
Консольные утилиты настройки сетевых компонентов Linux

Цель работы: получить практические навыки по конфигурированию сетевых интерфейсов (на примере протокола IPv4) в операционных системах Linux, ознакомиться с утилитами командной строки, освоить современные сетевые менеджеры Linux.

Необходимо:

- Установленная на компьютере среда виртуализации **ORACLE Virtual Box**
- Две виртуальные машины Linux Debian

Краткие теоретические сведения:

Несмотря на то, что в состав современных операционных систем входят утилиты конфигурирования сети с графическим интерфейсом, задачи по диагностике и настройке сети удобнее решать с помощью консольных утилит.

Linux – UNIX-подобная, многозадачная операционная система. Основным для нее является текстовый интерфейс, хотя для Linux разработаны (или портированы) графические оболочки, такие как KDE или Gnome.

Краткую справку по каждой команде можно получить с помощью команды `man`, краткую с помощью ключа `-h` (`--help`). Например: `man ifconfig`. Также полезными для получения справки могут оказаться команды `apropos` и `whatis`.

Сетевые интерфейсы в Linux именуются (для сетей Ethernet) `ethN`, где `N` — номер сетевого адаптера начиная с нуля (для первого адаптера — `eth0`).

В Linux, несмотря на то что в разных дистрибутивах методы хранения конфигурационной информации разнятся, утилиты настройки сети идентичны:

- `ifconfig` – отображение настроек и конфигурирование сети,
- `route` – управление таблицей маршрутизации (и, соответственно, настройками шлюза по умолчанию).
- настройки DNS хранятся в текстовом файле `/etc/resolv.conf`

Современным средствами конфигурирования сетевого интерфейса является утилита `ip`. Утилита `ip` объединяет в себе возможности традиционных утилит `ifconfig`, `arp`, `route`. Она поставляется во всех дистрибутивах.

Для автоматического получения адресов используется консольная утилита `dhclient`.

Все изменения, выполняемые с помощью этих утилит, изменяют только текущую конфигурацию. Для постоянного изменения конфигурации требуется изменить конфигурационные файлы, которые используются сетевой службой для их конфигурирования при запуске.

В семействе Linux RedHat эти файлы называются `ifcfg-ethX` (где X номер интерфейса) и располагаются в каталоге `/etc/sysconfig/network-scripts/`. В семействе Linux Debian - `/etc/network/interfaces`.

Параметры файлов применяются при старте системы, перезапуска сетевых интерфейсов или службы сети. Перезапустить службу сети можно по команде `service network restart`.

Исторически разные канальные интерфейсы и виртуальные интерфейсы настраивались в разных конфигурационных файлах. Для удобства и унификации управления разработали используют менеджеры сетей.

По умолчанию настройки сети в Linux семейства RedHat контролирует `NetworkManager` – специальная служба в Linux, упрощающая работу с сетью. Параллельно с ней работает служба `network`. Параметры `NetworkManager` можно менять через утилиту `nmcli`. Управление параметрами через `NetworkManager` доступно через TUI утилиту `nmtui`.

В семействе Linux Debian часто используют менеджер `netplan` (пакет `netplan.io`), который хранит конфигурацию в YAML файлах (например

/etc/netplan/01-network-manager-all.yaml). Управляется netplan, что, конечно, не удивительно, через утилиту netplan.

Для получения информации о сетевых устройствах служат утилиты lshw или lspci, а для конфигурирования параметров канального уровня – утилита ethtool.

Основными параметрами настройки сетевых интерфейсов являются:

- **IP-адрес**
- **Маска подсети**
- **Gateway** (шлюз по умолчанию)
- **DNS-сервер**

IP-адрес (сокращение от англ. *Internet Protocol Address*) — уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP. Имеет длину 4 байта.

В терминологии сетей TCP/IP **маской подсети** или **маской сети** называется битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети.

Шлюз по умолчанию (*Default gateway*), *шлюз последней надежды* (*Last hope gateway*) — в маршрутизируемых протоколах — адрес маршрутизатора, на который отправляется трафик, для которого невозможно определить маршрут исходя из таблиц маршрутизации.

DNS (*Domain Name System* — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства).

Все эти параметры можно настраивать вручную или при помощи специальной службы.

DHCP (*Dynamic Host Configuration Protocol* — протокол динамической конфигурации узла) — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP, запрашивая эти параметры с DHCP сервера.

Порядок выполнения работы:

Для выполнения работы рекомендуется виртуальные машины Linux Debian 12 или 13. Рекомендуется гипервизор VirtulaBox 7.

В работе используются два **сценария** конфигурации:

- 1) Статическая адресация IP=10.100.0.2, MASK=255.255.255.0, GATE=10.100.0.1, DNS = 8.8.8.8
- 2) Динамическая – все параметры получаются автоматически с dhcp сервера.

Часть 0. Подготовка инфраструктуры.

1. Создайте связанный клон вашей виртуальной машины. Не забудьте при клонировании сегрегировать новый MAC адрес
2. Создайте снимки виртуальных машин
3. Переименуйте виртуальные машины как **s1** и **s2**. Запустите виртуальные машины и задайте такие же имена хостов (компьютеров) в Linux
4. В Virtual Box настройте сетевые интерфейсы в режим **«Сеть NAT»**
5. Создайте снимки машин (давайте снимкам понятные названия, возможно вам придется откатываться к ним).

Часть 1. Работа с утилитами командной строки Linux

Запустите виртуальную машину **s1** и авторизоваться в системе под администраторской учётной записью.

1. В Virtual Box настройте сетевой интерфейс в режим «Сеть NAT»
2. С помощью команд lshw, ethtool и ip, узнайте:
 - a. модель сетевой карты, канальную скорость, режим работы (Full или Half Duplex). Наличие физического подключения (линка), MAC адрес
 - b. информацию о текущей конфигурации IPv4 (ip, mask, gate, dns).

3. Не внося изменения в системные конфигурационные файлы (кроме `resolv.conf`) и не используя менеджеры сети с помощью команд `ip` и `dhclient` настройте сетевой интерфейс сначала по сценарию #1, а затем по сценарию #2.
4. Перезапустите систему (`systemctl reboot`). Проверьте, что стало с настройками сети. Обдумайте результат.
5. Реализуйте настройку по сценариям #1 и #2, редактируя конфигурационный файл интерфейса. Сохраните результаты настройки в разных копиях конфигурационного файла (например `interfaces.static` и `interfaces.dhcp`).

Часть 2. Работа с интерфейсами Linux через Network Manager

1. Создайте снимок состояния машины **s2** с именем "before network-manager"
2. Внутри машины установите на сетевом интерфейсе получение адресов автоматически
3. Установите пакет `network-manager`
4. Закомментируйте символом `#` строки настройки реального сетевого интерфейса (**НЕ lo**) в файле `interfaces`, перезапустите службу сети.
5. Проверьте статус работы службы Network Manager (`systemctl status NetworkManager`) и если все ОК...
6. Создайте новое подключение с именем `din-con` (например, с DHCP на интерфейсе) и активируйте подключение
7. Выведите информацию о состоянии коннекции и сетевого интерфейса
8. Добавьте в виртуальную машину еще один сетевой адаптер (в свойства VM). На виртуальной машине настройте эмуляцию сети для нового адаптера в режим «**Внутренняя сеть**».
9. Используя консольную утилиту Network Manager `nmcli` новый сетевой интерфейс на статическую адресацию, назначив **ДВА** адреса:

10.100.0.3/24 и 10.100.0.4/24.

10. Выведите на консоль информацию обо всех адресах.

Часть 3. ARP

1. Добавьте в виртуальную машину **s1** еще один сетевой адаптер (в свойства VM). На виртуальной машине настройте эмуляцию сети для нового адаптера в режим **«Внутренняя сеть»**, причем внутренняя сеть должна быть та же, что и в п. 8, части 2
2. С помощью конфигурационного файла настройте первый адаптер на получение адресов автоматически, а добавленный выше на адрес 10.100.0.5/24
3. Выведите на консоль и в файл arp1 кэш arp с помощью команды ip
4. Используя утилиту ping, напишите команды проверки доступности узлов 10.100.0.3 и 10.100.0.4, пошлав по 5 запросов на каждый хост
5. Выведите на консоль и в файл arp2 кэш arp с помощью команды ip
6. Сравните файлы arp1 и arp2. В чем разница?

Содержание отчета

Требуется подготовить отчеты в формате DOC\DOCX или PDF. Отчет содержит титульный лист, артефакты выполнения и ответы на вопросы и задания.

Артефакты:

1. Скрипт Часть 1 п. 2
2. Файлы intrefaces.static и interfaces.dhcp из Части 1 п.5.
3. Команды Части 2 п. 6, 7
4. Команды и консольный вывод Части 2 п. 10
5. Файлы arp1 и arp2 из Части 1 п.5.

Вопросы и задания:

1. Как с помощью команды `ip`:
 - a. назначить новый IPv4 адрес?
 - b. назначить новый MAC адрес?
 - c. назначить новый gateway?
 - d. вывести информацию arp кэше?
 - e. очистить arp кэш?
 - f. включить интерфейс?
 - g. выключить интерфейс?
2. Как с помощью `nmcli` назначить на интерфейс статический IP адрес, маску и настроить default gateway?
3. Какой, по-вашему, практический смысл в возможности назначения нескольких IP адресов на один интерфейс?
4. Чем отличались файлы `arp1` и `arp2`. Почему?

Понятийный минимум по работе

1. IP адрес
2. Gate
3. Адрес DNS
4. DHCP клиент
5. DHCP сервер
6. Режим работы (duplex)
7. Linux
 - a. Имя интерфейса (`enp0s3` или `eth0` и т.п.)
 - b. `ifconfig` (получение информации о адресах)
 - c. `ip` (контекст `address`, `link`, `route`)
 - d. `dhclient`

- e. `/etc/resolv.conf`
- f. Nmcli (включение, выключение, назначение адресов статически, динамически)

Полезные ссылки

<https://losst.pro/nastrojka-seti-v-linux>

Отчет выслать в течение 4-х недель на адрес edu-net@yandex.ru.

В теме письма: №группы ФИО (латинскими буквами) №работы (например: 5555 Fedor Sumkin 1)

Поддержка работы

Дополнительные материалы по теме курса публикуются на Telegram-канале ITSMDao (t.me/itsmdao). Обсуждать работу и задавать вопросы можно в чате ITSMDaoChat (t.me/itsmdaochat).