

Семинар #10: Основы работы с сетями в Linux. Практика.

Как сдавать задачи

Для сдачи ДЗ вам нужно создать репозиторий на GitLab (если он ещё не создан) под названием `devtools-homework`. Структура репозитория должна иметь вид:

```
├── seminar10_network/  
│   ├── 01.sh  
│   ├── 02.sh  
│   └── ...  
└── ...
```

Для каждой задачи, если в самой задаче не сказано иное, нужно создать 1 скрипт с расширением `.sh` и шебангом в начале скрипта. Если задача делится на подзадачи нужно, если в самой задаче не сказано иное, создать скрипт для каждой подзадачи. Названия файлов решений для всех задач/подзадач должны начинаться с номера задачи, например `01.sh` или `04b.sh`, даже если в условии задачи используется другое имя для скрипта.

Если в задаче встречается вопрос, то на этот вопрос нужно ответить в комментариях (начинаются с `#`) скрипта.

Задача 1. Подсети

Для каждой из подсетей:

- a) 192.168.1.0/24
- b) 10.0.0.0/8
- c) 172.16.0.0/12

Определите:

- Диапазон адресов
- Маску
- Количество хостов
- Адрес сети
- Широковещательный адрес

Задача 2. Просмотр информации о сетевом интерфейсе

Используйте команду `ip address` (команду можно сокращать, то есть использовать `ip addr` или даже `ip a`), чтобы просмотреть информацию о сетевых интерфейсах. Для каждого из интерфейсов определите:

- Название интерфейса
- IP-адрес (IPv4 и/или IPv6)
- Маска подсети
- MAC-адрес
- Состояние интерфейса (UP/DOWN)
- Широковещательный адрес
- MTU (максимальный размер пакета)

Задача 3. Просмотр информации о маршрутах

Используйте команду `ip route` (команду можно сокращать, то есть использовать `ip r`), чтобы просмотреть информацию о маршрутах. Для каждого из маршрутов определите:

- Целевая сеть
- Шлюз
- Интерфейс, через который идёт маршрут
- Протокол, как был получен маршрут — вручную или через DHCP
- Источник

Задача 4. Пинг

Используйте команду `ping`, чтобы проверить доступность адресов:

- (a) Адрес `1.1.1.1`
- (b) Адрес `google.com`. Пошлите 20 пакетов по 1000 байт с интервалом 0.2 секунды на этот адрес.

Задача 5. traceroute

Используйте команду `traceroute`, чтобы определить адреса маршрутизаторов, через которые проходят пакеты на пути к адресу `8.8.8.8`.

! На виртуальной машине с NAT это может не работать и отображать один хоп или отображать все хопы звёздочками. Если у вас не работает, то закройте виртуальную машину и в настройках виртуальной машины, во вкладке "Сеть" установите "Тип подключения" – "Сетевой мост".

Задача 6. Основы работы с curl

Используйте `curl`, чтобы загрузить веб-страницу `google.com`. Сохраните эту веб-страницу в файл с расширением `.html`. Откройте этот файл с помощью браузера.

Задача 7. tcpdump

Запустите `tcpdump`:

```
$ sudo tcpdump -vvv
```

или

```
$ sudo tcpdump -i имяИнтерфейса -vvv
```

и проверьте, что он отображает при:

- (a) Использовании команды `ping`
- (b) Использовании `curl` для загрузки веб-страницы.
- (c) При переходе на веб-страницу в браузере.

Задача 8. wireshark

Установите программу `wireshark` и откройте её:

```
$ sudo wireshark
```

Используйте её для анализа трафика, проходящего через основной сетевой интерфейс. Проанализируйте, какие пакеты отправляются и приходят при следующих операциях:

- (a) Использовании команды `ping`
- (b) Использовании `curl` для загрузки веб-страницы.
- (c) При переходе на веб-страницу в браузере.
- (d) При загрузке YouTube видео в браузере.

Связь между двумя ВМ

Для решения следующих задач вам понадобится ещё одна виртуальная машина. Нужно будет установить связь между виртуальными машинами. Однако по умолчанию они запускается изолировано и не видят друг друга. Чтобы машины видели друг друга сделайте следующее:

1. Закройте все виртуальные машины.
2. В VirtualBox зайдите в менеджер сетей (Файл -> Инструменты -> Сеть).
3. В менеджере сетей перейдите во вкладку "Сети NAT" и создайте новую сеть.
4. Убедитесь, что стоит галочка на "Включить DHCP".
5. Продублируйте вашу ВМ или создайте новую ВМ. У вас должно быть две виртуальные машины.
6. Зайдите в опции самих виртуальных машин и в настройках сети установите "Тип подключения" на значение "Сеть NAT". Выберите сеть NAT, созданную ранее. Это нужно сделать для обеих ВМ.
7. Сначала запустите одну ВМ и дождитесь полной загрузки ВМ. Проверьте, что сеть на этой ВМ работает.
8. После этого запустите вторую ВМ и дождитесь её полной загрузки. Проверьте, что сеть на этой ВМ работает.
9. Выполните команду `ip address` на двух виртуальных машинах. IP адреса виртуальных машин должны различаться. Если это не так, то измените ip адрес одной из машин с помощью команды `ip`.

Задача 9. Пинг другой виртуальной машины

Используйте команду `ping`, для пинга другой виртуальной машины.

Задача 10. Простой чат

Используйте команду `nc`, чтобы создать простой чат между двумя виртуальными машинами.

Задача 11. Подключение по SSH

Создайте SSH подключение между виртуальными машинами. Для этого на одной из машин понадобится установить и запустить SSH-демон. Зайдите по SSH из одной машины на другую и создайте любой файл. Проверьте, что файл на другой машине действительно был создан.

Задача 12. Получение файлов с помощью scp

Создайте файл на одной ВМ. Используйте `scp`, чтобы скачать этот файл на другую ВМ. Скачайте также любую директорию из одной ВМ на другую.

Задача 13. Веб сервер

Запустите веб-сервер `nginx` на одной из машин. После этого зайдите на него через другую виртуальную машину.

Задача 14. Веб страница

Измените страницу веб сервера на простую страницу с сообщением "Hello from other VM". Скорей всего для этого будет достаточно поменять страницу в `/var/www/html`. Зайдите на эту страницу с другой ВМ.

Задача 15. Веб страница - curl

Используйте `curl`, чтобы получить веб-страницу из другой ВМ.