

# Лабораторная работа №1 Введение в Mininet

Леснухин Даниил Дмитриевич Российский университет дружбы  
народов Москва

# Цель работы

Основной целью работы является:

- Развёртывание Mininet в среде виртуализации
- Изучение основных команд Mininet
- Построение простейшей топологии
- Проверка связности узлов

# Задание

- Развернуть среду Mininet
- Изучить основы работы с эмулятором
- Построить простейшую топологию
- Проверить связность узлов

Mininet — программная среда для эмуляции компьютерных сетей.

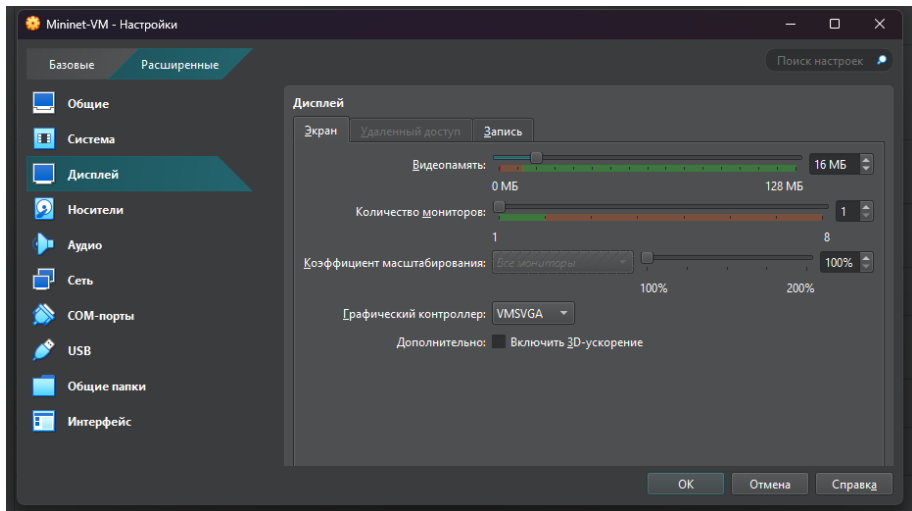
Позволяет:

- Создавать виртуальные хосты и коммутаторы
- Тестировать сетевые протоколы
- Исследовать SDN
- Анализировать задержки и пропускную способность

Использует реальный сетевой стек Linux.

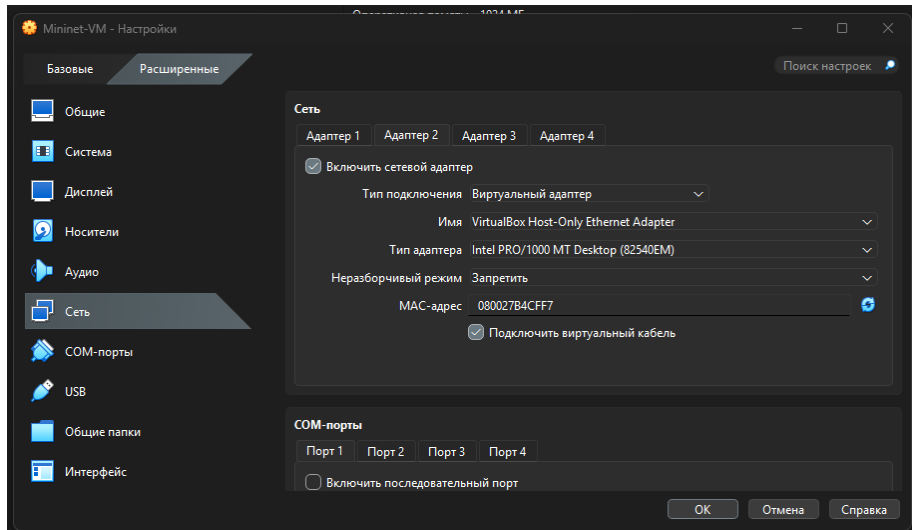
# Установка виртуальной машины

Устанавливаем образ Mininet в VirtualBox.



# Настройка сетевого адаптера

Настраиваем второй адаптер как виртуальный.



# Подключение к виртуальной машине

Вход под:

- login: mininet
- password: mininet

Команда для просмотра IP:

ifconfig

# Активация второго интерфейса

Для доступа в интернет активируем второй интерфейс.

```
mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.3 netmask 255.255.255.0 broadcast 192.168.56.255
    ether 08:00:27:27:42:c6 txqueuelen 1000 (Ethernet)
    RX packets 135 bytes 17744 (17.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 103 bytes 16704 (16.7 KB)
    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 08:00:27:2a:b6:e7 txqueuelen 1000 (Ethernet)
    RX packets 6 bytes 2137 (2.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7 bytes 1360 (1.3 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 284 bytes 21824 (21.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 284 bytes 21824 (21.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

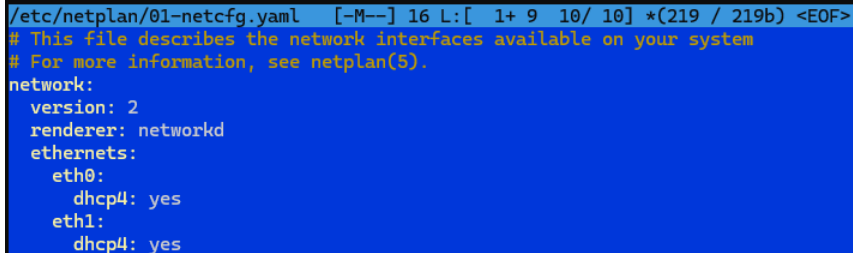
mininet@mininet-vm:~$
```



# Изменение netplan

Редактируем файл:

/etc/netplan/01-netcfg.yaml



```
/etc/netplan/01-netcfg.yaml [-M--] 16 L:[ 1+ 9 10/ 10] *(219 / 219b) <EOF>
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernet:
    eth0:
      dhcp4: yes
    eth1:
      dhcp4: yes
```

# Обновление Mininet

Команды обновления:

```
cd ~
```

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

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

```
cd ~/mininet
```

```
sudo make install
```

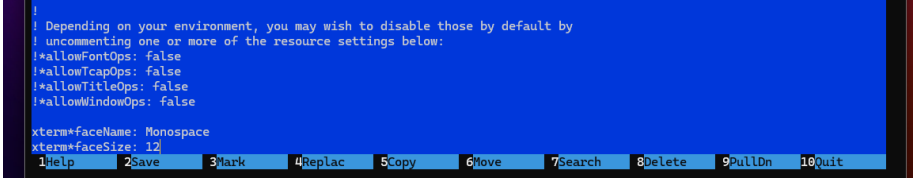
```
mininet@mininet-vm:~$ mv ~/mininet ~/mininet.orig
mininet@mininet-vm:~$ cd ~
mininet@mininet-vm:~$ git clone https://github.com/mininet/mininet.git
Cloning into 'mininet'...
remote: Enumerating objects: 10388, done.
remote: Counting objects: 100% (136/136), done.
remote: Compressing objects: 100% (64/64), done.
remote: Total 10388 (delta 109), reused 72 (delta 72), pack-reused 10252 (from 2)
Receiving objects: 100% (10388/10388), 3.36 MiB | 8.45 MiB/s, done.
Resolving deltas: 100% (6909/6909), 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\" mnexec.c -o mnexec
install -D mnexec /usr/bin/mnexec
PYTHONPATH=. help2man -N -n "create a Mininet network." \
--no-discard-stderr "python -B bin/mn" -o mn.1
help2man -N -n "execution utility for Mininet." \
-h "-h" -v "-v" --no-discard-stderr ./mnexec -o mnexec.1
install -D -t /usr/share/man/man1 mn.1 mnexec.1
python -m pip uninstall -y mininet || true
Found existing installation: mininet 2.3.0
Uninstalling mininet-2.3.0:
```

# Настройка XTerm

Редактируем: `/etc/X11/app-defaults/XTerm`

Добавляем: `xterm_faceName: Monospace`

`xterm_faceSize: 12`

A screenshot of a text editor window with a blue background and white text. The text shows the configuration of XTerm. It starts with a comment block explaining that users can disable certain features by uncommenting settings. Then, it shows the configuration changes: `xterm*faceName: Monospace` and `xterm*faceSize: 12`. At the bottom, there is a menu bar with 10 items: 1Help, 2Save, 3Mark, 4Replac, 5Copy, 6Move, 7Search, 8Delete, 9PullDn, and 10Quit. The 'Quit' item is highlighted.

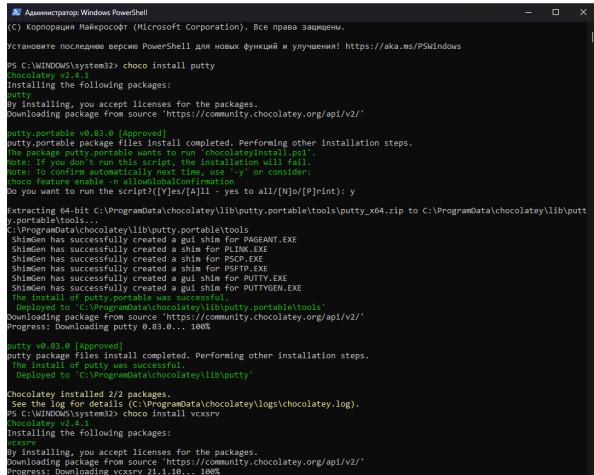
```
!  
! Depending on your environment, you may wish to disable those by default by  
! uncommenting one or more of the resource settings below:  
!*allowFontOps: false  
!*allowTcapOps: false  
!*allowTitleOps: false  
!*allowWindowOps: false  
  
xterm*faceName: Monospace  
xterm*faceSize: 12  
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

# Работа из-под Windows

## Устанавливаем:

choco install putty

choco install vcxsrv



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

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

PS C:\WINDOWS\system32> choco install putty
chocolatey v2.4.1
Installing the following packages:
putty
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'

putty.portable v0.83.0 [Approved]
putty.portable package files install completed. Performing other installation steps.
The package putty.portable wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y

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> choco install vcxsrv
chocolatey v2.4.1
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%
```

# Запуск Mininet

Запуск минимальной топологии: `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> 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
```

# Проверка связности

По умолчанию:

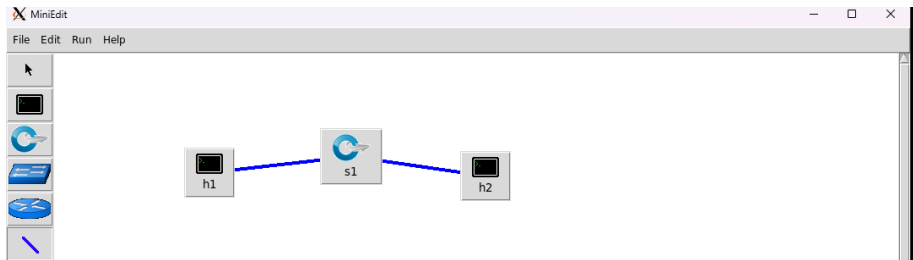
- h1 → 10.0.0.1
- h2 → 10.0.0.2

Проверка: ping 10.0.0.2

```
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=1.62 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.181 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.044 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.059 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.045 ms
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7153ms
rtt min/avg/max/mdev = 0.042/0.260/1.618/0.515 ms
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 159.878 seconds
mininet@mininet-vm:~$ |
```

# Построение топологии

## Создание собственной топологии.



# Проверка связи узлов

## Проверяем связь между узлами.

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

```
MiniEdit
File Edit Run Help
root@mininet-vm:/home/mininet# ifconfig
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 ba:4d:b4:1f:14:f4 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
```

Host: h1@mininet-vm

```
root@mininet-vm:/home/mininet# ifconfig
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 32:62:5c:cc:d6:e8 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 1488 bytes 279172 (279.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1488 bytes 279172 (279.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

root@mininet-vm:/home/mininet#

Conf

Host: h2@mininet-vm

```
root@mininet-vm:/home/mininet# ifconfig
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 ba:4d:b4:1f:14:f4 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 1122 bytes 250636 (250.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1122 bytes 250636 (250.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

root@mininet-vm:/home/mininet#



# Автоматическое назначение IP

Изменяем IP Base на:

15.0.0.0/8

Preferences

IP Base: 15.0.0.0/8

Default Terminal: xterm

Start CLI: ☐

Default Switch: Open vSwitch Kernel Mode

Open vSwitch

OpenFlow 1.0: ☒

OpenFlow 1.1: ☐

OpenFlow 1.2: ☐

OpenFlow 1.3: ☐

dpctl port:

sFlow Profile for Open vSwitch

Target:

Sampling: 400

Header: 128

Polling: 30

NetFlow Profile for Open vSwitch

Target:

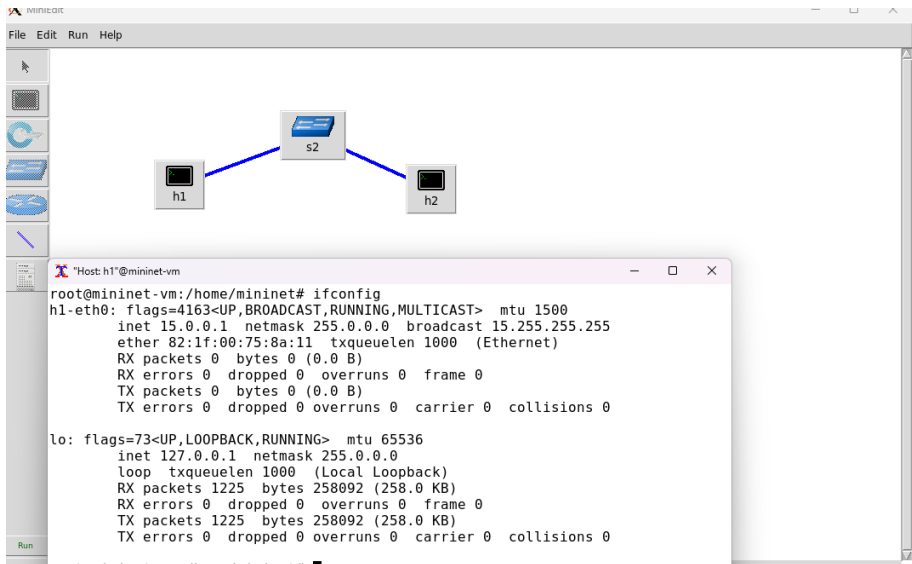
Active Timeout: 600

Add ID to Interface: ☐

OK Cancel

# Проверка новых адресов

## Запуск топологии с новыми IP.



The screenshot displays the Mininet VM interface. The main window shows a network topology with three nodes: a central switch labeled 's2' connected to two hosts, 'h1' and 'h2'. The nodes are represented by icons: a switch for 's2' and laptops for 'h1' and 'h2'. Blue lines indicate the network connections between 's2' and 'h1', and between 's2' and 'h2'.

Below the topology, a terminal window titled "Host: h1" is open, showing the output of the `ifconfig` command for host h1. The output shows the configuration for the `h1-eth0` interface and the loopback interface `lo`.

```
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 82:1f:00:75:8a:11 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 1225 bytes 258092 (258.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1225 bytes 258092 (258.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

В ходе работы:

- Развернута виртуальная среда Mininet
- Изучены базовые команды
- Построена топология сети
- Проверена связность узлов
- Освоен графический интерфейс MiniEdit

Получены практические навыки моделирования сетей передачи данных.