


Глава 9. Настройка сетевых параметров

9.1. Сетевой интерфейс



Сетевые интерфейсы

- Программный интерфейс между процессами в ОС и сетевым адаптером
- Для сетевых интерфейсов не создаются файлы устройств
- Создаются автоматически
- В каталоге `/sys/class/net` имеются ссылки на устройства

```
sa@debian12:~$ ls /sys/class/net
enp0s3  lo
```

Программный интерфейс между процессами в ОС и сетевым адаптером.

Сетевой адаптер может быть как физическим так и виртуальным.

При обнаружении сетевого адаптера, ядро автоматически создает для него сетевой интерфейс.

В Линукс для сетевых интерфейсов не создаются файлы устройств.

В ядрах начиная с 2.6 информация о сетевых интерфейсах находится в файловой системе `sysfs` – каталог `/sys/class/net`

Пример:

```
# ls /sys/class/net
eth0  lo  sit0
```

В классической системе именования Ethernet интерфейсы имеют следующий вид: `eth0`, `eth1`, `eth2` и т. д.

Примечание: названия других сетевых интерфейсов:

`lo` – закольцованный интерфейс (петля, `loopback`)

`tr0`, `tr1`, ... - Token Ring

`fddi0`, `fddi1`, ... FDDI

`sl0`, `sl1`, ... SLIP интерфейсы

`plip0`, `plip1`, ... сеть с использованием параллельных интерфейсов

9.2. Предсказуемые имена сетевых устройств



PNIDN

- PNIDN (Predictable Network Interface Device Names) используются в системе udev systemd.
- Примеры имен устройств:
 - Встроенные сетевые устройства: enoX (X — номер устройства)
 - Устройства PCI Express с горячим подключением: ensX (X — номер слота)
 - Имена по расположению оборудования: enpXsY (X — номер слота PCI, Y — номер устройства)
 - Именованное по MAC адресу: enx112233445566
 - Классический способ именования: ethX

В современных системах чаще используется система предсказуемых имен сетевых устройств (PNIDN - Predictable Network Interface Device Names).

Именованное устройств определяется настройками демона udev.

Если вы хотите вы можете использовать любое удобное вам имя.

Пример: Назначение имени сетевой карте в соответствии с MAC адресом.

```
[root@sl0 ~]# ifconfig myownnetname
myownnetname: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::4dbf:94b7:62c6:1b2b prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:e3:15:ae txqueuelen 1000 (Ethernet)
    RX packets 25 bytes 5288 (5.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 1650 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@sl0 ~]# cat /etc/udev/rules.d/70-persistent-net.rules
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="*", ATTR{address}
=="52:54:00:e3:15:ae", ATTR{type}=="1", KERNEL=="eth*", NAME="myownnetname"
```

В некоторых случаях, например, при использовании контейнеров LXC допускается задавать любые произвольные имена сетевых устройств в контейнерах, посредством настройки контейнера в гипервизоре.

9.3. Методы настройки сетевых параметров



Методы настройки сети

- Ручной — командами
- Сценарии настройки сети — `ifup/ifdown` + файл (файлы) конфигурации
- Служба для настройки сети (NetworkManager, conman, systemd-networkd)

В Линукс вы можете применять три метода настройки сети.

- Для временной настройки и/или тестирования написать команду, например `ip`. Такой метод не сохраняет настройки параметров при перезагрузках или изменениях в сети.
- Сценарии настройки сети. Обычно эти сценарии находятся в каталоге `/etc` и, как правило, совместно со сценариями работают команды `ifup` и `ifdown`, которые получают информацию о настройках в файле(ах) конфигурации.
- Служба для настройки сети. Например: NetworkManager, conman, systemd-networkd и ...

Преимущества применения специальной службы:

- Служба может оперативно реагировать на изменения.
- Можно предоставить возможность управления пользователям.

Важно помнить, что применение нескольких методов настройки одновременно на одной машине может приводить к непредсказуемым последствиям.

9.4. Настройка сети в Linux



Ручная настройка

- `ip` `ip [опции] объект [команда [параметры]]`
- `ifconfig` `ifconfig [имя_интерфейса] [адрес] [опции]`

В современных Linux системах используется команда `ip`, которая по своей функциональности заменяет команды `ifconfig` и `route`

Команда `ip` позволяет

- устанавливать адреса и параметры сетевым интерфейсам
- задавать маршруты
- настраивать простейший вариант NAT (Network Address Translation)
- выводить информацию о настройках

Формат команды `ip` таков

`ip [опции] объект [команда [параметры]]`

Поле объект может иметь следующие значения:

- `link` – сетевое устройство
- `addr` – адрес устройства
- `neighbour` – ARP таблица
- `route` – таблица маршрутизации
- `rule` – политика маршрутизации
- `tunnel` – туннель поверх IP
- `monitor` – мониторинг устройств

Для объекта `link` допустимы следующие команды:

- `set` – для установки параметров

Глава 9. Настройка сетевых параметров

- `show` (или `list`, или `ls`) – для вывода значений параметра

Пример:

```
[root@linux1 iproute-2.4.7]# ip addr show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:03:ff:11:5c:78 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.2/24 brd 192.168.1.255 scope global eth0
    inet 192.168.1.111/24 brd 192.168.1.255 scope global secondary eth0:1
```

В качестве параметров для объекта `link` наиболее часто используются:

- `dev` имя_интерфейса (`dev` можно опускать)
- `up/down`
- `mtu`
- `arp/arp off`

Пример:

```
[root@linux1 root]# ip link set eth0 down
[root@linux1 root]# ip link ls
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:03:ff:11:5c:78 brd ff:ff:ff:ff:ff:ff
```

Для объекта `addr` используются команды:

- `add`
- `del`
- `show`

В качестве параметров можно использовать следующие значения:

- `dev` имя_интерфейса
- `local` адрес/сетевая_маска – для задания адреса и сетевой маски
- `broadcast/brd` адрес
- `label` имя — создание виртуального интерфейса
- `peer` адрес — задание IP адреса на другом конце PPP соединения

Пример:

```
linux:~ # ip addr add dev eth0 label eth0:2 local 192.168.1.10/24
```

Устаревшее средство ручной настройки параметров TCP/IP утилита `ifconfig`

`ifconfig` [имя_интерфейса] [адрес] [опции]

Основные опции утилиты `ifconfig`

- `netmask` [адрес] – установка сетевой маски
- `[-]broadcast` [адрес] – установка или удаление широковещательного адреса
- `[-]pointopoint` [адрес] – установка IP адреса на удаленном конце PPP

Глава 9. Настройка сетевых параметров

соединения.

- `mtu` – установка MTU (в байтах)
- `[-]arp` - активация и деактивация ARP
- `up/down` – активация и деактивация интерфейса
- `tunnel [адрес]` – создание туннеля IPv6-IPv4

Команда `ifconfig -a` выводит информацию о всех, даже не сконфигурированных, сетевых интерфейсах

Для создания виртуального сетевого интерфейса после имени интерфейса нужно поставить двоеточие и номер виртуального интерфейса.

Пример :

```
[root@linux1 root]# ifconfig eth0:1 192.168.1.111
[root@linux1 root]# ifconfig eth0:1
eth0:1      Link encap:Ethernet  HWaddr 00:03:FF:11:5C:78
inet addr:192.168.1.111  Bcast:192.168.1.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
Interrupt:11  Base address:0xe000
```



Сценарии настройки сети

- `ifup` — включить интерфейс
- `ifdown` — выключить интерфейс
- Расположение постоянных параметров для сценариев
 - Red Hat `/etc/sysconfig/network-scripts/ifcfg-*`
 - Debian `/etc/network/interfaces`

Во многих Linux дистрибутивах для задания параметров сетевым интерфейсам при загрузке системы используются файлы `ifcfg-имя_интерфейса`

Примечание: В RedHat эти файлы находятся в каталоге `/etc/sysconfig/network-scripts`, в SUSE Linux – в каталоге `/etc/sysconfig/network`. Последние версии RedHat или Centos-stream уже не поддерживают такой метод настройки.

Форматы файлов `ifcfg-имя_интерфейса` в разных дистрибутивах могут различаться, но должны присутствовать параметры: `IPADDR`, `NETMASK`, `BROADCAST`

Пример:

```
[root@linux1 network-scripts]# less ifcfg-eth0
# DEC|DECchip 21140 [FasterNet]
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.1.2
NETMASK=255.255.255.0
GATEWAY=192.168.1.5
HWADDR=00:03:FF:10:6F:3A
```

Для активации и деактивации сетевых интерфейсов часто используют скрипты (иногда утилиты) `ifup` и `ifdown`, которые читают информацию из файлов `ifcfg-имя_интерфейса`

В Дебиан подобных дистрибутивах настройка описывается в файле `/etc/network/interfaces`

Пример:

```
root@sl:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*
```

Глава 9. Настройка сетевых параметров

```
# The loopback network interface
auto lo
iface lo inet loopback

auto enp0s3
iface enp0s3 inet manual

auto enp0s3.100
iface enp0s3.100 inet static
address 10.0.100.100/24
gateway 10.0.100.1
vlan-raw-device enp0s3
```

В файле описывается как активировать и настраивать интерфейсы.

Отдельно от `/etc/network/interfaces` настраивается разрешение имен, в файле `/etc/resolv.conf` или с помощью службы `systemd-resolved`.

NetworkManager

- Сервис NetworkManager — автоматизированная настройка сетевых интерфейсов
- Полезен на рабочих станциях
- В графических средах имеются удобные приложения для управления соединениями NetworkManager
- Для управления соединениями NetworkManager из командной строки удобно использовать утилиту `nmcli`

Во многих дистрибутивах Linux используется специальная служба для управления сетевыми соединениями - NetworkManager.

Особенно полезна эта служба при работе с мобильными устройствами, в которых очень часто может меняться сеть, к которой подключено устройство.

Если в системе установлена графическая оболочка, то как правило имеются специальные утилиты для настройки сетевых подключений.

Для управления подключениями без графической среды можно воспользоваться утилитой `nmcli`.

Пример: Подключение и отключение сетевых устройств.

```
[root@vlesk-nb ~]# nmcli dev status
УСТРОЙСТВО  ТИП      СОСТОЯНИЕ  СОЕДИНЕНИЕ
enp5s0      ethernet  подключено  LAN
wlp3s0      wifi      отключено   --
...
[root@vlesk-nb ~]# nmcli dev disconnect enp5s0
Выполнено: устройство «enp5s0» отключено.
[root@vlesk-nb ~]# nmcli dev status
УСТРОЙСТВО  ТИП      СОСТОЯНИЕ  СОЕДИНЕНИЕ
enp5s0      ethernet  отключено   --
wlp3s0      wifi      отключено   --
...
[root@vlesk-nb ~]# nmcli dev connect enp5s0
Устройство «enp5s0» успешно активировано с помощью «c913e7e4-fae7-4450-82b3-01e1ad35dcde».
[root@vlesk-nb ~]# nmcli dev status
УСТРОЙСТВО  ТИП      СОСТОЯНИЕ  СОЕДИНЕНИЕ
enp5s0      ethernet  подключено  LAN
wlp3s0      wifi      отключено   --
```

Пример: Мониторинг подключений

Глава 9. Настройка сетевых параметров

```
[root@vlesk-nb ~]# nmcli monitor
enp5s0: недоступен <-- Устройство отключили от сети
«WIFI» не является основным подключением.
Уровень соединения: «ограничено»
NetworkManager находится в состоянии «подключен (на узле)»
enp5s0: отключено <-- Устройство вновь подключили к сети
enp5s0: подключение (подготовка)
enp5s0: использует соединение «LAN».
enp5s0: подключение (настройка)
enp5s0: подключение (получение конфигурации IP)
enp5s0: подключение (проверка IP-подключения)
enp5s0: подключение (запуск второстепенных соединений)
enp5s0: подключено
«LAN» не является основным подключением.
Уровень соединения: «полностью»
NetworkManager находится в состоянии «подключено»
```



systemd-networkd

- Является частью systemd
- Основная команда управления - `networkctl`
- Автоматически создает интерфейсы при появлении устройств в системе, также имеется возможность создавать виртуальные устройства
- Конфигурационные файлы находятся в `/lib/systemd/network` и `/etc/systemd/network`
- Каталог временных файлов — `/run/systemd/network/`

`systemd-networkd` — системный демон для управления сетевыми настройками. Его задачей является обнаружение и настройка сетевых устройств по мере их появления, а также создание виртуальных сетевых устройств.

Файлы настроек могут находиться в следующих каталогах (в порядке увеличения приоритета):

1. `/usr/lib/systemd/network/` — системный сетевой каталог.
2. `/run/systemd/network/` — runtime-каталог сетевых программ.
3. `/etc/systemd/network/` — локальный каталог системного администрирования.

Существует три типа конфигурационных файлов:

1. `.network` — описывают параметры сети: IP, маршруты, DNS и другие;
2. `.link` — описывают физические параметры каждого интерфейса: имя, MAC, MTU и т.д.;
3. `.netdev` — описывают виртуальные интерфейсы, мосты.

Для этих файлов действуют следующие правила:

1. Профиль активируется, если совпали все условия в разделе `[Match]`.
2. Если в разделе `[Match]` не указано ничего, то профиль применяется для всех устройств (можно сравнить с подстановочным символом `*`).
3. Файлы настроек сортируются по названиям и обрабатываются в лексическом порядке, независимо от каталога, в котором они расположены.
4. Файлы с одинаковыми названиями заменяют друг друга — в порядке приоритета каталогов, в которых они находятся.

Глава 9. Настройка сетевых параметров

Примеры настроек:

Проводной интерфейс с DHCP

```
/etc/systemd/network/20-wired.network
```

```
[Match]
```

```
Name=enp1s0
```

```
[Network]
```

```
DHCP=yes
```

Проводной интерфейс со статическим IP-адресом

```
/etc/systemd/network/20-wired.network
```

```
[Match]
```

```
Name=enp1s0
```

```
[Network]
```

```
Address=10.1.10.9/24
```

```
Gateway=10.1.10.1
```

```
DNS=10.1.10.1
```