

---

# 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 с Linux CentOS и 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` (что выходит за рамки этой практической работы), но, чтобы `NetworkManager` не вмешивался в конфигурирование адаптера можно или остановить эту службу, или в файле конфигурации адаптера указать ключ `NM_CONTROLLED=no`. Но лучше будет свериться с документацией

вашего дистрибутива Linux. Управление параметрами через NetworkManager доступно через TUI утилиту nmtui. NetworkManager хранит конфигурацию в `/etc/NetworkManager/NetworkManager.conf`.

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 CentOS 7 и Linux Debian 11. Рекомендуется гипервизор VirtualBox 7.

В работе используются две 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 1. Linux (CentOS или Debian 11)

1. Запустить виртуальную машину и авторизоваться в системе под администраторской учётной записью.
2. Создать скрипт, который позволяет пользователю (!):
  - a. Узнать модель сетевой карты, канальную скорость, режим работы (Full или Half Duplex). Наличие физического подключения (линк), MAC адрес
  - b. Получить информацию о текущей конфигурации IPv4 (ip, mask, gate, dns).
  - c. Настроить сетевой интерфейс по сценарию #1
  - d. Настроить сетевой интерфейс по сценарию #2.
  - e. закрыть скрипт.
3. Скрипт не должен вносить изменения в системные конфигурационные файлы. Нельзя использовать менеджеры сети.

### Часть 2. Работа с виртуальными интерфейсами Linux CentOS через Network Manager

1. На виртуальной машине CentOS настройте эмуляцию сетевого интерфейса в Virtual Box на режим Внутренняя сеть.

2. Используя консольную утилиту Network Manager nmcli (!):
  - a. Настройте сетевой интерфейс по сценарию №1.
  - b. создайте виртуальный сетевой интерфейс с именем br0 и IP = IP=10.100.0.3.
  - c. активируйте виртуальный интерфейс.
3. С помощью утилиты ping проверьте связь между реальным и виртуальным интерфейсом.
4. Определите MAC адрес виртуального интерфейса (!).

### Часть 3. Работа с реальными интерфейсами Linux Debian 11 через netplan

1. На виртуальной машине Debian 11 заранее убедитесь, что установлен пакет netplan
2. Настройте эмуляцию сетевого интерфейса в Virtual Box на режим Внутренняя сеть (это должна быть та же сеть, что и в части 2).
3. Отредактируйте YAML файл конфигурации netplan так чтобы сетевой интерфейс имел два адреса 10.100.0.4 и 10.100.0.5, маска 255.255.255.0, а gateway 10.100.0.3 (!).
4. С помощью утилиты ping проверьте связь между адресами 10.100.0.2-10.100.0.5.
5. Выведите таблицу arp кэша. Найдите в ней записи обо всех адресах 10.100.0.2-10.100.0.5 (!).

### Часть 4. Настройка объединения реальных сетевых интерфейсов в Linux

В этой части вы исследуете возможности модуля ядра Linux – bonding. Этот модуль позволяет объединять сетевые интерфейсы в группы, в целях обеспечения отказоустойчивости или увеличения пропускной способности.

Выполнять задания части можно на любой из использованных виртуальных машин. Рекомендуется использовать менеджеры сети (nmcli или netplan), но

можно прямо настроить конфигурационные файлы интерфейсов и (или) воспользоваться утилитой `ifenslave`.

1. Выключите выбранную вами виртуальную машину, сделайте снапшот. В конфигурацию машины добавьте сетевой интерфейс. Оба интерфейса переведите в режим «Сеть NAT» (так же возможно использовать режим «Сетевой мост», только в этом случае нужно убедиться, что в реальной сети работает автоматическое конфигурирование IP по dhcp).
2. Убедитесь, что загружен модуль `bonding`. Проверить работу модуля можно командой `lsmod`. Если модуль не загружен, запустите его командой `modprobe` (если автоматической загрузки модуля нет, то учтите, что после перезагрузки модуль надо загружать заново вручную).
3. С помощью выбранного вами средства настройте объединение сетевых карт в одну группу, так чтобы (!):
  - a. карты чередовались при передаче пакетов,
  - b. виртуальный интерфейс группы назывался `bond007`,
  - c. интерфейс `bond007` получал конфигурацию IPv4 автоматически.
4. Активируйте интерфейс группы. Выведите на консоль информацию о IP и MAC адресах всех интерфейсов.
5. Выведите на консоль файл `/proc/net/bonding/bond007`, ознакомьтесь с его содержанием (этот динамический файл позволяет отлаживать `bonding`) (!).
6. На второй консоли выведите раза содержимое файла `/proc/net/dev`, ознакомьтесь с его содержимым.
7. Напишите скрипт, который выводит на консоль (!):
  - a. Отметку текущего времени
  - b. Имя интерфейса, значение `Receive-packets`, значение `Transmit-packets`.
8. На первой консоли запустите `ping -l bond007 8.8.8.8`
9. На второй консоли с помощью вашего скрипта соберите трижды данные.

Проанализируйте результаты (!).

## 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. Команды Части 2 п. 2
3. Команды и консольный вывод Части 2 п.4
4. YAML файл Части 3 п.3
5. Команды и консольный вывод Части 3 п.5
6. Команды и конфигурационные файлы (если таковые использовались) Части 4 п.5
7. Консольный вывод Части 4 п.5
8. Скрипт Части 4 п.7

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. Как с помощью netplan назначить на интерфейс статический IP адрес,



маску и настроить default gateway?

4. Какие режимы bonding стандартно существуют в Linux? Опишите их назначение, возможности по отказоустойчивости и необходимость поддержки со стороны оборудования.
5. Какие существуют и чем отличаются режимы работы адаптера (duplex) ?
6. Какой, по-вашему, практический смысл в возможности назначения нескольких IP адресов на один интерфейс?
7. Какой, по-вашему, практический смысл в возможности создания виртуальных интерфейсов?

#### 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)
  - g. Netplan (конфигурирование, назначение адресов статически, динамически)

- h. Bonding, общие принципы, файл текущей конфигурации в /proc
- i. Файл статистики интерфейсов в системе /proc

#### Полезные ссылки

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

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html/networking\\_guide/part-managing\\_ip\\_networking](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/networking_guide/part-managing_ip_networking)

<https://netplan.readthedocs.io/en/latest/examples/#using-dhcp-and-static-addressing>

Отчет выслать в течение 4-х недель на адрес [edu-net@yandex.ru](mailto:edu-net@yandex.ru).

В теме письма: Ногруппы ФИО (латинскими буквами) Номер работы (например: 5555 Fedor Sumkin 2)

#### Поддержка работы

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