После установки системы виртуализации VirtualBox и его плагина VirtualBox Extension Pack, я начал своё знакомство с системой анализа трафика Wireshark. Далее выбрал один из адаптеров сети для прослушивания (рисунок 1).

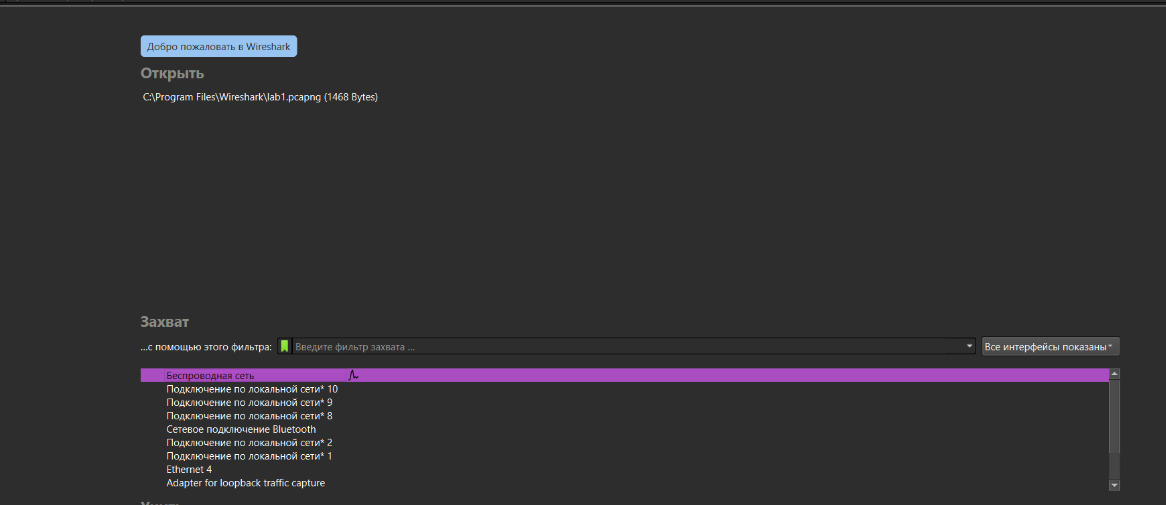


Рисунок 1

После этого запустил захват пакетов и настроил расположение фреймов так, как показано на рисунке в задании лабораторной работы.

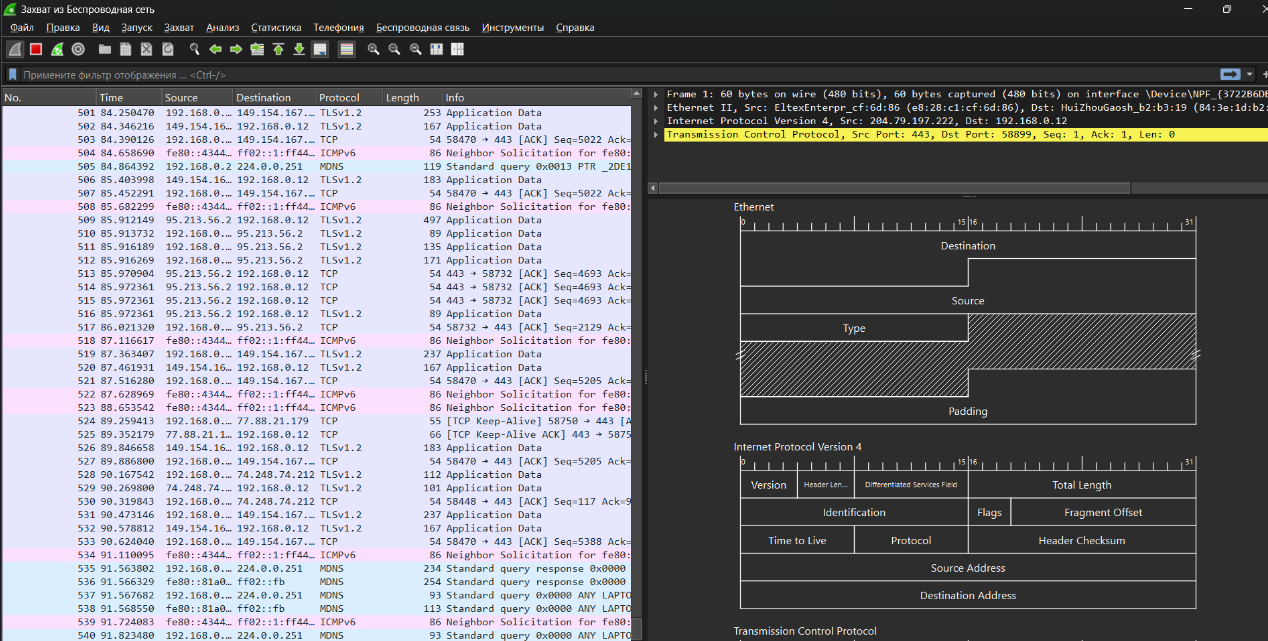


Рисунок 2. Окно анализатора трафика Wireshark

Остановил захват трафика, проанализировал несколько захваченных пакетов. Наблюдаемая информация о пакете напрямую связана с моделью OSI/ISO.

На **физическом** уровне происходит передача битов по физическому каналу связи и при этом этот уровень не вникает в смысл передаваемой информации. Мы просто представили биты информации в виде сигналов и тут это видно по первой строке, например, «480 bit».

На **канальном** уровне указывается, что Ethernet – это ключевой протокол, также прописаны сведения о девайсе, который используется.

На **сетевом** уровне обеспечивается адресация и поиск маршрутов. Например, мы можем видеть

source, destination.

( **Src:** EltexEnterpr\_cf:6d:86 (e8:28:c1:cf:6d:86), **Dst:** HuiZhouGaosh\_b2:b3:19 (84:3e:1d:b2:b3:19) )

На **транспортном** уровне обеспечивается передача данных между процессами на хостах. Это порты Source Port и Destination Port, номера последовательность для удобного отслеживания (Sequence Number);

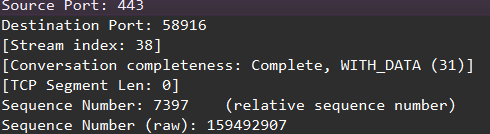


Рисунок 3.

На **сеансовом** уровне обеспечивается синхронизация, предотвращение одновременного выполнения критичной операции, а также очередность передачи сообщений. Я прочитал про TLS протоколы, что они обеспечивают поддержку шифрования и организации защищенных сессий, но какую-либо другую информацию про сеансовый уровень я не нашёл в данном окне.

На уровне **представления** происходит согласование синтаксиса и смысл передаваемых данных. Это может быть кодировка или разные форматы представления чисел. Сюда же можно отнести протокол TSL для шифрования, а также тут могут быть представлены данные, которые закодированы в определённых форматах (XML, JSON, Base64).

На **прикладном** уровне происходит взаимодействие пользователя с приложением и передача данных. Например, это может быть HTTP-запрос или DNS.



Рисунок 4.

В данном случае у меня попался HTTP протокол, который способствует передаче данных.

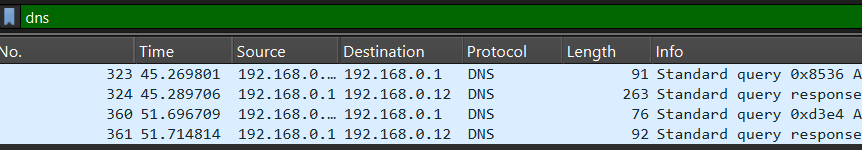


Рисунок 5. Отфильтровал пакеты DNS.

Утилита editcap (входит в состав Wireshark) используется для редактирования файлов с захваченным трафиком.

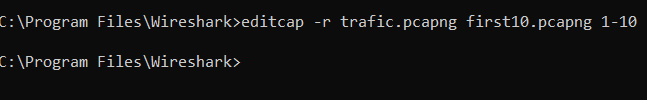


Рисунок 6. Сохранение первых 10 пакетов.

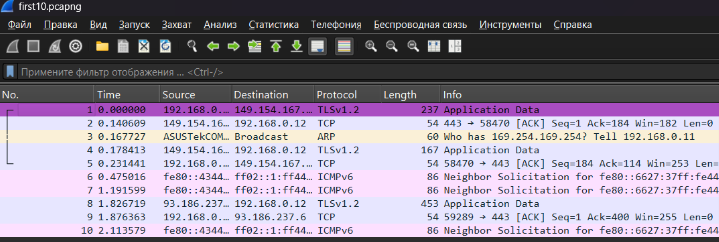


Рисунок 7. Первые 10 пакетов успешно сохранились.

Установим захват трафика с настройкой захвата таким образом, чтобы он автоматически остановился после 50ти захваченных пакетов.



Рисунок 8.

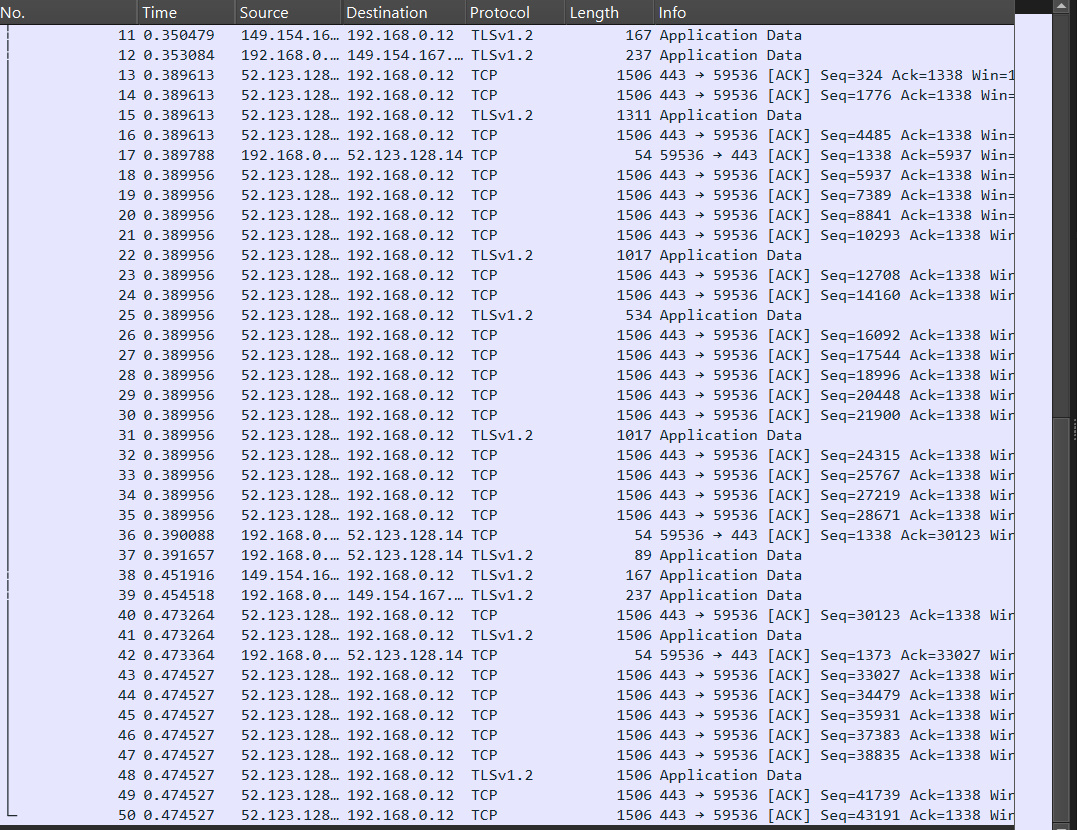


Рисунок 9. Пакетов действительно 50.

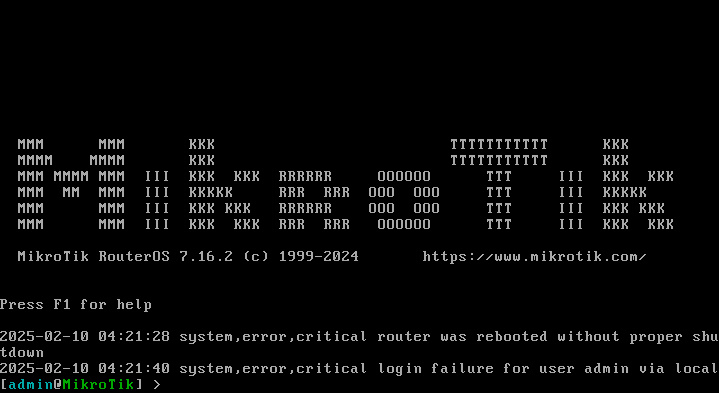


Рисунок 10. Создал виртуальную машину Mikrotik.

Установил информационное сообщение, выводимое на экран перед авторизацией.

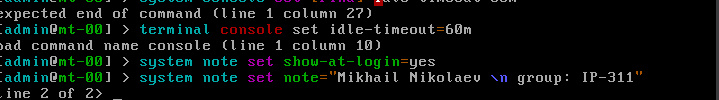


Рисунок 11. Установка информационного сообщения



Рисунок 12. Отображение сообщения

Также с помощью указания show-at-cli-login=yes я отобразил сообщение с фамилией, именем и группой перед логином.

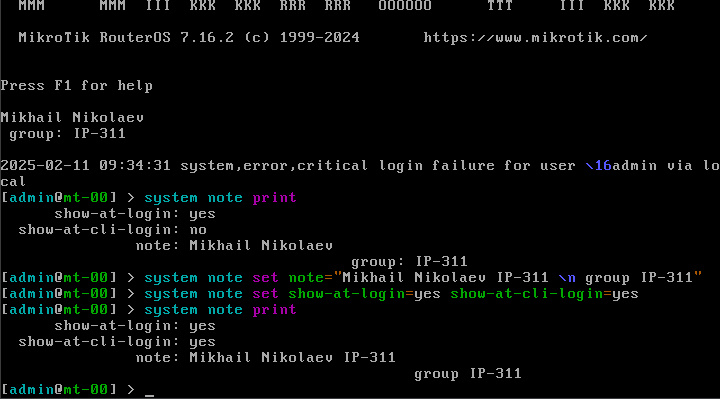


Рисунок 13. Команда для отображения сообщения перед и после логина.



Рисунок 14. Отображение сообщения перед логином.

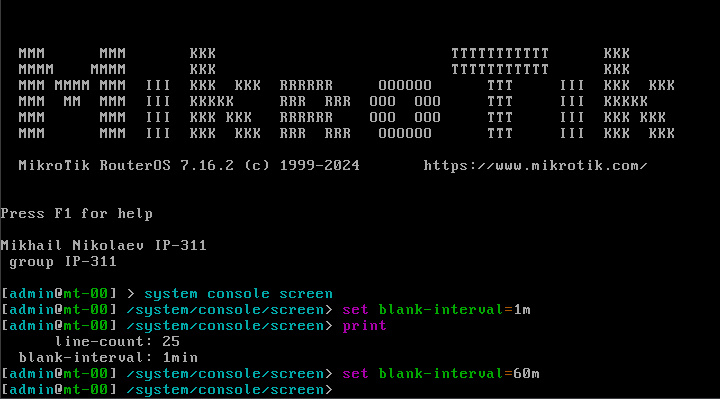


Рисунок 15. Установил режим выключения экрана спустя 60 минут бездействия пользователя.

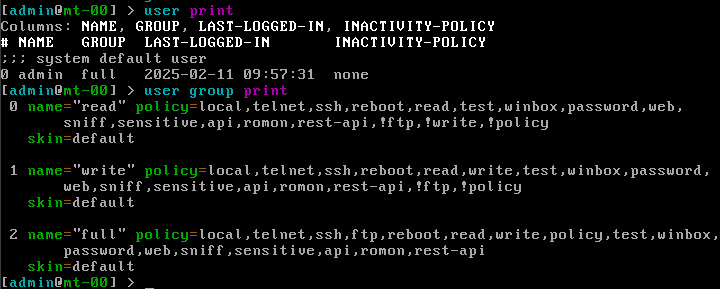


Рисунок 16.

Вывел информацию о пользователях. Команда user print выводит информацию о всех пользователях, пользователей также можно добавлять и удалять. Простыми словами, policy содержит список прав (что пользователь может или не может делать в системе). Например, пользователь может read, write, reboot и т.д. Группа full имеет полный список прав.



Рисунок 17. Безопасное выключение виртуальной машины.

Склонировал виртуальную машину с жёсткого диска и назвал её mikrotik\_clone.

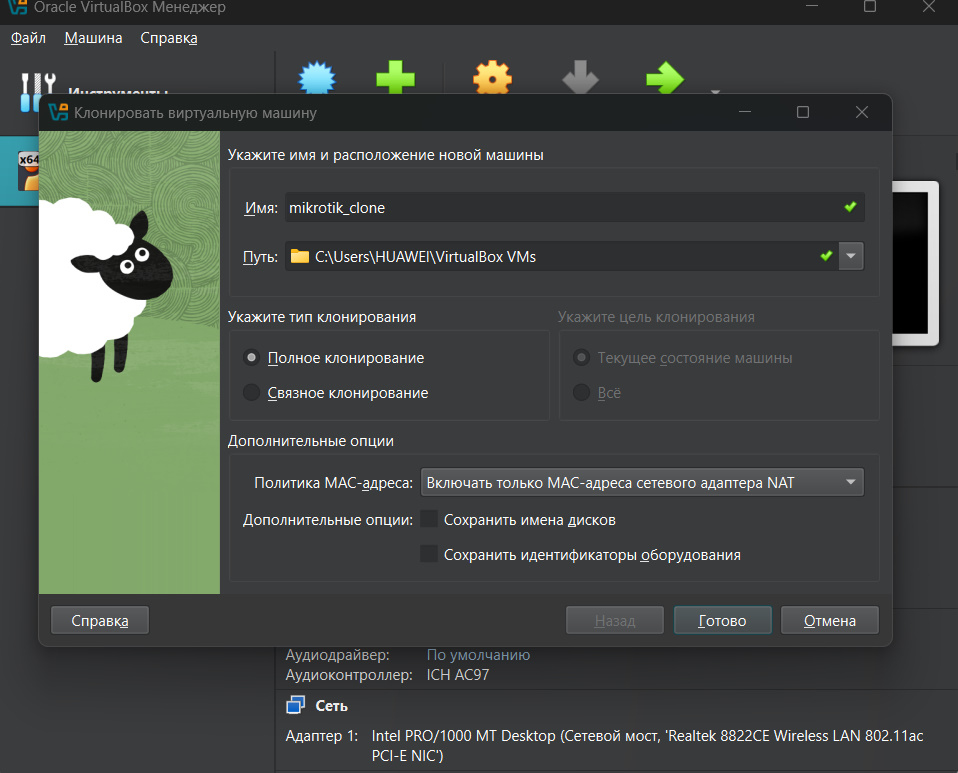


Рисунок 18. Создание mikrotik\_clone.

Проверим, что настройки у виртуальной машины и клонированной виртуальной машины разные.

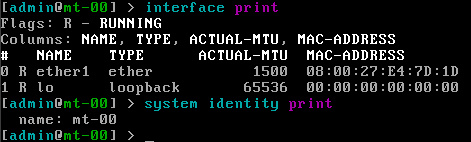


Рисунок 19.

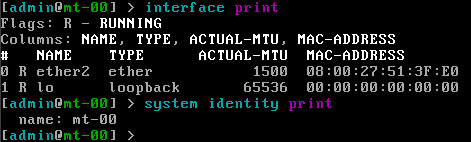


Рисунок 20.

Видно, что MAC-адрес сетевых интерфейсов у машин различается.

Забыл создать ещё одну копию виртуальной машины. Основная виртуальная машина называется mikrotik, а копия – mikrotik-base. Для проверки я вывел сообщение перед логином для другой группы.

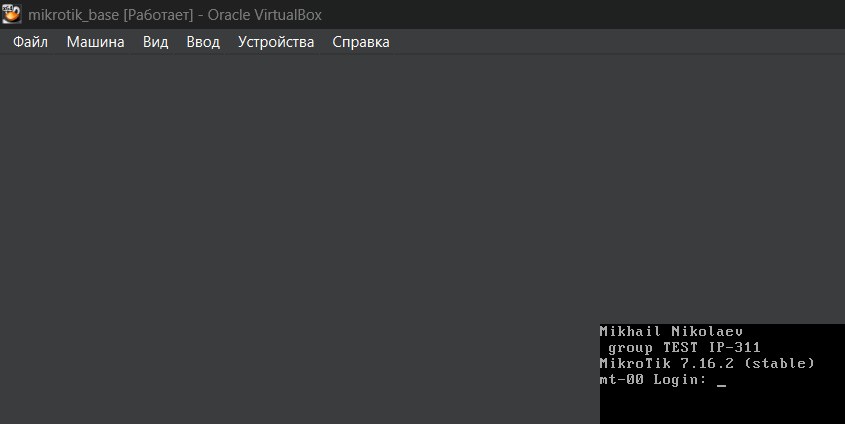


Рисунок 21. group TEST IP-311

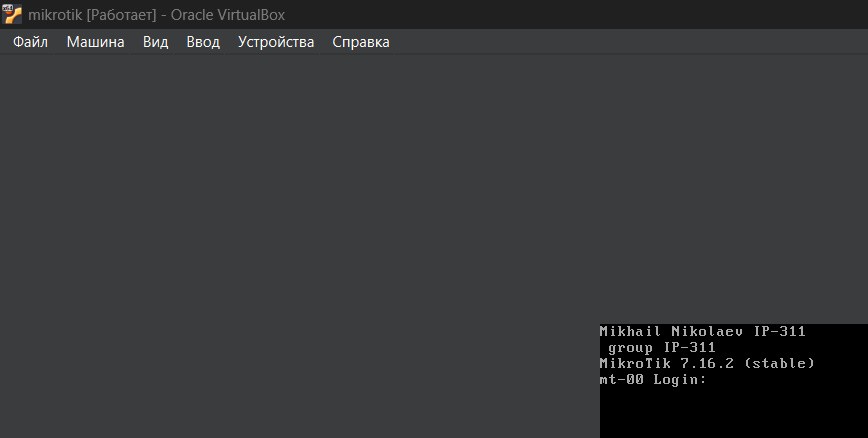


Рисунок 22. group IP-311

Далее я экспортировал клонированную виртуальную машину mikrotik-clone и получил резервную копию в .ova формате. Дальше удалил все файлы виртуальной машины mikrotik-clone.

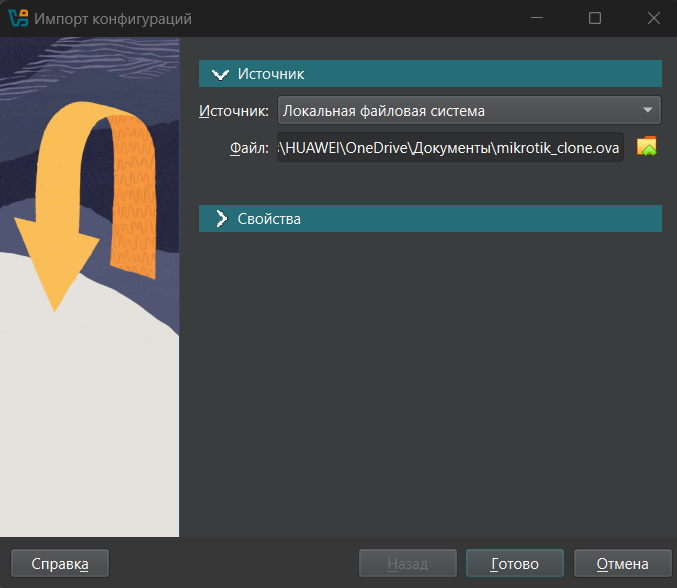


Рисунок 23. Импорт виртуальной машины из резервной копии.

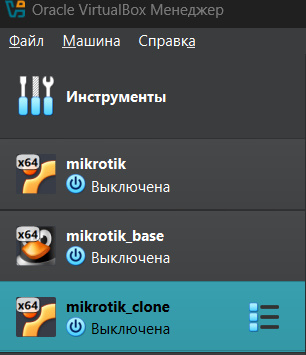


Рисунок 24. Виртуальная машина появилась в списке и готова к использованию.

После установки БазальтОС перевёл созданный жесткий диск в режим множественного использования.

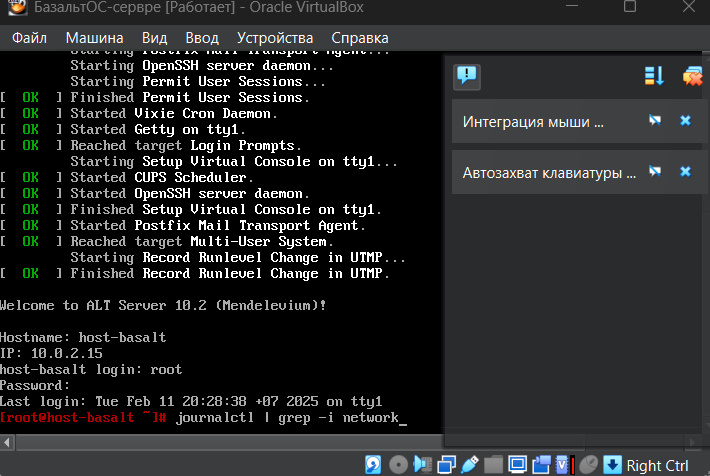


Рисунок 25. Регистрация от администратора и команда для вывода всех записей network

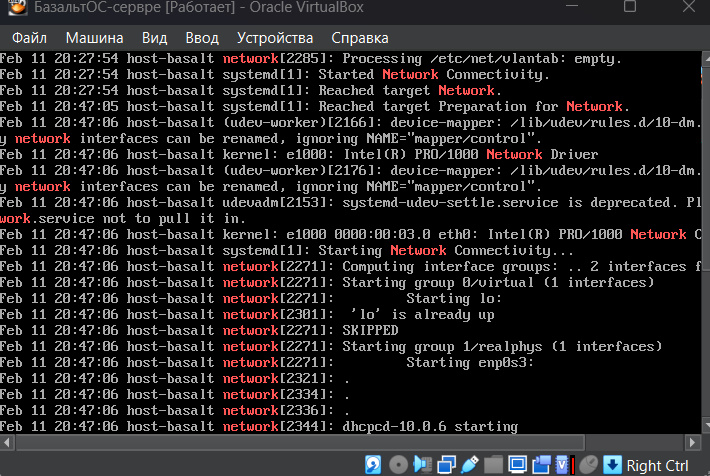


Рисунок 26. Все записи network

Далее зарегистрировавшись как обычный пользователь, используя конвейер команд, я вывел полную информацию об элементах каталога /usr/share/doc, изменил регистр у первых трёх букв.

Команда ls /usr/share/doc выводит список файлов в директории /usr/share/doc.

grep "lib" — фильтрует вывод

sed 's/lib/LIB/g' – замена.

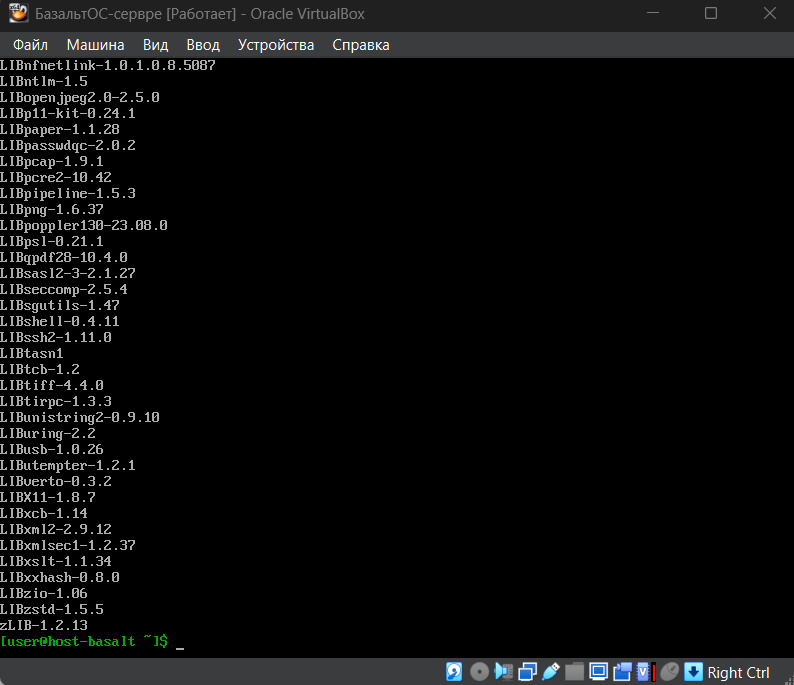


Рисунок 27. Вывод полной информации. Команда ls /usr/share/doc | grep "lib" | sed 's/lib/LIB/g'

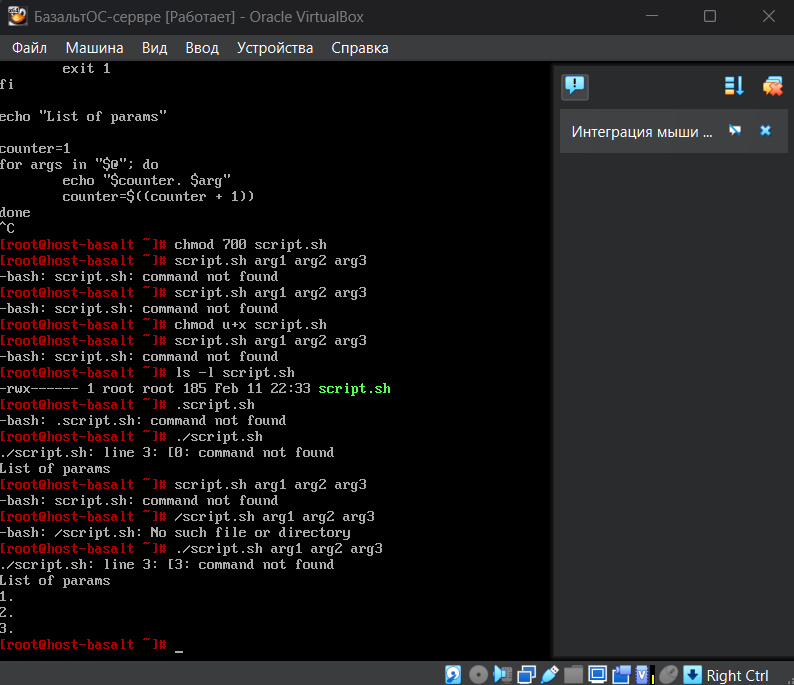


Рисунок 28. Написал скрипт, который выводит список порядковых номеров в строке. Доступно только владельцу.

После регистрации от имени администратора, используя потоковый редактор sed удалил в файле /etc/sudoers символы ”# “ (решетка и пробел) вначале строк содержащих подстроку SUDO\_USERS. Добавил группу пользователей sudo и включите в неё локального пользователя.

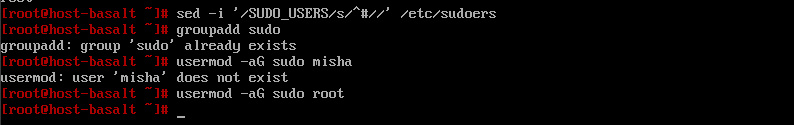


Рисунок 29.

После установки операционной системы запустил виртуальную машину. Создал в ней в каталоге /etc/network/interfaces.d текстовый файл eth0 следующего содержания auto eth0 iface eth0 inet dhcp

Перезагрузил ВМ, и перевел диск в режим множественного использования.

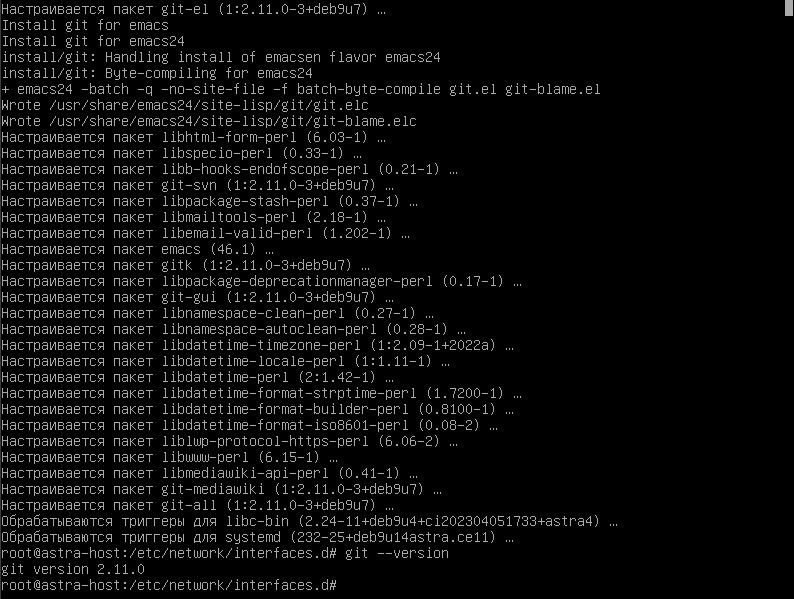


Рисунок 30. Установил git –version и все пакеты.

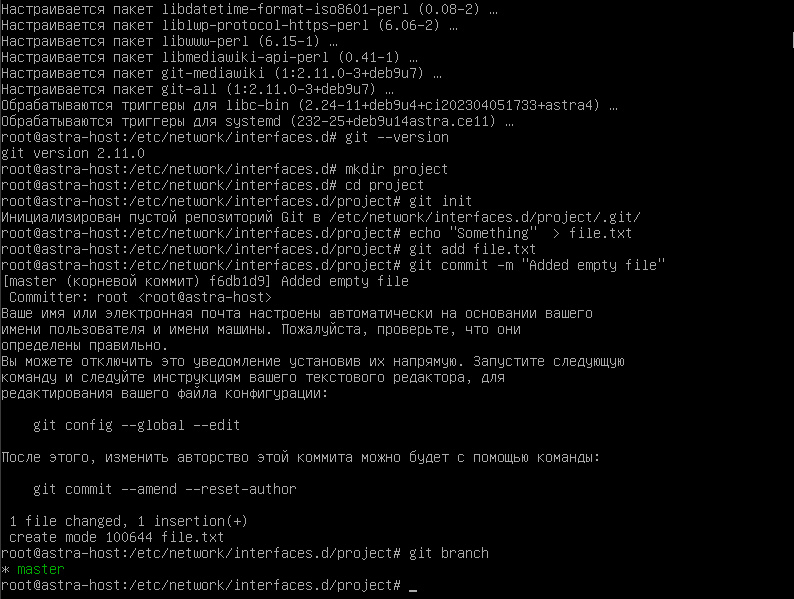


Рисунок 31. Создал проект и инициализировал в нём локальный репозиторий. По умолчанию создана ветка master.

Настроил приглашение cmd bash, чтобы показывалась текущая ветка.

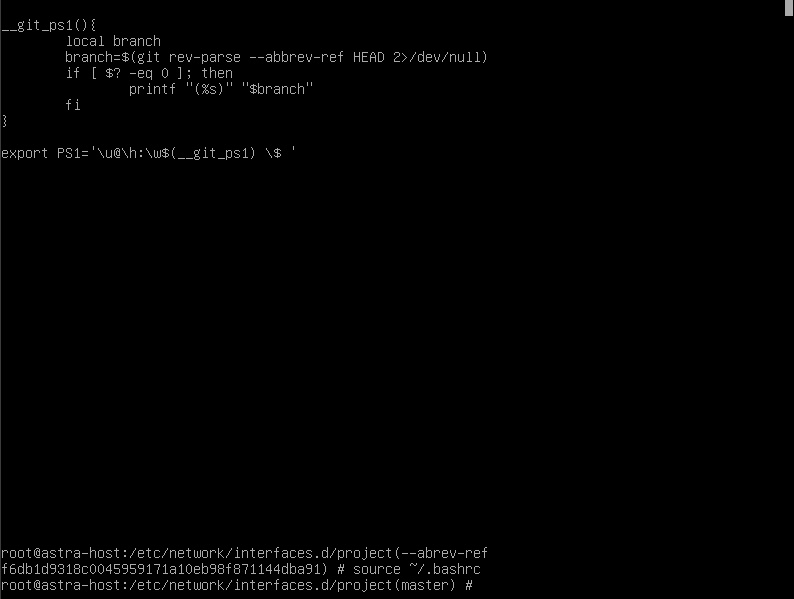


Рисунок 32. Отображение текущей ветки.