Лабораторная работа №1
Моделирование сетей передачи данных

Исаев Б. А.

2025

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

Настройка образа VirtualBox

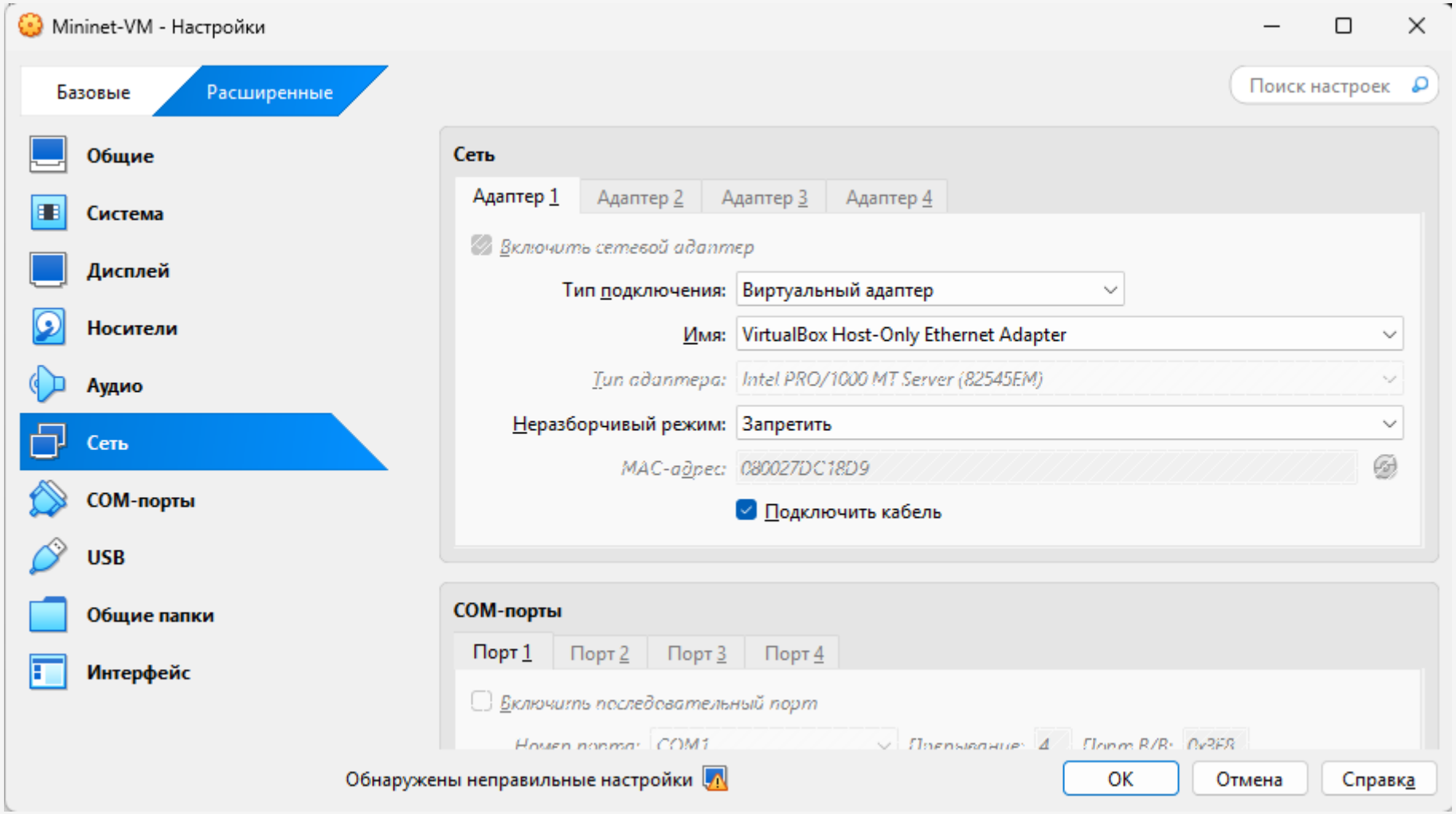

Рис. 1: Установка и настройка виртуальной машины

Подключение к виртуальной машине

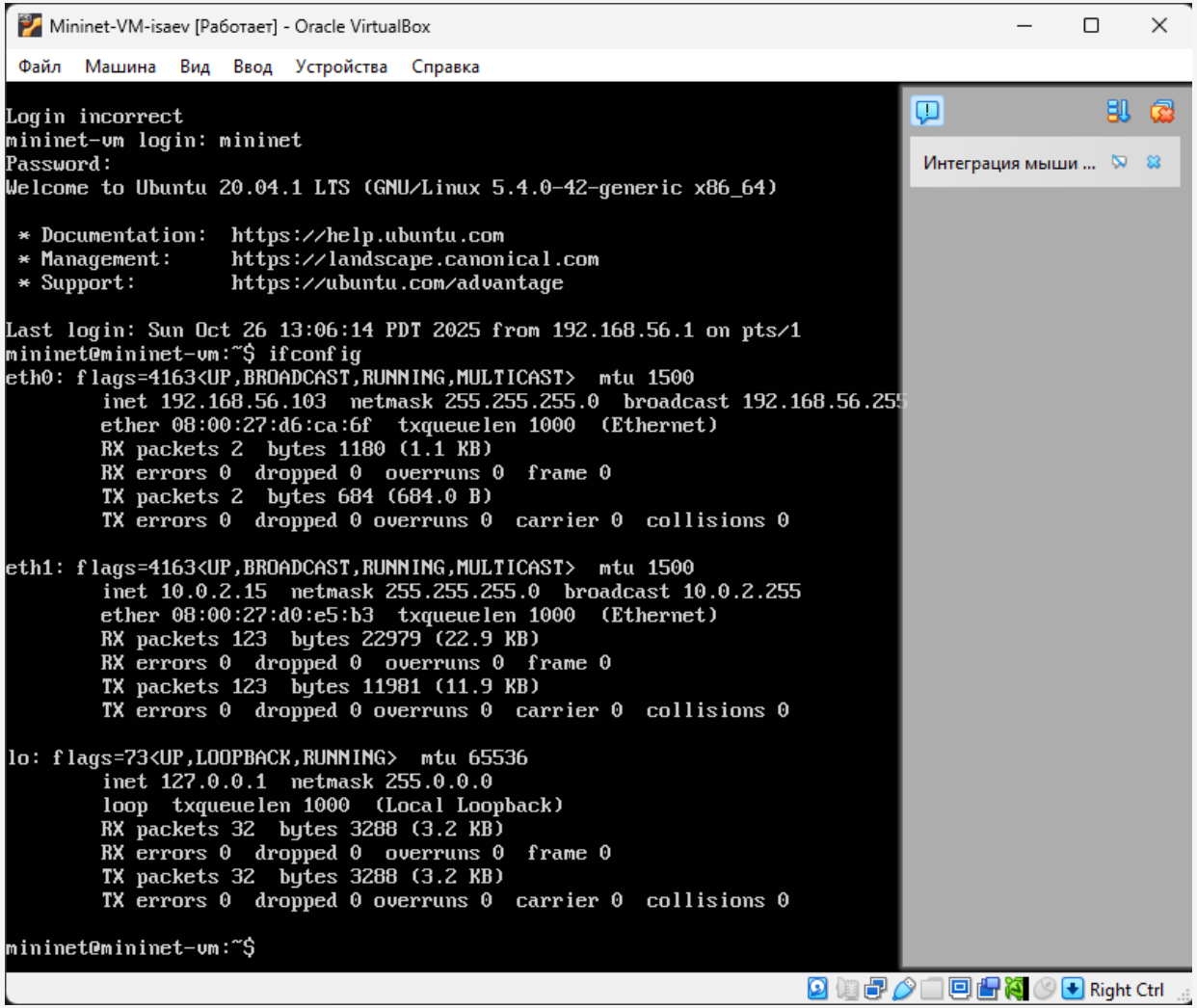

Рис. 2: Вход и просмотр адреса виртуальной машины

Подключение к виртуальной машине

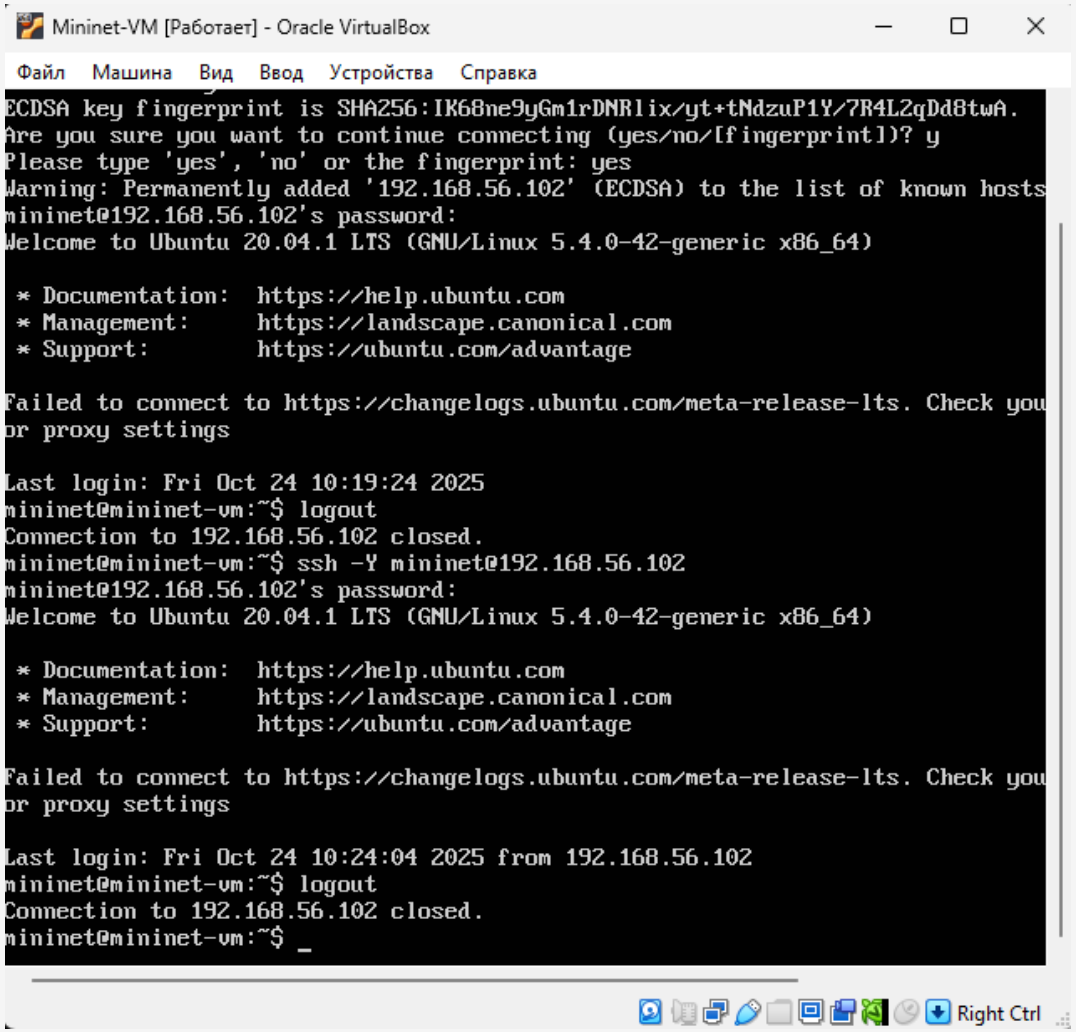

Рис. 3: Подключение к виртуальной машине из терминала хостовой машины

Работа с Mininet из-под Windows

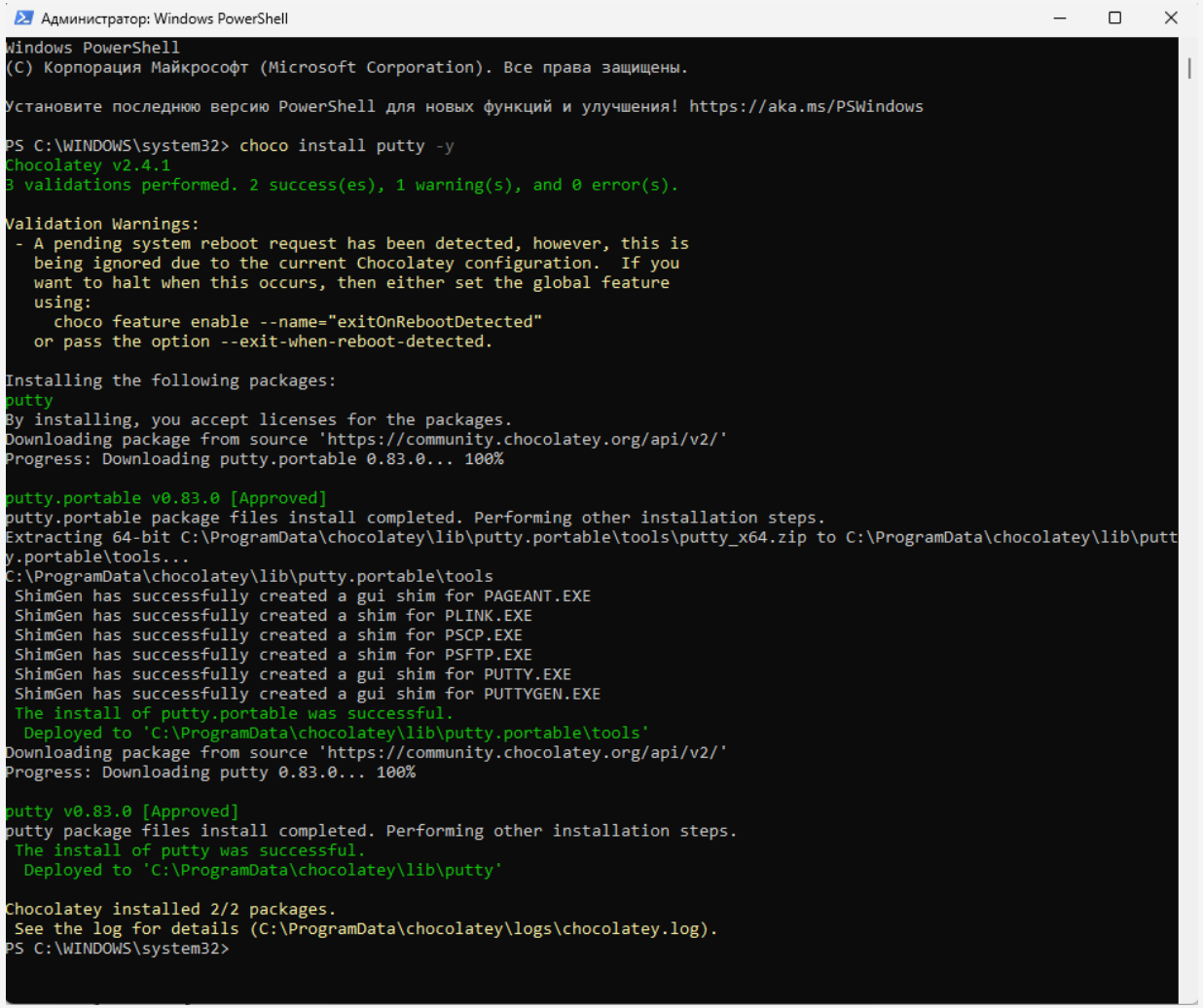

Рис. 4: Установка putty

Работа с Mininet из-под Windows

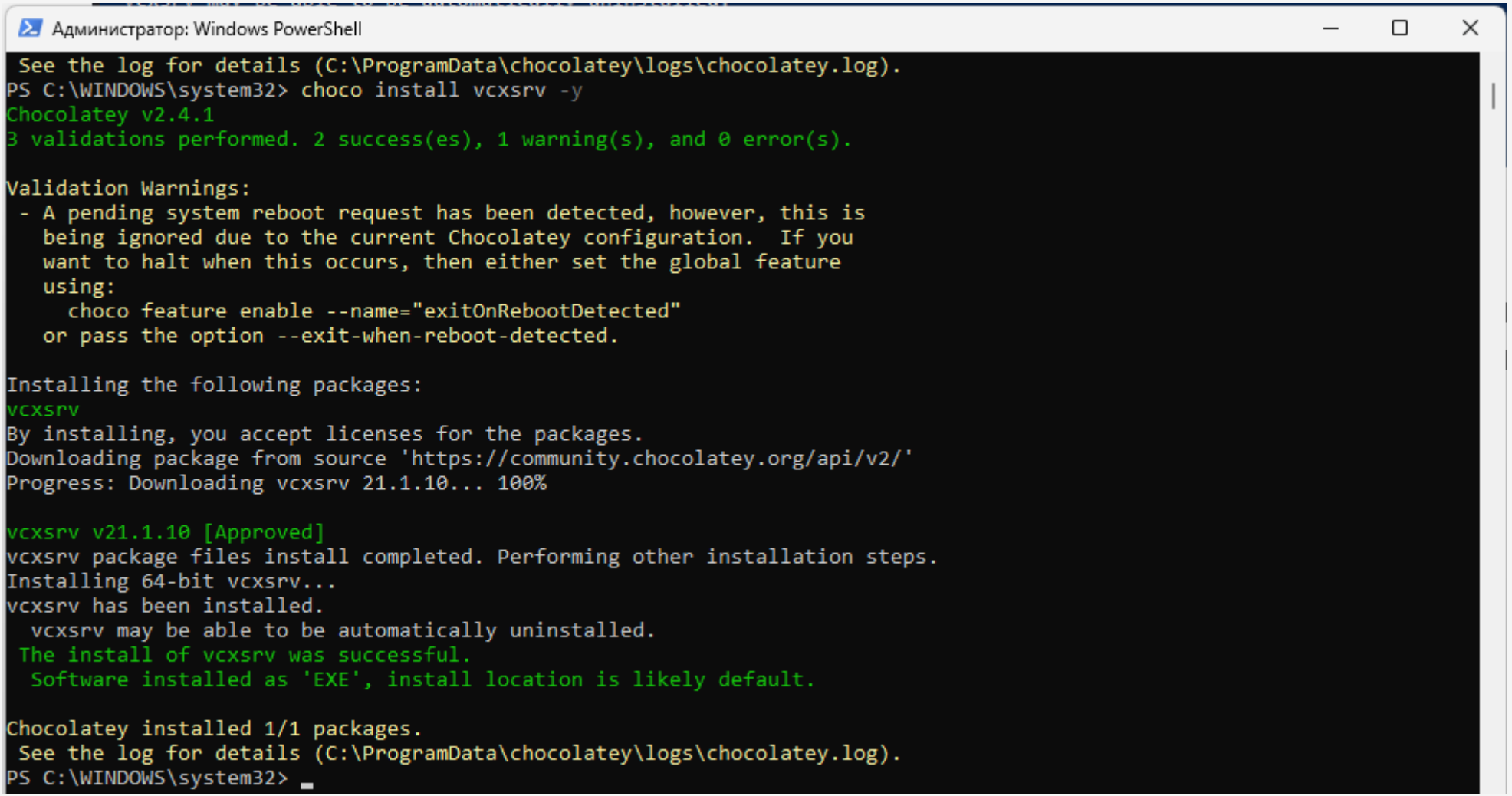

Рис. 5: Установка putty VcXsrv Windows X Server

Работа с Mininet из-под Windows

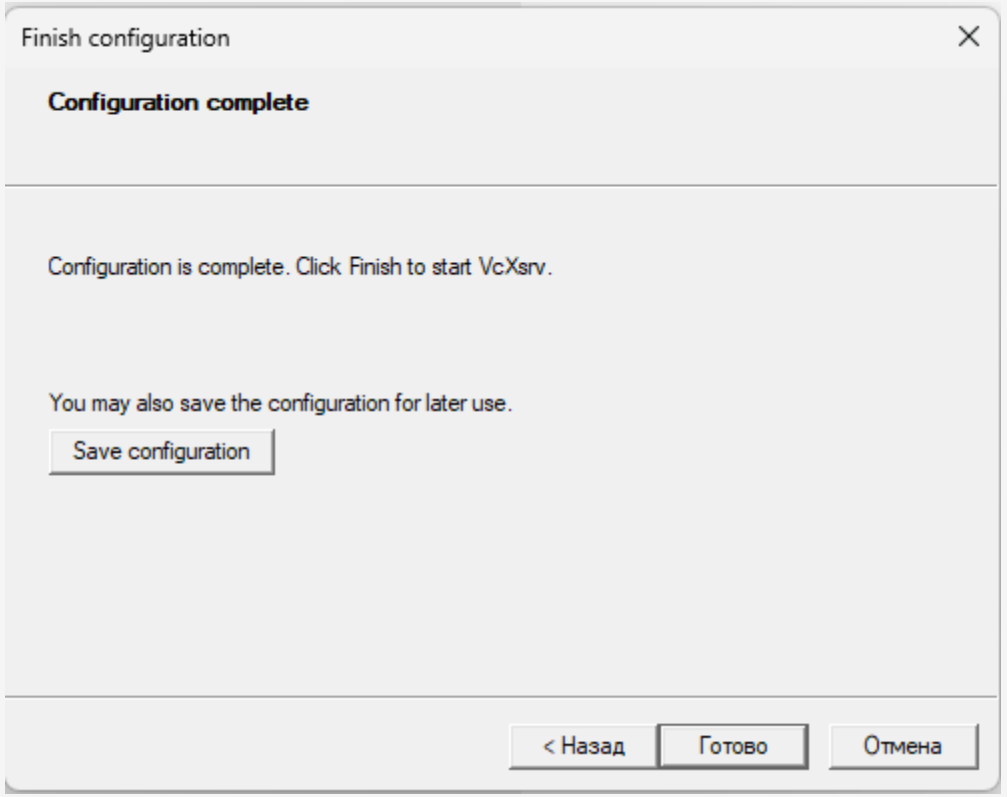

Рис. 6: Запуск и настройка Xserver

Работа с Mininet из-под Windows

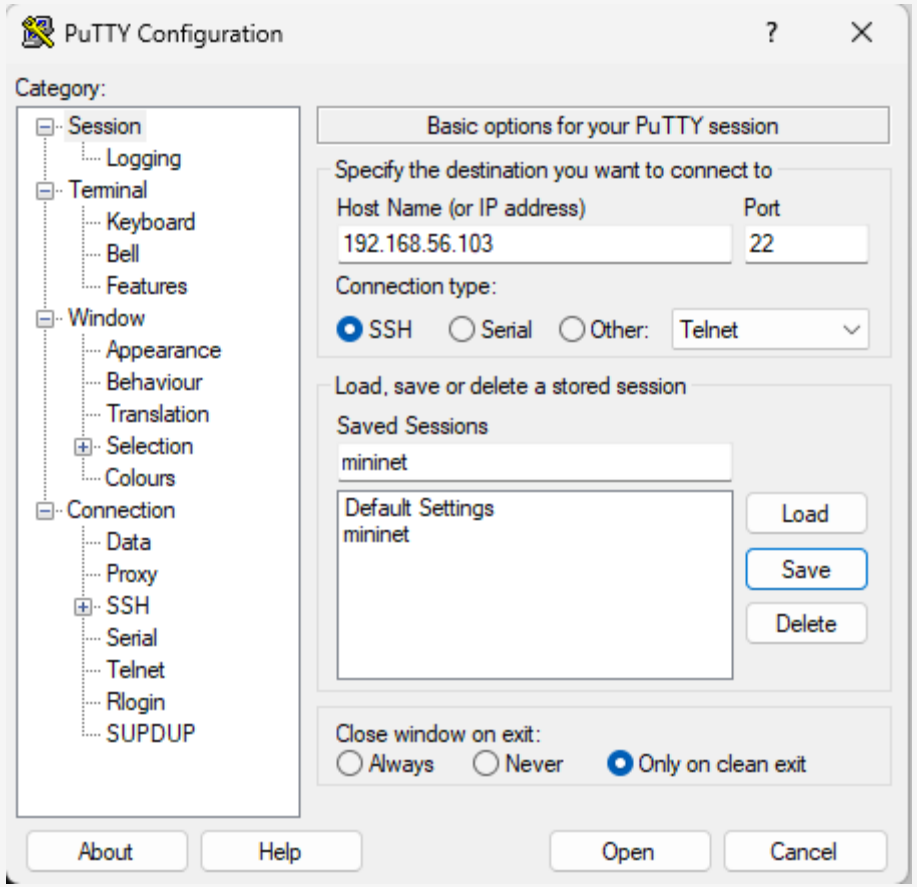

Рис. 7: Запуск putty и добавление опции перенаправления X11

Настройка параметров XTerm

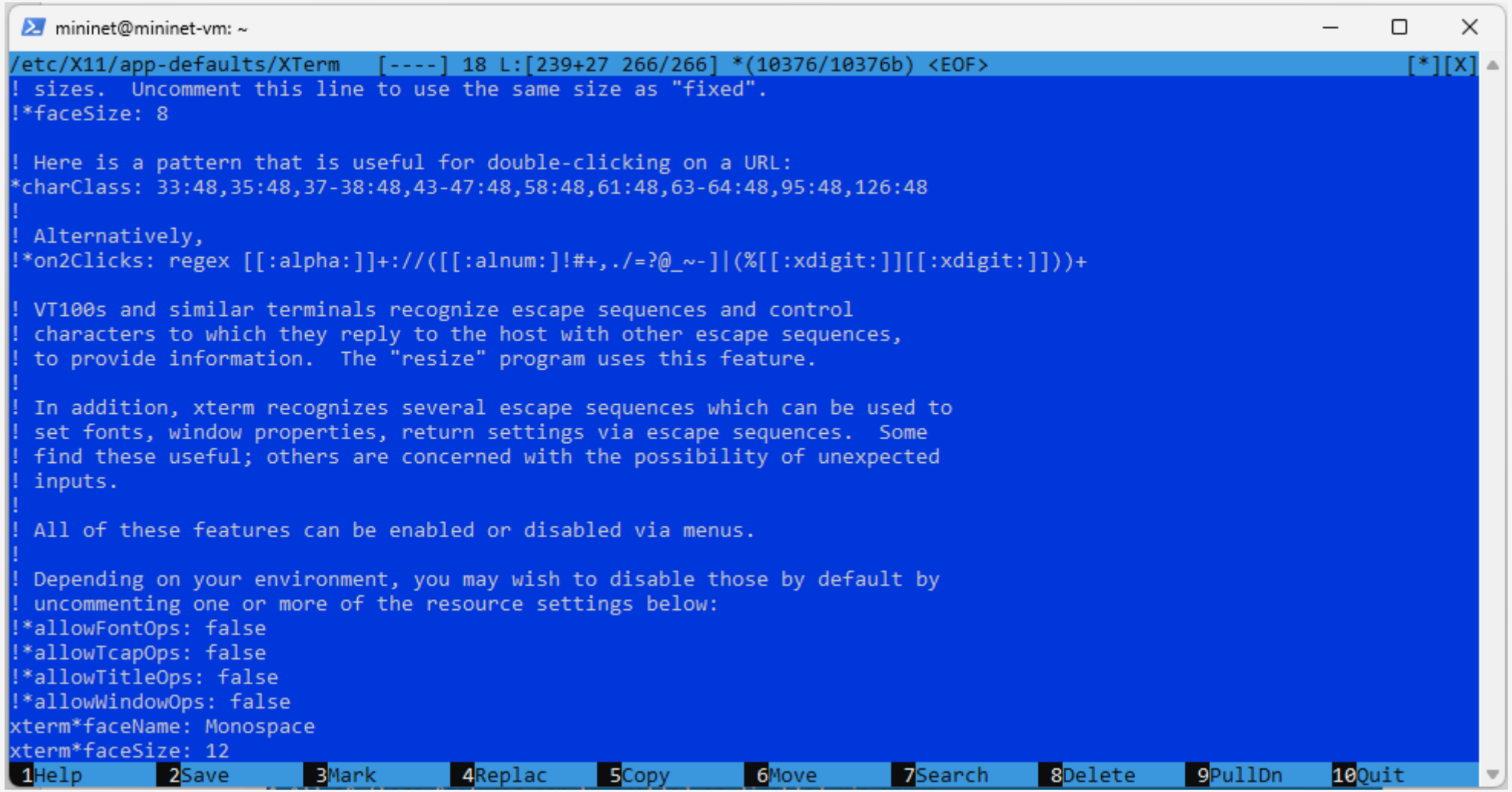

Рис. 8: Увеличение размера шрифта и применение векторного шрифта

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

```
root@mininet-vm: ~
mininet-vm/unix:10  MIT-MAGIC-COOKIE-1  69fa6826576af937fcefe92dc91d92ad
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

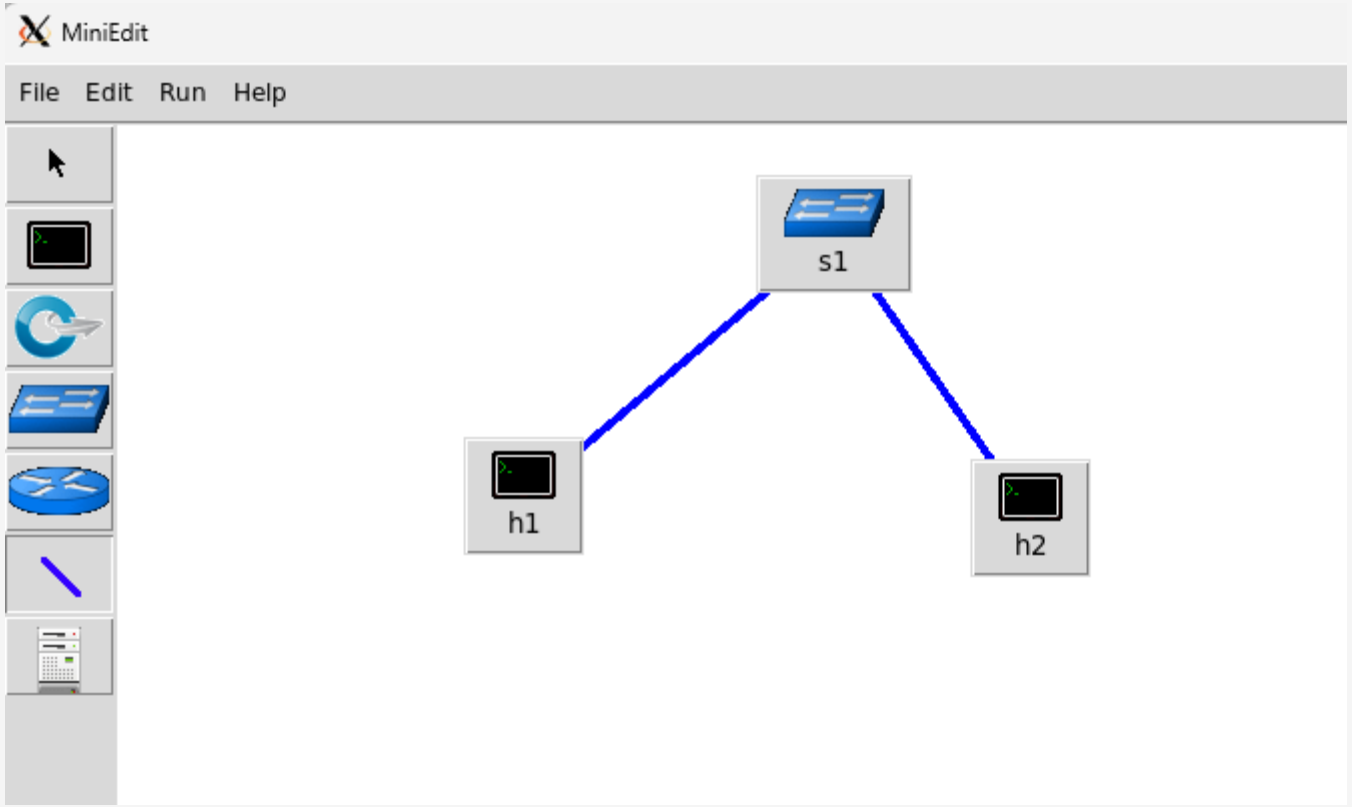

Рис. 17: Добавление двух хостов и одного коммутатора

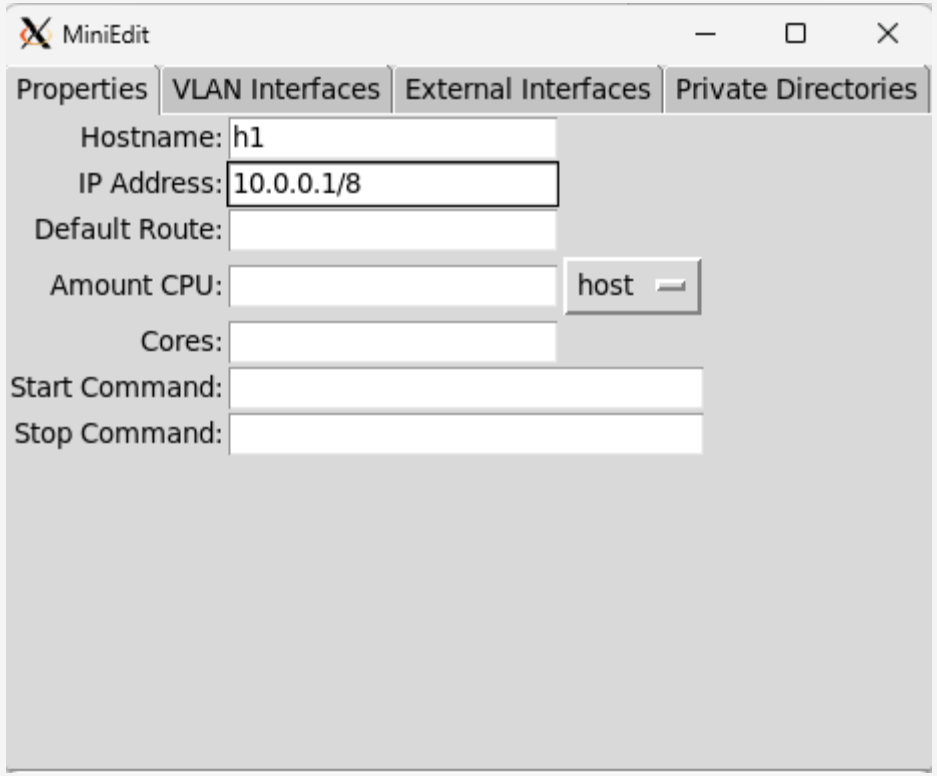

Рис. 18: Настройка IP-адреса на хосте h1

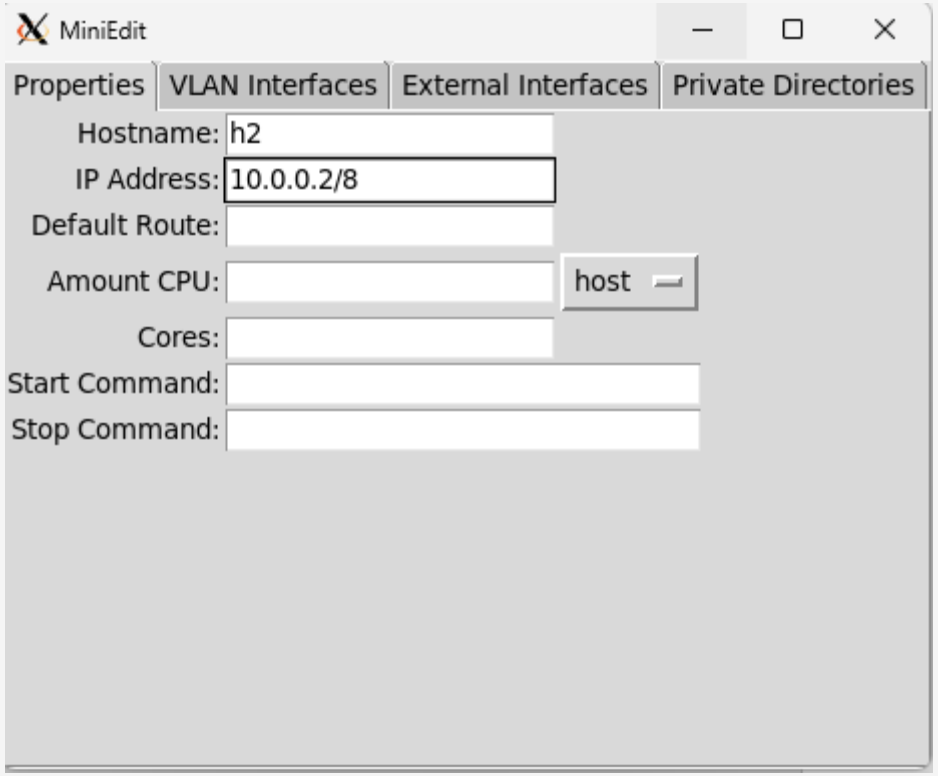

Рис. 19: Настройка IP-адреса на хосте h2

Построение и эмуляция сети в Mininet с использованием графического интерфейса

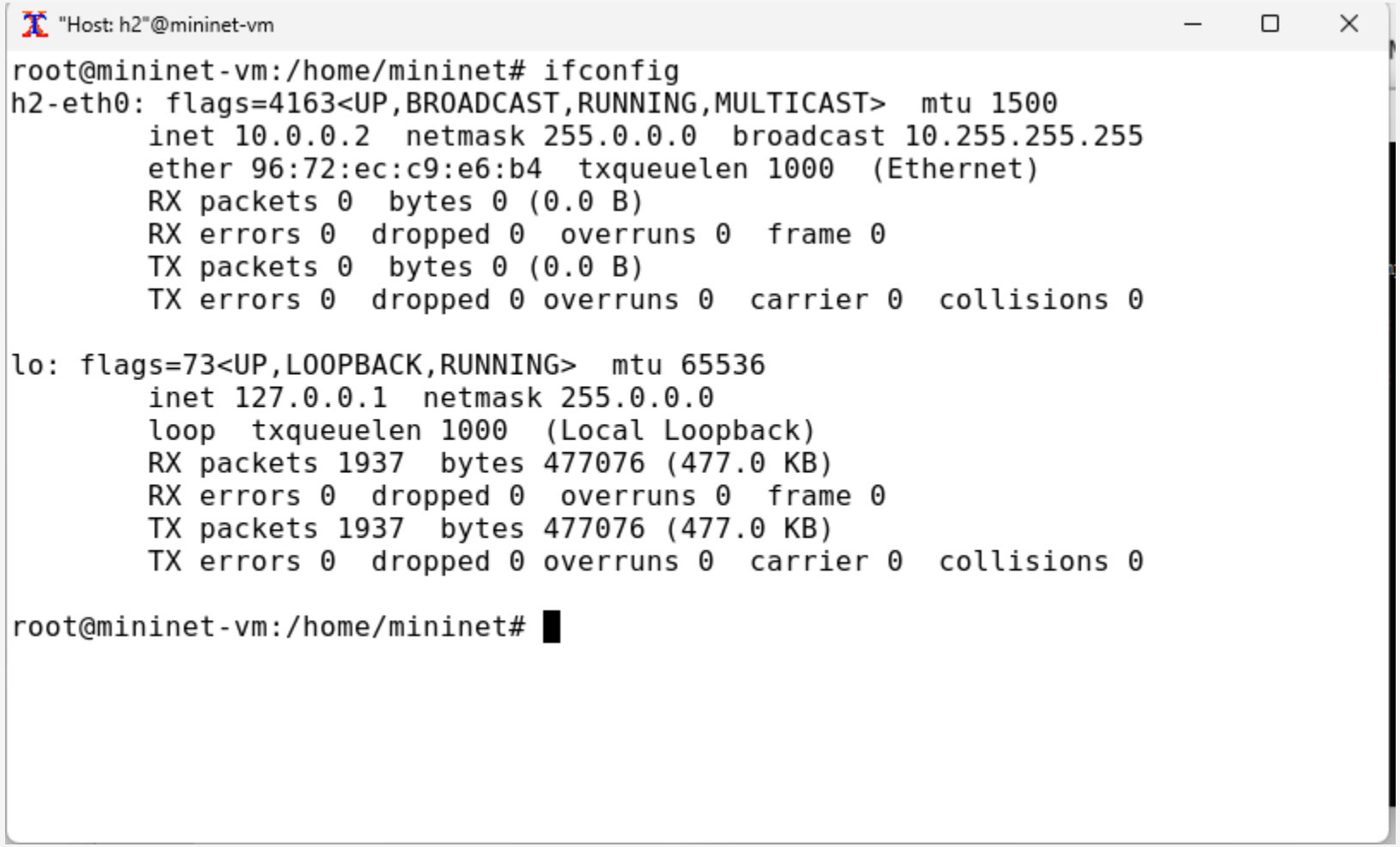

Рис. 20: Проверка назначенных IP-адресов для h2 и проверка соединения между хостами

Построение и эмуляция сети в Mininet с использованием графического интерфейса

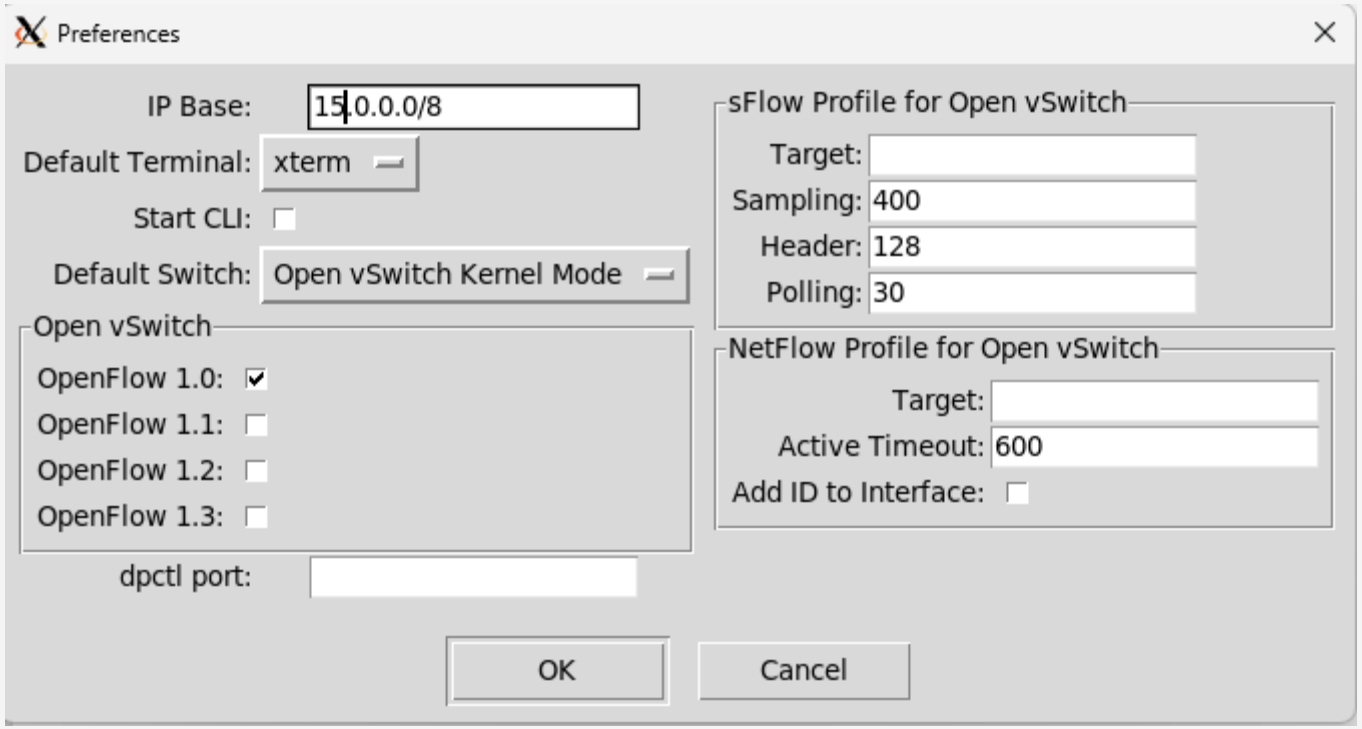

Рис. 21: Проверка автоматического назначения адресов

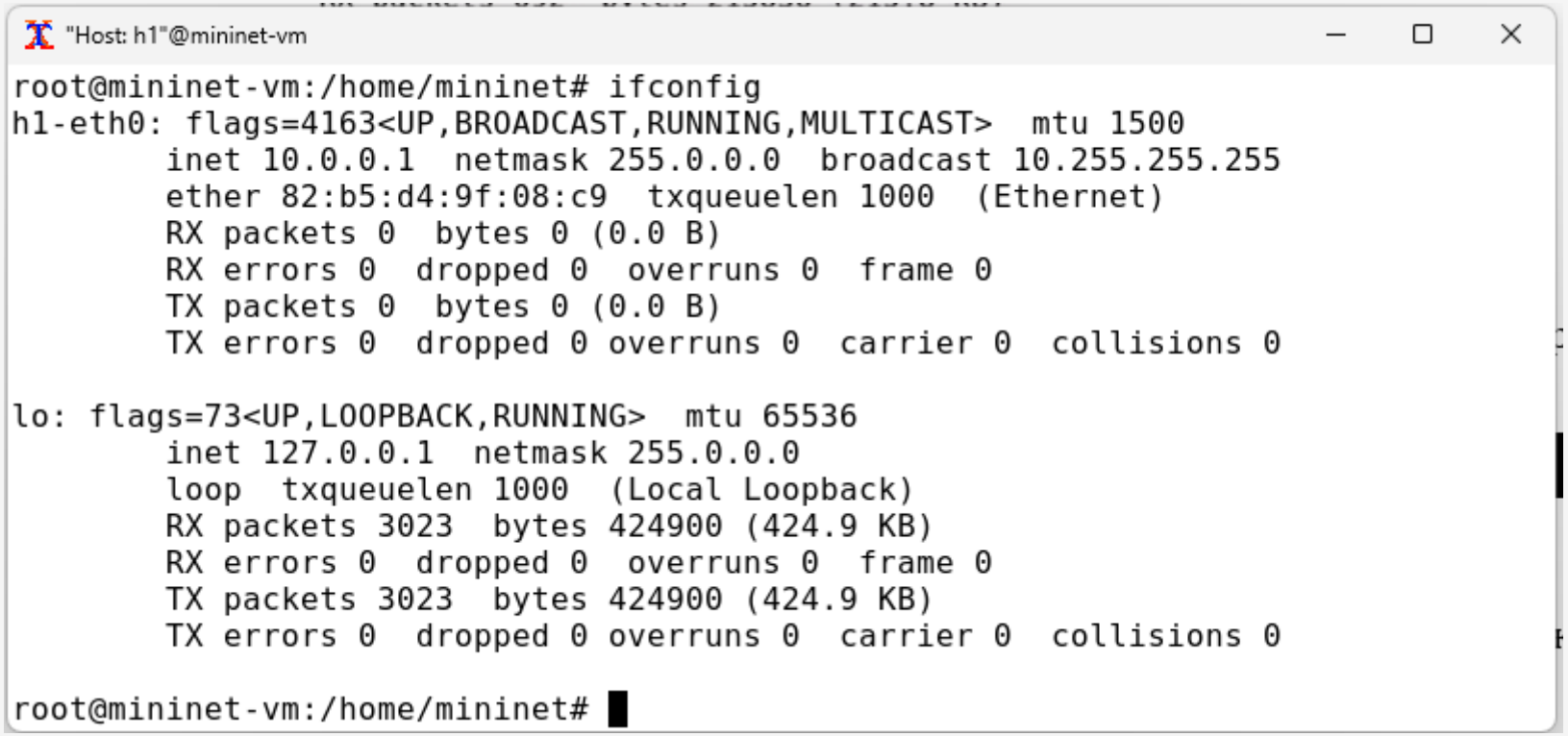

Рис. 22: Отображение IP-адреса, назначенного хосту h1

```
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

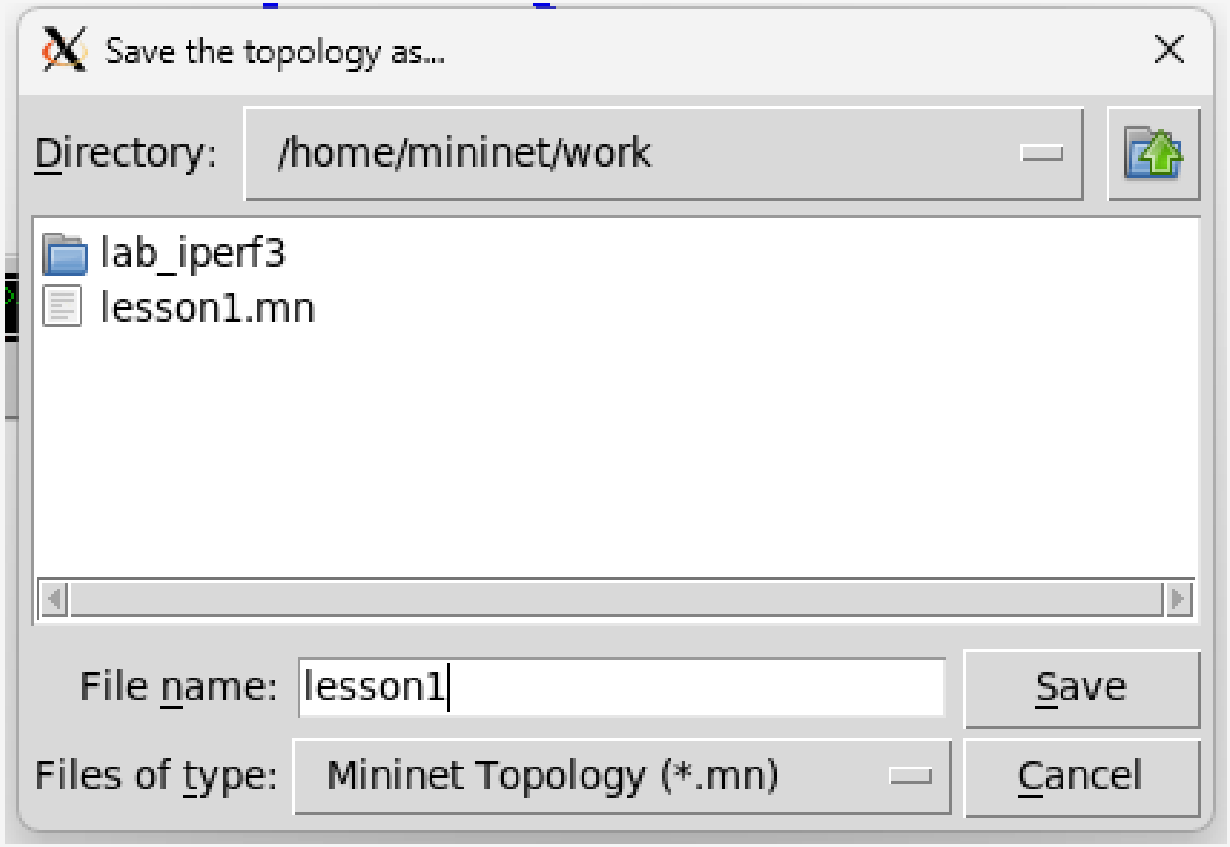

Рис. 24: Сохранение топологии

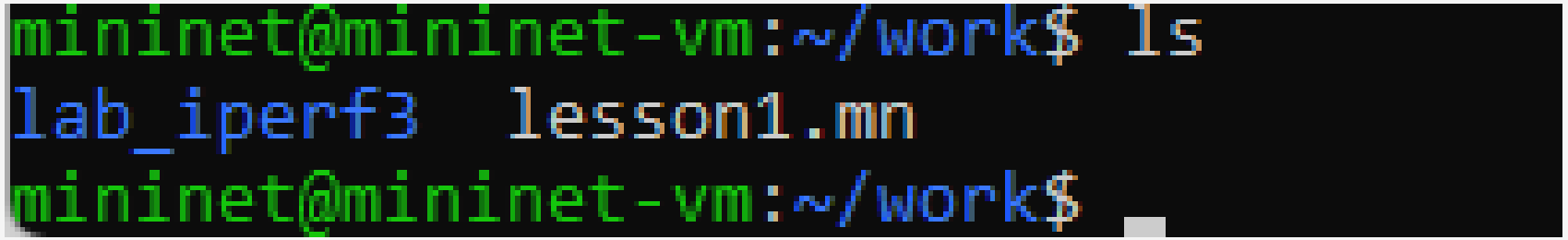

Рис. 25: Изменение прав доступа к файлам в каталоге проекта

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/