

Лабораторная работа №1  
Моделирование сетей передачи данных

---

Исаев Б. А.

2025

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

## Настройка образа VirtualBox

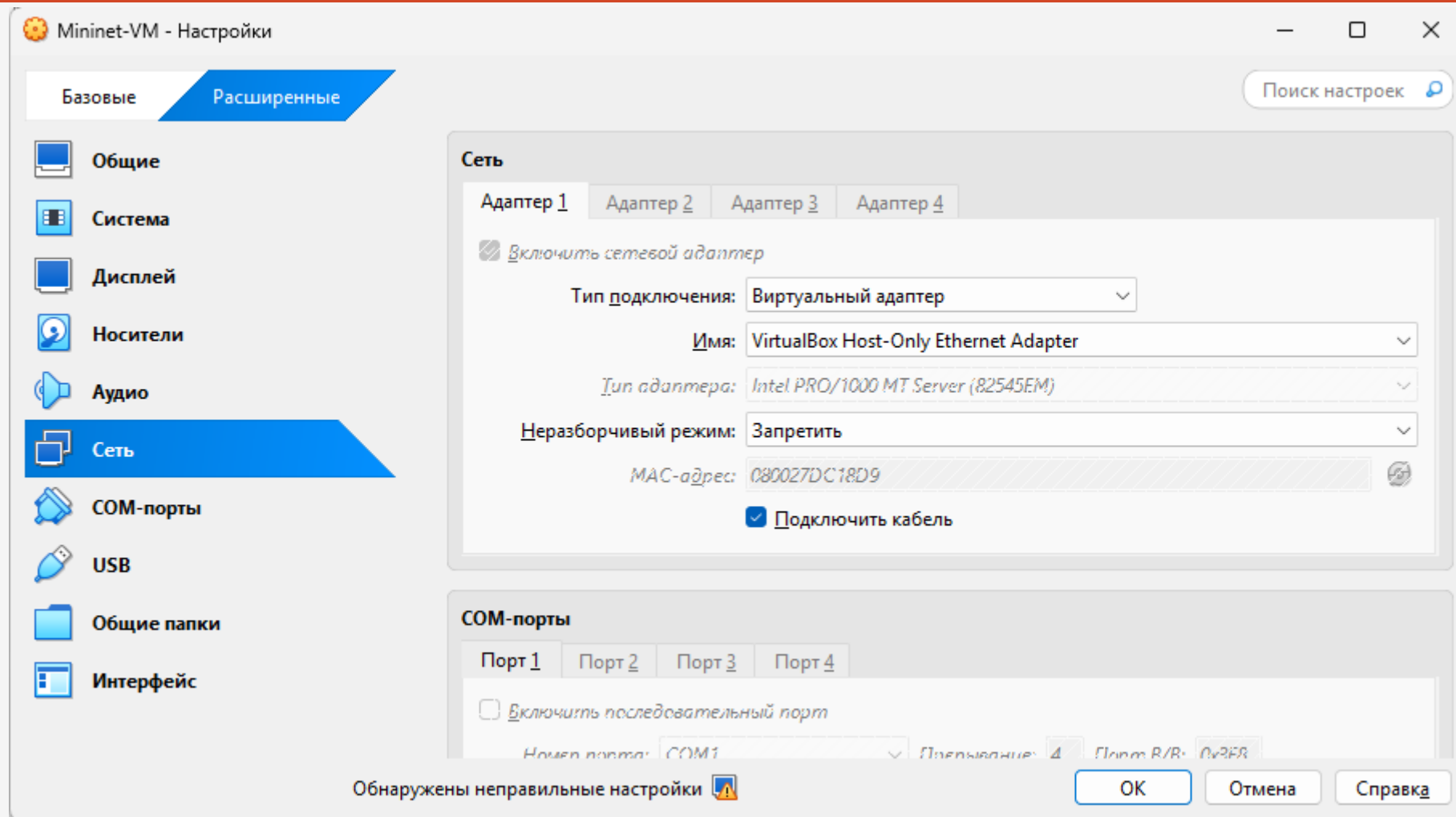
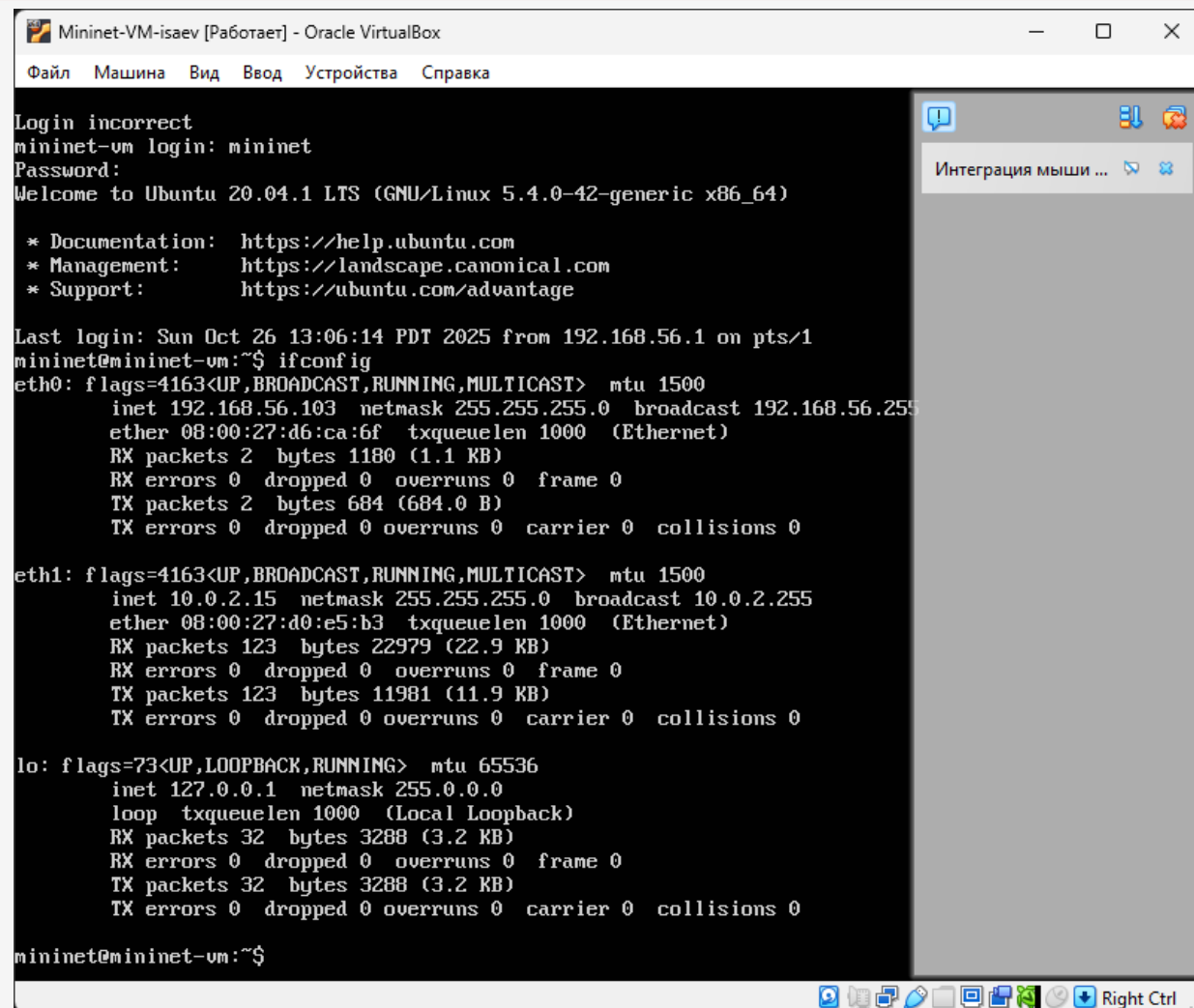


Рис. 1: Установка и настройка виртуальной машины

## Подключение к виртуальной машине



```
Mininet-VM-isaev [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

Login incorrect
mininet-vm login: mininet
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Last login: Sun Oct 26 13:06:14 PDT 2025 from 192.168.56.1 on pts/1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.103  netmask 255.255.255.0  broadcast 192.168.56.255
    ether 08:00:27:d6:ca:6f  txqueuelen 1000  (Ethernet)
    RX packets 2  bytes 1180 (1.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2  bytes 684 (684.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

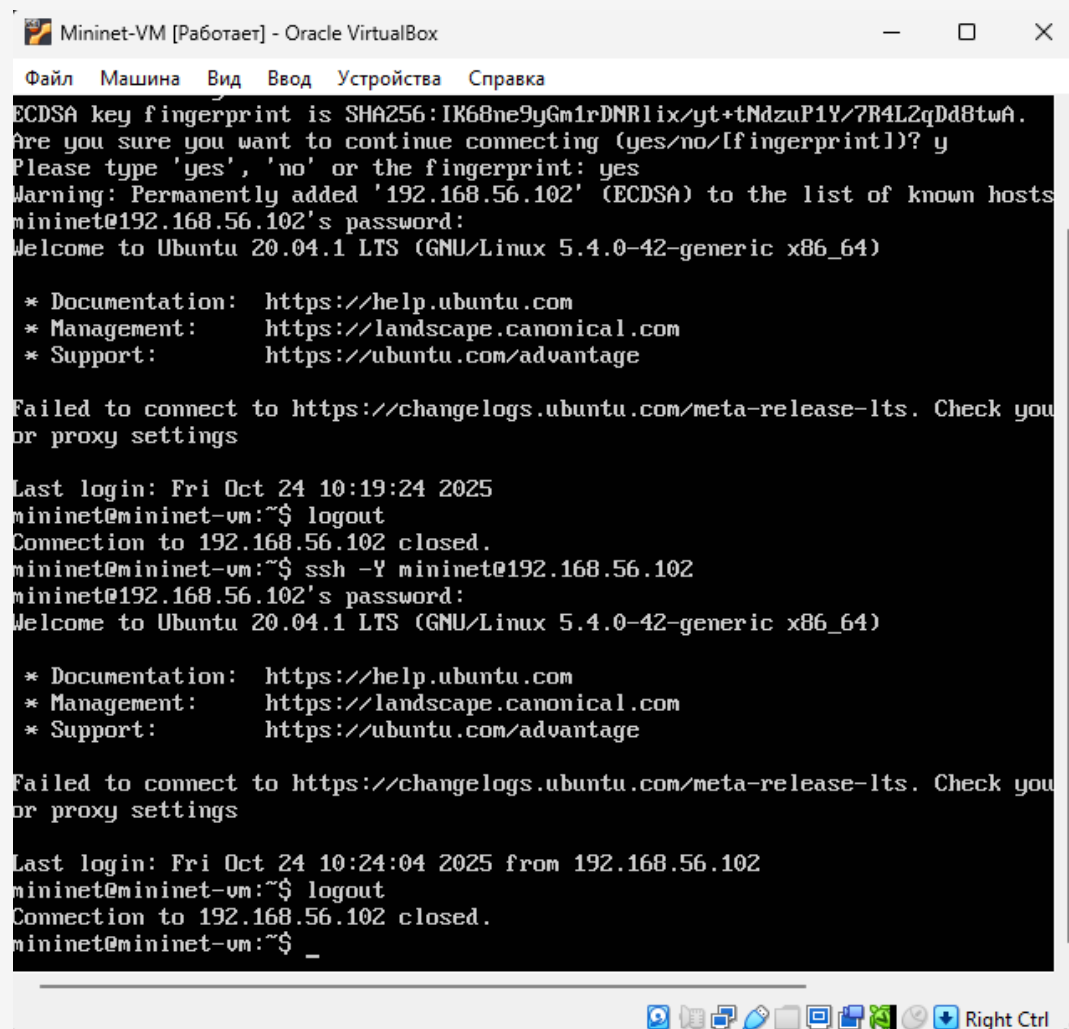
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    ether 08:00:27:d0:e5:b3  txqueuelen 1000  (Ethernet)
    RX packets 123  bytes 22979 (22.9 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 123  bytes 11981 (11.9 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 32  bytes 3288 (3.2 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 32  bytes 3288 (3.2 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

mininet@mininet-vm:~$
```

Рис. 2: Вход и просмотр адреса виртуальной машины

## Подключение к виртуальной машине



```
Mininet-VM [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
ECDSA key fingerprint is SHA256:IK68ne9yGm1rDNRLix/yt+tNdzuP1Y/7R4L2qDd8twA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.102' (ECDSA) to the list of known hosts
mininet@192.168.56.102's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
or proxy settings

Last login: Fri Oct 24 10:19:24 2025
mininet@mininet-vm:~$ logout
Connection to 192.168.56.102 closed.
mininet@mininet-vm:~$ ssh -Y mininet@192.168.56.102
mininet@192.168.56.102's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

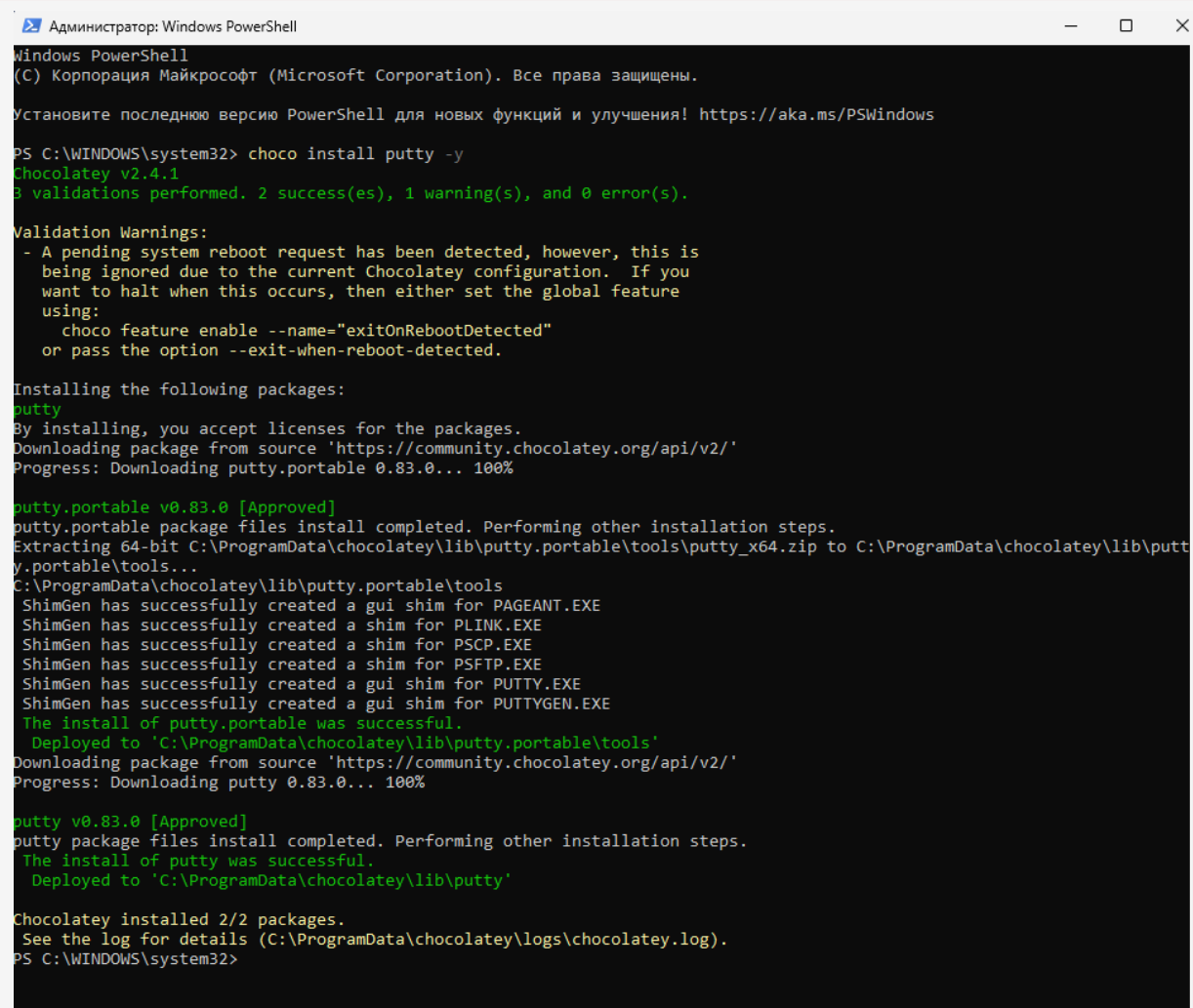
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
or proxy settings

Last login: Fri Oct 24 10:24:04 2025 from 192.168.56.102
mininet@mininet-vm:~$ logout
Connection to 192.168.56.102 closed.
mininet@mininet-vm:~$ _
```

Рис. 3: Подключение к виртуальной машине из терминала хостовой машины

## Работа с Mininet из-под Windows



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

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> choco install putty -y
Chocolatey v2.4.1
3 validations performed. 2 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:
- A pending system reboot request has been detected, however, this is
  being ignored due to the current Chocolatey configuration. If you
  want to halt when this occurs, then either set the global feature
  using:
    choco feature enable --name="exitOnRebootDetected"
  or pass the option --exit-when-reboot-detected.

Installing the following packages:
putty
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading putty.portable 0.83.0... 100%

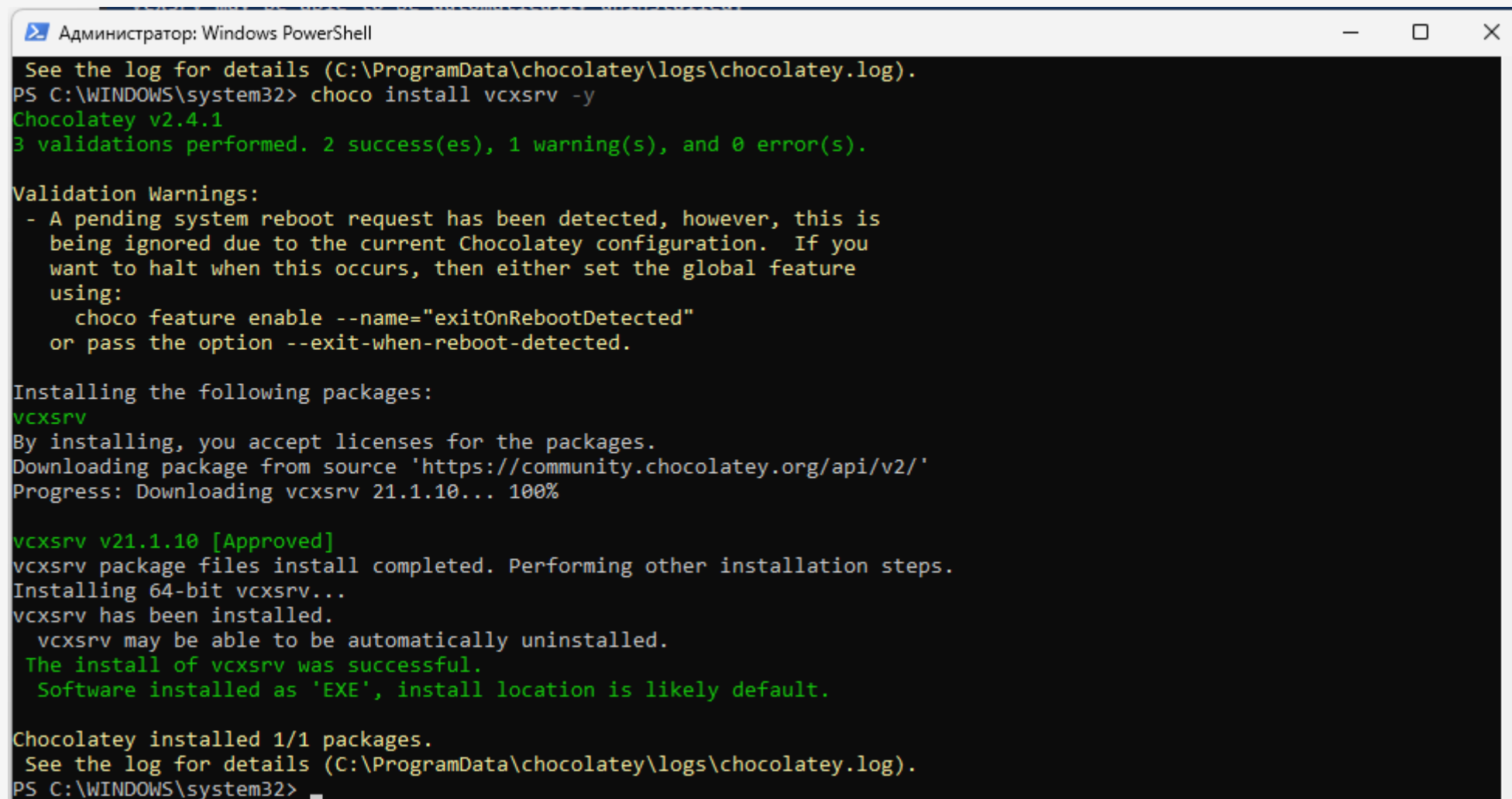
putty.portable v0.83.0 [Approved]
putty.portable package files install completed. Performing other installation steps.
Extracting 64-bit C:\ProgramData\chocolatey\lib\putty.portable\tools\putty_x64.zip to C:\ProgramData\chocolatey\lib\putty.portable\tools...
C:\ProgramData\chocolatey\lib\putty.portable\tools
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.
  Deployed to 'C:\ProgramData\chocolatey\lib\putty.portable\tools'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading putty 0.83.0... 100%

putty v0.83.0 [Approved]
putty package files install completed. Performing other installation steps.
The install of putty was successful.
  Deployed 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>
```

Рис. 4: Установка putty

## Работа с Mininet из-под Windows



```
Администратор: Windows PowerShell

See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32> choco install vcxsrv -y
Chocolatey v2.4.1
3 validations performed. 2 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:
- A pending system reboot request has been detected, however, this is
  being ignored due to the current Chocolatey configuration. If you
  want to halt when this occurs, then either set the global feature
  using:
    choco feature enable --name="exitOnRebootDetected"
  or pass the option --exit-when-reboot-detected.

Installing the following packages:
vcxsrv
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading vcxsrv 21.1.10... 100%

vcxsrv v21.1.10 [Approved]
vcxsrv package files install completed. Performing other installation steps.
Installing 64-bit vcxsrv...
vcxsrv has been installed.
vcxsrv may be able to be automatically uninstalled.
The install of vcxsrv was successful.
Software installed as 'EXE', install location is likely default.

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32>
```

Рис. 5: Установка putty VcXsrv Windows X Server

## Работа с Mininet из-под Windows

---

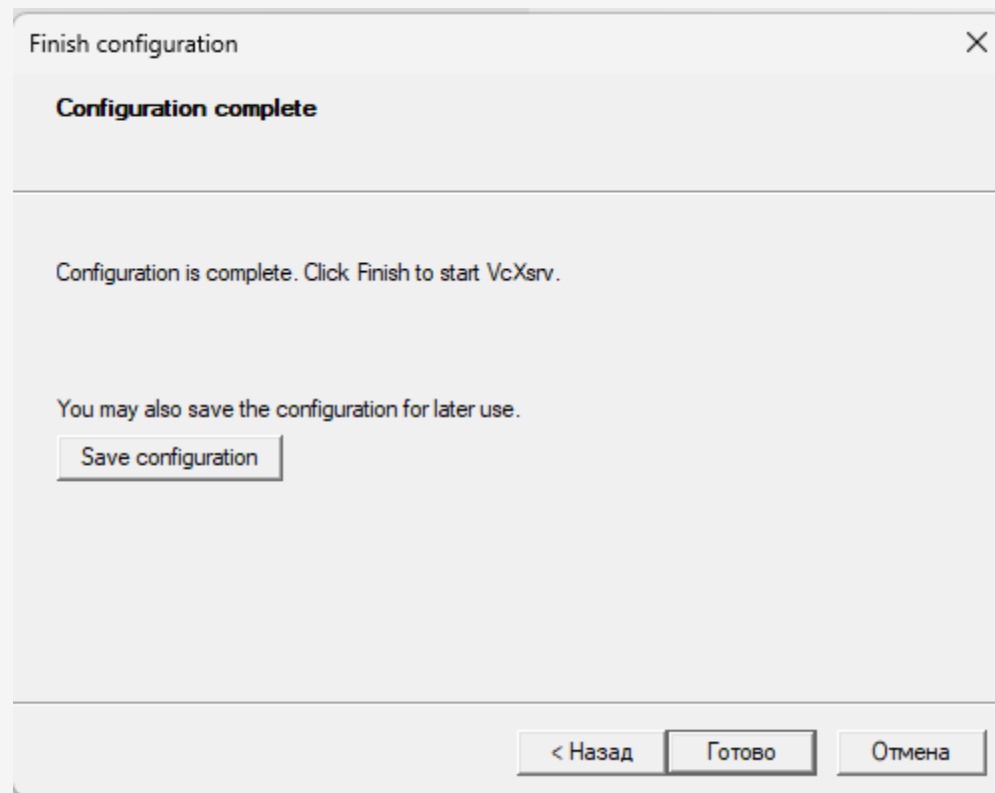


Рис. 6: Запуск и настройка Xserver

## Работа с Mininet из-под Windows

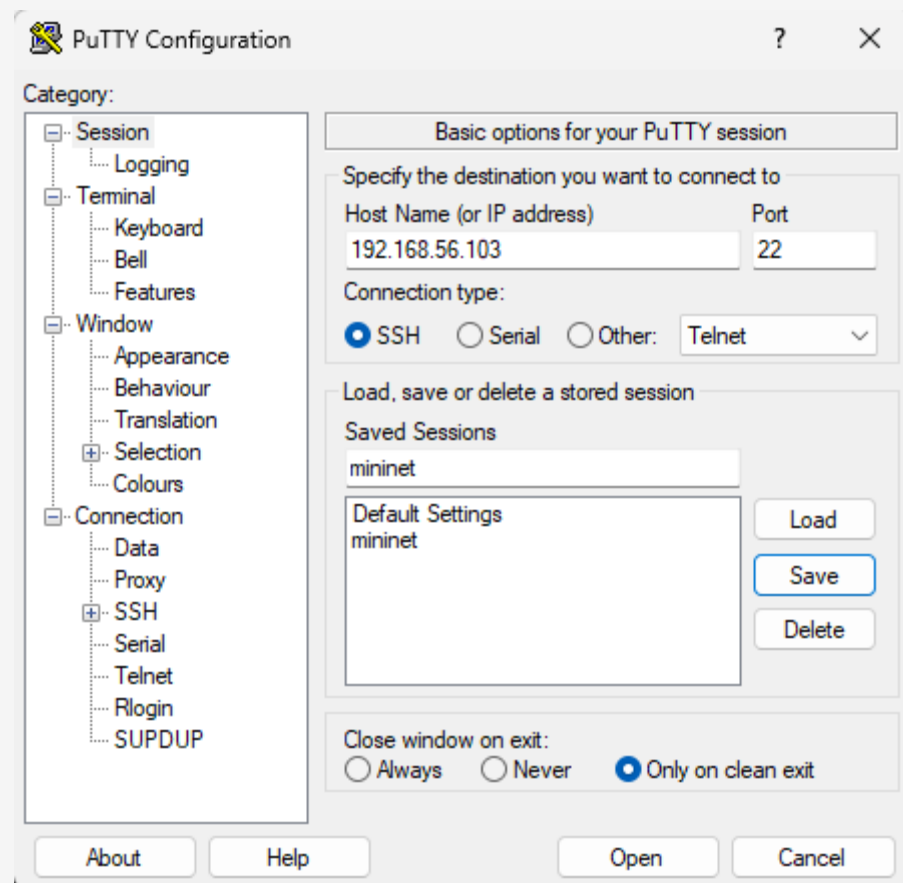
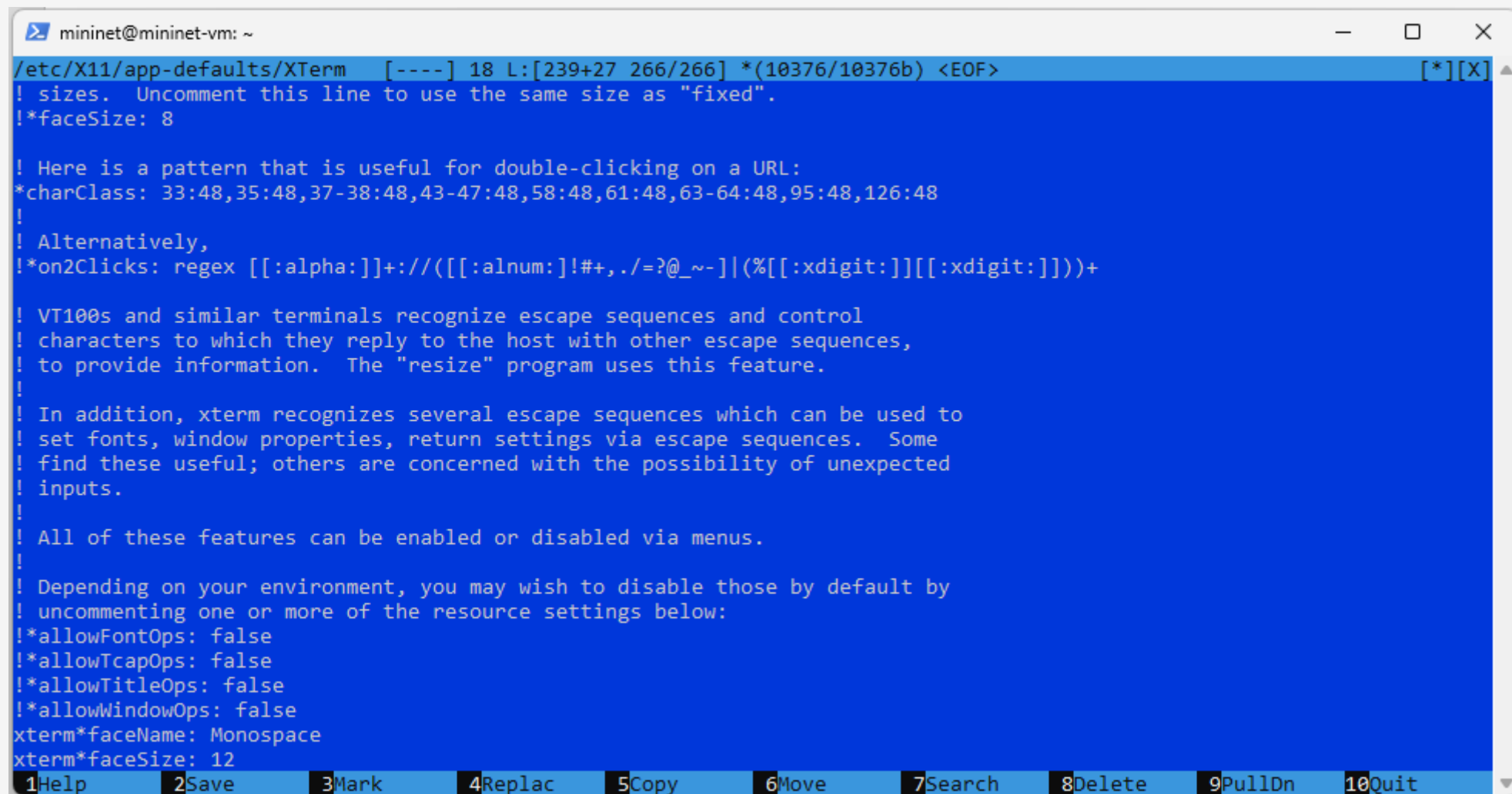


Рис. 7: Запуск putty и добавление опции перенаправления X11



## Настройка параметров XTerm

A screenshot of an XTerm window titled 'mininet@mininet-vm: ~'. The window has a blue background and white text. The text displays various configuration settings for XTerm, including font size, font name, and window properties. At the bottom of the window, there is a menu bar with 10 items: 1Help, 2Save, 3Mark, 4Replac, 5Copy, 6Move, 7Search, 8Delete, 9PullDn, and 10Quit. The text in the window is as follows:

```
/etc/X11/app-defaults/XTerm [----] 18 L:[239+27 266/266] *(10376/10376b) <EOF> [*][X]
! sizes. Uncomment this line to use the same size as "fixed".
!*faceSize: 8

! Here is a pattern that is useful for double-clicking on a URL:
*charClass: 33:48,35:48,37-38:48,43-47:48,58:48,61:48,63-64:48,95:48,126:48
!
! Alternatively,
!*on2Clicks: regex [[:alpha:]]+://([[:alnum:]]!#+,./=?@_~|)(%[[:xdigit:]][:xdigit:]]))+

! VT100s and similar terminals recognize escape sequences and control
! characters to which they reply to the host with other escape sequences,
! to provide information. The "resize" program uses this feature.
!
! In addition, xterm recognizes several escape sequences which can be used to
! set fonts, window properties, return settings via escape sequences. Some
! find these useful; others are concerned with the possibility of unexpected
! inputs.
!
! All of these features can be enabled or disabled via menus.
!
! Depending on your environment, you may wish to disable those by default by
! uncommenting one or more of the resource settings below:
!*allowFontOps: false
!*allowTcapOps: false
!*allowTitleOps: false
!*allowWindowOps: false
xterm*faceName: Monospace
xterm*faceSize: 12
```

Рис. 8: Увеличение размера шрифта и применение векторного шрифта

## Настройка соединения X11 для суперпользователя

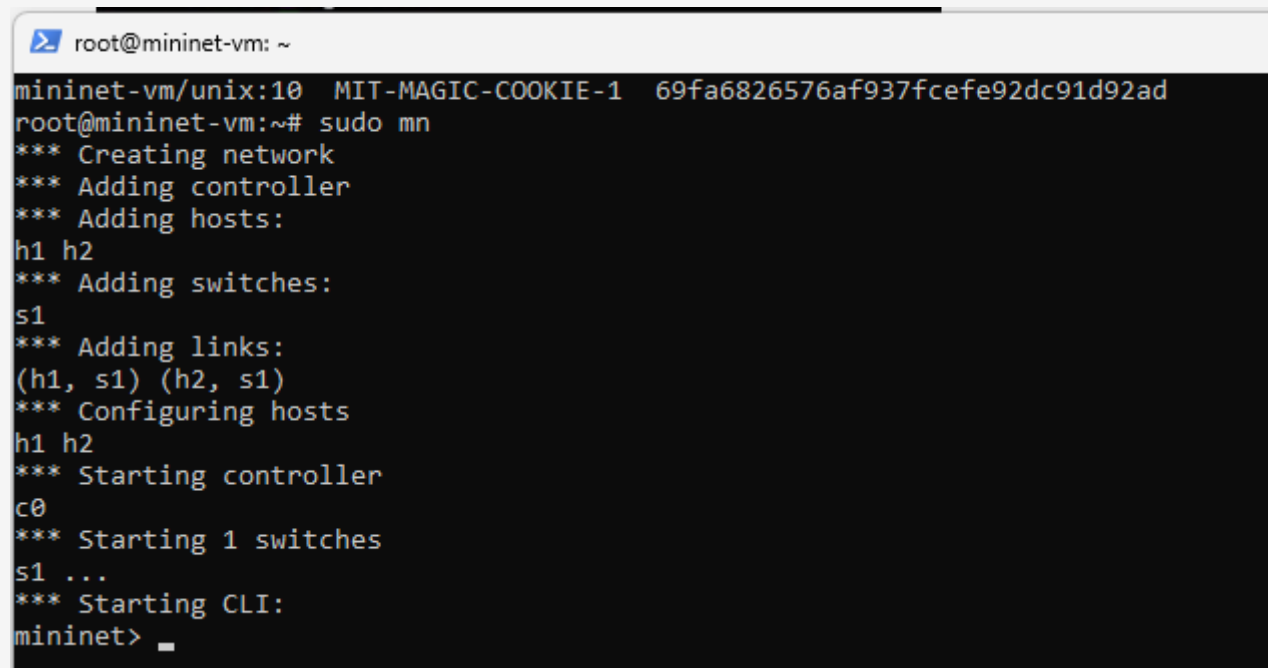
---

```
mininet@mininet-vm:~$ xauth list $DISPLAY
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  69fa6826576af937fcefe92dc91d92ad
mininet@mininet-vm:~$ sudo -i
root@mininet-vm:~# xauth list
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  69fa6826576af937fcefe92dc91d92ad
root@mininet-vm:~#
```

Рис. 9: Заполнения файла полномочий /root/.Xauthority

## Работа с Mininet с помощью командной строки

---

A terminal window titled 'root@mininet-vm: ~' showing the execution of the 'mn' command. The output displays the step-by-step creation of a Mininet network, including adding hosts (h1, h2), switches (s1), links, and starting the controller and CLI.

```
root@mininet-vm: ~
mininet-vm/unix:10 MIT-MAGIC-COOKIE-1 69fa6826576af937fcef92dc91d92ad
root@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. 10: Вызов Mininet с использованием топологии по умолчанию

## Работа с Mininet с помощью командной строки

---

```
mininet> help

Documented commands (type help <topic>):
=====
EOF      gterm  iperfudp  nodes      pingpair    py      switch  xterm
dpctl    help   link      noecho     pingpairfull  quit    time
dump     intfs  links     pingall    ports        sh      wait
exit     iperf  net       pingallfull  px          source  x

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2

mininet> _
```

Рис. 11: Отображение списка команд и примеров их использования

## Работа с Mininet с помощью командной строки

---

```
mininet> nodes  
available nodes are:  
c0 h1 h2 s1  
mininet>
```

Рис. 12: Отображение доступных узлов

## Работа с Mininet с помощью командной строки

---

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet>
```

Рис. 13: Просмотр доступных линков

## Работа с Mininet с помощью командной строки

---

```
mininet> h1 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 ca:29:e3:72:ab:3b 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 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

mininet>
```

Рис. 14: Выполнение команды для устройства h1

## Работа с Mininet с помощью командной строки

```
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=0.329 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.128 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.121 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.061 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.082 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.093 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.403 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.098 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.229 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.305 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.128 ms
^C
--- 10.0.0.2 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14319ms
rtt min/avg/max/mdev = 0.061/0.155/0.403/0.104 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 185.246 seconds
root@mininet-vm:~#
```

Рис. 15: Проверка связи между узлами h1 и h2



## Работа с Mininet с помощью командной строки

---

```
root@mininet-vm:~# sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller ovs-testcontroller udpbw
test mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller ovs-testcontroller ud
pbwtest mnexec ivs ryu-manager 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-_[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
root@mininet-vm:~#
```

Рис. 16: Очистка предыдущего экземпляра Mininet

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

---

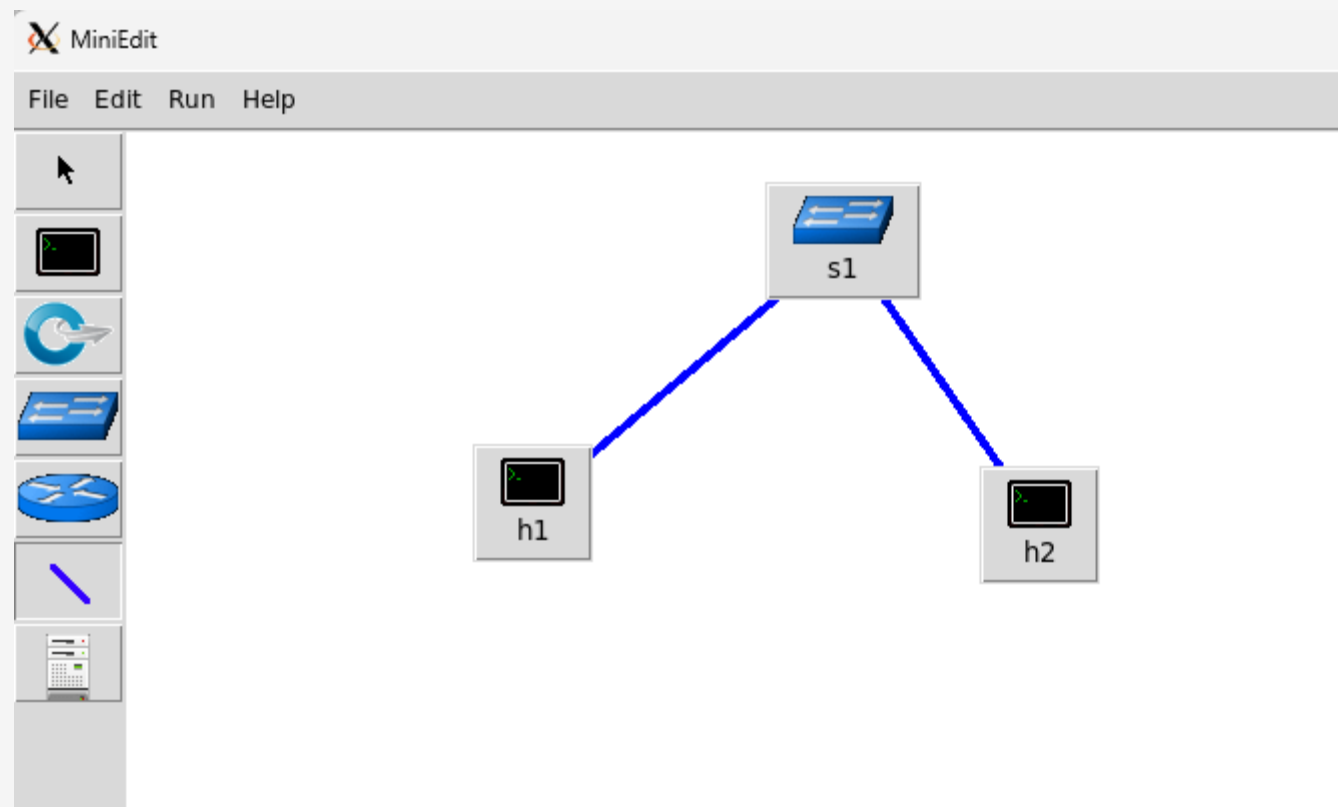


Рис. 17: Добавление двух хостов и одного коммутатора

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

---

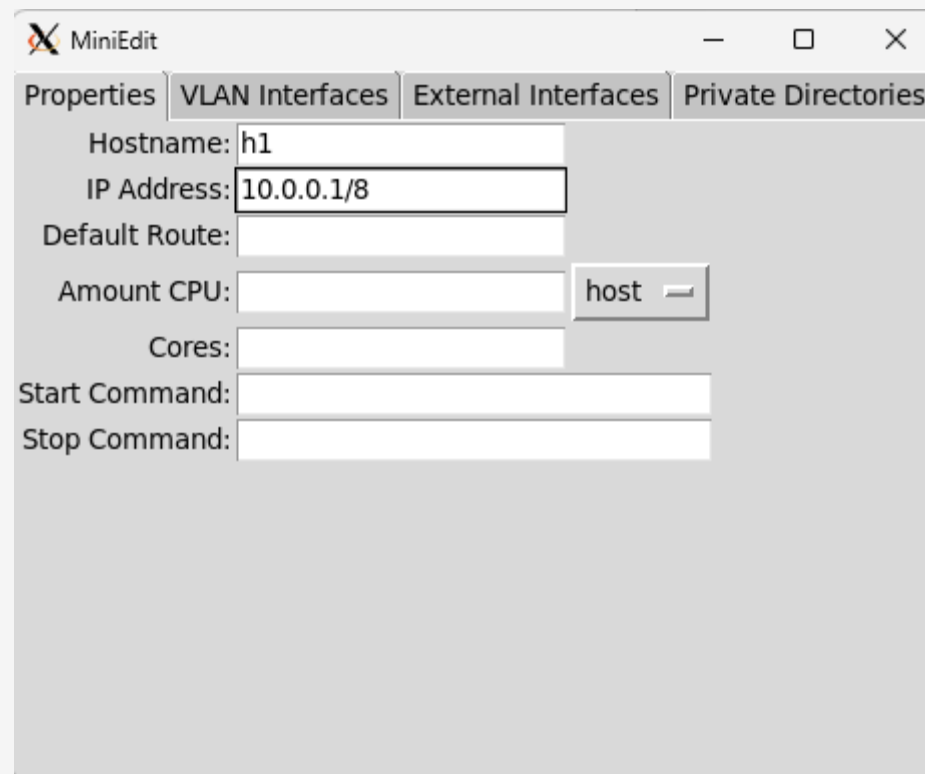


Рис. 18: Настройка IP-адреса на хосте h1

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

---

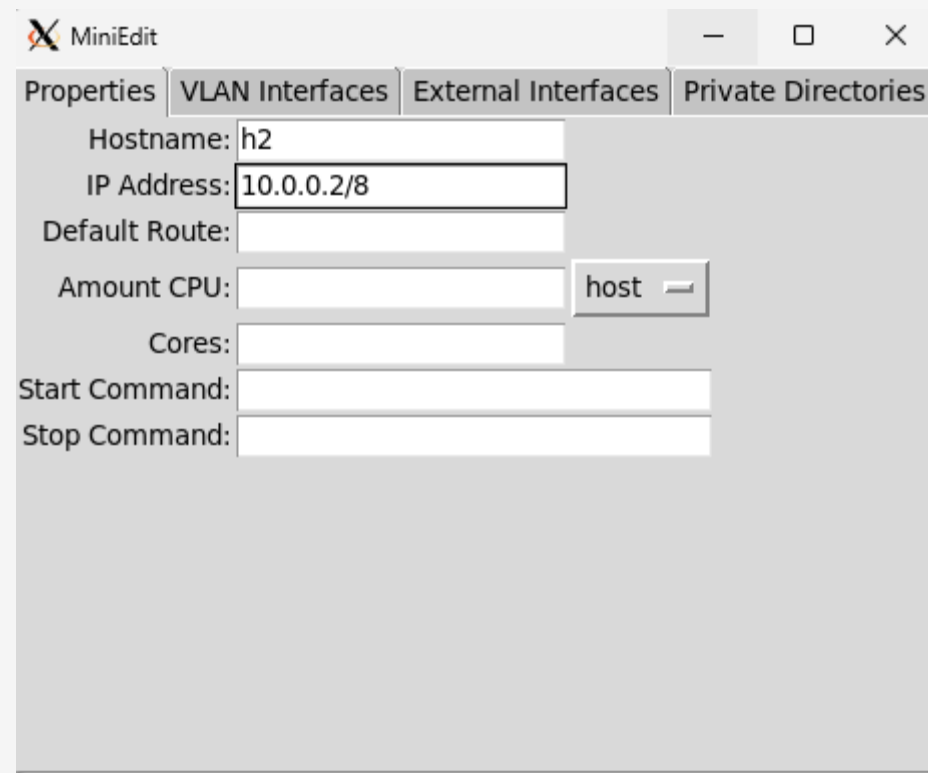
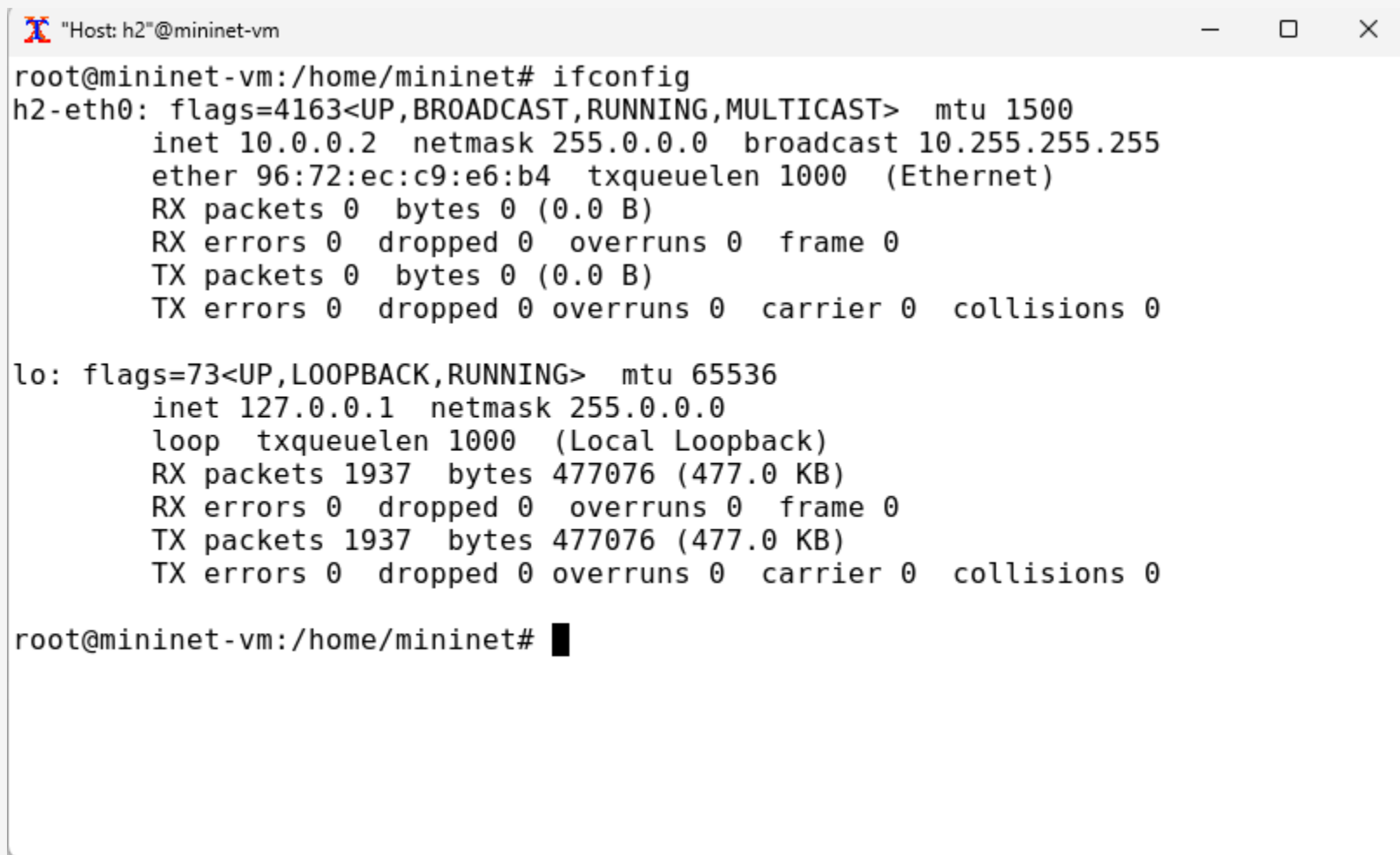


Рис. 19: Настройка IP-адреса на хосте h2

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



```
root@mininet-vm:/home/mininet# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.0.2  netmask 255.0.0.0  broadcast 10.255.255.255
    ether 96:72:ec:c9:e6:b4  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 1937  bytes 477076 (477.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1937  bytes 477076 (477.0 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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

Рис. 20: Проверка назначенных IP-адресов для h2 и проверка соединения между хостами

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

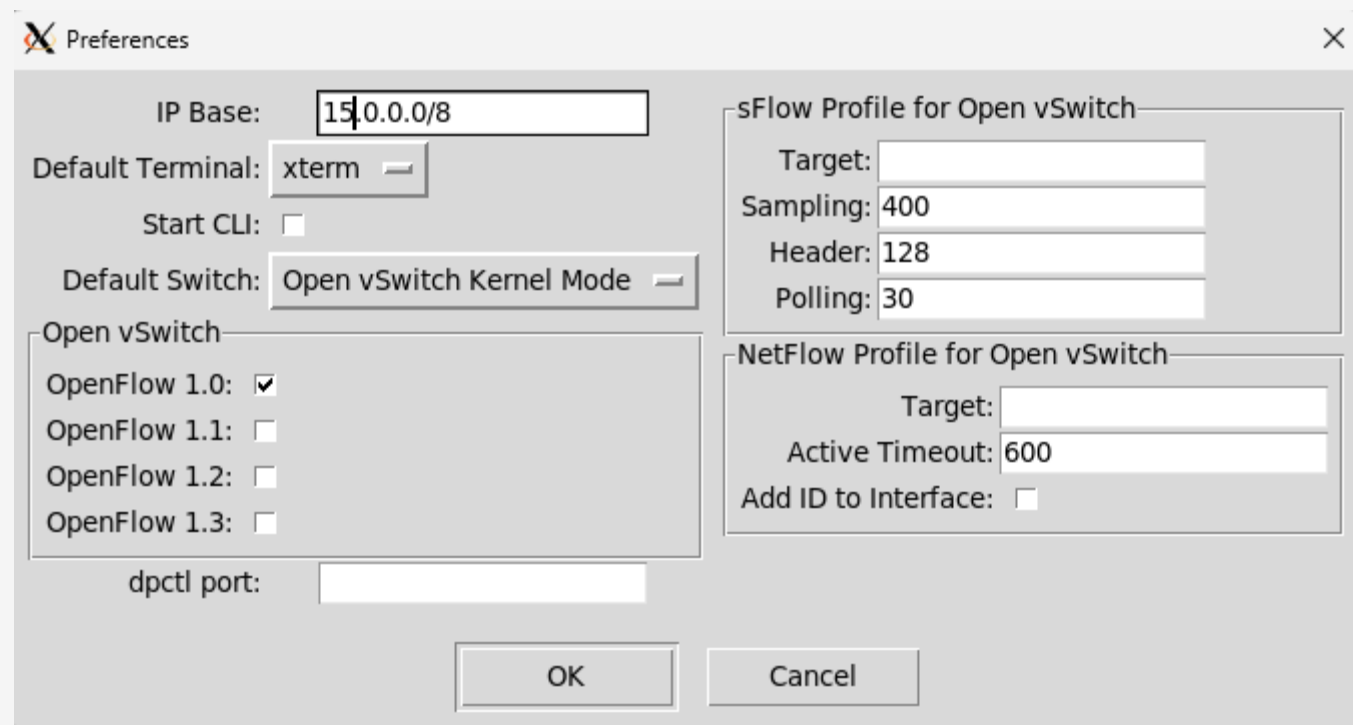
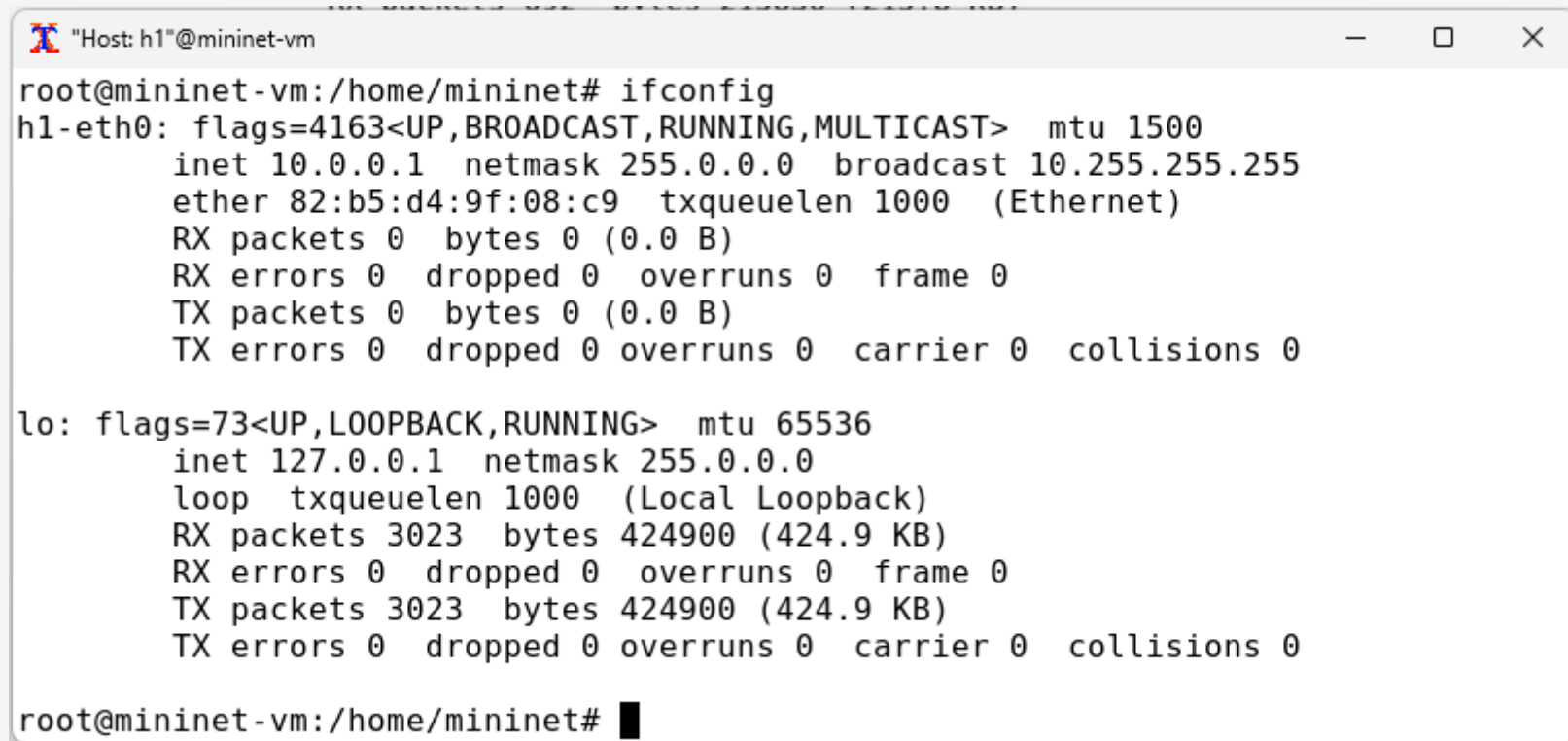


Рис. 21: Проверка автоматического назначения адресов

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

---



```
"Host: h1"@mininet-vm
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 82:b5:d4:9f:08:c9  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 3023  bytes 424900 (424.9 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 3023  bytes 424900 (424.9 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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

Рис. 22: Отображение IP-адреса, назначенного хосту h1

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

---

```
mininet@mininet-vm:~$ mkdir ~/work
mkdir: cannot create directory '/home/mininet/work': File exists
mininet@mininet-vm:~$ ls
mininet  mininet.orig  oflops  oftest  openflow  pox  work
mininet@mininet-vm:~$
```

Рис. 23: Создание нового каталога



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

---

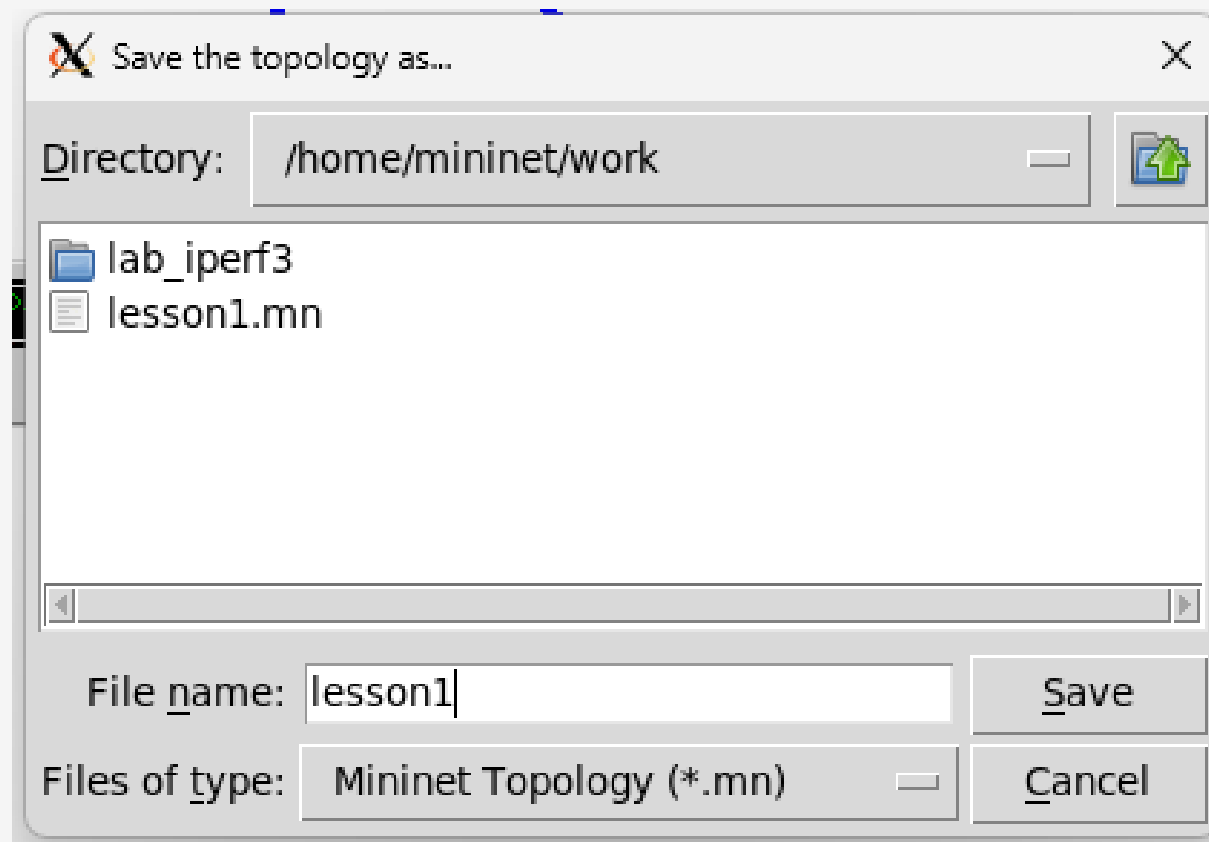


Рис. 24: Сохранение топологии

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

---

```
mininet@mininet-vm:~/work$ ls  
lab_iperf3  lesson1.mn  
mininet@mininet-vm:~/work$
```

Рис. 25: Изменение прав доступа к файлам в каталоге проекта

## Вывод

---

- В ходе выполнения лабораторной работы были получены навыки по развёртыванию в системе виртуализации (например, в VirtualBox) mininet, а также познакомились с основными командами для работы с Mininet через командную строку и через графический интерфейс.

## Список литературы. Библиография

---

[1] Mininet: <https://mininet.org/>