

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра теории вероятностей и кибербезопасности**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

*дисциплина: Моделирование сетей передачи данных*

Студент: Быстров Глеб

Группа: НПИбд-01-20

**МОСКВА**

2023 г.

## **Цель работы**

В данной лабораторной работе мне будет необходимо развернуть в системе виртуализации mininet, познакомиться с основными командами для работы с Mininet через командную строку и через графический интерфейс.

## Описание процесса выполнения работы

### 1.3.1. Настройка стенда виртуальной машины Mininet

#### 1. Настройка образа VirtualBox.

Перейдите в репозиторий Mininet. Скачайте актуальный релиз ovf-образа виртуальной машины.

При необходимости переместите скачанный образ в каталог для работы, затем распакуйте его.

Запустите систему виртуализации и импортируйте файл .ovf (рис. 1.1).

Перейдите в настройки системы виртуализации и уточните параметры настройки виртуальной машины. В частности, для VirtualBox выберете импортированную виртуальную машину и перейдите в меню Машина Настроить.

Перейдите к опции «Система». Если внизу этого окна есть сообщение об обнаружении неправильных настроек, то, следуя рекомендациям, внесите исправления. Например, может потребоваться увеличить видеопамять виртуальной машины и изменить тип графического контроллера на рекомендуемый. В настройках сети первый адаптер должен иметь подключение типа

NAT. Для второго адаптера укажите тип подключения host-only network adapter (виртуальный адаптер хоста), который в дальнейшем вы будете использовать для входа в образ виртуальной машины. В этом режиме адаптер хоста использует специальное устройство vboxnet0, создает подсеть и назначает IP-адрес сетевой карте гостевой операционной системы. Если данный режим не получается выбрать в настройках, то воспользуйтесь менеджером сетей хоста в VirtualBox для создания сети и настройки адаптера.

Запустите виртуальную машину с Mininet.

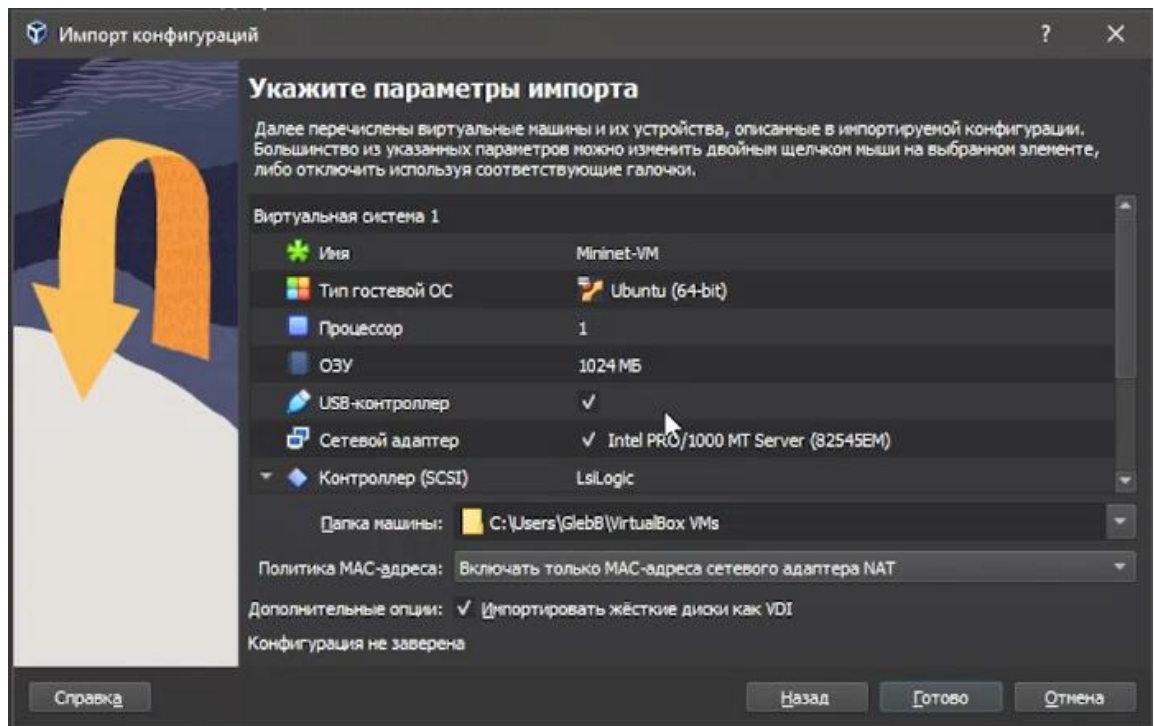


Рис. 1.1. Импортирование конфигурации

## 2. Подключение к виртуальной машине.

Залогиньтесь в виртуальной машине:

login: mininet

password: mininet

Посмотрите адрес машины:

ifconfig

Внутренний адрес машины будет иметь вид 192.168.x.y.

Подключитесь к виртуальной машине (из терминала хостовой машины):

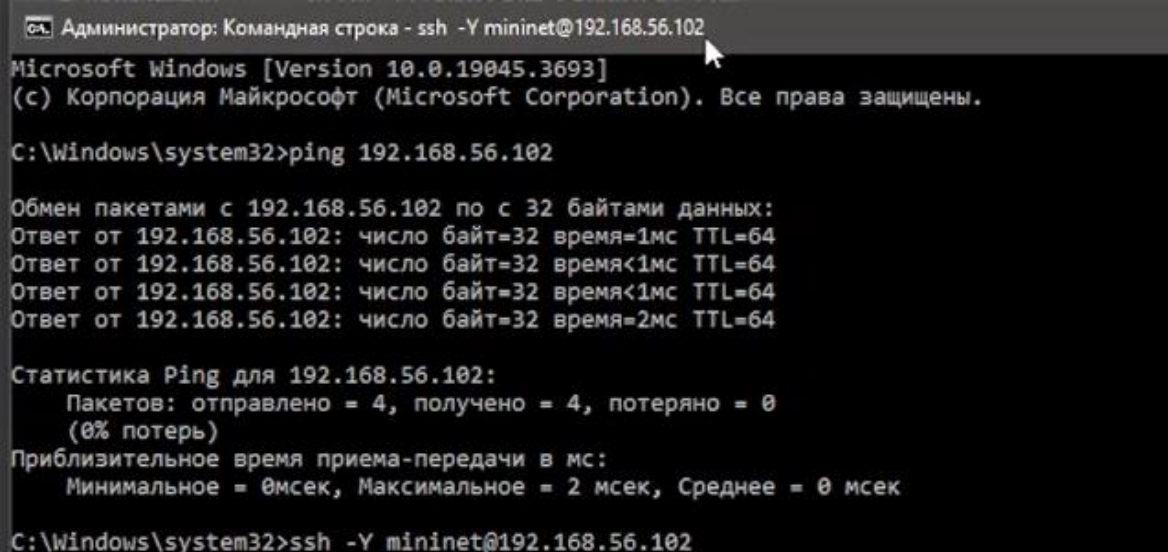
```
1 ssh -Y mininet@192.168.x.y
```

Для отключения ssh-соединения с виртуальной машиной нажмите Ctrl + d .

Настройте ssh-подсоединение по ключу к виртуальной машине, для чего в терминале основной Linux-машины перейдите в каталог .ssh своего домашнего каталога и введите (вместо 192.168.x.y укажите внутренний адрес виртуальной машины Mininet):

1 ssh-copy-id mininet@192.168.x.y

Вновь подключитесь к виртуальной машине и убедитесь, что подключение происходит успешно и без ввода пароля (рис. 1.2).



```
Администратор: Командная строка - ssh -Y mininet@192.168.56.102
Microsoft Windows [Version 10.0.19045.3693]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Windows\system32>ping 192.168.56.102

Обмен пакетами с 192.168.56.102 по 32 байтами данных:
Ответ от 192.168.56.102: число байт=32 время=1мс TTL=64
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64
Ответ от 192.168.56.102: число байт=32 время=2мс TTL=64

Статистика Ping для 192.168.56.102:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 2 мсек, Среднее = 0 мсек

C:\Windows\system32>ssh -Y mininet@192.168.56.102
```

Рис. 1.2. 2. Подключение к виртуальной машине

### 3. Настройка доступа к Интернет.

После подключения к виртуальной машине mininet посмотрите IP-адреса машины:

```
ifconfig
```

Для доступа к сети Интернет должен быть активен адрес NAT: 10.0.0.x.

Если активен только внутренний адрес машины вида 192.168.x.y, то активируйте второй интерфейс, набрав в командной строке:

```
sudo dhclient eth1
```

```
ifconfig
```

Для удобства дальнейшей работы установите mc:

```
sudo apt install mc
```

Для удобства дальнейшей работы добавьте для mininet указание на использование двух адаптеров при запуске. Для этого требуется перейти в режим суперпользователя и внести изменения в файл /etc/netplan/01-netcfg.yaml виртуальной машины mininet:

```
sudo mcedit /etc/netplan/01-netcfg.yaml
```

В результате файл /etc/netplan/01-netcfg.yaml должен иметь следующий вид:

network:

version: 2

renderer: networkd

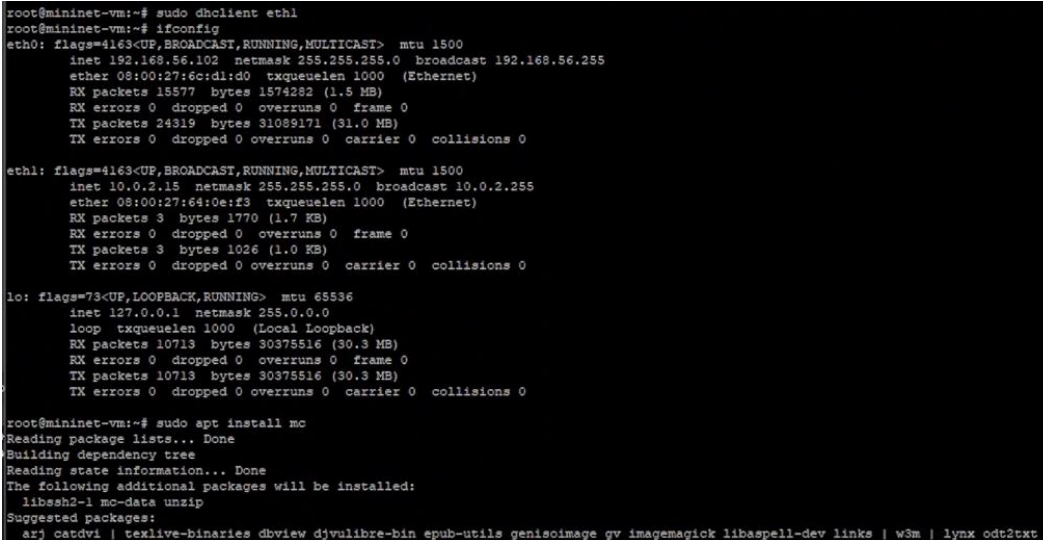
ethernets:

eth0:

dhcp4: yes

eth1:

dhcp4: yes (рис. 1.3).



```
root@mininet-vmr:~# sudo dhclient eth1
root@mininet-vmr:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:6c:d1:d0 txqueuelen 1000 (Ethernet)
    RX packets 18577 bytes 1574282 (1.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24319 bytes 31089171 (31.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    ether 08:00:27:64:0e:f3 txqueuelen 1000 (Ethernet)
    RX packets 3 bytes 1770 (1.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3 bytes 1026 (1.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 10713 bytes 30375516 (30.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10713 bytes 30375516 (30.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vmr:~# sudo apt install mc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libssh2-1 mc-data unzip
Suggested packages:
  arj catdvi | texlive-binaries dbview djvulibre-bin epub-utils genisoimage gv imagemagick libaspell-dev links | w3m | lynx odt2txt
```

Рис. 1.3. 3. Настройка доступа к Интернет

#### 4. Обновление версии Mininet.

В виртуальной машине mininet переименуйте предыдущую установку Mininet:

```
mv ~/mininet ~/mininet.orig
```

Скачайте новую версию Mininet:

```
cd ~
```

```
git clone https://github.com/mininet/mininet.git
```

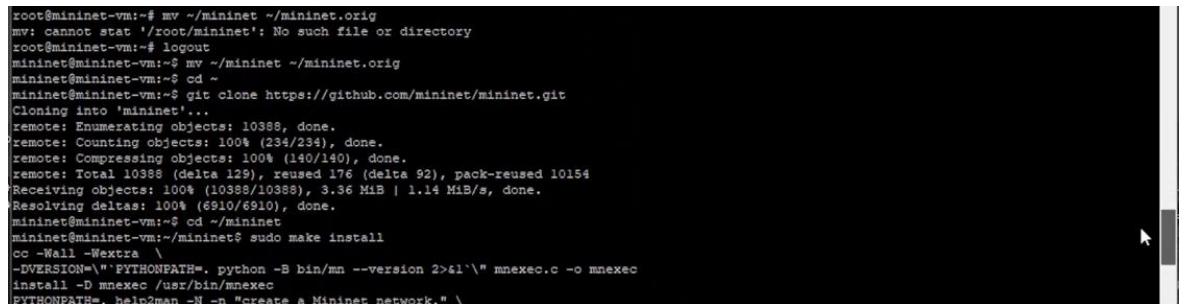
Обновите исполняемые файлы:

```
cd ~/mininet
```

`sudo make install`

Проверьте номер установленной версии mininet:

`mn --version` (рис. 1.4).



```
root@mininet-vm:~# mv ~/mininet ~/mininet.orig
mv: cannot stat '/root/mininet': No such file or directory
root@mininet-vm:~# logout
mininet@mininet-vm:~$ mv ~/mininet ~/mininet.orig
mininet@mininet-vm:~$ cd -
mininet@mininet-vm:~$ git clone https://github.com/mininet/mininet.git
Cloning into 'mininet'...
remote: Enumerating objects: 10388, done.
remote: Counting objects: 100% (234/234), done.
remote: Compressing objects: 100% (140/140), done.
remote: Total 10388 (delta 128), reused 176 (delta 92), pack-reused 10154
Receiving objects: 100% (10388/10388), 3.36 MiB | 1.14 MiB/s, done.
Resolving deltas: 100% (6910/6910), done.
mininet@mininet-vm:~$ cd ~/mininet
mininet@mininet-vm:~/mininet$ sudo make install
cc -Wall -Wextra \
-DVERSION="" PYTHONPATH=. python -B bin/mn --version 2>&1\'\' mnxec.c -o mnxec
install -D mnxec /usr/bin/mnxec
PYTHONPATH=. help2man -N -n "create a Mininet network." \'\"
```

Рис. 1.4. Обновление версии Mininet

## 5. Настройка параметров XTerm.

По умолчанию XTerm использует растровые шрифты малого кегля. Для увеличения размера шрифта и применения векторных шрифтов вместо растровых необходимо внести изменения в файл `/etc/X11/app-defaults/XTerm`. Для этого можно воспользоваться следующей командой:

`sudo mcedit /etc/X11/app-defaults/XTerm`

и затем в конце файла добавить строки:

`xterm*faceName: Monospace`

`xterm*faceSize: 12`

Здесь выбран системный моноширинный шрифт, кегль шрифта — 12 пунктов. (рис. 1.5).



```
mininet@mininet-vm:~/mininet$ sudo mcedit /etc/X11/app-defaults/XTerm
mininet@mininet-vm:~/mininet$ sudo ~/mininet/mininet/examples/miniedit.py
topo=none
^Z
[1]+  Stopped                  sudo ~/mininet/mininet/examples/miniedit.py
mininet@mininet-vm:~/mininet$ xterm
mininet@mininet-vm:~/mininet$ sudo ~/mininet/mininet/examples/miniedit.py
```

Рис. 1.5. 5. Настройка параметров XTerm

## 6. Настройка соединения X11 для суперпользователя.

При попытке запуска приложения из-под суперпользователя возникает ошибка:

X11 connection rejected because of wrong authentication.

Ошибка возникает из-за того, что X-соединение выполняется от имени пользователя mininet, а приложение запускается от имени пользователя root с использованием sudo. Для исправления этой ситуации необходимо заполнить файл полномочий /root/.Xauthority, используя утилиту xauth.

Скопируйте значение куки (MIT magic cookie)<sup>1</sup> пользователя mininet в файл для пользователя root:

```
mininet@mininet-vm:~$ xauth list $DISPLAY
```

```
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1
```

```
↪ 295acad8e35d17636924c5ab80e8462d
```

```
mininet@mininet-vm:~$ sudo -i
```

```
mininet@mininet-vm:~$ xauth list
```

```
xauth: file /root/.Xauthority does not exist (рис. 1.6).
```

```
root@mininet-vm:~# xauth add mininet-vm/unix:10
```

```
↪ MIT-MAGIC-COOKIE-1 295acad8e35d17636924c5ab80e8462d
```

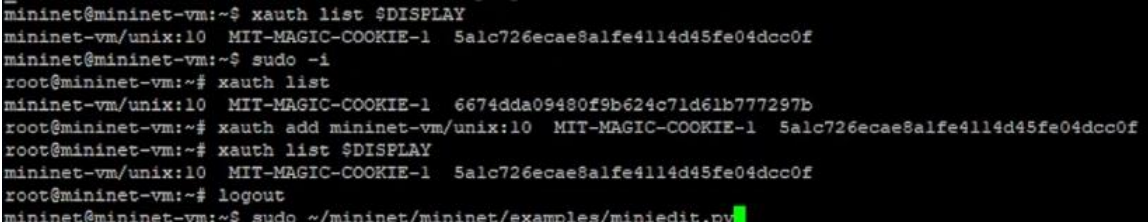
```
root@mininet-vm:~# xauth list $DISPLAY
```

```
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1
```

```
↪ 295acad8e35d17636924c5ab80e8462d
```

```
root@mininet-vm:~# logout
```

После выполнения этих действий графические приложения должны запускаться под пользователем mininet.



```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 5alc726ecae8alfe4114d45fe04dcc0f
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 6674dda09480f9b624c71d61b777297b
root@mininet-vm:~# xauth add mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 5alc726ecae8alfe4114d45fe04dcc0f
root@mininet-vm:~# xauth list $DISPLAY
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 5alc726ecae8alfe4114d45fe04dcc0f
root@mininet-vm:~# logout
mininet@mininet-vm:~$ sudo ~/mininet/mininet/examples/miniedit.py
```

Рис. 1.6. 6. Настройка соединения X11 для суперпользователя

## 7. Работа с Mininet из-под Windows (рис. 1.7).

Варианты Xserver. Использовать можно любой вариант Xserver.

VcXsrv Windows X Server:



Сайт: <https://sourceforge.net/projects/vcxsrv/>

Лицензия: GPLv3.

Компиляция: Visual C++ 2012 Express Edition

Xming X Server:

Сайт: <http://www.straightrunning.com/XmingNotes/>

Лицензия:

Public Domain Release: MIT

Website Release: за загрузку новых выпусков необходимо вносить плату; приобретение лицензии дает пользователю доступ к новым загрузкам в течение одного года.

Компиляция: MinGW, Pthreads-Win32.

Установка программного обеспечения.

Установите putty:

choco install putty

Установите VcXsrv Windows X Server:

choco install vcxsrv

Запуск Xserver.

Запустите XLaunch.

Выберите опции:

Multiple windows;

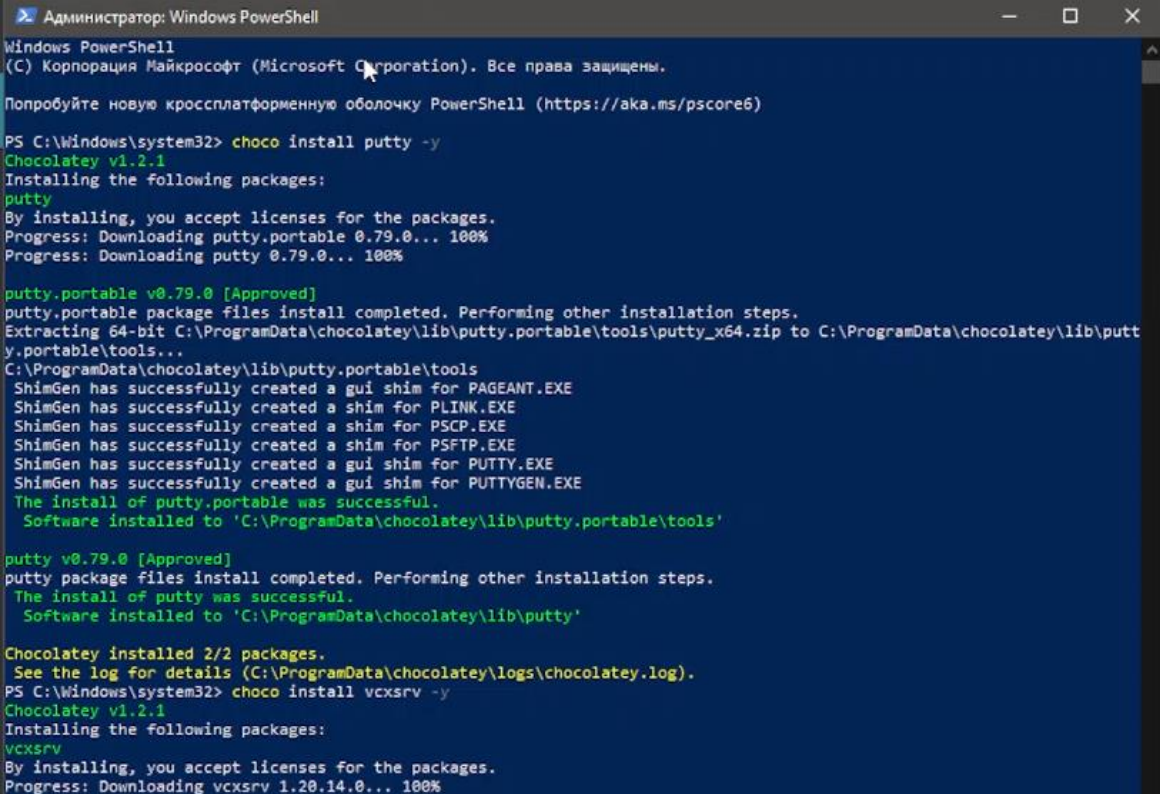
Display number: -1;

Start no client.

Можно сохранить параметры, тогда при следующем запуске не нужно будет отмечать эти опции.

Запуск putty. При подключении добавьте опцию перенаправления X11:

Connection SSH X11 : Enable X11 forwarding.



```
Администратор: Windows PowerShell
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Windows\system32> choco install putty -y
Chocolatey v1.2.1
Installing the following packages:
putty
By installing, you accept licenses for the packages.
Progress: Downloading putty.portable 0.79.0... 100%
Progress: Downloading putty 0.79.0... 100%

putty.portable v0.79.0 [Approved]
putty.portable package files install completed. Performing other installation steps.
Extracting 64-bit C:\ProgramData\chocolatey\lib\putty.portable\tools\putty_x64.zip to C:\ProgramData\chocolatey\lib\putt
y.portable\tools...
C:\ProgramData\chocolatey\lib\putty.portable\tools
ShimGen has successfully created a gui shim for PAGEANT.EXE
ShimGen has successfully created a shim for PLINK.EXE
ShimGen has successfully created a shim for PSCP.EXE
ShimGen has successfully created a shim for PSFTP.EXE
ShimGen has successfully created a gui shim for PUTTY.EXE
ShimGen has successfully created a gui shim for PUTTYGEN.EXE
The install of putty.portable was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\putty.portable\tools'

putty v0.79.0 [Approved]
putty package files install completed. Performing other installation steps.
The install of putty was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\putty'

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32> choco install vcxsrv -y
Chocolatey v1.2.1
Installing the following packages:
vcxsrv
By installing, you accept licenses for the packages.
Progress: Downloading vcxsrv 1.20.14.0... 100%
```

Рис. 1.7. Установка Putty и Xserver

### 1.3.2. Основы работы в Mininet

#### 8. Работа с Mininet с помощью командной строки

##### 8.1. Вызов Mininet с использованием топологии по умолчанию.

Для запуска минимальной топологии введите в командной строке (рис. 1.8):

```
sudo mn
```

```
mininet@mininet-vm:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> █
```

Рис. 1.8. Запуск минимальной топологии

Эта команда запускает Mininet с минимальной топологией, состоящей из коммутатора, подключённого к двум хостам.

Для отображения списка команд интерфейса командной строки Mininet и примеров их использования введите команду в интерфейсе командной строки Mininet:

help

Для отображения доступных узлов введите:

nodes

Вывод этой команды показывает, что есть два хоста (хост h1 и хост h2) и коммутатор (s1).

Иногда бывает полезно отобразить связи между устройствами в Mininet, чтобы понять топологию. Введите команду net в интерфейсе командной строки Mininet, чтобы просмотреть доступные линки:

net

Вывод этой команды показывает:

Хост h1 подключён через свой сетевой интерфейс h1-eth0 к коммутатору на интерфейсе s1-eth1.



Рис. 1.10. Команда ifconfig на хосте h1

### 8.2. Проверка связности.

По умолчанию узлам h1 и h2 назначаются IP-адреса 10.0.0.1/8 и 10.0.0.2/8 соответственно. Чтобы проверить связь между ними, вы можете использовать команду ping. Команда ping работает, отправляя сообщения эхо-запроса протокола управляющих сообщений Интернета (ICMP) на удалённый компьютер и ожидая ответа.

Например, команда

```
h1 ping 10.0.0.2
```

проверяет соединение между хостами h1 и h2. Для остановки теста нажмите Ctrl + c (рис. 1.11).

```
mininet> h1 ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.08 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.269 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.074 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.180 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.045 ms
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 510lms
rtt min/avg/max/mdev = 0.045/0.781/4.076/1.475 ms
```

Рис. 1.11. Проверка соединения между хостами h1 и h2

### 8.3. Остановка эмуляции.

Для завершения работы режима эмуляции mininet используйте команду exit (рис. 1.12).

Заметим, что команда sudo mn -с часто используется в терминале для очистки предыдущего экземпляра Mininet (например, после сбоя).

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 129.809 seconds
```

Рис. 1.12. Завершение работы режима эмуляции

## 9. Построение и эмуляция сети в Mininet с использованием графического интерфейса.

### 9.1. Построение топологии сети.

В терминале виртуальной машины mininet запустите MiniEdit:

```
sudo ~/mininet/mininet/examples/miniedit.py
```

Основные кнопки:

Select: позволяет выбирать/перемещать устройства. Нажатие Del на клавиатуре после выбора устройства удаляет его из топологии.

Host: позволяет добавить новый хост в топологию. После нажатия этой кнопки щелкните в любом месте пустого холста, чтобы вставить новый хост.

Switch: позволяет добавить в топологию новый коммутатор. После нажатия этой кнопки щелкните в любом месте пустого холста, чтобы вставить переключатель.

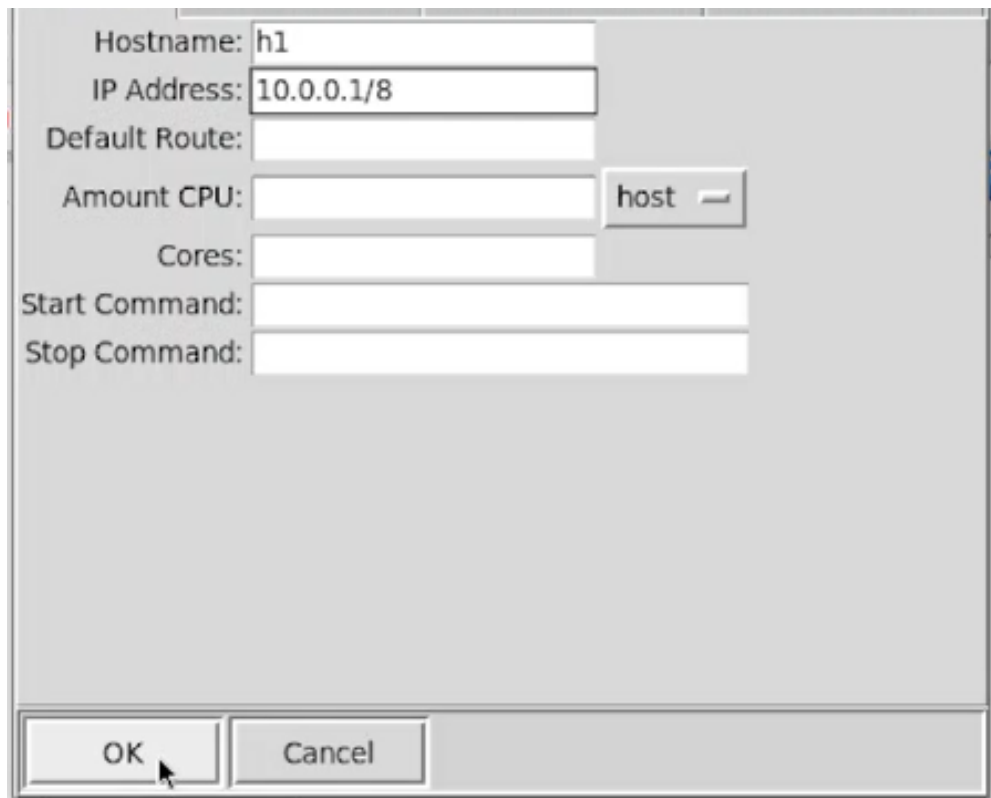
Link: соединяет устройства в топологии. После нажатия этой кнопки щелкните устройство и перетащите его на второе устройство, с которым необходимо установить связь.

Run: запускает эмуляцию. После проектирования и настройки топологии нажмите кнопку запуска.

Stop: останавливает эмуляцию.

Добавьте два хоста и один коммутатор, соедините хосты с коммутатором.

Настройте IP-адреса на хостах h1 и h2. Для этого удерживая правую кнопку мыши на устройстве выберите свойства. Для хоста h1 укажите IP-адрес 10.0.0.1/8, а для хоста h2 — 10.0.0.2/8 (рис. 1.13).



Hostname: h1

IP Address: 10.0.0.1/8

Default Route:

Amount CPU: host

Cores:

Start Command:

Stop Command:

OK Cancel

Рис. 1.13. Настройка IP-адреса на хосте h1

## 9.2. Проверка связности.

Перед проверкой соединения между хостом h1 и хостом h2 необходимо запустить эмуляцию. Для запуска эмуляции нажмите кнопку Run. После начала эмуляции кнопки панели MiniEdit станут серыми, указывая на то, что в настоящее время они отключены.

Откройте терминал на хосте h1, удерживая правую кнопку мыши на хосте h1 и выбрав Terminal. Это действие позволит выполнять команды на хосте h1.

Откройте терминал на хосте h2.

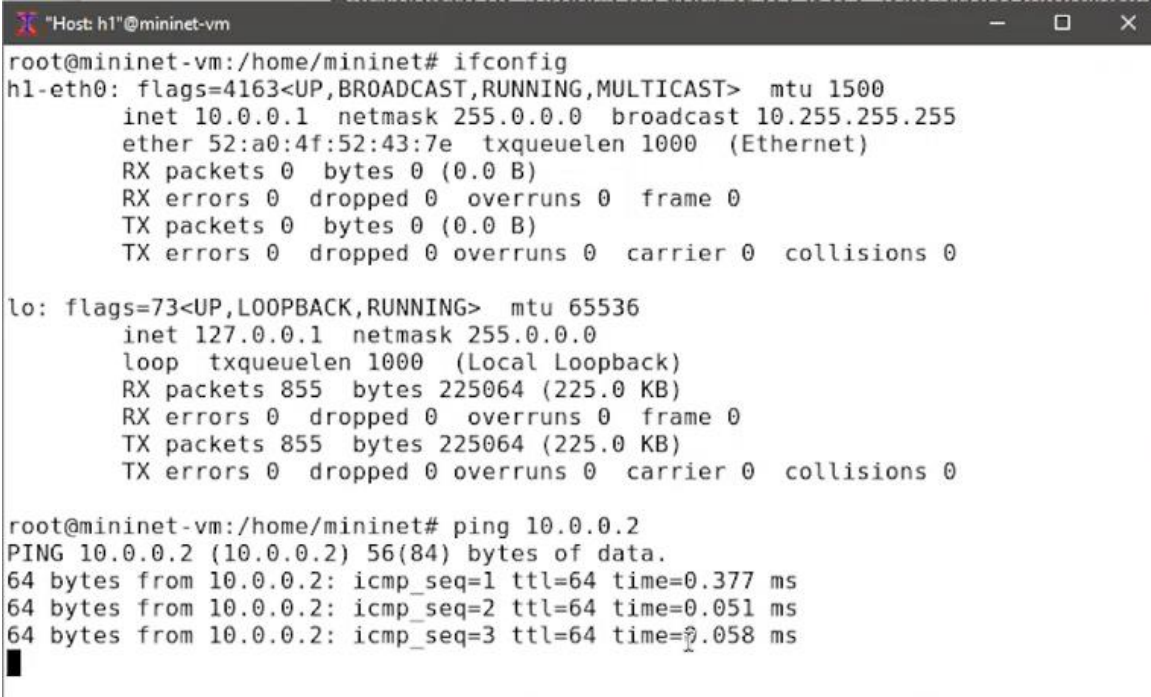
На терминале хоста h1 введите команду `ifconfig`, чтобы отобразить назначенные ему IP-адреса. Интерфейс h1-eth0 на хосте h1 должен быть настроен с IP-адресом 10.0.0.1 и маской подсети 255.0.0.0.

Повторите эти действия на хосте h2. Его интерфейс h2-eth0 должен быть настроен с IP-адресом 10.0.0.2 и маской подсети 255.0.0.0.

Проверьте соединение между хостами, введя в терминале хоста h1

команду `ping 10.0.0.2`. Для остановки теста нажмите `Ctrl + c`.

Остановите эмуляцию, нажав кнопку Stop (рис. 1.14).



```
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    ether 52:a0:4f:52:43:7e txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 855 bytes 225064 (225.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 855 bytes 225064 (225.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.377 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.058 ms
■
```

Рис. 1.14. Просмотр IP-адреса и проверка соединения

### 9.3. Автоматическое назначение IP-адресов.

Ранее IP-адреса узлов `h1` и `h2` были назначены вручную. В качестве альтернативы можно полагаться на Mininet для автоматического назначения IP-адресов.

Удалите назначенный вручную IP-адрес с хостов `h1` и `h2`.

В MiniEdit нажмите `Edit Preferences`. По умолчанию в поле базовые значения IP-адресов (`IP Base`) установлено `10.0.0.0/8`. Измените это значение на `15.0.0.0/8`.

Запустите эмуляцию, нажав кнопку `Run`.

Откройте терминал на хосте `h1`, удерживая правую кнопку мыши на хосте

`h1` и выбрав `Terminal`.

Чтобы отобразить IP-адреса, назначенные хосту `h1`, введите команду `ifconfig`

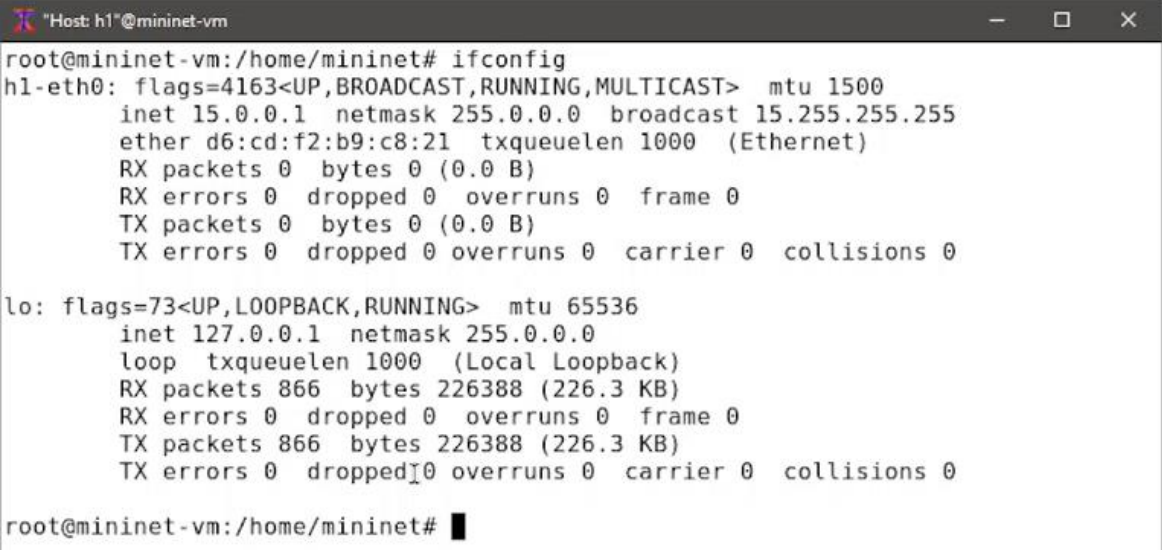
Интерфейс `h1-eth0` на узле `h1` теперь имеет IP-адрес `15.0.0.1` и маску



подсети 255.0.0.0 (рис. 1.15)..

Вы также можете проверить IP-адрес, назначенный хосту h2. Соответствующий интерфейс h2-eth0 на хосте h2 должен иметь IP-адрес 15.0.0.2 и маску подсети 255.0.0.0.

Остановите эмуляцию, нажав кнопку Stop.



```
root@mininet-vm:/home/mininet# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 15.0.0.1 netmask 255.0.0.0 broadcast 15.255.255.255
    ether d6:cd:f2:b9:c8:21 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 866 bytes 226388 (226.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 866 bytes 226388 (226.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@mininet-vm:/home/mininet#
```

Рис. 1.15. Просмотр IP-адреса

#### 4. Сохранение и загрузка топологии Mininet.

В домашнем каталоге виртуальной машины mininet создайте каталог для работы с проектами mininet:

```
mkdir ~/work
```

Для сохранения топологии сети в файл нажмите в MiniEdit File Save.

Укажите имя для топологии и сохраните на своём компьютере.

После сохранения проекта поменяйте права доступа к файлам в каталоге проекта:

```
sudo chown -R mininet:mininet ~/work
```

Для загрузки топологии в MiniEdit нажмите File Open (рис. 1.16).

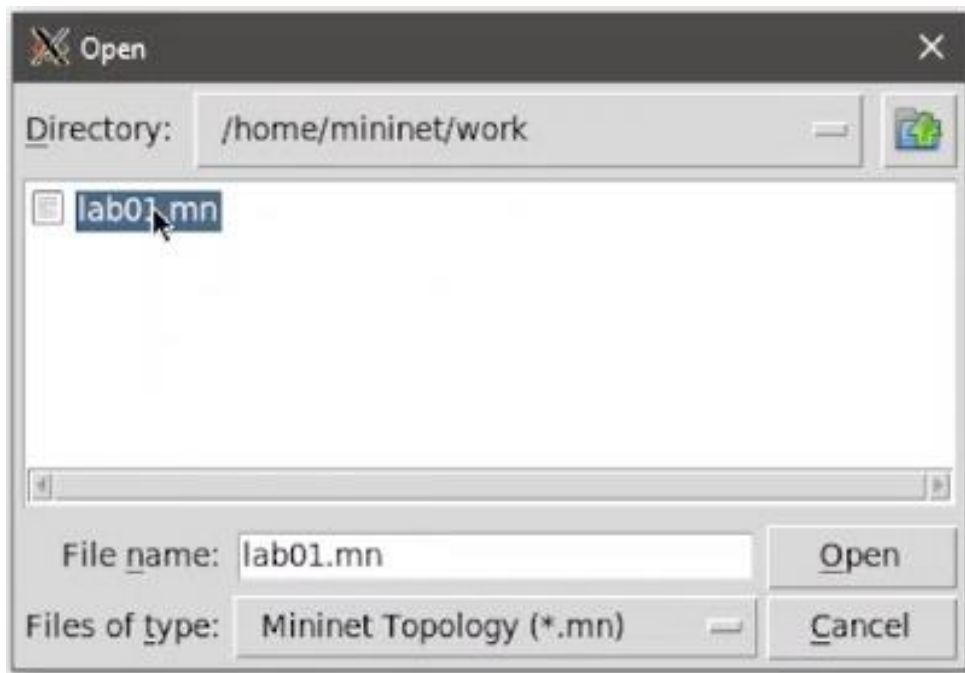


Рис. 1.16. Открытие файла

9.5. Завершите соединение с виртуальной машиной mininet и выключите её (рис. 1.17).

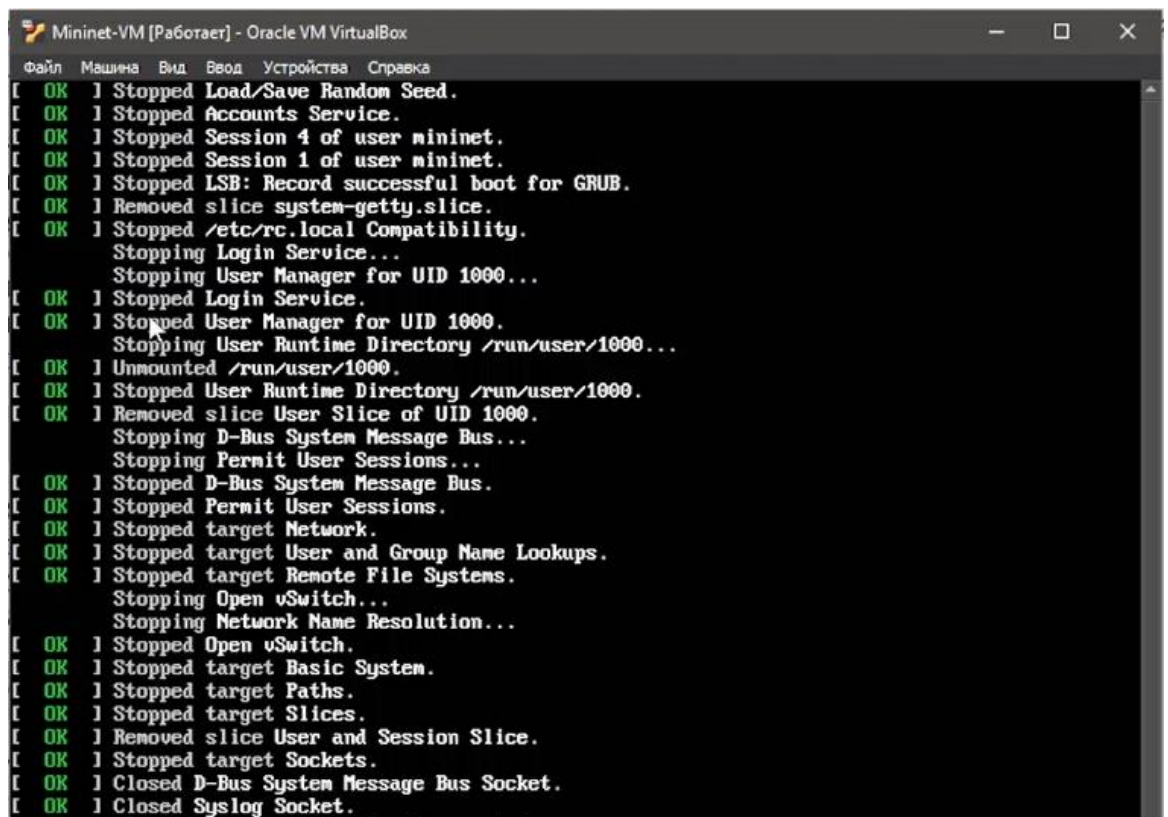


Рис. 1.17. Выключение виртуальной машины

## **Вывод**

В данной лабораторной работе мне успешно удалось развернуть в системе виртуализации mininet, познакомиться с основными командами для работы с Mininet через командную строку и через графический интерфейс.