Лабораторная работа 1. Консольные утилиты настройки сетевых компонентов Linux.

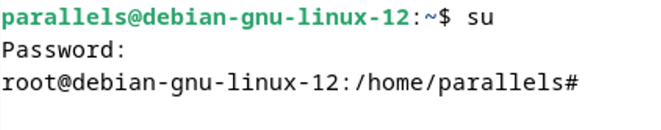
Выполнил: Михалев Никита Романович М3311

## Цели и задачи:

получить практические навыки по конфигурированию сетевых интерфейсов (на примере протокола IPv4) в операционных системах Linux, ознакомится с утилитами командной строки, освоить современные сетевые менеджеры Linux.

## Часть 1. Linux (Debian 12):

Запустить виртуальную машину и авторизоваться в системе под администраторской учётной записью.



Создать скрипт, который позволяет пользователю:

1. Узнать модель сетевой карты, канальную скорость, режим работы (Full или Half Duplex). Наличие физического подключения (линка), MAC адрес
2. Получить информацию о текущей конфигурации IPv4 (ip, mask, gate, dns).
3. Настроить сетевой интерфейс по сценарию #1
4. Настроить сетевой интерфейс по сценарию #2.
5. закрыть скрипт.

Скрипт part-one.sh:

Изображение выглядит как текст, чек, снимок экрана, алгебра

Автоматически созданное описание

Изображение выглядит как текст, чек, документ, цвета слоновой кости

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, документ, чек

Автоматически созданное описание

Демонстрация работы первых двух пунктов:

Изображение выглядит как текст, Шрифт, алгебра

Автоматически созданное описание

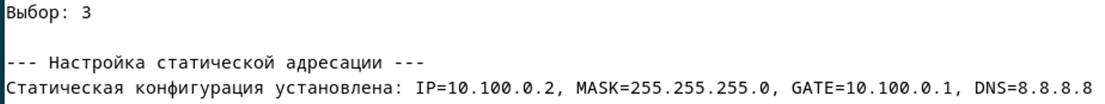
Изображение выглядит как текст, снимок экрана, документ, чек

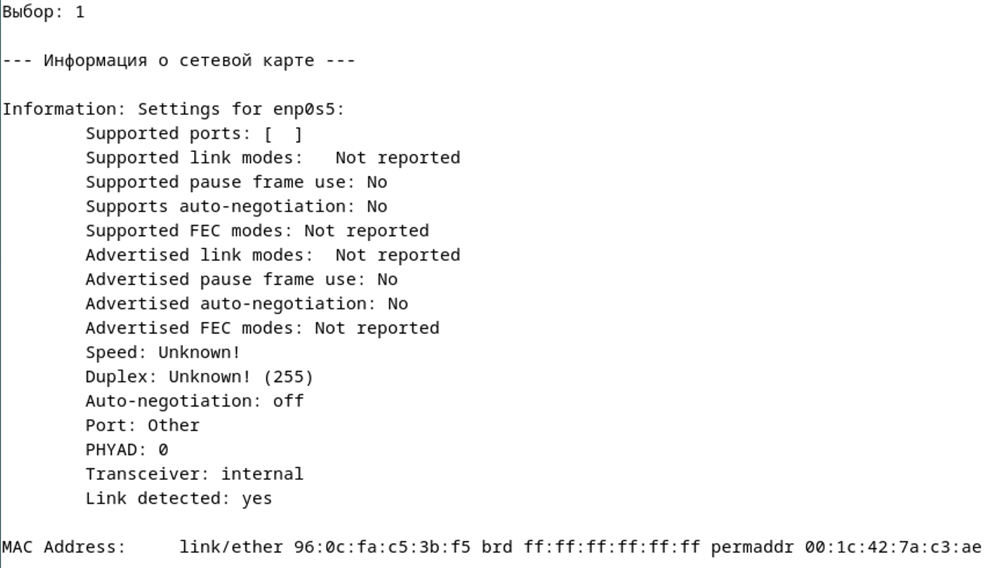
Автоматически созданное описание

Изображение выглядит как текст, Шрифт, чек, алгебра

Автоматически созданное описание

**Настройка статической адресации и вывод значений после ее установки:**

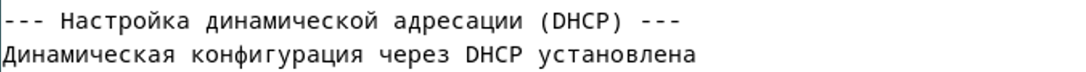




Изображение выглядит как текст, Шрифт, алгебра, снимок экрана

Автоматически созданное описание

**Настройка динамической адресации и вывод значений после ее установки:**



Изображение выглядит как текст, снимок экрана, документ, Шрифт

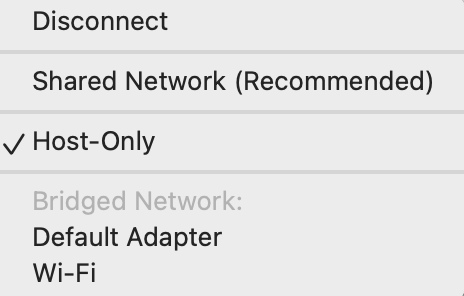
Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

## Часть 2. Работа с виртуальными интерфейсами Linux CentOS через Network Manager:

Для выполнения этой части эмуляция сетевого интерфейса была изменена на Host-Only, что ближе всего по описанию похоже на режим Внутренняя сеть в Virtual Box:

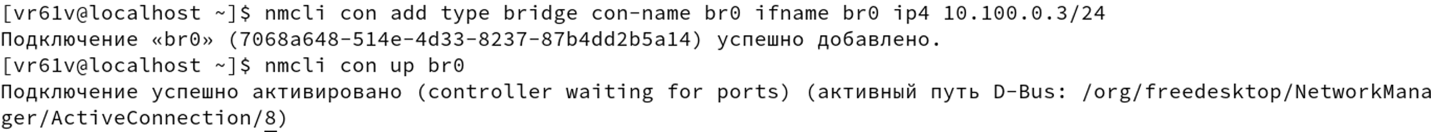


**Создание нового подключения part2-connection с типом ethernet, с ip4 = 10.100.0.2/24 и шлюзом 10.100.0.1 и его активация:**

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

**Создание** виртуального сетевого интерфейса с именем br0 и IP = 10.100.0.3 и его активация:



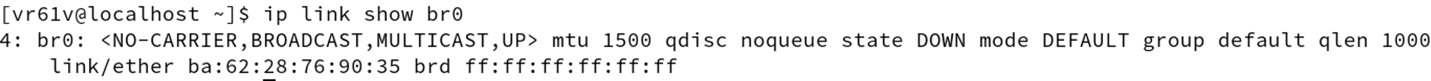
**Проверка созданных интерфейсов, команда ping успешно отработала, следовательно интерфейсы успешно подключены:**Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

**Получение mac адреса созданного интерфейса br0:**



## Часть 3. Работа с реальными интерфейсами Linux Debian 11 через netpaln:

Отредактировал YAML файл конфигурации netplan так чтобы сетевой интерфейс имел два адреса 10.100.0.4 и 10.100.0.5, маска 255.255.255.0, а gateway 10.100.0.3:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

С помощью утилиты ping проверил связь между адресами 10.100.0.2–10.100.0.5:

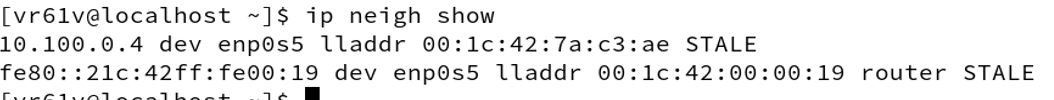
Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

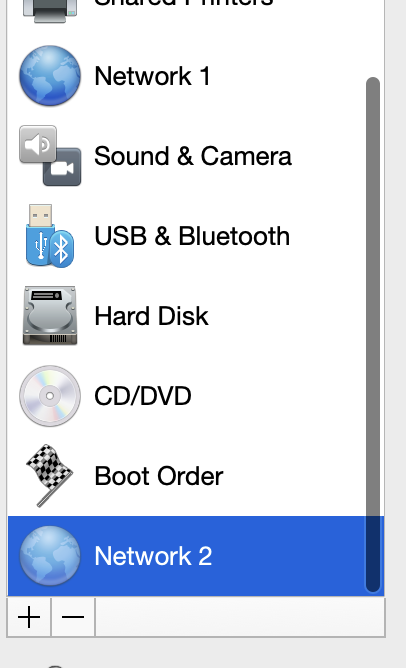
Вывел таблицу arp кэша: Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание



## Часть 4. Настройка объединения реальных сетевых интерфейсов в Linux:

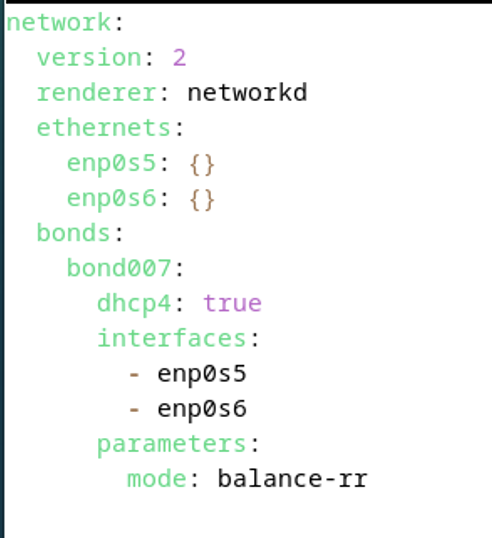
Для начала выполнения работы создал второй интерфейс, с идентичными настройками:



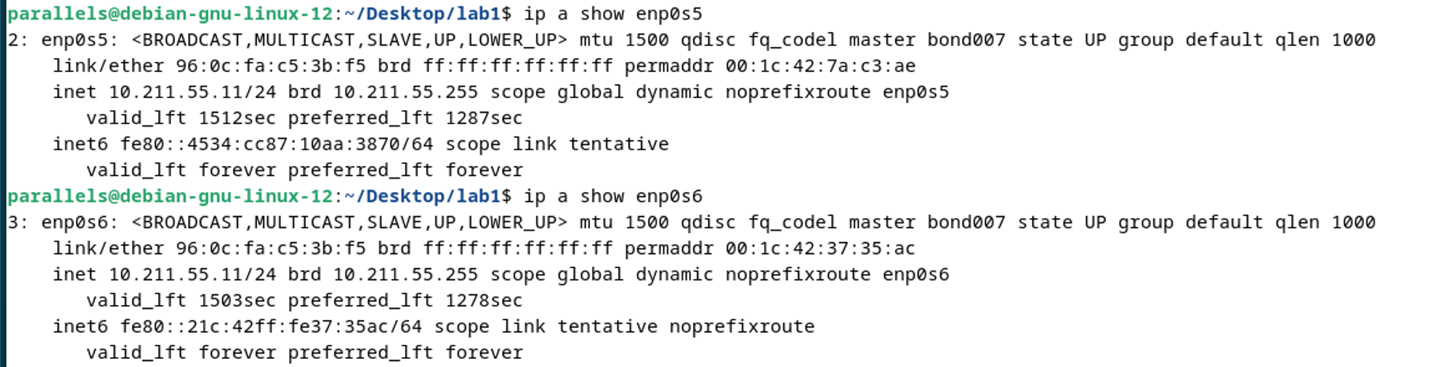
Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

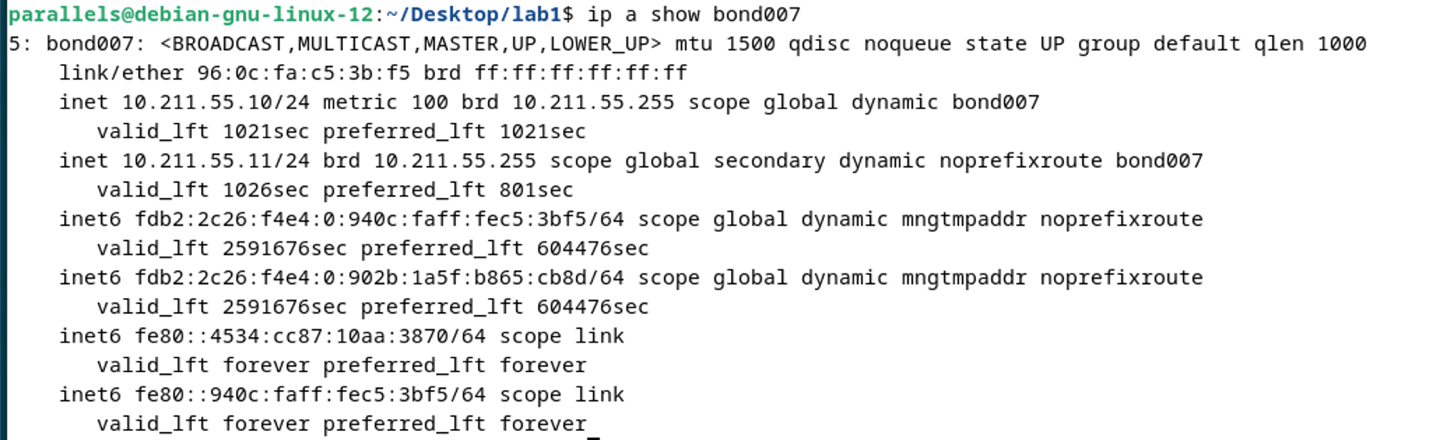
Автоматически созданное описание

Дальше настроил файл конфига 01-netconf.yaml, в котором описываю новый bond007, с автоматическим получением конфигурации ipv4, двумя объединенными интерфейсами enp0s5, enp0s6 и распределением нагрузки по Round Robin:

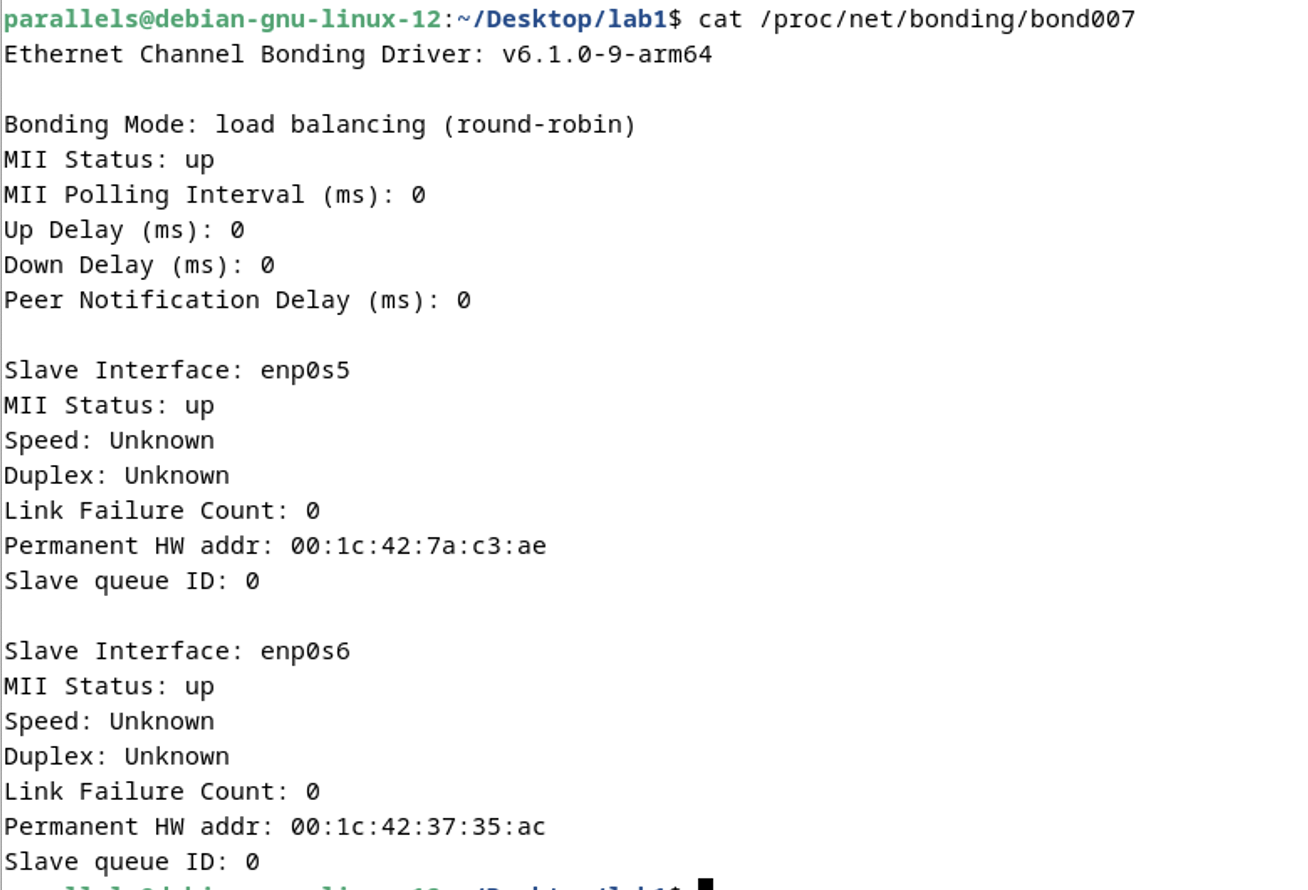


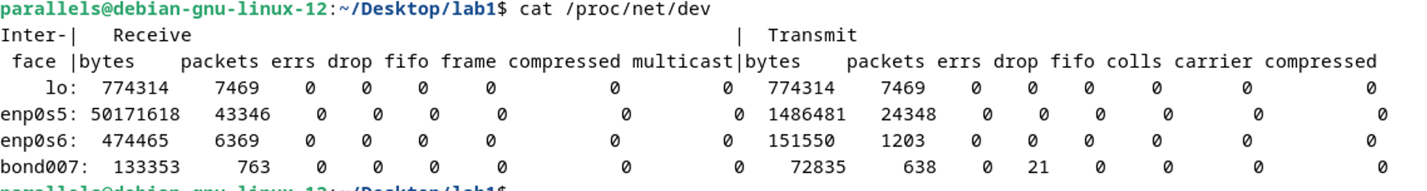
Вывод в терминал mac и ip адресов созданного bond и его интерфейсов:





Вывод в терминал содержание файла /proc/net/bonding/bond007 и /pcroc/net/dev:

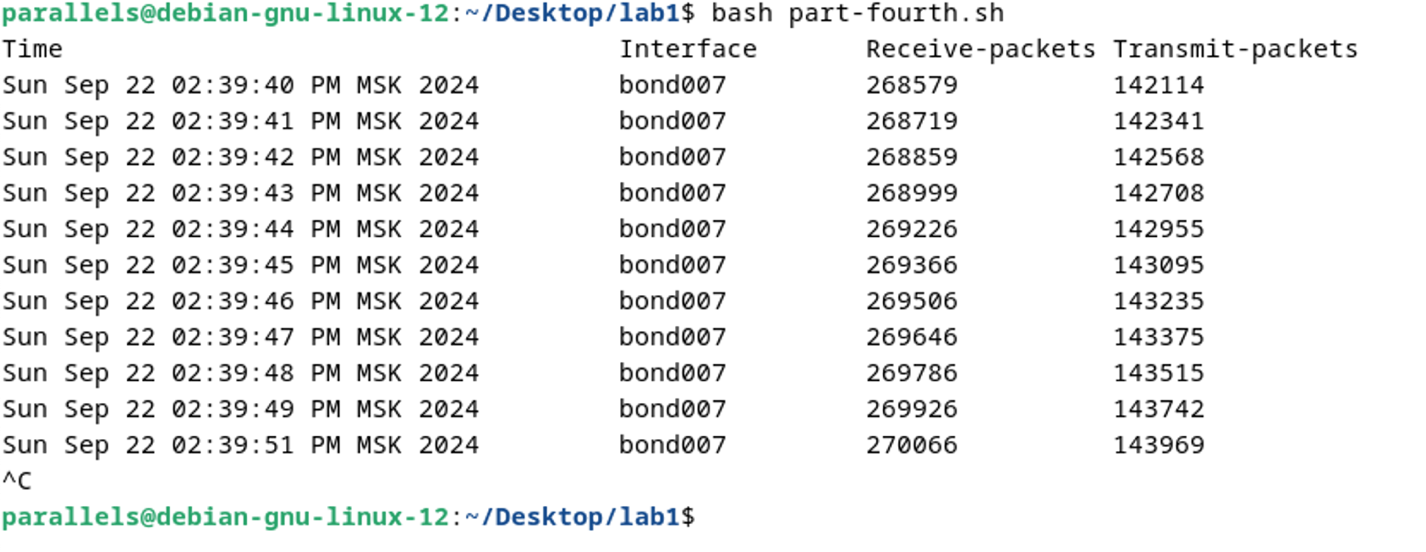




Код скрипта для вывода текущего времени, имени интерфейса, значения Receive-packets и Transmit-packet:



Проведение измерений:



Изображение выглядит как текст, число, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Часть вывода команды ping:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание