

Техника работы в виртуальной лаборатории

Удалённый доступ	1
Работа с Ubuntu	2
Установка и работа с операционной системы RouterOS	4
Настройка GNS3	9
Первая топология	11
GNS3автоматически генерирует команды для Qemu	15
Шаблон для топологий	16
Закладки	18
Работа с RouterOS из командной строки	19
Импорт и экспорт топологий	20
О персональных компьютерах и свичах в топологиях	22
Синхронизация проектов между лабораториями	23
Контрольные вопросы	24
Порядок выполнения работы	25
Для сдачи работы необходимо	25

Удалённый доступ

Коллективная работа в виртуальной лаборатории организовывается путём удалённого доступа пользователей к рабочему столу операционной системы Ubuntu из семейства Linux.

При работе на кафедре используйте преимущественно лабораторию по адресу 192.168.14.56. При работе из дома используйте лабораторию labs.mikrotik.com.ua. Последняя доступна с кафедральных компьютеров по адресу 192.168.3.253. Связь по адресу 192.168.3.253 завязана на сеть университета, которая не всегда стабильно работает.

Графический интерфейс операционных систем семейства Linux организован с помощью X-протокола. Приложения осуществляют графический вывод на X-сервер, выступая по отношению к нему как клиенты. Для доступа из Windows к удалённому рабочему столу Ubuntu можно использовать обычный X-сервер для Windows, например Xming. Опыт показал, что этот подход приемлем только для работы в локальной сети. Может быть организована удалённая работа и по протоколу VNC, но при этом требуется множество дополнительных настроек.

Мы используем наиболее продвинутую технологию фирмы NoMachine, основанную на X-протоколе и протоколе безопасной консоли SSH (Secure Shell). Для организации удалённого доступа к рабочему столу Ubuntu должен быть поднят NX-сервер. Мы используем бесплатную реализацию freeNX-сервер.

Для удалённого доступа пользователей Windows к рабочему столу Ubuntu они используют бесплатный NX-Client for Windows фирмы NoMachine.

Для организации соединений NX-Client с NX-сервером используется мастер подключения, в котором вводится адрес удалённого Ubuntu, устанавливается менеджер окон GNOME и создаётся ярлык. Ярлык используется для старта сеанса доступа к удалённому рабочему столу.

Аутентификация пользователя осуществляется с помощью ввода имени и пароля, которые можно разрешить системе запомнить. Последовательно в появившемся окошке видим сообщения:

Setup the environment -установка окружения

Connecting to ... - соединение с Ubuntu

Connected to ... - соединились с Ubuntu

Waiting authentication - ждём проверку подлинности

Authentication complete - проверку прошли

Или

Authentication failed for user ... -это если вы неправильно ввели имя или пароль.

Если вы в ходе предыдущей сессии некорректно завершили работу или не вышли из Ubuntu , то появится окно, извещающее о незакрытых сессиях (Рис. 1)

Если в этом окне недоступна кнопка Resume, то нажимайте Terminate и затем New.

Вы увидите сообщения

Download session information - загрузка информации о сессии

Negotiation the link parameters - соглашение о параметрах связи

Esteblish a display connection - установка соединения к дисплею

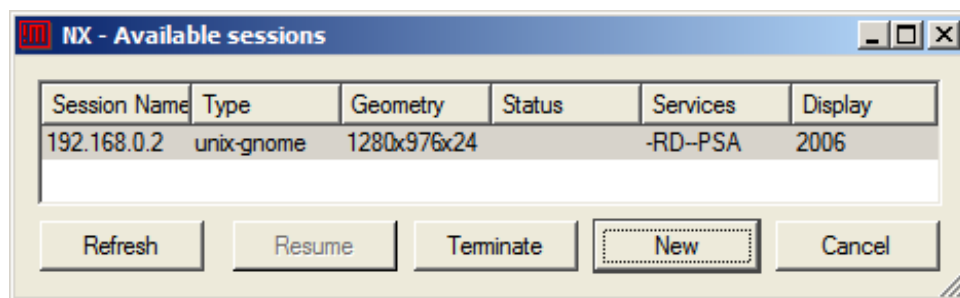


Рис. 1. Незакрытые сессии

Появится окно во весь экран с большим изображением символа Nomachine !m. Возможно придётся ждать пару минут и вы в Ubuntu.

Для завершения своей удалённой работы в Ubuntu не закрывайте в Windows окно NX-client for windows. Закройте все свои программы и выйдите из Ubuntu (System-Logout).

Всегда держите открытыми окна пингов в сторону адресов лабораторий и вашего провайдера Интернет. Если велика задержка (более 300 мс), то работа с nxclient станет некомфортной. Причиной задержки может быть то, что вы что-то качиваете или выкачиваете.

Работа с Ubuntu

Всё необходимое программное обеспечение Ubuntu берёт из Интернета и инсталляции программных продуктов в виде файлов или дисков (как в Windows) практически не применяются.

Опыт подсказывает, что графический интерфейс версии 10.0.43 больше

подходит для организации сетевой лаборатории, чем интерфейс более новых версий.

Мы часто используем терминал (консоль). Добавим иконку запуска терминала в верхнюю панель (applications-accessories-terminal-правая кнопка мыши -add to panel). Не запускайте много экземпляров. Используйте закладки в терминале (file-open tab) для открытия в одном окне множества консолей. Внутри консоли работает поддержка мыши для выделения текста и копирования/вставки из/в буфер обмена.

В консоли можно запустить файловый менеджер, набрав `mc` и далее Enter.

Для редактирования текстовых файлов пользуйтесь редактором **nano** **ИмяФайла** или встроенным редактором файлового менеджера `mc`, который вызывается нажатием клавиши F4.

В менеджера файлов `mc` для поиска используйте клавиатурную последовательность F9-c-f. Для скрывания-показа панелей – комбинацию клавиш `ctrlO` (удерживая Ctrl, нажимаем o).

Нажмите на верхней панели правую кнопку мыши, выберете `add to panel` и добавьте программы System monitor и Windows selector

Загружать свою сетевую топологию в виртуальной лаборатории может любой студент. В период загрузки сетевой топологии маршрутизаторы сильно грузят процессоры. Стартовать свою топологию нет смысла, пока загружены процессоры. Поэтому за правило надо взять наблюдение за окном системного монитора. С его помощью можно узнать, когда загрузка маршрутизаторов у вас или других студентов завершилась. За загрузкой системы можно наблюдать и с помощью консольной команды `top`. Утилита `top` покажет, кто занимает компьютер.

При работе вы можете открыть массу окон, чтобы в них не запутаться используйте закреплённый на панели Windows selector.

Сделаем краткий обзор команд linux. В отличие от Windows Linux различает маленькие и большие буквы. Смена папок осуществляется командой `cd` ИмяПапки. Родительская папка имеет имя «`..`», текущая - «`.`», а корневая – «`/`». Просмотр содержимого папки выполняется командой `ls` или, если подробно `ls -l`. Последняя команда выводит права доступа к файлам. Если вы имеете право доступа на выполнения файла `abc` в текущей папке `def`, то запустить его на выполнение можно, введя `./abc`. Перейдя в родительскую папку (`cd ..`), нам для выполнения `abc` надо будет ввести `def/abc`.

В Linux есть переменная окружения `PATH`. Посмотреть её значение можно командой `echo $PATH`. Это значение содержит перечень папок. Все исполняемые файлы из папок, указанных в `PATH` запускаются на выполнение без указания полного пути к ним. Так, если бы файл `abc` лежал бы в такой папке, то для его выполнения следовало бы просто ввести `abc`. Текущая папка при этом значения не имеет.

Копирование (перемещение) файлов и папок осуществляется командами **cp (mv) источник приёмник**

Используются шаблоны имён типа `abc*cde*`. Для работы с конкретным

файлом или папкой можно не вводить всего его имени, а попытаться подобрать уникальный шаблон. Например, если в папке файл abc, является единственным файлом, имя которого начинается с буквы «а», то вместо имени abc можно использовать шаблон a*.

Установка и работа с операционной системы RouterOS

1. Используем последнюю версию file.iso образа RouterOS Mikrotik для Интел-платформы, который берём в свою рабочую папку с сайта производителя mikrotik.com или из папки /home/4all. Создадим пустой образ виртуального диска формата qcow2 размером 111Мб:

```
qemu-img create -f qcow2 file.img 111M
```

С помощью менеджера виртуальных машин qemu установим операционную систему RouterOS на образ виртуального диска file.img из CD-образа file.iso

```
qemu file.img -cdrom file.iso -boot d
```

видим

```
[1]11424
```

```
VNC server running on `127.0.0.1:5900`
```

Виртуальная машина Qemu общается с внешним миром через протокол VNC- Virtual Network Computing.

Запускаем VNC–клиент: applications->Internet->Remote desktop viewer->Connect. Выбираем протокол VNC и Host – согласно выводу qemu - 127.0.0.1:5900.

Видим в VNC-клиенте начальное окно установки RouterOS. В VNC-клиенте мышь не поддерживается и он её захватывает. Захватывается и ввод с клавиатуры. Освобождение осуществляется комбинацией клавиш ctrlAlt. С помощью клавиш со стрелками и пробела выбираем модули для установки: system, ppp, dhcp, hotspot, advanced-tools, mpls, routerboard, routing, security. Нажимая i, начинаем установку. На вопросы отвечаем по умолчанию. После завершения установки, закрываем qemu (и RouterOS) , нажав комбинацию CtrlC в том терминале Ubuntu, откуда была запущена qemu, а не в VNC-клиенте.

Проверяем установку:

```
qemu file.img
```

Запускаем VNC–клиент. Вводим Login: admin, Password: оставляем пустым. Для остановки RouterOS используйте команду RouterOS system shutdown. При обилии маршрутизаторов ввод этой команды для каждого маршрутизатора утомляет. Поэтому иногда будем практиковать неправильную остановку RouterOS с помощью ввода комбинации CtrlC в консоли Ubuntu.

Важно помнить, что в VNC–клиент Qemu работает в двух режимах: системы и монитора. Переключение ctrl+alt 1 и ctrl+alt 2. Перейдя в монитор, посмотрите версию Qemu командой info version. Полный список команд с описанием смотри на сайте Qemu.org.

Можно стартовать RouterOS и так

qemu file.img&

Знак & в конце команды важен: Qemu не захватит терминал. Если мы введём такую последовательность

qemu file.img

CtrlZ

^Z

[1]+ Stopped qemu file.img

то второй экземпляр Qemu остановился. Запустим его

bg

[1]+ qemu file.img &

Qemu освободил терминал. В обоих случаях комбинация CtrlC теперь не пойдёт в программу Qemu. Нам для неправильной остановки RouterOS надо убить процесс **Qemu**. Находим его номер

ps aux|grep qemu

Команда **ps aux** выводит подробные сведения обо всех процессах в системе от всех пользователей. Этот вывод проходит через фильтр **grep**. Видим, кто запустил процесс и, какой у процесса номер

User5 9718 61.3 0.9 247212 80336 pts/3 Sl 21:09 0:08 qemu file.img

User5 9719 61.3 0.9 247212 80336 pts/3 Sl 21:09 0:08 qemu file.img

User5 9723 0.0 0.0 7624 944 pts/3 S+ 21:09 0:00 grep --color=auto qemu

Убиваем процессы

kill 9718

kill 9719

проверяем

ps aux|grep qemu

видим

12020 pts/3 S+ 0:00 grep --color=auto qemu

Если процесс убить не удалось, то попробуйте использовать

kill -KILL 9718

Убить все процессы qemu можно так

killall qemu

Убить чужие процессы вы не сможете.

Всегда проверяйте наличие нежеланных версий процесса. Набирая **ps aux|grep qemu|grep user**, вы узнаете, кто сейчас запустил qemu.

Узнать, кто сейчас занимает процессорное время можно с помощью консольной утилиты top.

RouterOS с сайта производителя после установки имеет время непрерывной работы равное 24 часам. В папке /home/4all лежит лицензионный образ RouterOS **mikrotik-5.21.img**, имеющий неограниченное время работы. Скопируйте лицензионный образ себе в папку и дальше работайте с ним.

2. В VNC-клиенте в окне Qemu не работает copy-paste. Добьемся этого другими средствами.

Реальные железные маршрутизаторы не имеют клавиатуры и монитора,

и настройка производится через консоль и последовательный порт. Qemu позволяет перенаправить последовательный порт на соединение по протоколу telnet. Наберём в терминале Ubuntu команду

qemu mikrotik-5.21.img -serial telnet:127.0.0.1:3000,server,nowait

Если qemu не запустилась, то проверьте не занял ли порт 3000

netstat -atn|grep 3000

Если вывод команды **netstat** не будет пустым, то порт занят и вам следует его поменять на заведомо незанятый.

В другой закладке окна терминала введём

telnet 127.0.0.1 3000

Мы в консоли RouterOS. Copy-paste работает с помощью мыши. Проверьте.

Сделайте копию R0.img для **mikrotik-5.21.img**.

cp mikrotik-5.21.img R0.img

и пока работайте с копией. Чистая версия нам понадобится для GNS3.

3. RouterOS внутри Qemu можно связать с внешним миром многими способами (см. документацию на сайте wiki.qemu.org). Мы для связи с хост-машиной Ubuntu используем тап-интерфейсы, а для связи между RouterOS — протокол UDP.

Связь между хост-машиной Ubuntu и устройствами внутри виртуальной лаборатории осуществляется как через консоль по протоколу Telnet, так и с использованием тап-интерфейсов. В лаборатории под Qemu запускается множество операционных систем RouterOS маршрутизаторов фирмы Mikrotik. В Ubuntu созданы тап-интерфейсы tapV00, tapV01 ... tapV07, tapV10, где V — различные целые числа (как правило — номера судентов). Интерфейсов tapV08 и tapV09 нет, по техническим причинам. Тап-интерфейсы помещены в мосты mV00, ... mV10. На мосты назначены адреса 10.V.0.2, 10.V.1.2 ... 10.V.7.2, 10.V.10.2. Маска /24. Для каждого V все адреса мостов образуют сеть 10.V.0.0/16. Например, для V=5 это сеть 10.5.0.0/24.

Тап-интерфейсы и мосты создаются при старте Ubuntu путём выполнения следующего скрипта из файла /etc/rc.local

```
taps="0 5 7 22 32 43 59"
ifs="00 01 02 03 04 05 06 07 10"
for s in $taps; do
for i in $ifs; do
openvpn --mktun --dev tap$s$i
brctl addbr m$s$i
brctl addif m$s$i tap$s$i
ifconfig tap$s$i up
ifconfig m$s$i 10.$s.$i.2 netmask 255.255.255.0 up
done
done
```

Мосты можно посмотреть командой **brctl show**, а адреса командой **ifconfig [ИмяИнтерфейса]**. Номер своей тап-сети пользователь лаборатории получает перед началом работы.

Возьмём тап-сеть с номером 0. Соединим маршрутизатор **R0** с тап-интерфейсом **tap000**. (Вы работаете со своей сетью и интерфейсом). В дальнейшем старайтесь придерживаться соглашения - номер в имени маршрутизатора определяется двумя последним цифрам в имени тап-интерфейса (**R0 - tap000**).

qemu R0.img -serial telnet:127.0.0.1:3000,server,nowait -net nic -net tap,script=no,downscript=no,ifname=tap000&

После ввода команды видим

VNC server running on `127.0.0.1:5900'

Запускаем VNC –клиент к 127.0.0.1:5900. Переключаемся в монитор ctrl+alt 2. Вводим **info network** и наблюдаем (Рис. 2), что Qemu увидела для RouterOS тап-интерфейс как Ethernet-карточку модели e1000 с MAC-адресом 525400123456.

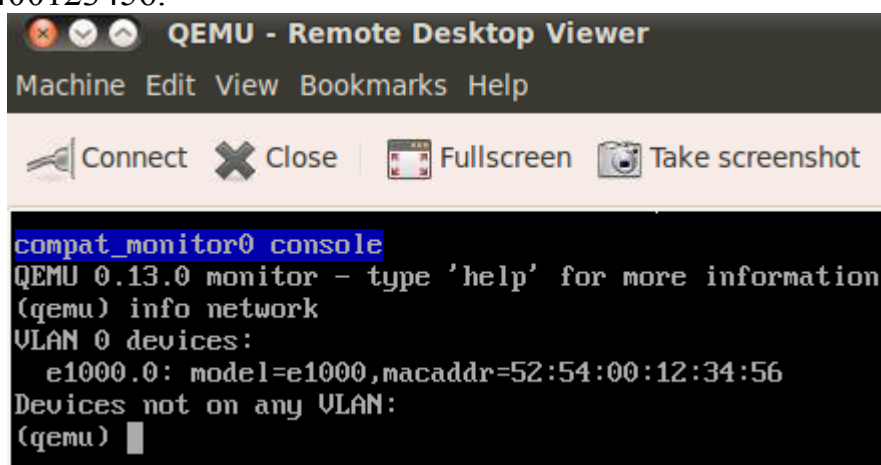


Рис. 2. Монитор Qemu

В новой закладке окна терминала введём

telnet 127.0.0.1 3000

Мы в консоли RouterOS. Смотрим интерфейсы

[admin@MikroTik] > **interface ethernet print**

Flags: X - disabled, R - running, S - slave

#	NAME	MTU	MAC-ADDRESS	ARP
0	R ether1	1500	52:54:00:12:34:56	enabled

Видим, что наша Ethernet-карточка модели e1000 с MAC-адресом 525400123456 называется в RouterOS как ether1. Со стороны Ubuntu тап-интерфейс **tap000** помещён в мост m000 с адресом 10.0.0.2/24. Поэтому в RouterOS назначим другой адрес из сети 10.0.0.0/24 например

[admin@MikroTik] > **ip address add address=10.0.0.1/24 interface=ether1**

Проверим связь MikroTik с хост-машиной Ubuntu

[admin@MikroTik] > **ping 10.0.0.2**

HOST	SIZE	TTL	TIME	STATUS
10.0.0.2	56	64	10ms	
10.0.0.2	56	64	1ms	

sent=2 received=2 packet-loss=0% min-rtt=1ms avg-rtt=5ms max-rtt=10ms

Для надёжности проверим связь и со стороны хост-машины Ubuntu. В

новой закладке окна терминала введём

```
User0@u10034d64bbb071:~$ ping 10.0.0.1
```

```
User0@u10034d64bbb071:~$ telnet 10.0.0.1
```

Мы опять в консоли RouterOS. Выход – CtrlD.

Тап-устройства имеют двоякую природу. С одной стороны это сетевые интерфейсы, а с другой стороны это устройство ввода-вывода /dev/net/tun. Сетевые пакеты, пришедшие на Тап-интерфейс можно прочитать как данные из устройства /dev/net/tun. Данные записанные в устройство /dev/net/tun исходят из Тап-интерфейса в виде сетевых пакетов. Опция

-net nic -net tap,script=no,downscript=no,ifname=tapXXX

командной строки qemu организует сетевой обмен данными между Ethernet-адаптером виртуальной машины и Тап-интерфейсом **tapXXX** хост-машины. Qemu производит этот обмен путём записи-чтения устройства /dev/net/tun в Ubuntu.

Подключимся с помощью безопасной консоли ssh. В новой закладке окна терминала введём

```
User0@u10034d64bbb071:~$ ssh admin@10.0.0.1
```

Согласитесь с выведенным предложением. Если вы получите сообщение о нарушении безопасности, то удалите файл в папке .ssh и попробуйте снова. Если вам будет отказано в подключении, то переставьте операционную систему RouterOS. Без поддержки доступа по ssh, вы не сможете быстро делать резервные копии и восстанавливаться из них.

Выход из безопасной консоли – CtrlD.

Mikrotik для настройки своих маршрутизаторов использует графическую Windows-утилиту winbox.exe. Для её запуска в Ubuntu установлена поддержка выполнения Windows-программ под Linux (технология **wine**)

Поместите в верхнюю панель ссылку на этот файл winbox.exe (/usr/local/bin). Запустите Winbox и соединитесь с маршрутизатором, введя в адрес 10.0.0.1 и (рекомендуется сохраниться , нажав save).

4. Сделайте ещё одну копию R1.img образа диска маршрутизатора mikrotik-5.21.img. Соединим два маршрутизатора, используя протокол UDP. Командой **netstat -aun** проверьте, что порты 33333 и 22222 свободны

```
qemu R0.img -serial telnet:127.0.0.1:3000,server,nowait -net nic -net udp,sport=33333,dport=22222,daddr=127.0.0.1&
```

```
Warning: vlan 0 is not connected to host network
```

```
VNC server running on `127.0.0.1:5900'
```

```
qemu R1.img -serial telnet:127.0.0.1:3001,server,nowait -net nic -net udp,sport=22222,dport=33333,daddr=127.0.0.1&
```

```
Warning: vlan 0 is not connected to host network
```

```
VNC server running on `127.0.0.1:5901'
```

Запускаем VNC –клиент и подключаемся к 127.0.0.1:5900 и 127.0.0.1:5901. Переключаемся в монитор ctrl+alt 2. Вводим

info network

и видим, для R0 (Рис. 3)


```
(qemu) info network
LAN 0 devices:
  udp.0: dcap: 33333->127.0.0.1:22222
  ne2k_pci.0: model=ne2k_pci,macaddr=52:54:00:12:34:56
```

Рис. 3. UDP для R0.

и для R1 (Рис. 4)

```
(qemu) info network
LAN 0 devices:
  udp.0: dcap: 22222->127.0.0.1:33333
  ne2k_pci.0: model=ne2k_pci,macaddr=52:54:00:12:34:56
```

Рис. 4. UDP для R1.

Ethernet-адаптеру поставлен в соответствие UDP-канал. Подключаемся к консолям роутеров через telnet

```
telnet 127.0.0.1 3000
```

```
telnet 127.0.0.1 3001
```

и назначаем роутерам имена R0 и R1

```
[admin@MikroTik] > system identity set name=R0
```

```
[admin@MikroTik] > system identity set name=R1
```

Промпт роутеров отразил смену имени

```
[admin@R0]>
```

```
[admin@R1]>
```

Назначаем роутерам адреса

```
[admin@R0]> ip address add address=1.1.1.1/24 interface=ether1
```

```
[admin@R1]> ip address add address=1.1.1.2/24 interface=ether1
```

Пингуем из R1 в R0.

```
[admin@MikroTik] > ping 1.1.1.1
```

Связь есть.

Все пакеты из Ethernet-интерфейса R0 поступают на UDP порт 22222 хост-машины Ubuntu и Ethernet-интерфейс R0 принимает все пакеты из UDP порта 33333 хост-машины. Для R1 наоборот. Все пакеты из Ethernet-интерфейса R1 поступают на UDP порт 33333 хост-машины ubuntu и Ethernet-интерфейс R1 принимает все пакеты из UDP порта 22222 хост-машины. Таким образом сетевой Ethernet-кабель моделируется двунаправленным UDP-каналом с использованием двух портов 22222 и 33333.

Попытаться из командной строки организовать сложную топологию – дело громоздкое и чревато ошибками. Тут на помощь приходит контейнер виртуальных машин GNS3

Настройка GNS3

В Ubuntu установлена GNS3 версии 0.7.4. Более новые версии приводят к проблемы совместимости GNS3 и нашей нестандартной установки qemu.

С помощью менеджера файлов mc распакуйте архив GNS3-0.7.4-src.tar.gz из /home/4all в отдельную папку (это будет GNS3-0.7.4-src). Для этого нажмите Enter на архиве и далее нажмите клавиши F5 и Enter. Дайте всем права на чтение всех файлов из GNS3-0.7.4-src

chmod -R a+r GNS3-0.7.4-src

и запись

chmod -R a+w GN.S3-0.7.4-src

Поместите в верхнюю панель ссылку на файл gns3 из этой папки. В папке qemuwrapper папки GNS3-0.7.4-src находится файл qemuwrapper.py. Перейдите в папку qemuwrapper и запустите в терминале программу qemuwrapper.py:

./qemuwrapper.py -p ВашПорт

Значение **ВашПорт** см. Ниже.

Убедитесь в отсутствии ошибок и остановите программу с помощью комбинацией клавиш Ctrl+c.

Напомним, что переключение менеджера файлов mc в режим терминала осуществляется комбинацией клавиш Ctrl+o.

GNS3 используется для быстрого сбора сетевых топологий. Эта оболочка генерирует командные строки для запуска Qemu, которые можно увидеть, запустив перед запуском GNS3 программу qemuwrapper.py.

Создайте в Вашей рабочей Папке две папки (рабочую и проектов)

mkdir tmp

mkdir projects

Запустим GNS3. Закроем окошко настройки GNS3 или окно создания проекта. Выберем для настройки Edit - Preferences. Устанавливаем путь к своей папке проектов projects и путь к папке, куда вы поместили образ диска **mikrotik-5.21.img** операционной системы RouterOS (Image Directory).

Настроим терминал: Edit - Preferences - General - Terminal setting. Возьмём для Preconfigured terminal commands - Gnome Terminal. Для Terminal command возьмём- gnome-terminal -t %d -e 'telnet %h %p' >/dev/null 2>&1 & и жмём OK.

Выберем для настройки Edit - Preferences - Qemu - General setting. Устанавливаем путь к своему файлу qemuwrapper.py и путь к своей рабочей папке tmp.

Важно, чтобы у всех одновременно работающих не было общих открытых портов TCP и UDP. Возьмём порты

10525+N - Qemuwrapper

20000+N*100 -UDP

3000+N*100 - консоль

N - ваш номер. Например, если ваш номер 7, то вписываем порты

10532 - Qemuwrapper

20700 -UDP

3700 – консоль

Хотя заметим, что в этой версии GNS3 затирает порт консоли и его придётся указывать перед каждым запуском сетевой топологии путём редактирования текстового файла topology.net конфигурации сетевой топологии.

Удалите внешний Qemuwrapper или дайте ему тот же порт. Нажимаем применить. Должен пройти Test. Жмём OK. Закроем GNS3.

Периодически проверяйте целостность GNS, нажимая Test. При проблемах- настройте GNS3 снова.

Запустим в терминале qemuwrapper.py, указав свой порт, например для номера 7

```
./qemuwrapper.py -p 10532
```

Снова запустим GNS3. Не обращайтесь внимание на окно с предупреждениями. Смотрите на сообщения в терминальном окне Qemuwrapper и анализируйте их.

Перейдём Edit - Preferences - Qemu - Qemu host и добавим в **Binary image** путь к образу диска операционной системы RouterOS **mikrotik-5.21.img**. Галочки kQemu , kvm не ставим. Галочку kvm нужно поставить, если у вас в Ubuntu установлена поддержка KVM. Дадём произвольное имя для образа. Нажимаем Save, Apply и Ok. Остановим GNS3.

Для проверки поддерживает ли процессор аппаратную виртуализацию KVM выполните. Для Intel

```
cat /proc/cpuinfo|grep vmx
```

Для Amd

```
cat /proc/cpuinfo|grep svm
```

Если вы увидите число строк более 0, то процессор поддерживает KVM.

Первая топология

Перед созданием и запуском топологии запустим qemuwrapper.py для своего порта. Когда вы наберётесь опыта, то это можно не делать.

Создадим первую топологию first (Рис. 5). Запускаем GNS3. В появившемся окошке вводим имя нового проекта first и ставим галочку Save pngams. Нажимаем Ok. Если проект уже создан, то выбираем Open. Для создания проекта можно открыть и пункт меню File-NewBlankProject. Перетащим два роутера (устройства Qemu host) с левой панели на центральную и меняем их имена на R0 и R1 (пункт change the hostname в контекстном меню устройства). Изменим иконку, выбрав в контекстном меню ChangeSymbol роутеров иконку Router. Соединим интерфейс e0 маршрутизатора R0 с интерфейсом e1 маршрутизатора R1 (Значок «add a link» на верхней панели GNS3, пункт Manual). После завершения соединения снова нажмите значок «Add a link». Для видимости имён интерфейсов выбираем View/Show interface labels. Параметры запуска маршрутизатора и используемые порты можно узнать, наведя на роутер мышью.

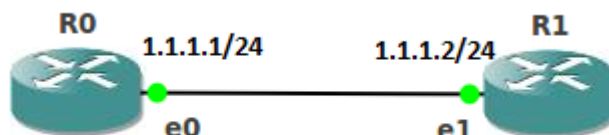


Рис. 5. Первая топология

Заметим о соотношении названий интерфейсов в GNS3и RouterOS

GNS	RouterOs
e0	ether1
e1	ether2
...	

По умолчанию GNS3 назначает маршрутизатору шесть сетевых интерфейсов. Менять не будем. Добавляем в маршрутизаторы тап-интерфейсы. Пусть у вас номер тап-сети 7. Начнём. В контекстном меню R0 выбираем Configure - R0 - Qemu options и вводим

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap700

Заметим, что во вводе присутствуют только три пробела. Нажимаем Ok. Для R1 вводим

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap701

Это приводит к созданию внутри маршрутизатора сетевой карточки ether7 которая связывается с сетевым тап-интерфейсом Ubuntu с именем tap700 (tap701). Тап-интерфейса в GNS3 не видно. Нажмите Apply и Ok. Если окно не закроется, то закройте GNS3, убейте папку .gns3 и заново настройте GNS3, начиная с распаковки архива. Закройте и запустите GNS3. В окне New Project выберите Open a project, откройте папку first, выберите файл topology.net и Open. Вы должны увидеть топологию first (рис.5). В контекстном меню R0 выбираем Configure - R0(R1) - Qemu options вы должны увидеть:

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap700

(-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap701)

Если это не так, то закройте GNS3, убейте папку .gns3 и заново настройте GNS3, начиная с распаковки архива.

Перед запуском топологии проверьте не заняты ли ваши порты TCP (UDP)

netstat -atn|grep начальные-цифры-порта (netstat -aun)

Используя команду ps aux и фильтр grep по начальным цифрам портов, можно определить, кто забрал ваш порт.

Запустите проект (зелёный треугольник вверх). Наблюдайте загрузенность системы с помощью System monitor. Запустите консоли маршрутизаторов (контекстное меню маршрутизатора или меню console to all devices).

Если проект или консоли не запустились, то анализируйте вывод программы Qemuwrapper.

Сохраните проект. Откройте в папке проектов projects появившуюся папку проекта first. Если папки first нет, то проверьте настройки GNS3 (Adit-Preference-Genereal setting-Project direrctory должно быть установлено на папку projects. Откройте двойным щелчком (программой gedit) файл topology.net. Изучите его. Поняв его устройство можно менять топологию без GNS3. Перед запуском топологии, сделанной на другой машине, обязательно проверьте и исправьте этот файл. К сожалению GNS3 забыл наши порты консолей. Добавьте порты для консолей в возрастающем порядке номеров роутеров

```

autostart = False
[Qemu 127.0.0.1:10537] -согласно номеру пользователя 7
workingdir = working – обязательная строка
udp = 20700 -согласно номеру пользователя 7
[[QemuDevice]]
image = /home/user7/mikrotik-5.21.img -измените при переносе с другой
машины
netcard = e1000
kvm = True - эта строка есть, если только работаете на чистом железе
с установленным kvm
[[QEMU R0]] – начало конфигурации для R0
Console = 3700- прописываем вручную
options = -net nic,vlan=6 -net
tap,script=no,downscript=no,vlan=6,ifname=tap700
e0 = R1 e1 -связь
[[QEMU R1]] – начало конфигурации для R1
Console = 3701- прописываем вручную
options = -net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap701
e1 = R0 e0 -связь
[GNS3-DATA]
workdir = working -обязательная строка

```

К сожалению после сохранения топологии в GNS3 настройки портов консоли удаляются. Поэтому перед запуском GNS3 сделайте копию файла topology.net или заново назначайте порты консоли в этом файле.

Улучшив момент, когда система не занята, стартуйте все маршрутизаторы. Старт-стоп отдельного маршрутизатора производится из его контекстного меню. Не используйте рестарт – не всегда работает. Дождитесь окончания загрузки.

В папке first вы увидите папку working, а в ней папки R0 и R1, а в этих папках два образа диска FLASH и SWAP. Если это не так, то остановитесь и проверьте настройки, так как в дальнейшем возникнут трудности.

Осуществлять конфигурацию маршрутизатора через консоль с помощью интерфейса командной строки можно тремя способами: либо через VNC-клиент, либо через консоль маршрутизатора в GNS3, либо через telnet. В первом случае надо знать порт VNC-сервера. Порт VNC можно узнать в консоли Qemuwrapper, если запускать маршрутизаторы по одному. Во втором случае консоль маршрутизатора вызывается двойным щелчком мыши на иконке маршрутизатора. В третьем случае консоль маршрутизатора вызывается через консоль Ubuntu с помощью команды **telnet 127.0.0.1 порт-консоли**. Заметим, что второй случай также приводит к неявному вызову команды **telnet 127.0.0.1 порт-консоли**.

Во втором и третьем случае работать удобнее всего, так как работает CopyPaste.

Если консоль маршрутизатора не запустилась, то анализируйте вывод в консоли Qemuwrapper и занятые порты (netstat, ps aux). Если там всё

нормально, то останавливаем маршрутизатор, убиваем папку маршрутизатора в папке working проекта first и снова запускаем маршрутизатор.

Запускаем консоли маршрутизаторов, два раза щёлкнув на их иконках в GNS3 (Рис. 6). Если терминальное окно будет отличное от приведенного на Рис. 6, то исправьте настройки терминала в GNS3 (см. выше)

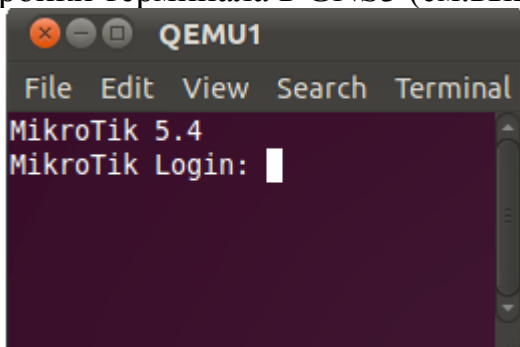


Рис. 6. Gnome Terminal

Вводим имя admin без пароля. Назначаем имена (используйте CopyPaste)

system identity set name=R0

system identity set name=R1

Каждый маршрутизатор шлёт соседям широковещательную информацию о себе и обрабатывает такую же информацию, принятую от соседей. Результат можно посмотреть командой **/ip neighbor print** и **/ip neighbor print detail**. Мы увидим соседей, достижимых по Ethernet-протоколу, в том числе через сетевые мосты.

[admin@R0] > **ip neighbor print**

```
#INTERFACE ADDRESS MAC-ADDRESS IDENTITY VERSION BOARD
0 ether1      00:AA:00:28:C5:01 R1          5.2      x86
```

[admin@R0] > **ip neighbor print detail**

```
0 interface=ether1 mac-address=00:AA:00:28:C5:01 identity="R1"
platform="MikroTik" version="5.2" unpack=none age=17s uptime=10m5s
software-id="1ZGH-GE5Q" board="x86" ipv6=yes interface-name="ether2"
```

Этот вывод надо понимать так. Этот маршрутизатор R0 через интерфейс **ether1** обнаружил соседа **R1**. Сосед R1 послал эту информацию через свой интерфейс **ether2** у которого MAC -адрес равен **00:AA:00:28:C5:01**

Проверим, что интерфейс ether2 у R1 имеет этот MAC-адрес

[admin@R1] > **interface ethernet print where name=ether2**

```
# NAME MTU MAC-ADDRESS ARP
1 R ether2 1500 00:AA:00:28:C5:01 enabled
```

Аналогично для R1

[admin@R1] > **ip neighbor print**

```
#INTERFACE ADDRESS MAC-ADDRESS IDENTITY VERSION BOARD
0 ether2      00:AA:00:F4:68:00 R0          5.2      x86
```

[admin@R1] > **ip neighbor print detail**

```
0 interface=ether2 mac-address=00:AA:00:F4:68:00 identity="R0"
platform="MikroTik" version="5.2" unpack=none age=1m uptime=10m6s
software-id="1ZGH-GE5Q" board="x86" ipv6=yes interface-name="ether1"
```

Этот вывод надо понимать так. R1 через интерфейс ether2 обнаружил

соседа R0. Сосед послал эту информацию через свой интерфейс **ether1** у которого MAC -address= **00:AA:00:F4:68:00**

Проверим, что интерфейс ether1 у R0 имеет этот MAC-адрес

[admin@R0] > **interface ethernet print where name=ether1**

```
#  NAME                               MTU  MAC-ADDRESS  ARP
0 R  ether1                           1500 00:AA:00:F4:68:00 enabled
```

Возьмите за привычку давать маршрутизаторам имена и набирать эти команду обнаружения соседей. Если вдруг маршрутизатор не видит соседей, то остановите его, убейте его папку в папке working проекта и стартуйте маршрутизатор заново.

Проверим правильность подключения тап-интерфейсов, набрав в консоли

interface ether pint

Должны увидеть в последней строке

```
6 R  ether7                           1500 52:54:00:12:34:5C enabled
```

Если это не так, то остановите маршрутизатор, убейте его папку в папке working проекта и стартуйте заново. Хотя, в принципе, если MAC-адрес 52:54:00:12:34:5C будет назначен не на интерфейс ether7, то можно работать с этим интерфейсом в качестве интерфейса, соответствующего тап-сети.

Назначаем адреса согласно своего номера тап-сети 7. На R0

ip ad ad address=10.7.0.1/24 interface=ether7

ip ad ad address=1.1.1.1/24 interface=ether1

На R1

ip ad ad address=10.7.1.1/24 interface=ether7

ip ad ad address=1.1.1.2/24 interface=ether2

Должны пойти пинги в консолях маршрутизаторов

[admin@R0] > **ping 10.7.0.2**

[admin@R0] > **ping 1.1.1.2**

[admin@R1] > **ping 10.7.1.2**

[admin@R1] > **ping 1.1.1.1**

И в консоли Ubuntu **ping 10.7.0.1** и **ping 10.7.1.1**.

Если пинги непошли, то остановите нужный маршрутизатор, убейте его папку в папке working проекта и стартуйте заново.

Помните, что если в GNS3 назначить маршрутизатору один сетевой интерфейс, а не шесть, как у нас сейчас, то тап-интерфейс в RouterOS будет называться ether2 и настройки адресов для тап-интерфейса ether7 будут относиться к несуществующему интерфейсу.

Наличие тап-интерфейсов с настроенными внутри RouterOS адресами даёт четвёртый способ доступа к консоли RouterOS из Ubuntu **telnet 10.7.0.2** и **telnet 10.7.1.2**. Пятый способ, позволяющий без пароля вызывать терминал предоставляет безопасная консоль **ssh admin@10.7.0.2** и **ssh admin@10.7.1.2**.

Запускаем winbox. Вводим адрес 10.7.0.2, имя admin. Сохраняемся и соединяемся. Делаем то же для адреса 10.7.0.2. Совершите обзор возможностей маршрутизатора. Они поражают, если посмотреть в Ubuntu на размер виртуального диска командой **ls -l *img**. В winbox есть пункт меню

NewTermial, что даёт шестой способ доступа к консоли RouterOS из Ubuntu.

Маршрутизаторы доступны по протоколам FTP и SSH. Более того маршрутизаторы имеет веб-интерфейс и доступны в Ubuntu из Firefox по адресам <http://10.7.0.1> и <http://10.7.1.1>.

GNS3 автоматически генерирует команды для Qemu

Запустим топологию first. Смотрим в окно консоли предварительно запущенного Qemuwrapper. Для каждого маршрутизатора в топологии GNS3 делает для mikrotik-5.21.img разностные образы с именем FLASH. SWAP – это образ диска для свопинга:

```
Formatting '/home/user0/projects/First/working/R0/FLASH', fmt=qcow2  
size=103809024 backing_file='/home/user0/LIC/mikrotik-5.21.img'  
encryption=off cluster_size=0
```

```
Formatting '/home/user0/projects/First/working/R0/SWAP', fmt=qcow2  
size=1073741824 encryption=off cluster_size=0
```

Берём содержимое консоли Qemuwrapper в карман и отформатируем согласно правилам записи командной строки Qemu (см. <http://wiki.Qemu.org/Manual>). MAC-адреса мы не указываем и убираем несоединённые интерфейсы. Получим

```
qemu -name R0 -m 256 /home/user7/projects/first/working/R0/FLASH  
-hdb /home/user7/projects/first/working/R0/SWAP -net nic,vlan=0 -net  
udp,vlan=0,sport=20702,dport=20703,daddr=127.0.0.1 -serial  
telnet:127.0.0.1:3700,server,nowait -net nic,vlan=6 -net nic -net  
tap,script=no,downscript=no,vlan=6,ifname=tap700
```

```
qemu -name R1 -m 256 /home/user7/projects/first/working/R1/FLASH  
-hdb /home/user7/projects/first/working/R1/SWAP -net nic,vlan=1 -net  
udp,vlan=1,sport=20703,dport=20702,daddr=127.0.0.1 -serial  
telnet:127.0.0.1:3701,server,nowait -net nic,vlan=6 -net nic -net  
tap,script=no,downscript=no,vlan=6,ifname=tap701
```

Заметим, что здесь мы не видим образ диска RouterOS mikrotik-5.21.img, а видим разностные образы с именем FLASH и образ диска для свопинга SWAP.

Если ввести эти команды в консоли ubuntu, то получим тот же эффект, что и в GNS3. Делать это не обязательно. Если решитесь, то лучше для этого создайте командный файл.

Строки **-serial telnet:127.0.0.1:3700,server,nowait** и **-serial telnet:127.0.0.1:3701, server,nowait** отвечают за связь с консолью.

Для R0 строка **-net nic,vlan=0 -net udp,vlan=0,sport=20702,dport=20703,daddr=127.0.0.1** говорит, что все пакеты из интерфейса e0 (ether1) поступают на UDP порт 20702 хост-машины ubuntu и интерфейс e0 (ether1) принимает все пакеты из UDP порта 20703 хост-машины.

Для R1 наоборот. Строка

-net nic,vlan=1 -net udp,vlan=1,sport=20703,dport=20702,daddr=127.0.0.1

говорит, что все пакеты из интерфейса e1 (ether2) поступают на UDP порт 20703 хост-машины ubuntu и интерфейс e1 (ether2) принимает все пакеты из UDP порта 20701 хост-машины.

Таким образом сетевой Ethernet-кабель моделируется двунаправленным UDP-каналом с использованием двух портов **20702** и **20703**. Ethernet-адаптеру поставлен в соответствие двусторонний UDP-канал

И, наконец, строки

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap700

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap701

приводят к созданию внутри маршрутизатора сетевой карточки ether7, которая связывается с сетевым тап-интерфейсом с именем tap700 (tap701).

В принципе можно обойтись и без GNS. Однако для сложных топологий легко запутаться. Остановите все маршрутизаторы (красный квадрат). Закройте GNS. Не сохраняйте топологию – вы затрёте порты консолей.

Шаблон для топологий

Создадим топологию template. Поместим на панель несколько роутеров. Дадим роутерам имена от R0, R1 ... R7, R10. В названиях роутеров должны быть отражены номера ваших тап-интерфейсов. Пусть у нас тап-сеть 7. Поставим в соответствие роутеру R0 тап-интерфейс **tap700**. Роутеру R1 тап-интерфейс **tap701** и т.д. Для чего в контекстном меню роутеров R0, R1 ... R16 в поле options вводим настройки для тап-интерфейсов:

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap700

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap701

...

-net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap710

Сохранимся и закроем GNS3. Проверим, что никто не занял нашу тап сеть 5

ps ax|grep tap7

Вручную впишем в файл топологии порты для консолей. Сделайте копию файла, так как GNS3 при следующем сохранении затрёт порты. Проверим, что никто не занял наши порты консоли, например

netstst -ant|grep 300

Откроем топологию в GNS3. В папке working проекта появятся подпапки R0, R1 ... R10. Запускаем все роутеры, нажимая соответствующую иконку на верхней панели GNS3. В папках R0, R1 ... R10 появятся файлы FLASH и SWAP. Ждём окончания загрузки. Запускаем все консоли, нажимая соответствующую иконку на верхней панели GNS3.

В селекторе окон (вы поместили селектор ранее на верхнюю панель Ubuntu) выбираем консоль роутера R0. Жмём Enter и вводим имя admin и пустой пароль. Мышью берём в карман имя admin до конца строки. В селекторе окон выбираем консоль роутера R1. Жмём Enter и извлекаем имя admin из кармана. Жмём Enter 2 раза. Повторяем так для всех роутеров.

Проверяем, что роутеры имеют интерфейс ether7 для связи тап-интерфейсами Ubuntu. В селекторе окон выбираем консоль роутера R0. Вводим **int print**. Берём команду в карман (до конца строки). Вставляем команду из кармана для остальных роутеров. Командой из контекстного меню роутеров в GNS3 останавливаем роутеры у которых нет ether7. Проверяем для этих роутеров в GNS3 строку **-net nic -net tap**, определяющую связь тап-интерфейса Ubuntu с роутером. В папке роутера в папке working проекта удаляем файлы FLASH и SWAP. Запускаем роутеры и снова проверяем наличие ether7.

В селекторе окон выбираем консоль роутера R0. Мы ему назначили тап-интерфейс tap700 (тап-сеть 7). Назначаем адрес на ether7 из соответствующей сети **10.7.0.0/24**

```
ip address add address=10.7.0.1/24 interface=ether7
```

Берём команду в карман (не до конца строки)

В селекторе окон выбираем консоль роутера R1. Мы ему назначили тап-интерфейс tap701. Вставляем команду из кармана и изменяем её. Назначаем адрес на ether7 из соответствующей сети **10.7.1.0/24**

```
ip address add address=10.7.1.1/24 interface=ether7
```

Назначаем адрес на ether7 для остальных роутеров.

Из консоли Ubuntu проверим доступность роутеров

```
ping 10.7.0.1
```

```
ping 10.7.1.1
```

```
...
```

```
ping 10.7.10.1
```

Остановите недоступные роутеры, удалите для них файлы FLASH и SWAP и переназначьте адреса.

Свяжемся с роутерами с помощью безопасной оболочки ssh, которая позволит входить в роутеры без ввода пароля.

```
ssh admin@10.7.0.1
```

На запрос ответьте yes и вы попадёте в роутер. Выход Ctrl+d.

Для других роутеров

```
ssh admin@10.7.1.1
```

```
...
```

```
ssh admin@10.7.7.1
```

Остановите недоступные роутеры, удалите для них файлы FLASH и SWAP и переназначьте адреса. Если и это не поможет, то переставьте заново RouterOS.

Не двигайтесь дальше, пока не заработает **ssh** для всех роутеров.

Остановите топологию. Закройте GNS3. Далее при выполнении очередной лабораторной работы следует лишь сделать копию папки Template с топологией шаблона.

Закладки

Для роутеров R0, R1 ... R7 создадим в консоли Ubuntu восемь профилей с именами 0 1 2 ... 7 соответственно. (Edit-Profiles-New). В профиле зададим

заголовок, команду и другие настройки, например для R0 и тап-сети 0 (Рис. 7)

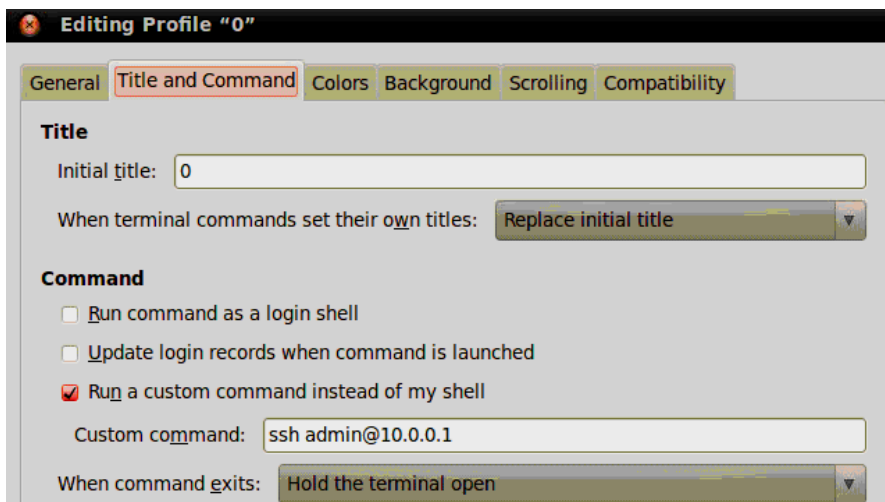


Рис. 7. Настройка закладки

В верхней панели десктопа Ubuntu выберем AddToPanel и добавим Custom Application Launcher, где введём команду запуска консоли с 8 закладками

gnome-terminal --tab-with-profile=0 --tab-with-profile=1 --tab-with-profile=2 --tab-with-profile=3 --tab-with-profile=4 --tab-with-profile=5 --tab-with-profile=6 --tab-with-profile=7

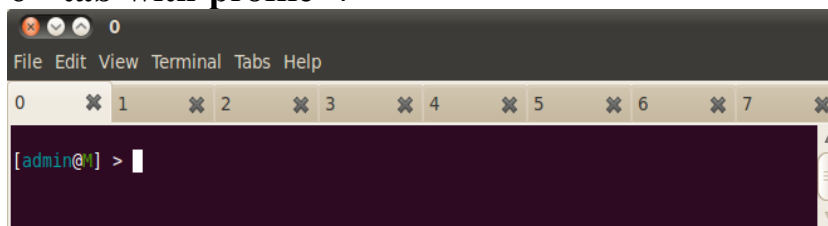


Рис. 7. Закладки

Запустите эту консоль с восемью закладками. Вы увидите в консоли 8 закладок (Рис. 7). Каждая закладка связана с отдельным роутером.

Казалось, было бы легче создать профили с командами для связи с роутерами через консоль, например **telnet 127.0.0.1 3000**, где 3000 – порт консоли. К сожалению, GNS3 не сохраняет и затирает назначенные вручную порты консоли.

Использование закладок в консоли Ubuntu представляется наиболее эффективным способом одновременной настройки нескольких роутеров. Удобство работы в основном достигается использованием технологии Copy/Paste.

Работа с RouterOS из командной строки

Запустите топологию first. Обратимся к R0 из Ubuntu через telnet или ssh. Для получения списка пунктов меню нажмите табуляцию. Если два раза нажать клавишу табуляции, то увидим список команд и подменю. Более подробная справка выводится после нажатия знака ?. Синие пункты — это меню, а остальное — команды. Перейдём в меню interface, набирая строку с этим же именем interface и нажимая Enter. Команда или подменю

идентифицируется по первым набранным символам и, если они уникальны, то текст зеленеет. Поэтому достаточно набрать `in`. При желании видеть всю команду нажмите табуляцию. Команда `print` выведет интерфейсы. Команда `print detail` выведет более детальную информацию. Вернёмся уровнем выше «..». Переход в главное меню `/`. Тоже самое можно сделать, набрав команду целиком

```
[admin@ R0]> interface print detail
```

Команды можно вводить из разных уровней подменю. Зайдём в меню **ip address** и выполним команду **print**

```
[admin@ R0]> /ip address> pr
```

Или то же самое

```
[admin@ R0]> /ip address pr
```

Команды работают в режиме полного ввода

```
[admin@R0]>/ip address>add add=2.2.2.2/24 interface=ether2
```

или диалога

```
[admin@ R0]> /ip address> add
```

```
address: 3.3.3.3/24
```

```
interface: ether3
```

Всегда помогает табуляция и знак вопроса. Настройки можно поменять. Например, для адреса `2.2.2.2/24` поменяем интерфейс на `ether4`. В начале, узнаем номер

```
[admin@ R0]> /ip address> pr
```

```
Flags: X - disabled, I - invalid, D - dynamic
```

	#	ADDRESS	NETWORK	INTERFACE
0		10.0.0.1/24	10.0.0.0	ether7
1		1.1.1.1/24	1.1.1.0	ether1
2		2.2.2.2/24	2.2.2.0	ether2
3		3.3.3.3/24	3.3.3.0	ether3

Это №2. Меняем

```
[admin@ R0]> /ip address> set 2 interface=ether4
```

```
[admin@ R0]> /ip address> pr
```

```
Flags: X - disabled, I - invalid, D - dynamic
```

	#	ADDRESS	NETWORK	INTERFACE
0		10.0.0.1/24	10.0.0.0	ether7
1		1.1.1.1/24	1.1.1.0	ether1
2		2.2.2.2/24	2.2.2.0	ether4
3		3.3.3.3/24	3.3.3.0	ether3

Убиваем созданное

```
[admin@ R0]> /ip address> rem 2,3
```

```
[admin@ R0]> /ip address> pr
```

	#	ADDRESS	NETWORK	INTERFACE
0		10.0.0.1/24	10.0.0.0	ether7
1		1.1.1.1/24	1.1.1.0	ether1

Это основной принцип. Интерфейс чрезвычайно мощный. Например, можно фильтровать вывод

```
/ip address print where interface=ether1
```

Уничтожим все IP-адреса на интерфейсе ether2

```
/ip address remove [find where interface=ether2]
```

Кроме того поддерживаются скрипты. Вот пример скрипта, который наряду с IP-адресами выводит и MAC-адреса

```
foreach i in=[/ip address find] do={  
:local intr [/ip address get $i interface];  
:local addr [/ip address get $i address];  
:local MAC [/int eth get [/int eth find name=$intr] mac-address] ;  
:put ($addr." ".$intr." ".$MAC); }
```

При желании, изучите скриптинг самостоятельно по материалам официального сайта wiki.mikrotik.com. Покажем лишь как добавить и запустить скрипт

```
[admin@MikroTik]>/system script
```

```
[admin@MikroTik]> /system script> add
```

```
[admin@MikroTik]> /system script> pr
```

```
0I name="script1" owner="admin" policy=ftp,reboot,read,write,policy,test,  
winbox, password ,sniff, sensitive, api run-count=0 source=""
```

Наберём тест скрипта в редакторе gedit. Выделим его, поместим в карман и вставим в редакторе скриптов RouterOS

```
[admin@MikroTik]> /system script> edit 0 source
```

Сохраняем скрипт командой Ctrl o и выполняем

```
[admin@MikroTik]> /system script> run 0
```

Видим IP- и MAC-адреса интерфейсов

```
10.0.0.1/24 ether7 52:54:00:12:34:5C
```

```
1.1.1.1/24 ether1 00:00:AB:18:47:00
```

Импорт и экспорт топологий

Периодически делайте резервную копию, следуя нижеизложенной методике.

1. В RouterOs есть пара команд **export** и **import**, позволяющая сохранять конфигурацию в виде читаемого тестового файла команд и затем восстанавливать их. Рассмотрим скрипт **makeGetExport** из папки /usr/local/bin для создания в Ubuntu резервной копии настроек маршрутизаторов топологии

```
#!/bin/bash
```

```
for ((i=$2;i<=$3;i++)) ;do
```

```
ssh admin@10.$1.$i.1 "export compact file=ESi"
```

```
sftp admin@10.$1.$i.1:/E*
```

```
done
```

Цикл for организует перебор целых чисел в диапазоне от \$2 до \$3. Здесь \$2 и \$3 – 2-й и 3-й аргументы этого скрипта. В цикле утилита ssh (secure shell) даёт команду **export compact file=ESi** маршрутизатору Mikrotik с адресом **10.\$1.\$i.1** на сохранение конфигурации маршрутизатора в файл **ESi**, а команду sftp забирает этот файл из Mikrotik в локальный файл в Ubuntu.

Здесь **\$1** - 1-й аргумент этого скрипта, определяющий номер тап-сети.

Запустим в GNS3 топологию first. В нашей топологии номер тап-сети равен 7. Чтобы забрать конфигурацию в Ubuntu надо выполнить команду

```
makeGetExport 7 0 1
```

Отвечайте yes на вопросы Ubuntu относительно принятия сертификатов. В случае проблем удалите файл known_hosts в папке .ssh.

В текущей папке появятся файлы резервных копий E0.rsc и E1.rsc.

Для восстановления из Ubuntu конфигураций операционных систем маршрутизаторов предлагается скрипт **Import** из папки /usr/local/bin:

```
#!/bin/bash  
for ((i=$2;i<=$3;i++)) ;do  
scp E$i.rsc admin@10.$1.$i.1:  
ssh admin@10.$1.$i.1 "import E$i"  
done
```

Здесь команда scp безопасно копирует по сети файл резервной копии из Ubuntu в Mikrotik. Команда ssh удалённо выполняет в Mikrotik команду **import E\$i** восстановления конфигурации из резервной копии.

Роутеры не должны иметь настроек, присутствующих в импортируемой конфигурации. Иначе вы не получите сообщения об успешном выполнении **Script file loaded and executed successfully**. Поэтому перед использованием скрипта **Import** следует в файлах команд E*.rsc закомментировать команду назначения IP-адреса на тап-интерфейс **ether7**. Эти адреса не могут быть не назначены заранее.

Готовим полученные файлы E0.rsc и E1.rsc для импорта, закомментировав команды назначения адресов на тап-интерфейсы

```
E0.rsc  
/ip address  
#add address=10.7.0.1/24 interface=ether7  
add address=1.1.1.1/24 interface=ether1  
/system identity  
set name=R0  
E1.rsc  
/ip address  
#add address=10.7.1.1/24 interface=ether7  
add address=1.1.1.2/24 interface=ether2  
/system identity  
set name=R1
```

Помещаем в архив файлы topology.net, E0.rsc и E1.rsc. На этом или другом компьютере распаковываем архив в новую папку firstCopy и создаём там папку working. Возможно, изменяем в файле topology.net путь к образу RouterOs и имена тап-интерфейсов. Запускаем топологию. В папке working появятся папки R0 и R1. Останавливаем топологию.

Можно было назначить адреса на тап-интерфейсы роутеров R0 и R1 топологии firstCopy и восстановить конфигурацию командой **Import 7 0 1**.

Поступим иначе, воспользовавшись тем, что в топологии шаблона `template` уже назначены адреса на тап-интерфейсы. Помним, что в папке `working` созданного нами шаблона `template`

подпапка `R0` содержит роутер у которого на тап-интерфейс назначен адрес `10.7.0.1/24`,

подпапка `R1` содержит роутер у которого на тап-интерфейс назначен адрес `10.7.1.1/24`.

Заменяем содержимое папки `R0` из `firstCopy` (файлы `FLASH` и `SWAP`) содержимым папки `R0` из `template` (файлы `FLASH` и `SWAP`) и содержимое папки `R1` из `firstCopy` содержимым папки `R1` из `template`. Тем самым мы назначим для роутеров `R0` и `R1` топологии `firstCopy` адреса на тап-интерфейсы. Стартуем топологию `firstCopy`. Адреса Проверим из `Ubuntu` доступность адресов тап-интерфейсов роутеров через протокол `ssh`

```
ssh admin@10.7.0.1
```

```
ssh admin@10.7.1.1
```

Восстанавливаем конфигурацию командой

```
Import 7 0 1
```

Отвечайте `yes` на вопросы `Ubuntu` относительно принятия сертификатов. Маршрутизаторы подхватят конфигурацию и перегрузятся. В случае проблем удалите файл `known_hosts` в папке `.ssh`.

Убедитесь, что конфигурация восстановлена. В частности из роутера `R1` должны пойти пинги в сторону `R0`

```
ping 1.1.1.1
```

Таким же образом сколь угодно сложная топология из весьма большого числа `RouterOs` может быть сохранена и далее воспроизведена в `GNS3` на другом компьютере.

О персональных компьютерах (ПК) и свичах в топологиях

Не пользуйтесь стандартным свичом в `GNS3`, расположенным на левой панели.

Традиционно в `GNS3` в качестве модели ПК используется отдельная программа `Virtual PC Simulator`, которая моделирует 10 ПК. Модель ПК можно соединять с роутером только через свич `GNS3`. Для этого модель ПК подключают по протоколу `UDP` к облаку в `GNS3`. Облако соединяют со стандартным свичом в `GNS3`. И наконец, свич соединяют с роутером.

В лаборатории в качестве роутеров, ПК и свичей мы будем использовать те же виртуальные машины под управлением `Mikrotik RouterOS`. В топологии они будут различаться только внешним видом (иконкой). Роутер `Mikrotik` можно превратить в свич, поместив в мост его интерфейсы. При таком подходе роутер и ПК можно соединять друг с другом как без свича, так и через свич (тот же `Mikrotik`) без использования облака `GNS3`.

ПК в `Virtual PC Simulator` потребляют меньше ресурсов, чем ПК в виде `RouterOS`. Но при этом тратится более ценный ресурс – `UDP` порты, которые нам крайне необходимы для организации многопользовательской работы.

Синхронизация проектов между лабораториями

Для передачи файлов используют протоколы FTP и SSH. FTP-клиент есть и в Windows и вызывается через командную строку cmd. Для работы по FTP введите

ftp Адрес

Далее введите имя и пароль. Попадёте в командную строку FTP. Перейдите в бинарный режим с помощью команды bin. Выгружать файлы следует из командной строки FTP с помощью команды mput маска. Забирать файлы следует с помощью команды mget маска. Пример маски *.txt.

Попробуйте с компьютера выгрузить и загрузить в/из лабораторию 192.168.14.56 (192.168.3.253, labs.mikrotik.com.ua) произвольный небольшой файл.

Попробуйте из лаборатории выгрузить и загрузить в/из виртуальную машину Mikrotik произвольный небольшой файл.

Для передачи файлов часто используют безопасное копирование с помощью протокола SSH. В мире linux безопасное копирование осуществляется с помощью команды scp. Скопируйте произвольный небольшой файл из сетевой лаборатории в виртуальную машину Mikrotik, например так

scp abc.txt admin@:10.0.0.1:abc.txt

Заберите этот файл обратно в лабораторию

scp admin@:10.0.0.1:abc.txt abc1.txt

По окончании работы с помощью nxclient в удалённой сетевой лаборатории 192.168.14.56 (192.168.3.253, labs.mikrotik.com.ua) сделайте как указано выше с помощью утилиты makeGetExport резервную копию своей работы.

Восстановите топологию из резервной копии в лаборатории 192.168.3.253, labs.mikrotik.com.ua (192.168.14.56).

Перенос архива резервной копии осуществляйте с помощью протокола FTP или SSH, с помощью электронной почты или с использованием флеш-накопителя.

Если вы организовали лабораторию на своём домашнем компьютере, то можно восстановить свою работу из резервной копии и продолжать работать локально. Сделайте на своём компьютере новую резервную копию и отправьте её архив в удалённую лабораторию. Зайдите с помощью nxclient в удалённую лабораторию, восстановите свою работу и продолжайте работать удалённо.

Контрольные вопросы

1. Как организован графический интерфейс операционных систем семейства Linux?
2. Как в технологии фирмы NoMachine организован доступ к удалённому рабочему столу Linux?
3. Перечислите основные клавиатурные комбинации файлового менеджера mc.

4. Как в Linux просмотреть содержимое папки, изменить текущую папку, выполнить программу, скопировать (переместить) файлы?
5. Что в Linux определяет переменная окружения PATH?
6. Как в Linux использовать шаблоны имён?
7. Что такое qemu и как его использовать?
8. По какому протоколу виртуальная машина qemu общается с внешним миром? Как подсоединиться к qemu по этому протоколу?
9. Перечислите основные клавиатурные комбинации VNC-клиента qemu.
10. Объясните значение комбинации клавиш CtrlC
11. Объясните значение знака & в конце командной строки Linux.
12. Объясните значение комбинации клавиш CtrlZ и команды bg.
13. Как в Linux убить процесс?
14. Как узнать какие пользователи Linux запустили qemu?
15. Как использовать фильтр grep?
16. Как подсоединиться к виртуальной машине qemu через последовательный порт?
17. Объяснить назначение тап-интерфейсов и их использование в qemu.
18. Объяснить двоякую природу Тап-устройств.
19. Указать в тексте примеры использования безопасной консоли ssh.
20. Что такое winbox и что надо сделать чтобы начать его использовать?
21. Как в Linux называется поддержка выполнения Windows-программ?
22. Как в qemu моделируется связь двух виртуальных машин с помощью Ethernet-кабеля? Укажите в тексте пример и объясните его.
23. Укажите два способа определения занятости TCP(UDP)-порта.
24. Что такое GNS3, что в нём надо настраивать и как он работает?
25. Чем полезен qemuwrapper?
26. Как в GNS3 обеспечить уникальность своих портов консоли в своей топологии?
27. Что GNS3 помещает в папку проекта после старта соответствующей топологии. Объясните структуру файла topology.net. (с. 12-13)
28. Что автоматически генерирует GNS3 при старте топологии? Объяснить пример на с. 15-16.
29. Зачем в работе создаётся шаблон для топологий.
30. Объяснить назначение команд RouterOS /ip neighbor print и /ip neighbor print detail.
31. Объясните назначение и алгоритм работы скриптов makeGetExport и Import (с. 21)
32. Что помещают в архив резервной копии топологии? Что изменяют в rsc файлах перед помещением их в архив.
33. Объясните процедуру восстановления топологии из резервной копии.
34. Объясните роль шаблона для топологий при восстановлении из резервной копии.

35. Почему не следует пользоваться стандартным свичом в GNS3 и чем его заменить?
36. Почему в качестве модели персонального компьютера не следует в GNS3 пользоваться программой Virtual PC Simulator и чем её заменить?
37. Перечислите способы переноса архивов резервных копий между сетевыми лабораториями.

Порядок выполнения работы

На кафедре работаем в лаборатории 192.168.14.56. Дома работаем в лаборатории labs.mikrotik.com.ua. При отсутствии проблем с сетью университета лаборатория labs.mikrotik.com.ua доступна из кафедры по адресу 192.168.3.253.

1. Установите RouterOS Mikrotik (с. 4)
2. Подсоединиться к RouterOS, запущенной из qemu с помощью командной строки (с. 7) с помощью VNC, telnet (2 способа), ssh, winbox, ftp и http.
3. Соедините два маршрутизатора, используя протокол UDP (с. 8). Добейтесь связи между маршрутизаторами по протоколу IP и проверьте её с помощью утилиты ping.
4. Настройте GNS3 согласно своего номера
5. Создайте и запустите в GNS3 топологию first (Рис. 5). Назначьте адреса. Подсоединиться с помощью winbox к обоим роутерам. Проверьте в winbox связь между роутерами по протоколу IP (tools/ping).
6. Создайте шаблон для топологий template (с. 16). Покажите доступность из Ubuntu роутеров топологии по протоколу ssh. Создайте закладки (Рис. 7.)
7. В лаборатории 192.168.14.56 (labs.mikrotik.com.ua, 192.168.3.253) сделайте резервную копию топологии first и восстановите её под другим именем в лаборатории labs.mikrotik.com.ua, 192.168.3.253 (192.168.14.56). В случае проблем с сетью (недоступности из кафедры адреса 192.168.3.253) восстановить резервную копию под другим именем на том же компьютере 192.168.14.56.

Для сдачи работы необходимо

1. Уметь отвечать на контрольные вопросы.
2. Показать умение создавать резервную копию топологии first и восстанавливаться из неё
3. Продемонстрировать умение подсоединяться к роутерам топологии first с помощью VNC, telnet (2 способа), ssh, winbox, ftp и http.
4. Уметь в маршрутизаторе Mikrotik назначить, изменить и удалить IP-адрес.
5. На момент сдачи работы топологии first и template должны присутствовать в двух сетевых лабораториях 192.168.14.56 и labs.mikrotik.com.ua (192.168.3.253).

