МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по лабораторной работе №2

«Настройка сетевой системы ОС Linux»

по дисциплине

«Администрирование систем и сетей»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Кочешков А. А.\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сухоруков В.А.\_\_\_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_19-ВМ\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2023

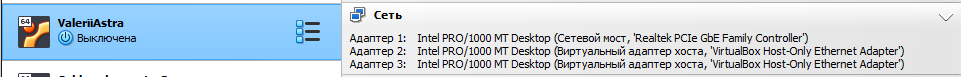
# Цель работы

Изучение установки и конфигурирования сетевого интерфейса, маршрутизации и контроля сетевых связей в ОС Linux

# Ход работы

## Конфигурирование сетевого интерфейса.

Спланируем работу трёх виртуальных машин Linux. Виртуальная машина ValeriiAstra будет иметь 3 сетевых интерфейса: Bridged и 2 Host-Only.



Виртуальная машина SukhorukovAstra2 будет иметь один сетевой интерфейс: Host-Only.



Виртуальная машина Sukhorukov-аstra3 будет иметь один сетевой интерфейс: Host-Only.

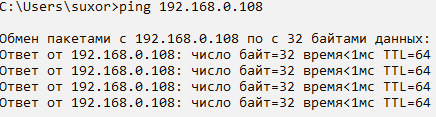


С помощью команды ifconfig зададим параметры IP-адреса и сетевой маски. Начнем с машины ValeriiAstra.

Интерфейс eth0, подключенный как сетевой мост, служит для связи с основной машиной Windows и доступа к Internet. Зададим этому интерфейсу статический IP адрес 192.168.0.108, маску сети 255.255.255.0, broadcast адрес 192.168.0.255.



Проверим соединение из основной машины, умеющую адрес 192.168.0.104/24. Соединение доступно.



Настроим интерфейс eth1, который служит для связи с машиной SukhorukovAstra2. Зададим этому интерфейсу IP адрес 192.168.11.2, маску сети 255.255.255.128, broadcast адрес 192.168.11.127.



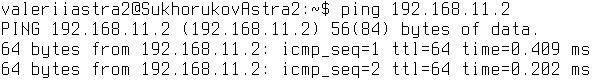
Настроим интерфейс eth2, который служит для связи с машиной Sukhorukov-аstra3. Зададим этому интерфейсу IP адрес 192.168.11.130, маску сети 255.255.255.128, broadcast адрес 192.168.11.255.



Настроим интерфейс на машине SukhorukovAstra2. Эта машина входит в сеть 192.168.11.0/25, broadcast -192.168.11.127 с адресом 192.168.11.1.



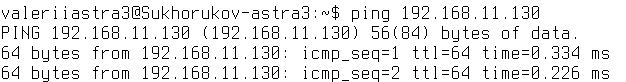
Проверим соединение с машиной ValeriiAstra по адресу 192.168.11.2. Сеть доступна.



Настроим интерфейс на машине Sukhorukov-аstra3. Эта машина входит в сеть 192.168.11.0/25, broadcast -192.168.11.255 с адресом 192.168.11.129.



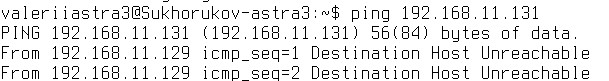
Проверим соединение с машиной ValeriiAstra по адресу 192.168.11.130. Сеть доступна.



Проверим соединение с машиной SukhorukovAstra2по адресу 192.168.11.1. Сеть не доступна. Эта проблема возникла из-за того, что нет настроенного пути в другую сеть.

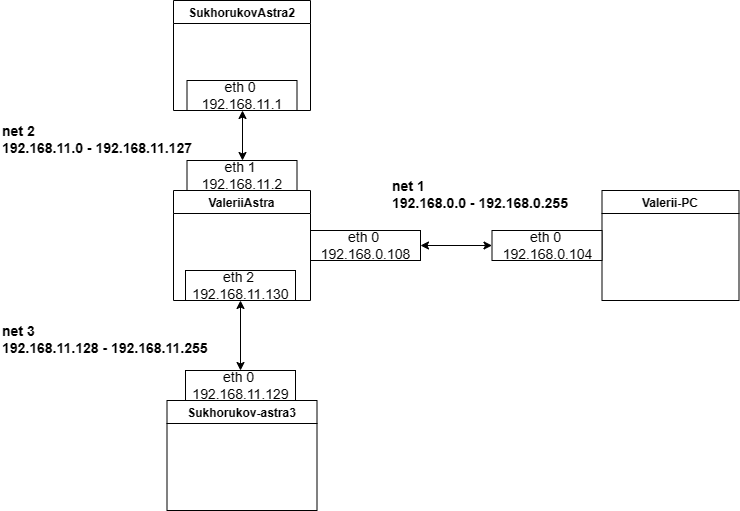


Попробуем установить соединение по IP адресу, который отсутствует в сети. Было получено сообщение о недоступности узла.





В результате настроек была реализована следующая модель сетей.



## Управление пространством сетевых имен.

В UNIX-сетях используется доменная система именования узлов сети DNS. В соответствии с DNS каждый компьютер имеет полное доменное имя FQDN, которое состоит из локального имени хоста и имени домена. При работе в локальной сети можно использовать сокращенное локальное имя. В системе должен быть реализован механизм разрешения (трансляции) сетевого имени хоста в IP-адрес. В Linux могут быть задействованы разные способы решения:

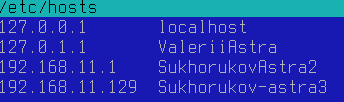
* Локальные файлы hosts
* Запросы к серверу DNS
* Службы NIS
* Современные системы также могут применять методы авто конфигурирования в локальных сетях.

Рассмотрим файлы конфигурации имен.

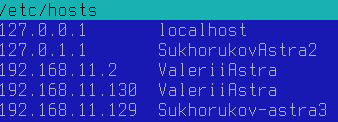
* hostname- содержится символьное имя локального узла.
* hosts- связывает хост-имена с IP-адресами. Каждому IP-адресу ставится в соответствие доменное имя.

Заполним файлы сопоставления IP- адреса и имени хоста трёх машин, причем для машины ValeriiAstra необходимо внести два разных IP-адреса, но одно и то же имя

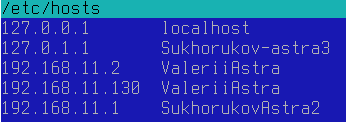
ValeriiAstra



SukhorukovAstra2

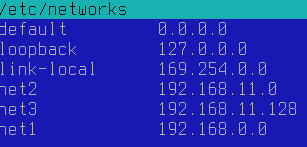


Sukhorukov-аstra3



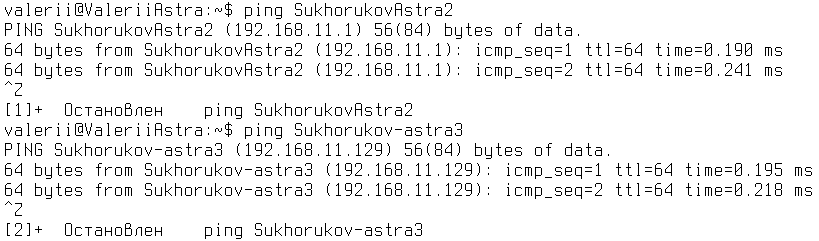
* networks- связывает имена сетей с IP адресами.

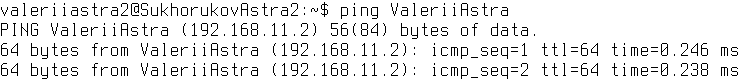
Зададим имена сетям и укажем их IP-адреса на трёх машинах.



* hosts.allow и hosts.deny- содержат разрешение входов и отрицание доступа к услугам и хостам. Просматриваются сначала hosts.allow, затем hosts.deny. Если запись была найдена в hosts.allow, доступ предоставляется независимо от hosts.deny. Если соответствие найдено в hosts.deny, то запрос будет отклонен и связь разорвана. Если никакое соответствие не найдено вообще, запрос будет принят
* host.conf- содержит настройки для библиотеки resolver. resolver- механизм преобразования имен узлов сети в IP адреса и обратно (прямое и обратное преобразование). Данный файл должен содержать в каждой строке одно ключевое слово, за которым следует информация о соответствующей этому ключевому слову настройке.
* resolv.conf- определяет, как механизм преобразования имен узлов получает доступ к системе доменных имён

Проверим доступ к хостам по их заданным нами именам с помощью команды ping.



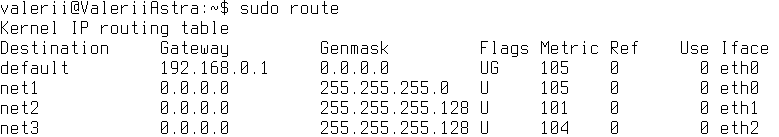




Доступ был получен ко всем узлам, которые находятся в одной подсети.

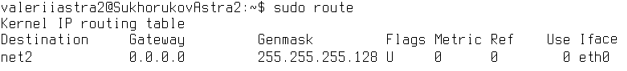
## Формирование подсетей и маршрутизация.

Рассмотрим таблицы маршрутизации, с помощью команды route на узле ValeriiAstra.



В таблице можно увидеть имя сети, шлюз, маску, флаги и интерфейс, использующийся для этой сети. На данной машине есть пути во всех трёх сетях.

Рассмотрим таблице узле с одним сетевым интерфейсом типа host only - SukhorukovAstra2.



На данном узле настроен путь только в одной сети. Добавим маршруты к сетям net1 и net3.

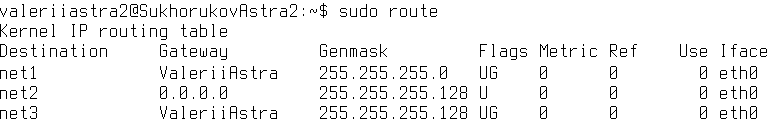






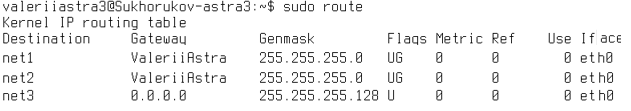


Получим новую таблицу с помощью route.

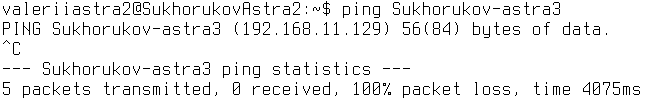


В столбце флагов у добавленный сетей присутствует флаг G, который означает, что используется шлюз.

Аналогично настроим пути на узле Sukhorukov-astra3. Получим таблицу сетей.



Проверим связь между узлом SukhorukovAstra2 и Sukhorukov-astra3.

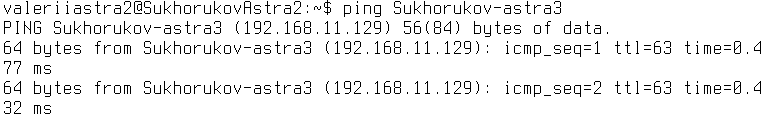




При сравнении результата, который мы получили в п.1, где узлы из разных сетей были недоступны, получаем уже другую реакцию. В данном случае узлы доступны, но пакеты не доходят. Это происходит, потому что функция маршрутизации в ядре Linux на узле-маршрутизаторе ValeriiAstra не включена. Включим данную функцию, для этого запишем «1» в содержимое файла: /proc/sys/net/ipv4/ip\_forward.

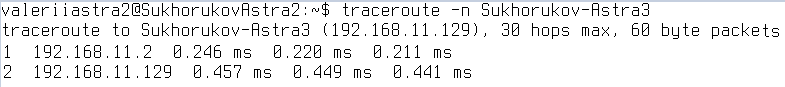


Пропингуем еще раз и посмотрим результат.



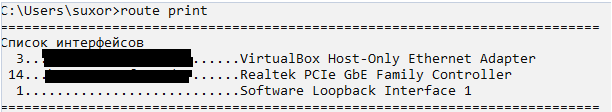
Теперь соединение доступно, и пакеты проходят.

Выполним трассировку прохождения пакетов от узла SukhorukovAstra2 до узла Sukhorukov-astra3 с помощью команды traceroute.

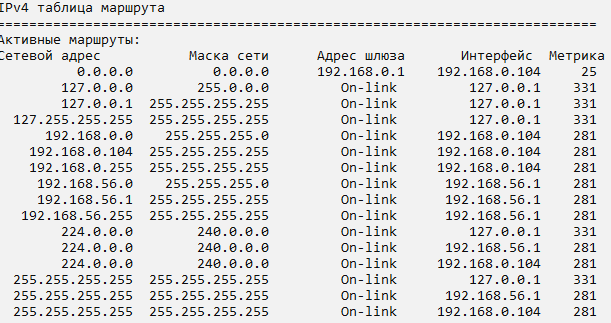


Cначала пакет проходит через шлюз, затем доходят до указанного узла в другой подсети. Таким образом, статическая маршрутизация между подсетями была настроена верно.

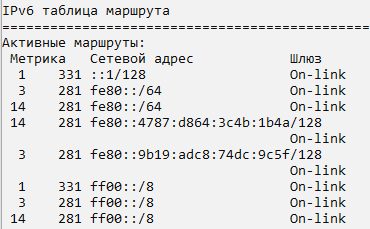
Рассмотрим таблицу маршрутизации на Windows с помощью команды route print. Сначала данная команды выводит список сетевых интерфейсов, отображая их ID, MAC-адреса и названия.



Далее следует таблица маршрута сетей IPv4, которая схожа с таблицей в Linux. Вместо флагов в маршруте используется столбец «Метрика», который хранит числовое значение флагов.



При наличии и настройки IPv6 адресации также выводится таблица маршрутизации по данному протоколу.



## Реализовать настройку IP-сетей по внешним правилам для использования шлюза во внешние сети и Internet

Реализуем функцию трансляции адресов NAT в ядре узла ValeriiAstra с помощью команды iptables, которая указывает, что нужно добавить правило цепочки, относящейся к таблице NAT.





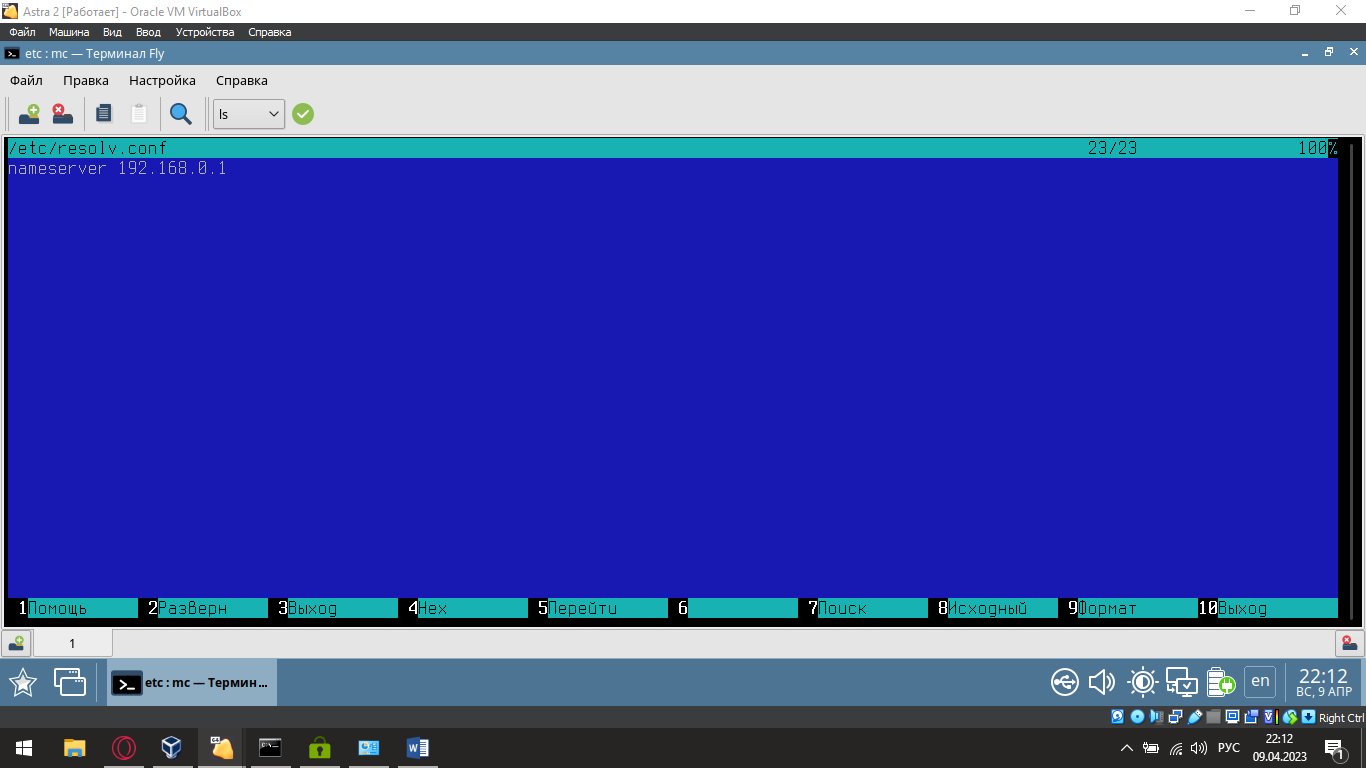
Теперь настроим шлюзы по умолчанию на двух других машинах.



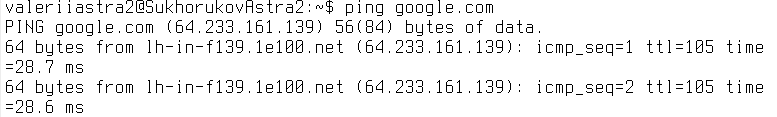




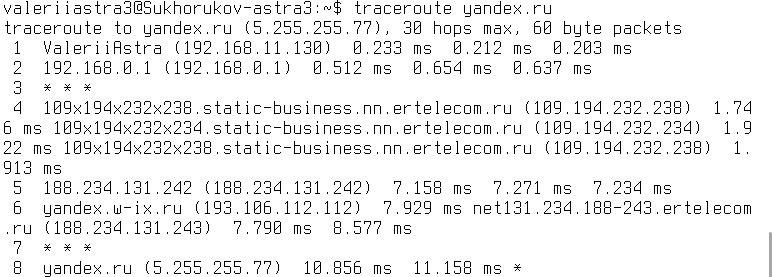
Теперь настроим IP адреса сервера DNS в конфигурационном файле resolv.conf на хостах. Добавим туда IP адрес моего роутера 192.168.0.1



Попробуем получить доступ к сайту google.com с машины SukhorukovAstra2. Доступ к Интернет ресурсам возможен.



Выполним трассировку к серверу yandex.ru с хоста Sukhorukov-astra3 по протоколу TCP.



Сначала происходит запрос к узлу ValeriiAstra, далее к роутеру и в сеть Internet. Для достижения нужно ресурса потребовалось 8 шагов.

Теперь выполним трассировку по протоколу ICMP. Можно увидеть, что при использовании разных протоколов маршрутизация происходит по разному.

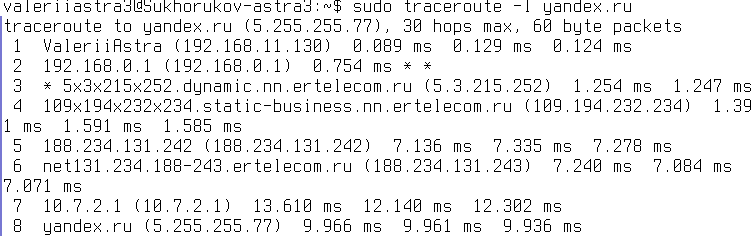


Схема сети на канальном уровне:

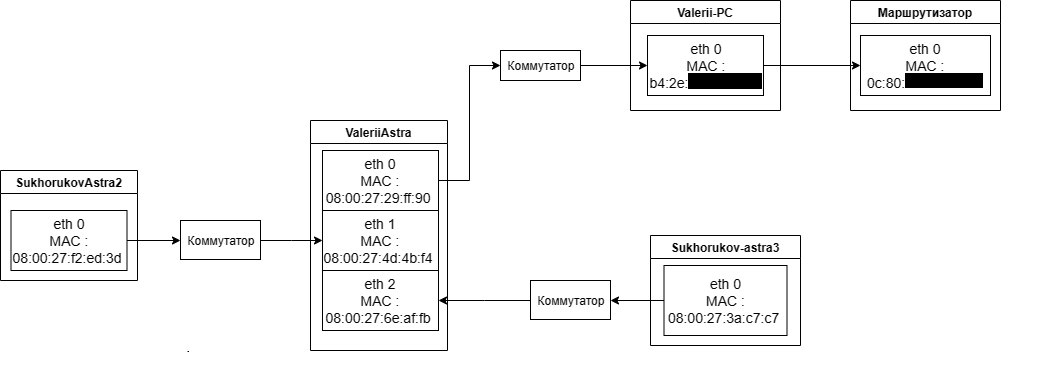
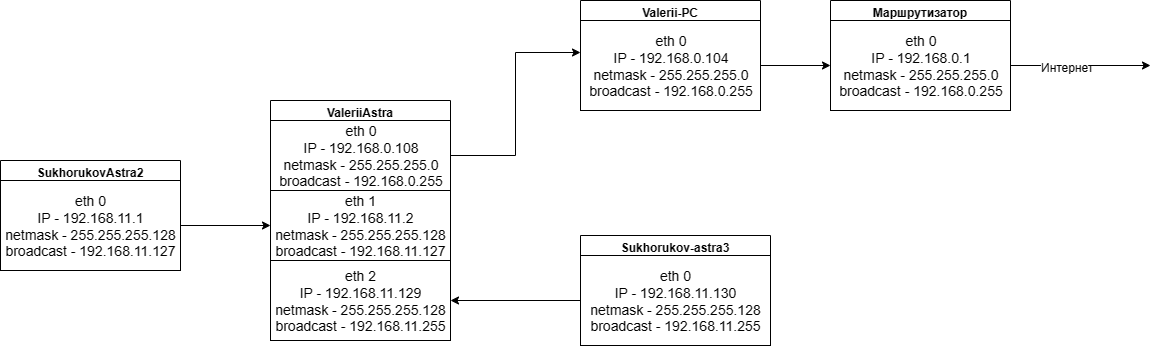


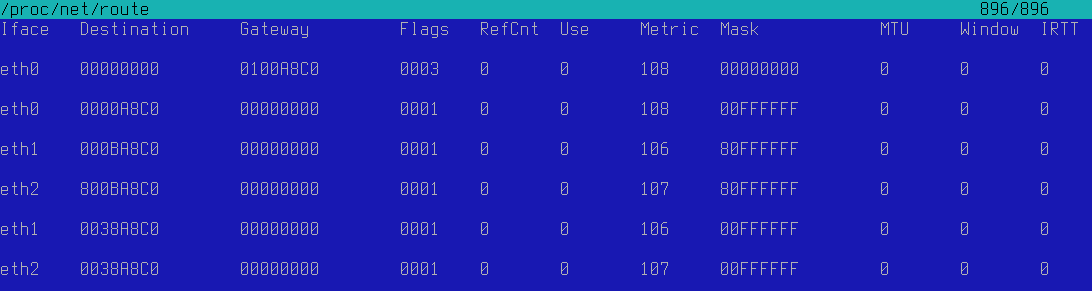
Схема сети на сетевом уровне с выходом в Интернет:



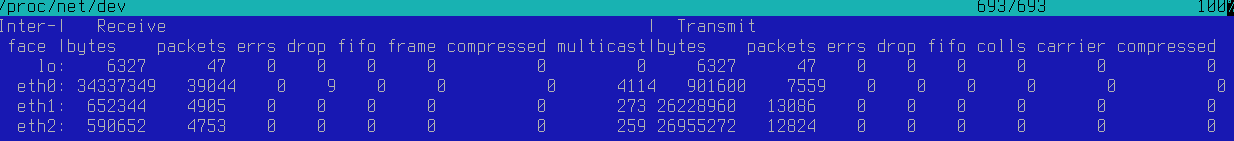
## Контроль за сетью, получение статистики и другой сетевой информации.

В псевдофайловой системе ядра /proc/net/ и /proc/sys/net найдем отдельные примеры текущей информации о сетевой подсистеме во внутреннем представлении:

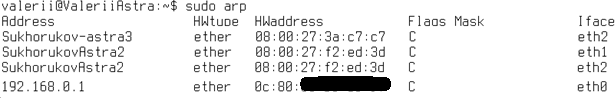
Таблица маршрутизации (файл proc/net/route):



Статистика сетевых интерфейсов (файл /proc/net/dev)



Используем команду arp, чтобы получить кэш-таблицу протокола arp



Здесь мы можем увидеть MAC адреса всех хостов, с которыми устанавливали соединения.

Команда **netstat** предоставляет различную информацию о сетевых соединениях, портах, маршрутизации и т.д. С помощью различных опций можно получить разную статистическую информацию. Например:

* **netstat -i** выводит статистику по интерфейсам, такую как количество отправленных и принятых пакетов, ошибки и т.д.
* **netstat -r** выводит таблицу маршрутизации.
* **netstat -n** отображает IP-адреса и номера портов без их преобразования в имена хостов и служб.
* **netstat -a** отображает все активные сетевые соединения.

Команда **nstat** также предоставляет статистику сетевых подключений, но использует протокол SNMP для сбора этой информации. **nstat** позволяет получить более подробную информацию, например, статистику для каждого протокола, типа трафика, IP-адресов и т.д.

Команда **ss** также позволяет получать информацию о сетевых соединениях и маршрутизации, но в отличие от **netstat** и **nstat** предоставляет информацию о сокетах, а не о процессах, которые используют эти сокеты. **ss** также имеет более удобный и понятный формат вывода.

Как правило, использование команд **netstat**, **nstat** и **ss** зависит от конкретной задачи, которую необходимо выполнить. Команда **netstat** является стандартной утилитой и обычно доступна на большинстве операционных систем. Команда **nstat** может быть установлена дополнительно и предоставляет более подробную информацию с помощью протокола SNMP. Команда **ss** является более современной и удобной утилитой для работы с сокетами, которая может быть использована вместо **netstat**.

# Вывод

По выполнению данной лабораторной работы были созданы три виртуальные машины, которые составляли сеть из двух клиентских узлов и одного узла-маршрутизатора.

На каждой из виртуальных машин были настроены сетевые интерфейсы и таблицы маршрутизации.

С помощью таблиц маршрутизации удалось связать два узла из разных подсетей и проверить связь между ними и реализовать с узла-клиента доступ в интернет.

Также рассмотрены важные конфигурационные файлы: hosts, который связывает хост-имена с IP-адресами и networks, который связывает доменные имена с адресами сетей.