptables-translate -A INPUT -i ens3 -m state --state ESTABLISHED,RELATED -j ACCEPT

```
nft add rule ip filter INPUT iifname "ens3" ct state related,established counter accept
```

root@debian-11-host:~# nft add rule ip filter INPUT iifname "ens3" ct state related,established counter accept

Выводим набор правил в консоль для проверки изменений:

root@debian-11-host:~# nft -a list ruleset

```
table ip filter { # handle 1
        chain INPUT { # handle 1
                type filter hook input priority filter + 1; policy drop;
                iifname "ens3" ct state established,related counter packets 0
                bytes 0 accept # handle 2
        }
}
```

Данные команды можно прямо ввести в консоли, и они начнут действовать сразу. Но будьте внимательны, если вы выполните данные команды удаленно, подключаясь через внешний интерфейс – то полностью потеряете связь с сервером до его перезагрузки. В этом случае добавьте между этими правилами еще одно и не удаляйте до тех пор, пока не убедитесь, что Port Knocking работает как требуется.

Разрешаем подключения на порту 2222, чтобы проверить пропускает ли брандмауэр SSH-трафик:

root@debian-11-host:~# nft add rule ip filter INPUT iifname "ens3" tcp dport 2222 counter accept
root@debian-11-host:~# nft -a list ruleset

```
table ip filter { # handle 1
        chain INPUT { # handle 1
                type filter hook input priority filter + 1; policy drop;
                iifname "ens3" ct state established,related counter packets 0
                bytes 0 accept # handle 2
                iifname "ens3" tcp dport 2222 counter packets 0 bytes 0 accept
                # handle 3
        }
}
```

Разрешаем все локальные (loopback) соединения:

root@debian-11-host:~# nft insert rule ip filter INPUT iif lo accept
root@debian-11-host:~# nft -a list ruleset

```
table ip filter {
	chain INPUT { # handle 1
		type filter hook input priority filter + 1; policy drop;
		iif lo accept # handle 4
		iifname "ens3" ct state established,related counter packets 0
		bytes 0 accept # handle 2
		iifname "ens3" tcp dport 2222 counter packets 0 bytes 0 accept
		# handle 3
	}
}
```

Входим на виртуальную машину Debian 11 – Server.

Попробуем подключиться к виртуальной машине Debian 11 – Host, чтобы убедиться в работоспособности наших правил:

root@debian-11-server:~# ssh -p 2222 guest@debian-11-host.ksd.su

Входим на виртуальную машину Debian 11 – Host.

Разрешим ICMP-пакеты:

root@debian-11-host:~# nft add rule ip filter INPUT position 4 ip saddr 0.0.0.0/0 icmp type echo-request counter accept

(*iif lo accept* – Номер позиции может отличаться, так что будьте внимательны и выставьте число, которое выведется вам после выполнения команды *nft -a list ruleset*)

root@debian-11-host:~# nft -a list ruleset

```
table ip filter {
	chain INPUT { # handle 1
		type filter hook input priority filter + 1; policy drop;
		iif lo accept # handle 4
		ip saddr 0.0.0.6/0 icmp type echo-request counter packets 0
		bytes 0 accept # handle 5
		iifname "ens3" ct state established,related counter packets 0
		bytes 0 accept # handle 2
		iifname "ens3" tcp dport 2222 counter packets 0 bytes 0 accept
		# handle 3
	}
}
```

Удалим предыдущее правило, которое мы создавали на порту 2222, так как оно нам больше не понадобится:

root@debian-11-host:~# nft delete rule ip filter INPUT handle 3

(*iifname "ens3" tcp dport 2222 counter packets 0 bytes 0 accept* – Номер позиции может отличаться)

root@debian-11-host:~# nft -a list ruleset (В конечном итоге правила в таблице должны быть размещены именно в таком порядке)

```
table ip filter {
        chain INPUT {
                type filter hook input priority filter + 1; policy drop;
                iif lo accept
                ip saddr 0.0.0.6/0 icmp type echo-request counter packets 0
                bytes 0 accept
                iifname "ens3" ct state established,related counter packets 0
                bytes 0 accept
        }
}
```

Установим пакет netfilter-persistent для автоматической загрузки правил nftables:

```
root@debian-11-host:~# apt install netfilter-persistent
```

Для обновите и сохранения текущую конфигурации отдайте команду:

```
root@debian-11-host:~# netfilter-persistent reload
root@debian-11-host:~# netfilter-persistent save
```

Сохранение текущего набора правил

Выходные данные команды nft list ruleset также являются допустимым входным файлом для нее. Текущий набор правил можно сохранить в файл, а затем загрузить обратно:

```
root@debian-11-gateway:~# nft -s list ruleset | tee /etc/nftables.conf
root@debian-11-host:~# nft -a list ruleset
```

Установка knockd

Для реализации технологии Port Knocking нам потребуется пакет knockd, установим его:

```
root@debian-11-host:~# apt install knockd
```

Затем откроем файл /etc/default/knockd и установим следующую опцию:

```
root@debian-11-host:~# nano -c /etc/default/knockd
```

```
START_KNOCKD=1
```

Затем создадим файл юнита systemd:

```
root@debian-11-host:~# touch /etc/systemd/system/knockd.service
```

Внесем в него следующие строки:

root@debian-11-host:~# nano /etc/systemd/system/knockd.service

```
[Unit]
Description=Port-Knock Daemon
After=network.target
Requires=network.target
Documentation=man:knockd(1)

[Service]
EnvironmentFile=-/etc/default/knockd
ExecStartPre=/usr/bin/sleep 1
ExecStart=/usr/sbin/knockd $KNOCKD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
Restart=always
SuccessExitStatus=0 2 15
ProtectSystem=full
CapabilityBoundingSet=CAP_NET_RAW CAP_NET_ADMIN

[Install]
WantedBy=multi-user.target
```

Затем перечитаем изменения и добавим сервис knockd в автозагрузку:

```
root@debian-11-host:~# systemctl daemon-reload
root@debian-11-host:~# systemctl enable knockd
```

Не спешите запускать саму службу, с большой долей вероятности вы получите ошибку, связанную с неправильными настройками.

Настройка knockd

Откроем файл /etc/knockd.conf и удалим оттуда все, кроме первой секции options, которую приведем к следующему виду:

root@debian-11-host:~# nano /etc/knockd.conf

```
[options]
        UseSyslog
        Interface = ens3
```

Первая опция указывает записывать события в системный лог, вторая задает интерфейс, на котором будет слушать knockd.

Теперь создадим секцию для доступа к SSH:

```
[SSH]
        sequence = 7000,9000,8000
        seq_timeout = 5
        tcpflags = syn
        start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 2222 -j ACCEPT
        stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 2222 -j ACCEPT
cmd_timeout = 60
```

Опция sequence задает последовательность «стуков», по умолчанию используется протокол tcp, если вы хотите использовать udp, то следует указать его через двоеточие, например:

```
sequence = 7000,9000:udp,8000
```

seq_timeout – это время в секундах за которое клиент должен совершить всю последовательность «стуков», следует сделать его минимальным, но при этом следует учитывать возможные задержки каналов связи как сервера, так и клиента, tcpflags – определяет TCP-флаги пакетов, которые будут участвовать в последовательности, syn рекомендуется использовать совместно с SSH, т.к. SSH-трафик может мешать knockd, делая последовательность недействительной.

Опции start_command и stop_command определяют команды, которые должен выполнить knockd с интервалом указанном в опции cmd_timeout. Принцип действия здесь прост – получив указанную последовательность «стуков» сервис выполняет команду из опции start_command, а по истечении заданного таймаута команду stop_command.

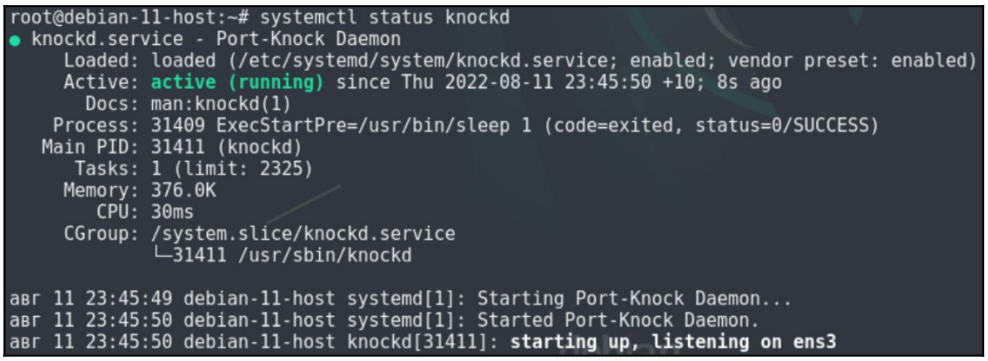
Наша задача – открыть доступ к порту SSH на время достаточное для установления подключения. Поэтому первой командой мы добавляем самым первым правилом в цепочку INPUT разрешение подключиться к SSH с адреса источника «стука». Обратите внимание, что для этого мы используем ключ -I, который без указания номера добавляет правило первой строкой, а также подстановочную конструкцию -s %IP%, которая подставит в правило текущий адрес «стучавшего». Вторая команда удаляет уже добавленное правило.

Таким образом порт будет открыт на подключение только для адреса клиента и только на время, указанное в таймауте. Этот момент обычно вызывает у начинающих некоторое недоумение: а каким образом будет работать соединение, если мы закрыли порт? Все очень просто, доступ к порту 22 нам нужен только для установления соединения, после чего оно перейдет в состояние ESTABLISHED и будет разрешено первым заданным нами правилом в цепочке INPUT, которое разрешает установленные и связанные подключения.

```
[options]
        UseSyslog
        Interface = ens3

[SSH]
        sequence = 7000,9000,8000
        seq_timeout = 5
        tcpflags = syn
        start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 2222 -j ACCEPT
        stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 2222 -j ACCEPT
        cmd_timeout = 60
```

Сохраним конфигурацию и запустим сервис knockd:

root@debian-11-host:~# systemctl start knockd



Для надежности перезагрузим виртуальную машину и затем продолжим работу:

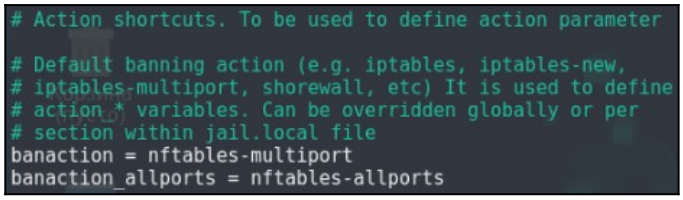
root@debian-11-host:~# systemctl reboot

Теперь вы можете включить и сервис fail2ban:

root@debian-11-host:~# systemctl enable fail2ban
root@debian-11-host:~# systemctl start fail2ban
root@debian-11-host:~# fail2ban-client set sshd unbanip debian-11-host.ksd.su (Выполните эти команды, чтобы нашей проверке не мешали блокировки со стороны fail2ban)
root@debian-11-host:~# fail2ban-client set sshd unbanip windows-10-host.ksd.su

Не забудьте подправить конфигурационный файл fail2ban, чтобы он мог работать одновременно с nftables и knockd:

root@debian-11-host:~# nano -c /etc/fail2ban/jail.local



Теперь вдобавок к нашим правилам будут добавляться правила, которые создает fail2ban:

root@debian-11-host:~# nft -a list ruleset (Вывод может отличаться, так как fail2ban не вписывает изменения в брандмауэр пока не произошло ни одной блокировки)

```
table ip filter {
        chain INPUT {
                type filter hook input priority filter + 1; policy drop;
                iif lo accept
                ip saddr 0.0.0.6/0 icmp type echo-request counter packets 0
                bytes 0 accept
                iifname "ens3" ct state established,related counter packets 0
                bytes 0 accept
        }
}
table inet f2b-table {
        set addr-set-sshd {
                type ipv4 add r
        }
```

```
        chain f2b-chain { # handle 1
                type filter hook input priority filter - 1; policy accept;
                tcp dport { 2222 } ip saddr @addr-set-sshd reject
        }
}
```

Давайте попытаемся войти по SSH на виртуальную машину Debian 11 – Host, чтобы убедиться, что брандмауэр работает как надо.

Входим на виртуальную машину Debian 11 – Server.

root@debian-11-server:~# ssh -p 2222 guest@debian-11-host.ksd.su

Входим на виртуальную машину Windows 10 – Host.

C:\Users\Win10>ssh -p 2222 guest@debian-n11-host.ksd.su

Как видим – все закрыто, подключиться к серверу невозможно.

Входим на виртуальную машину Debian 11 – Server.

Теперь постучимся, в Linux для этого можно использовать команду knock. Перед этим проверим состояние брандмауэра. Мы использовали самый простой набор из двух правил: разрешающего установленные соединения и запрещающего остальное.

В Windows постучаться немного сложнее, из стандартных средств можно использовать PowerShell, например:

```
PS C:\Windows\system32> Test-NetConnection debian-11-host.ksd.su -Port ××××
```

Теперь попробуем постучать с другого компьютера по определенным портам, чтобы нужный нам порт 2222 открылся:

root@debian-11-server:~# apt install knockd
root@debian-11-server:~# knock debian-11-host.ksd.su 7000 9000 8000

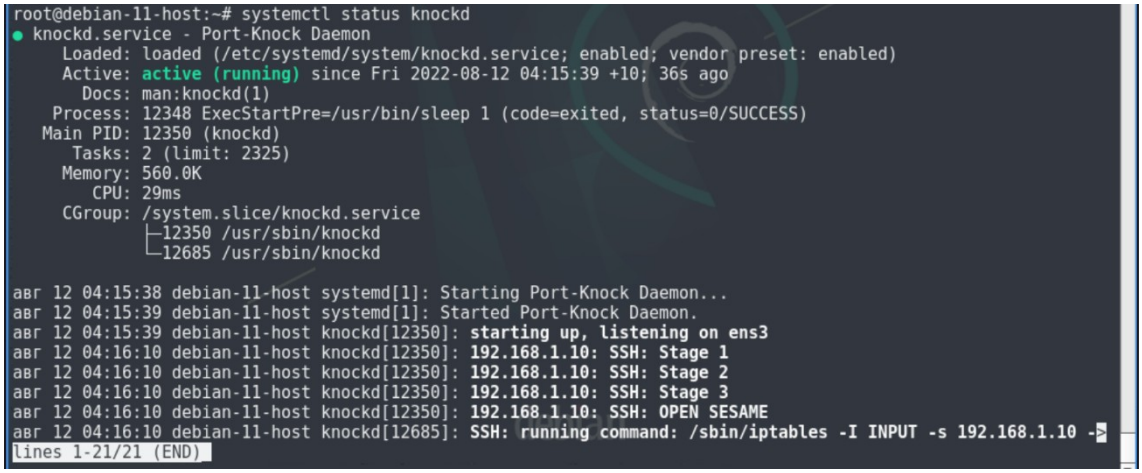
Входим на виртуальную машину Debian 11 – Host.

И снова проверяем брандмауэр, самым первым в цепочке INPUT появилось правило, разрешающее доступ к SSH с адреса «стучавшего» клиента.

root@debian-11-host:~# nft -a list ruleset

```
table ip filter {
        chain INPUT {
                type filter hook input priority filter + 1; policy drop;
                meta l4proto tcp ip saddr 192.168.1.10 tcp dport 2222 counter
                packets 0 bytes 0 accept
                iif lo accept
                ip saddr 0.0.0.6/0 icmp type echo-request counter packets 0
                bytes 0 accept
                iifname "ens3" ct state established,related counter packets 0
                bytes 0 accept
        }
}
```

```
table inet f2b-table {
	set addr-set-sshd {
		type ipv4 add r
	}

	chain f2b-chain { # handle 1
		type filter hook input priority filter - 1; policy accept;
		tcp dport { 2222 } ip saddr @addr-set-sshd reject
	}
}
```



Входим на виртуальную машину Debian 11 – Server.

Ровно через указанное в таймауте время, в нашем случае 60 секунд, правило исчезнет, поэтому выбирайте данное значение небольшим, но достаточным для выполнения всех действий по подключению.

Давайте попробуем подключиться по SSH пока функционирует правило:

root@debian-11-server:~# ssh -p 2222 guest@debian-11-host.ksd.su

Входим на виртуальную машину Debian 11 – Host.

Дополнительные возможности Port Knocking

Возможности knockd не исчерпываются управлением брандмауэром, сама возможность выполнить произвольную команду открывает довольно широкие перспективы, ограниченные фантазией и здравым смыслом. Например, мы можем запустить произвольный скрипт, скажем, для резервного копирования.

Снова откроем файл /etc/knockd.conf и добавим в конец файла еще одну секцию:

root@debian-11-host:~# nano /etc/knockd.conf

```
[knockd-script]
	sequence = 9000,7000,8000
	seq_timeout = 5
	tcpflags = syn
	command = /home/guest/knockd-script.sh
```

root@debian-11-host:~# mkdir /home/guest/experiment
root@debian-11-host:~# touch /home/guest/knockd-script.sh

root@debian-11-host:~# chmod ugo+x /home/guest/knockd-script.sh
root@debian-11-host:~# nano /home/guest/knockd-script.sh (Необходимо заполнить скрипт самостоятельно)

```
#!/bin/sh
touch /home/guest/experiment/knockd-script
nft -a list ruleset > /home/guest/experiment/knockd-script
```

root@debian-11-host:~# systemctl restart knockd

Основным отличием от предыдущей секции является опция command, при верной последовательности стуков служба просто выполняет указанную в ней команду. Таким образом в руки администратора попадает достаточно мощный и удобный инструмент, который может оказаться полезным в некоторых ситуациях.

Входим на виртуальную машину Debian 11 – Server.

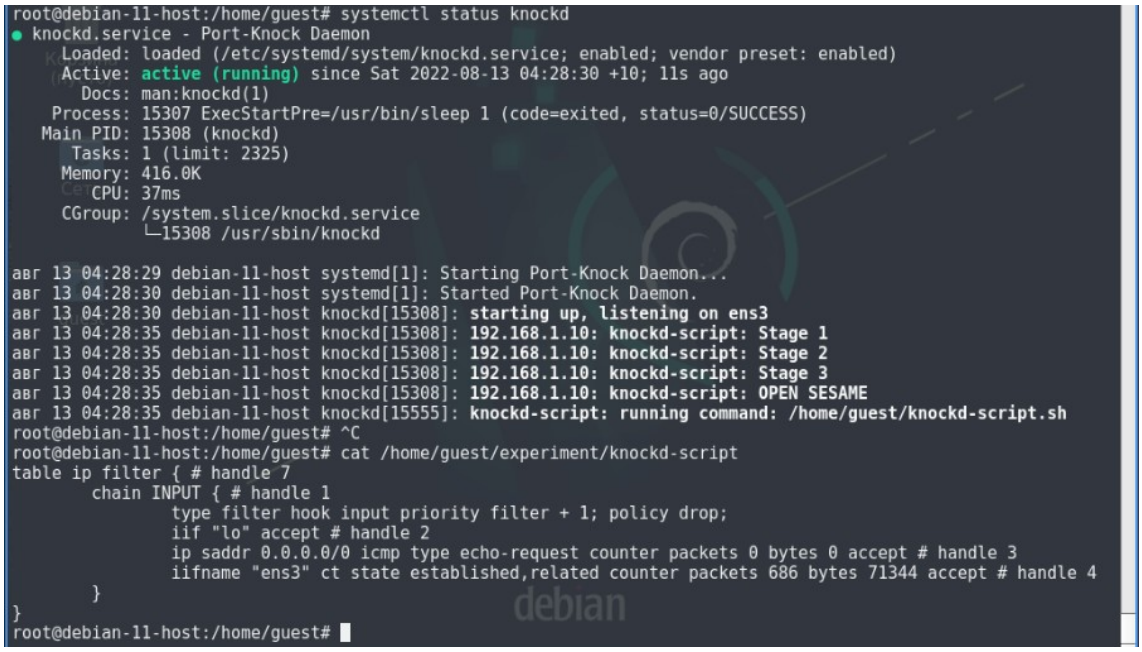
Теперь попробуем постучать с другого компьютера по определенным портам, чтобы запустить наш скрипт:

root@debian-11-server:~# knock debian-11-host.ksd.su 9000 7000 8000

Входим на виртуальную машину Debian 11 – Host.

Проверим, что скрипт действительно отработал после того как мы «простучали» по определенным портам:

root@debian-11-host:~# systemctl status knockd
root@debian-11-host:~# cat /home/guest/experiment/knockd-script



Существует у Port Knocking и существенный недостаток: если кто-то имеет возможность прослушивать ваш трафик (например, на публичной точке доступа), то вполне может перехватить вашу последовательность и чем чаще вы «стучитесь», тем быстрее это можно сделать. Как нам в таком случае поступить? В этом случае следует воспользоваться еще одной возможностью knockd – одноразовыми последовательностями.

Для этого замените опцию sequence в нужной секции опцией one_time_sequences с указанием пути к файлу одноразовых последовательностей:

root@debian-11-host:~# nano /etc/knockd.conf

```
[knockd-script]
        one_time_sequences = /root/ssh-sequences
        seq_timeout = 5
        tcpflags = syn
        command = /home/guest/knockd-script.sh
```

root@debian-11-host:~# nano /root/ssh-sequences

```
1234,2234,3234
1134,1234,1334
1214,1224,1234
1231,1232,1233
```

root@debian-11-host:~# systemctl restart knockd

Важно! Не следует располагать данный файл в директории /etc , в этом случае knockd будет выдавать ошибку при запуске службы.

Входим на виртуальную машину Debian 11 – Server.

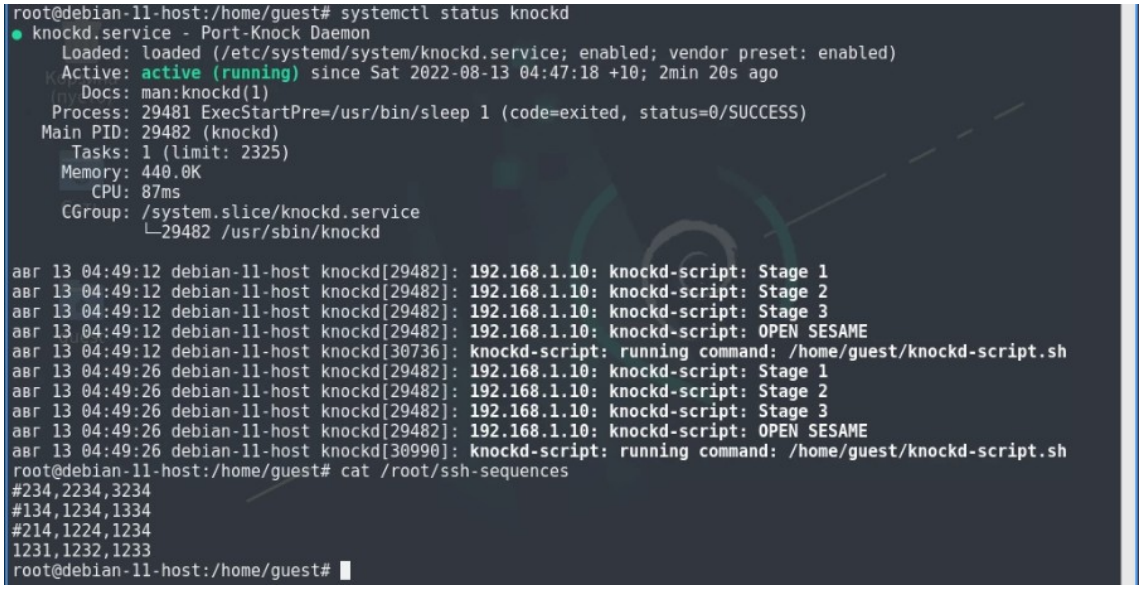
Попробуйте постучаться, после каждого успешного «стука» knockd будет ставить комментарий в начале сработавшей строки:

root@debian-11-server:~# knock debian-11-host.ksd.su 1234 2234 3234
root@debian-11-server:~# knock debian-11-host.ksd.su 1134 1234 1334
root@debian-11-server:~# knock debian-11-host.ksd.su 1214 1224 1234

Входим на виртуальную машину Debian 11 – Home.

root@debian-11-host:~# systemctl status knockd
root@debian-11-host:~# cat /root/ssh-sequences

```
#1234,2234,3234
#1134,1234,1334
#1214,1224,1234
1231,1232,1233
```



Будьте внимательны, используя данный метод, если у вас закончатся одноразовые последовательности, то вы рискуете остаться полностью без доступа к серверу.

Самостоятельная работа

Попробуйте «постучаться» и войти в виртуальную машину Debian-11 – Host используя компьютер Windows 10 – Host.

ЛАБОРАТОРНО-ПРАКТИЧЕСКАЯ РАБОТА №27

Тема: Развертывание RADIUS-сервера (FreeRADIUS)

Цель работы: Сконфигурировать виртуальную машину в работу с FreeRADIUS:

– Обновление локального индекса пакетов;

– Установка пакета с программой OpenSSH-Server и его базовая настройка;

– Запуск службы и локальная проверка работы сервера RADIUS;

– Проверка работы сервера RADIUS с использованием маршрутизатора Cisco.

Материальное обеспечение:

– Компьютер;

– Доступ в Интернет.

Порядок проведения работ:

RADIUS (аббревиатура от Remote Authentication Dial In User Service) – это сетевой протокол, обеспечивающий централизованную Аутентификацию (Authentication), Авторизацию (Authorization) и Учет используемых сетевых ресурсов (Accounting).

В настоящее время протокол RADIUS используется для доступа к виртуальным частным сетям (VPN), точкам беспроводного (Wi-Fi) доступа, Ethernet-коммутаторам и для удаленного хранения учетных записей. Благодаря открытости, простоте внедрения, постоянному усовершенствованию, протокол RADIUS сейчас является фактически стандартом для удаленной аутентификации.

FreeRADIUS – это RADIUS-сервер с открытым исходным кодом. Он является альтернативой других коммерческих RADIUS серверов, поскольку он модульный и функциональный на сегодняшний день. Кроме того, он входит в пятерку RADIUS серверов мира с точки зрения развертывания и количества пользователей, которых этот сервер авторизует ежедневно.

Входим на виртуальную машину Debian 11 – Server.

Обновление локального индекса пакетов

```
root@debian-11-server:~# su -
Пароль: guest
root@debian-11-server:~# apt update
```

Установка пакета с программой OpenSSH-Server и его базовая настройка

root@debian-11-server:~# apt install freeradius

Создайте пользователя для работы с RADIUS-сервером:

root@debian-11-server:~# adduser radius

Откройте конфигурационный файл sites-available/default и раскомментируйте строку unix, что RADIUS-сервер мог использовать базу данных пользователей и паролей из файла /etc/passwd:

root@debian-11-server:~# cd /etc/freeradius/3.0/
root@debian-11-server:~# ls -all
root@debian-11-server:~# nano -c sites-available/default

unix (Строка 400)

Откройте конфигурационный файл clients.conf:

client * { (Строка 30 – Разрешение подключения для всех клиентов RADIUS-сервера)
…
ip addr = * (Строка 42 – Разрешение подключаться к RADIUS-серверу с любых IP-адресов)
…
secret = cisco (Строка 37 – Пароль на вход в привилегированный режим в системе Cisco IOS)

Запуск службы и локальная проверка работы сервера RADIUS

root@debian-11-server:~# systemctl enable freeradius
root@debian-11-server:~# systemctl start freeradius
root@debian-11-server:~# systemctl status freeradius
[ctrl+c]
root@debian-11-server:~# nano -c /etc/hosts

192.168.1.10 debian-11-server (Строка 7)

root@debian-11-server:~# radtest radius cisco 192.168.1.10 0 cisco

```
root@debian-11-server:/etc/freeradius/3.0# radtest radius cisco 192.168.1.10 0 cisco
Sent Access-Request Id 171 from 0.0.0.0:35369 to 192.168.1.10:1812 length 76
        User-Name = "radius"
        User-Password = "cisco"
        NAS-IP-Address = 192.168.1.10
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "cisco"
```

root@debian-11-server:~# systemctl restart freeradius

Проверка работы сервера RADIUS с использованием маршрутизатора Cisco

Разрешите на брандмауэре любые подключения из подсети 172.16.1.0/24, что маршрутизатор смог подключиться к серверу RADIUS:

root@debian-11-server:~# ufw allow from 172.16.1.0/24
root@debian-11-server:~# ufw status

Входим на виртуальную машину Cisco vIOS Router и прописываем в терминале следующие команды.

cisco-vios-router>enable
cisco-vios-router#show ip interface brief
cisco-vios-router#conf t
cisco-vios-router(config)#line vty 0 15
cisco-vios-router(config-line)#exec-timeout 15 0
cisco-vios-router(config-line)#transport input telnet (Включить возможность подключения по telnet)
cisco-vios-router(config-line)#end
cisco-vios-router#conf t
cisco-vios-router(config)#aaa new-model (Настроить доступа к серверу RADIUS по умолчанию)
cisco-vios-router(config)#radius server RS
cisco-vios-router(config-radius-server)#address ipv4 192.168.1.10 auth-port 1812
cisco-vios-router(config-radius-server)#key cisco
cisco-vios-router(config-radius-server)#exit

cisco-vios-router(config)#aaa authentication login default group radius local (Настроить клиент аутентификации через RADIUS)
cisco-vios-router(config)#line vty 0 4
cisco-vios-router(config-line)#login authentication default
cisco-vios-router(config-line)#exit
cisco-vios-router(config)#line console 0
cisco-vios-router(config-line)#login authentication default
cisco-vios-router(config-line)#exit
cisco-vios-router(config)#exit
cisco-vios-router#show running-config (Сохранить конфигурацию и проверить работу маршрутизатора Cisco)
cisco-vios-router#write memory
cisco-vios-router#reload

Входим на виртуальную машину Debian 11 – Host.

Подключитесь к Cisco vIOS Router и попробуйте авторизоваться, используя учетную запись, находящуюся на сервере RADIUS:

root@debian-11-host:~# telnet 172.16.1.254

```
root@debian-11-host:~# telnet 172.16.1.254
Trying 172.16.1.254...
Connected to 172.16.1.254.
Escape character is '^]'.

**************************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**************************************************************************

User Access Verification

Username: radius
Password:

**************************************************************************
* IOSv is strictly limited to use for evaluation, demonstration and IOS  *
* education. IOSv is provided as-is and is not supported by Cisco's      *
* Technical Advisory Center. Any use or disclosure, in whole or in part, *
* of the IOSv Software or Documentation to any third party for any       *
* purposes is expressly prohibited except as otherwise authorized by     *
* Cisco in writing.                                                      *
**************************************************************************
```

4 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основные

1. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы. Учебник для вузов. / В.Г. Олифер, Н.А. Олифер. – СПб: Питер, 2019. – 992 с.
2. Таненбаум, Э. Компьютерные сети. / Э. Таненбаум, Д. Уэзеролл. – СПб: Питер, 2019. – 960 с.
3. Левицкий Н.Д. Справочник системного администратора. Полное руководство по управлению Windows-сетью. / Н.Д. Левицкий. – СПб: Наука и Техника, 2020. – 464.
4. Немет Э. Unix и Linux: руководство системного администратора. / Немет Э., Снайдер Г., Хейн Трен Р., Уэйли Б., Макин Д. – М.: Вильямс, 2020. – 1168 с.

Дополнительные

1. Уймин А.Г. Сетевое и системное администрирование. Демонстрационный экзамен КОД 1.1. Учебно-методическое пособие. / А.Г. Уймин. – СПб.: Лань, 2021. – 480 с.