

Руководство администратора операционной системы Скейлер версии 1.0.

Москва 2023

Оглавление

Аннотация	6
Просмотр информации о системе	7
Настройка базовых компонентов системы	8
Установка языка системы	8
Отображение текущего состояния языка	8
Список доступных языков	8
Установка языковых предпочтений	9
Настройка раскладки клавиатуры	9
Отображение текущих настроек	9
Список доступных раскладок клавиатуры	10
Настройка раскладки клавиатуры	11
Установка даты и времени	11
Использование команды <code>timedatectl</code>	12
Отображение текущей даты и времени	12
Синхронизация системных часов с удаленным сервером	12
Изменение текущей даты	13
Изменение текущего времени	13
Изменение часового пояса	14
Использование команды <code>date</code>	15
Отображение текущей даты и времени	15
Изменение текущего времени	16
Изменение текущей даты	17
Использование команды <code>hwclock</code>	17
Часы реального времени и системные часы	17
Отображение текущей даты и времени	17
Установка даты и времени	18
Установка <code>kdump</code>	18
Установка зарезервированной памяти для <code>kdump</code>	18
Формат параметров зарезервированной памяти для <code>kdump</code>	18
Рекомендуемая зарезервированная память	19
Отключение сетевых драйверов	20
Настройка сети	21
Настройка IP-адреса	21
Использование команды <code>nmcli</code>	21
Введение в <code>nmcli</code>	21
Управление устройствами	22
Подключение к устройству	22
Отключение от устройства	22
Настройка сетевых подключений	23

Настройка динамических IP-подключений	23
Настройка IP-адресов	23
Активация подключения сетевого устройства и проверка состояния подключения	24
Настройка статических IP-подключений	24
Настройка IP-адресов	24
Активация сетевого подключения и проверка состояния подключения устройства	25
Добавление подключения Wi-Fi	26
Изменение атрибутов	27
Настройка статического маршрута	28
Использование команды ip	28
Настройка IP-адресов	28
Настройка статического IP-адреса	28
Настройка нескольких IP-адресов	29
Настройка статического маршрута	29
Настройка сети с помощью файла ifcfg	30
Настройка статической адресации сети	30
Настройка динамической адресации сети	31
Конфигурация сетевого шлюза по умолчанию	32
Настройка имени хоста	32
Введение	32
Настройка имени хоста с помощью команды hostnamectl	33
Просмотр всех имен хостов	33
Задание всех имен хостов	33
Задание конкретного имени хоста	33
Сброс определенного имени хоста на значение по умолчанию	33
Удаленное изменение имени хоста	34
Настройка имени хоста с помощью команды nmcli	34
Настройка объединения сетевых интерфейсов	34
Выполнение команды nmcli	35
Настройка объединения сетевых интерфейсов с помощью командной строки	36
Проверка установки модуля ядра объединения интерфейсов	36
Создание интерфейса объединения каналов	37
Создание ведомого интерфейса	37
Активация объединения каналов	38
Создание нескольких объединяющих связей	39
Отличия IPv6 от IPv4	40
Ограничения	40
Описание конфигурации	40
Настройка MTU устройства интерфейса	40
Обзор	40

Настройка MTU устройства интерфейса	40
Автонастройка IPv6-адреса с отслеживанием состояния	41
Обзор	41
Установка класса поставщика для dhclient при автонастройке IPv6-адреса с отслеживанием состояния	42
Ядро с поддержкой системных вызовов, связанных с сокетами	44
Обзор	44
Различия между адресом link-local и глобальным адресом во время вызова сокета	44
Настройка устойчивости процесса управляющей программы dhclient для IPv4-адресов	45
Обзор	46
Ограничения	46
Различия в конфигурации между IPv4 DHCP и IPv6 DHCPv6	46
Различия между конфигурациями IPv4 и IPv6 при использовании команды iproute	47
Обзор	47
Жизненный цикл IPv6-адреса	47
Команда ip link	48
Команда ip addr	49
Command ip route	50
Команда ip rule	51
Различия в конфигурации сервиса NetworkManager	51
Обзор	51
Различия в конфигурации	51
Часто задаваемые вопросы	54
iscsi-initiator-utils не поддерживает IPv6-адрес fe80	54
Симптом	54
Возможная причина	54
Потеря IPv6-адреса после отключения сетевой карты	55
Симптом	55
Возможная причина	55
Долгое добавление или удаление IPv6-адреса для интерфейса объединения с несколькими IPv6-адресами	55
Симптом	55
Возможная причина	55
Решение	56
Задержка при передаче журнала rsyslog в сценарии, где используются как IPv4-адреса, так и IPv6-адреса	56
Симптом	56
Возможная причина	56
Решение	56
Управление сервисами	57

Введение в systemd	57
Компоненты systemd	57
Возможности	59
Быстрая активация	60
Активация по требованию	60
Управление жизненным циклом сервисов с помощью Sgroups	60
Управление точками монтирования и автоматического монтирования	61
Управление транзакционными зависимостями	61
Совместимость со скриптами SysVinit	61
Моментальные снимки и восстановление состояния системы	62
Управление системными сервисами	62
Сравнение команд SysVinit и systemd	62
Перечисление сервисов	64
Отображение состояния сервисов	65
Запуск сервиса	68
Остановка сервиса	68
Перезапуск сервиса	69
Включение сервиса в автозагрузку	69
Изменение уровня выполнения	69
Уровни выполнения и цели	69
Просмотр цели запуска по умолчанию	71
Просмотр всех целей запуска	71
Изменение цели по умолчанию	71
Изменение текущей цели	71
Переход в режим восстановления	71
Переход в аварийный режим	72
Команда systemctl	73
Завершение работы операционной системы	73
Перезагрузка операционной системы	74
Приостановка операционной системы	74
Гибернация операционной системы	74

Аннотация

Данный документ содержит описание базовых операций администрирования системы Скейлер ОС (Scaler OS) и призван помочь системным администраторам эффективнее использовать данную систему. Документ предназначен для всех администраторов, а также инженеров группы эксплуатации и сопровождения продукта, которые используют Скейлер ОС (Scaler OS).

Просмотр информации о системе

- Вы можете просмотреть информацию о системе.

```
$ cat /etc/os-release
```

Например, команда и выходные данные имеют следующий вид.

```
# cat /etc/os-release  
  
NAME="scaler"  
  
VERSION="1.0"  
  
ID="scaler"  
  
VERSION_ID="1.0"  
  
PRETTY_NAME="scaler 1.0"  
  
ANSI_COLOR="0;31"
```

- Просмотрите информацию о системных ресурсах.

Выполните следующую команду, чтобы просмотреть информацию о ЦП:

```
# lscpu
```

Выполните следующую команду, чтобы просмотреть информацию о памяти:

```
$ free
```

Выполните следующую команду, чтобы просмотреть информацию о диске:

```
$ fdisk -l
```

- Просмотр информации о системных ресурсах в режиме реального времени.

```
$ top
```

Настройка базовых компонентов системы

Установка языка системы

Настройки языка системы хранятся в файле `/etc/locale.conf` и могут быть изменены командой `localectl`. Эти настройки считываются демоном `systemd` при загрузке системы.

Отображение текущего состояния языка

Чтобы отобразить текущее состояние языковых настроек, выполните следующую команду:

```
$ localectl status
```

Пример вывода команды:

```
# localectl status

System Locale: LANG=ru_RU.UTF-8
               VC Keymap: us
               X11 Layout: us,ru
               X11 Variant: ,
               X11 Options: grp:alt_shift_toggle
```

Список доступных языков

Чтобы отобразить доступные языки, выполните следующую команду:

```
$ localectl list-locales
```

Вы можете проверить это, перечислив все варианты русского языка с помощью следующей команды:

```
localectl list-locales | grep ru
```

```
ru_RU.UTF-8
```

```
ru_UA.UTF-8
```


Установка языковых предпочтений

Чтобы настроить языковую среду, выполните от имени пользователя **root** следующую команду. В этой команде *locale* указывает тип языка, который необходимо установить. Запустите команду **localectl list-locales**, чтобы получить список доступных языков и кодировок которые могут быть использованы в системе. Измените значение необходимым образом.

```
# localectl set-locale LANG=locale
```

Например, чтобы использовать “Русский язык в кодировке UTF-8”, выполните от имени пользователя **root** следующую команду:

```
# localectl set-locale LANG=ru_RU.UTF-8
```



ПРИМЕЧАНИЕ.

Внеся изменение, войдите в систему снова или выполните команду **source /etc/locale.conf** от имени пользователя **root**, чтобы обновить файл конфигурации, и тогда изменения вступят в силу.

Настройка раскладки клавиатуры

Настройки раскладки клавиатуры хранятся в файле **/etc/locale.conf** и могут быть изменены командой **localectl**. Эти настройки считываются на раннем этапе загрузки демоном **systemd**.

Отображение текущих настроек

Чтобы отобразить текущие настройки раскладки клавиатуры, выполните следующую команду:

```
$ localectl status
```

Пример вывода команды:

```
# localectl status

System Locale: LANG=ru_RU.UTF-8
               VC Keymap: us
               X11 Layout: us,ru
               X11 Variant: ,
               X11 Options: grp:alt_shift_toggle
```

Список доступных раскладок клавиатуры

Чтобы получить список всех доступных раскладок клавиатуры, которые можно настроить в Scaler OS, выполните следующую команду:

```
$ localectl list-keymaps
```

Например, команда вывода раскладок для русского языка выглядит так:

```
# localectl list-keymaps | grep ru
```

```
cz-rus
```

```
dvorak-ru
```

```
ge-ru
```

```
ng-yoruba
```

```
ru
```

```
ru-cp1251
```

```
ru-cv_latin
```

```
ru-ms
```

```
ru-yawerty
```

```
ru1
```

```
ru2
```

```
ru3
```

```
ru4
```

```
ru_win
```

```
ruwin_alt-CP1251
```

```
ruwin_alt-KOI8-R
```

```
ruwin_alt-UTF-8
```

```
ruwin_alt_sh-UTF-8
```

```
ruwin_cplk-CP1251
```

```
ruwin_cplk-KOI8-R
```

```
ruwin_cplk-UTF-8
```

`ruwin_ct_sh-CP1251`

`ruwin_ct_sh-KOI8-R`

`ruwin_ct_sh-UTF-8`

`ruwin_ctrl-CP1251`

`ruwin_ctrl-KOI8-R`

`ruwin_ctrl-UTF-8`

`sunt5-ru`

Настройка раскладки клавиатуры

Чтобы задать раскладку клавиатуры, выполните следующую команду от имени пользователя **root**. В этой команде *map* указывает раскладку клавиатуры, которую необходимо установить. Чтобы получить диапазон возможных значений, выполните команду **localectl list-keymaps**. Измените значение необходимым образом.

```
$ localectl set-keymap map
```

Раскладка клавиатуры будет аналогичным образом применяться к графическим пользовательским интерфейсам.

После этого можно проверить, применена ли установка, узнав состояние:

```
# localectl status

System Locale: LANG=ru_RU.UTF-8
                VC Keymap: us
                X11 Layout: us,ru
                X11 Variant: ,
                X11 Options: grp:alt_shift_toggle
```

Установка даты и времени

В этом разделе описано задание системной даты, времени и часового пояса с помощью команд **timedatectl**, **date** и **hwclock**.

Использование команды `timedatectl`

Отображение текущей даты и времени

Чтобы отобразить текущую дату и время, выполните следующую команду:

```
$ timedatectl
```

Пример вывода команды:

```
# timedatectl

          Local time: Пт 2023-04-28 10:33:01 MSK
    Universal time: Пт 2023-04-28 07:33:01 UTC
           RTC time: Пт 2023-04-28 07:33:01
        Time zone: Europe/Moscow (MSK, +0300)

System clock synchronized: yes
           NTP service: active
        RTC in local TZ: no
```

Синхронизация системных часов с удаленным сервером

Ваши системные часы могут автоматически синхронизироваться с удаленным сервером, используя протокол NTP (Network Time Protocol — протокол сетевого времени). Для включения или отключения NTP выполните от имени пользователя **root** следующую команду. Логическое значение (*boolean*) равно **yes** (да) или **no** (нет), что характеризует включение или отключение NTP для автоматической синхронизации системных часов. Измените значение необходимым образом.

ПРИМЕЧАНИЕ.

Если на удаленном NTP-сервере включена автоматическая синхронизация системных часов, вы не сможете изменить дату и время вручную. Если вам необходимо вручную изменить дату или время, убедитесь, что автоматическая синхронизация системных часов с помощью NTP отключена. Отключить службу NTP можно командой **`timedatectl set-ntp no`**.

```
# timedatectl set-ntp boolean
```

Например, чтобы включить автоматическую удаленную синхронизацию времени, выполните следующую команду:

```
# timedatectl set-ntp yes
```

Изменение текущей даты



ПРИМЕЧАНИЕ.

Перед изменением даты убедитесь, что автоматическая синхронизация системных часов с помощью NTP отключена.

Для изменения текущей даты выполните от имени пользователя **root** следующую команду. В этой команде *YYYY* указывает год, *MM* — месяц, а *DD* — день. Измените эти значения необходимым образом.

```
# timedatectl set-time YYYY-MM-DD
```

Например, чтобы изменить текущую дату на 28 апреля 2023 г., от имени пользователя **root** выполните следующую команду:

```
# timedatectl set-time '2023-04-28'
```

Изменение текущего времени



ПРИМЕЧАНИЕ.

Перед изменением времени убедитесь, что автоматическая синхронизация системных часов с помощью NTP отключена.

Чтобы изменить текущее время, выполните от имени пользователя **root** следующую команду. В этой команде *HH* указывает час, *MM* — минуты, а *SS* — секунды. Измените эти значения необходимым образом.

```
# timedatectl set-time HH:MM:SS
```

Например, чтобы изменить текущее время на 15:57:24, выполните следующую команду:

```
# timedatectl set-time 15:57:24
```

Изменение часового пояса

Чтобы вывести список всех доступных часовых поясов, выполните следующую команду:

```
$ timedatectl list-timezones
```

Чтобы изменить текущий часовой пояс, выполните от имени пользователя **root** следующую команду. В этой команде *time_zone* указывает устанавливаемый часовой пояс. Измените значение необходимым образом.

```
# timedatectl set-timezone time_zone
```

Предположим, вы находитесь в Европе и хотите определить, какой часовой пояс к вам ближе всего. Для этого можно вывести список всех доступных часовых поясов в Европе, используя следующую команду:

```
# timedatectl list-timezones | grep Europe
```

```
Europe/Amsterdam
```

```
Europe/Andorra
```

```
Europe/Astrakhan
```

```
Europe/Athens
```

```
Europe/Belfast
```

```
Europe/Belgrade
```

```
Europe/Berlin
```

```
Europe/Bratislava
```

```
Europe/Brussels
```

```
Europe/Bucharest
```

```
Europe/Budapest
```

```
Europe/Busingen
```

```
Europe/Chisinau
```

```
.....
```

Чтобы изменить часовой пояс на Europe/Moscow (Европа/Москва), выполните следующую команду:

```
# timedatectl set-timezone Europe/Moscow
```

Использование команды date

Отображение текущей даты и времени

Чтобы отобразить текущую дату и время, выполните следующую команду:

```
$ date
```

По умолчанию команда **date** отображает местное время. Чтобы отобразить время в формате всемирного координированного времени (UTC), запустите команду с параметром командной строки **-utc** или **-u**:

```
$ date --utc
```

Вы также можете настроить формат отображаемых сведений, указав в командной строке параметр **+"format"**:

```
$ date +"format"
```

Таблица 1. Параметры форматирования

Параметр формата	Описание
%H	Час в формате HH (например, 17).
%M	Минуты в формате MM (например, 37).
%S	Секунды в формате SS (например, 25).
%d	День месяца в формате DD (например, 15).
%m	Месяц в формате MM (например, 07).
%Y	Год в формате YYYY (например, 2023).

%Z	Аббревиатура часового пояса (например, CEST).
%F	Полная дата в формате YYYY-MM-DD (например, 2023-04-28). Этот параметр эквивалентен %Y-%m-%d.
%T	Полное время в формате HH:MM:SS (например, 18:30:25). Этот параметр эквивалентен %H:%M:%S.

Примеры команд и выходных данных:

- Отображение текущей даты и времени:

```
# date
```

```
Пт 28 апр 2023 10:41:32 MSK
```

- Отображение текущей даты и времени в формате UTC:

```
# date --utc
```

```
Пт 28 апр 2023 07:44:07 UTC
```

- Настройка вывода команды date:

```
$ date +"%Y-%m-%d %H:%M"
```

```
2023-04-28 17:24
```

Изменение текущего времени

Чтобы изменить текущее время, запустите команду **date** с параметром **-set** или **-s** от имени пользователя **root**. В этой команде *HH* указывает час, *MM* — минуты, а *SS* — секунды. Измените эти значения необходимым образом.

```
# date --set HH:MM:SS
```

По умолчанию команда **date** устанавливает местное время. Чтобы вместо этого установить системные часы в формате UTC, запустите команду с параметром командной строки **-utc** или **-u**:

```
# date --set HH:MM:SS --utc
```


Например, чтобы изменить текущее время на 23:26:00, выполните от имени пользователя **root** следующую команду:

```
# date --set 23:26:00
```

Изменение текущей даты

Чтобы изменить текущую дату, запустите команду с параметром командной строки **-set** или **-s** от имени пользователя **root**. В этой команде *YYYY* указывает год, *MM* — месяц, а *DD* — день. Измените эти значения необходимым образом.

```
# date --set YYYY-MM-DD
```

Например, чтобы изменить текущую дату на 28 апреля 2023 г., от имени пользователя **root** выполните следующую команду:

```
# date --set 2023-04-28
```

Использование команды hwclock

Команда **hwclock** позволяет установить часы реального времени (RTC).

Часы реального времени и системные часы

Операционная система Linux выделяет следующие типы часов.

- Системные часы: часы текущего ядра Linux.
- Аппаратные RTC-часы: аппаратные часы материнской платы с питанием от батареи. Эти часы можно установить в BIOS с помощью параметра **Standard BIOS Feature** (Стандартный компонент BIOS).

При запуске Linux считывает RTC и устанавливает время системных часов на основе времени RTC.

Отображение текущей даты и времени

Чтобы отобразить текущую дату и время RTC, выполните следующую команду от имени пользователя **root**:

```
# hwclock
```

Пример вывода команды:

```
# hwclock
```

```
2023-04-28 11:31:13.609433+03:00
```

Установка даты и времени

Чтобы изменить дату и время на текущей операционной системе, запустите от имени пользователя **root** следующую команду. В этой команде *dd* указывает день, *mm* — месяц, *yyyy* — год, *HH* — час, а *MM* — минуты. Измените эти значения необходимым образом.

```
# hwclock --set --date "dd mm yyyy HH:MM"
```

Например, чтобы изменить текущее время на 21:17 и дату на 28 апреля 2023г., выполните следующую команду:

```
# hwclock --set --date "28 Apr 2023 21:17" --utc
```

Установка kdump

В этом разделе описана установка зарезервированной памяти для kdump и изменение параметров в файле конфигурации kdump.

Установка зарезервированной памяти для kdump

Формат параметров зарезервированной памяти для kdump

Память, зарезервированная для kdump, должна быть добавлена к загрузочным аргументам в файле **/boot/efi/EFI/openEuler/grub.cfg** (режим загрузки UEFI) или **/boot/grub2/grub.cfg** (традиционный режим загрузки). Зарезервированная память для kdump была добавлена в выпуски Scaler OS по умолчанию и может быть изменена по мере необходимости. После добавления или изменения загрузочных аргументов перезапустите систему, чтобы настройки вступили в силу. Параметры памяти, зарезервированной для kdump, имеют следующий формат:

Загрузочный аргумент	Описание	Значение по умолчанию	Примечания
crashkernel=x	Если размер физической памяти меньше 4 ГБ, то для kdump резервируется объем памяти x.	Значение по умолчанию — 512 МБ для x86.	Данный метод конфигурации используется, только если доступный размер памяти меньше 4 ГБ. В этом случае убедитесь, что доступно достаточно последовательных блоков памяти для резервирования.

Загрузочный аргумент	Описание	Значение по умолчанию	Примечания
crashkernel=x@y	Для kdump резервируется объем памяти x по начальному адресу y.	Не используется.	Убедитесь, что объем памяти x по начальному адресу y не зарезервирован для других модулей.
crashkernel=x,high	Если размер физической памяти меньше 4 ГБ, резервируется память объемом 256 МБ. Если размер физической памяти больше 4 ГБ, для kdump резервируется объем памяти x.	Значение по умолчанию – 1024M,high для ARM64.	Если размер памяти меньше 4 ГБ, то убедитесь, что физический размер доступных последовательных блоков памяти составляет как минимум 256 МБ, а если больше 4 ГБ, – как минимум x. Фактический объем зарезервированной памяти составит 256 МБ + x.
crashkernel=x,low crashkernel=y,high	Если размер физической памяти меньше 4 ГБ, для kdump резервируется объем памяти x, а если больше 4 ГБ – объем памяти y.	Не используется.	Если физический размер памяти меньше 4 ГБ, то убедитесь, что физический размер доступных последовательных блоков памяти составляет как минимум x, а если больше 4 ГБ – как минимум y.

Рекомендуемая зарезервированная память

Рекомендуемое решение	Параметр резервирования	Описание
Общее решение	crashkernel=2048M,high	Если размер памяти меньше 4 ГБ, для kdump резервируется 256 МБ. Если размер памяти больше 4 ГБ, для kdump резервируется 2048 МБ. Итого: 256 + 2048 МБ.
Экономичное решение	crashkernel=1024M,high	Если размер памяти меньше 4 ГБ, для kdump резервируется 256 МБ. Если размер памяти больше 4 ГБ, для kdump резервируется 1024 МБ. Итого: 256 + 1024 МБ. В случаях, когда размер системной памяти меньше 512 ГБ, не рекомендуется создавать дампы файлов kdump по сети. В сценариях с виртуальными машинами зарезервированную память можно уменьшить. Рекомендуется установить для crashkernel значение 512M или 256M, high.

ПРИМЕЧАНИЕ.

Если дампы файлов `kdump` по сети не создаются, необходимо настроить файловую систему `kdump` без упаковки сетевых драйверов. Для загрузки сетевого драйвера требуется большой объем памяти. В результате зарезервированной для `kdump` памяти может оказаться недостаточно, и в `kdump` возможен сбой. В связи с этим сетевые драйверы рекомендуется отключить.

Отключение сетевых драйверов

В файле конфигурации `kdump` `/etc/kdump.conf` можно использовать параметры `dracut` для установки специализированного драйверного модуля. Вы можете настроить сетевой драйвер на специальный список драйверов, чтобы загрузка драйвера файловой системой `kdump` не производилась. После изменения файла конфигурации перезапустите службу `kdump`, чтобы изменения вступили в силу. Установите параметры `dracut` следующим образом:

```
dracut_args --omit-drivers "mdio-gpi usb_8dev et1011c rt2x00usb bcm-phy-lib
mac80211_hwsim rtl8723be rndis_host hns3_cae amd vrf rtl8192cu mt76x02-lib int51x1
ppp_deflate team_mode_loadbalance smsc911x aweth bonding mwifiex_usb hnae dnet
rt2x00pci vaser_pci hdlc_ppp marvell rtl8xxxu mlxsw_i2c ath9k_htc rtl8150 smc91x cortina
at803x rockchip cxgb4 spi_ks8995 mt76x2u smsc9420 mdio-cavium bnxt_en ch9200
dummy macsec ice mt7601u rtl8188ee ixgbev net1080 liquidio_vf be2net mlxsw_switchx2
gl620a xilinx_gmii2rgmii ppp_generic rtl8192de sja1000_platform ath10k_core
cc770_platform realte igb c_can_platform c_can ethoc dm9601 smsc95xx lg-vl600 ifb enic
ath9 mdio-octeon ppp_mppe ath10k_pci cc770 team_mode_activebackup marvell10g hinic
rt2x00lib mlx4_en iavf broadcom igc c_can_pci alx rtl8192se rtl8723ae microchip lan78xx
atl1c rtl8192c-common almia ax88179_178a qed netxen_nic brcmsmac rt2800usb e1000
qla3xxx mdio-bitbang qsemi mdio-mscc-miim plx_pci ipvlan r8152 cx82310_eth slhc
mt76x02-usb ems_pci xen-netfront usbnet pppoe mlxsw_minimal mlxsw_spectrum cdc_ncm
rt2800lib rtl_usb hnae3 ath9k_common ath9k_hw catc mt76 hns_enet_drv ppp_async
huawei_cdc_ncm i40e rtl8192ce dl2 qmi_wwan mii peak_usb plusb can-dev slcan amd-xgbe
team_mode_roundrobin ste10Xp thunder_xcv pptp thunder_bgx ixgbe davicom icplus tap tun
smc75xx smc dlsi hns_dsaf mlxsw_core rt2800mmi softing uPD60620 vaser_usb
dp83867 brcmfmac mwifiex_pcie mlx4_core micrel team macvlan bnx2 virtio_net rtl_pci
zaurus hns_mdi libcxgb hv_netvsc nicvf mt76x0u teranetics mlxfw cdc_eem qcom-emas
pppox mt76-usb sierra_net i40evf bcm87xx mwifiex pegasus rt2x00mmi sja1000 ena
hclgevf cnic cxgb4vf ppp_synctty iwlmvm team_mode_broadcast vxlan vsockmon hdlc_cisc
rtl8723-common bsd_comp fakelb dp83822 dp83tc811 cicada fm10 8139t sfc hs geneve
```

```
hclge xgene-enet-v2 cdc_mbim hdhc asix netdevsim rt2800pci team_mode_random lxt
ems_usb mlxsw_pci sr9700 mdio-thunder mlxsw_switchib macvtap atlantic cdc_ether
mcs7830 nicpf mdi peak_pci atl1e cdc_subset iptap btcoexist mt76x0-common veth slip
iwldvm bcm7xxx vitesse netconsole epic100 myri10ge r8169 qede microchip_t1 liquid
bnx2x brcmutil mwifiex_sdi mlx5_core rtlwifi vmxnet3 nlmon hns3 hdhc_raw esd_usb2 atl2
mt76x2-common iwlwifi mdio-bcm-unimac national ath rtwpci rtw88 nfp rtl8821ae fjes
thunderbolt-net 8139cp atl1 mscv vcan dp83848 dp83640 hdhc_fr e1000e ipheth net_failover
aquantia rtl8192ee igbvf rocker intel-xway tg3" --omit "ramdisk network ifcfg qemu-net"
--install "chmod" --nofscks
```

Настройка сети

Настройка IP-адреса

Использование команды nmcli



ПРИМЕЧАНИЕ.

Конфигурация сети, настроенная с помощью команды **nmcli**, вступает в силу немедленно и не будет потеряна после перезагрузки системы.

Введение в nmcli

nmcli (NetworkManager Command Line Interface) — это утилита командной строки для настройки сети с помощью NetworkManager. Для работы с **nmcli** используется следующий основной формат:

```
nmcli [OPTIONS] OBJECT { COMMAND | help }
```

В предыдущей команде вместо **OBJECT** можно указать один из следующих параметров: **general**, **networking**, **radio**, **connection** и **device**. Вместо **OPTIONS** можно указать необязательные параметры, такие как **-t**, **-terse** (для обработки скриптов), **-p**, **-pretty** (для вывода в понятном для человека формате), **-h** и **-help**. Для получения дополнительной информации выполните команду **nmcli help**.

```
$ nmcli help
```

Основные команды перечислены ниже.

- Чтобы отобразить общее состояние NetworkManager, выполните следующую команду:

```
$ nmcli general status
```
- Чтобы отобразить все подключения, выполните следующую команду:

```
$ nmcli connection show
```
- Чтобы отобразить только текущие активные подключения, добавьте параметр **-a** или **-active** следующим образом:

```
$ nmcli connection show --active
```
- Чтобы отобразить устройство, идентифицированное NetworkManager, и состояние его подключения, выполните следующую команду:

```
$ nmcli device status
```
- Например, чтобы запустить или остановить сетевые интерфейсы, выполните команды nmcli от имени пользователя **root**:

```
# nmcli connection up id enp3s0  
# nmcli device disconnect enp3s0
```

Управление устройствами

Подключение к устройству

Выполните следующую команду, чтобы подключить NetworkManager к соответствующему сетевому устройству. Попробуйте найти подходящую конфигурацию подключения и активировать ее.

“\$IFNAME” - замените на имя интерфейса требуемой сетевой карты

```
$nmcli device connect "$IFNAME"
```

Если соответствующая конфигурация подключения не существует, NetworkManager создает и активирует файл конфигурации с использованием настроек по умолчанию.

Отключение от устройства

Выполните следующую команду, чтобы отключить NetworkManager от сетевого устройства и предотвратить автоматическую активацию устройства.

```
$nmcli device disconnect "$IFNAME"
```

Настройка сетевых подключений

Выполните следующую команду, чтобы отобразить все доступные сетевые подключения:

```
$ nmcli con show
```

```
NAME UUID                TYPE DEVICE
enp4s0 5afce939-400e-42fd-91ee-55ff5b65deab ethernet enp4s0
enp3s0 c88d7b69-f529-35ca-81ab-aa729ac542fd ethernet enp3s0
virbr0 ba552da6-f014-49e3-91fa-ec9c388864fa bridge virbr0
```



ПРИМЕЧАНИЕ.

Значение **NAME** в выходных данных команды указывает идентификатор (имя) подключения.

После добавления сетевого подключения создается соответствующий файл конфигурации, который связывается с соответствующим устройством. Чтобы проверить доступные устройства, выполните следующую команду:

```
$ nmcli dev status
```

```
DEVICE    TYPE STATE      CONNECTION
enp3s0    ethernet connected enp3s0
enp4s0    ethernet connected enp4s0
virbr0    bridge connected virbr0
lo        loopback unmanaged --
virbr0-nic tun      unmanaged --
```

Настройка динамических IP-подключений

Настройка IP-адресов

Если для предоставления IP адреса сети используется протокол DHCP, выполните следующую команду, чтобы добавить файл конфигурации сети:

```
nmcli connection add type ethernet con-name connection-name ifname interface-name
```

Например, чтобы создать файл конфигурации динамического подключения с именем **net-test**, выполните следующую команду от имени пользователя **root**:

```
# nmcli connection add type ethernet con-name net-test ifname enp3s0
Connection 'net-test' (a771baa0-5064-4296-ac40-5dc8973967ab) successfully added.
```

NetworkManager устанавливает для **connection.autoconnect** значение **yes** и сохраняет параметр в файле **/etc/sysconfig/network-scripts/ifcfg-net-test**. В файле **/etc/sysconfig/network-scripts/ifcfg-net-test** для параметра **ONBOOT** установлено значение **yes**.

Активация подключения сетевого устройства и проверка состояния подключения

Выполните следующую команду от имени пользователя **root**, чтобы активировать сетевое подключение:

```
# nmcli con up net-test
Connection successfully activated (D-Bus active
path:/org/freedesktop/NetworkManager/ActiveConnection/5)
```

Выполните следующую команду, чтобы проверить состояние подключения устройств:

```
$ nmcli device status
```

DEVICE	TYPE	STATE	CONNECTION
enp4s0	ethernet	connected	enp4s0
enp3s0	ethernet	connected	net-test
virbr0	bridge	connected	virbr0
lo	loopback	unmanaged	--
virbr0-nic	tun	unmanaged	--

Настройка статических IP-подключений

Настройка IP-адресов

Чтобы добавить статическое сетевое подключение IPv4, выполните следующую команду:

```
nmcli connection add type ethernet con-name connection-name ifname interface-name ip4
address gw4 address
```

 **ПРИМЕЧАНИЕ.**

Чтобы добавить IPv6-адрес и соответствующую информацию о шлюзе, используйте параметры **ip6** и **gw6**.

Например, чтобы создать файл конфигурации статического подключения с именем **net-static**, выполните следующую команду от имени пользователя **root**:

```
# nmcli con add type ethernet con-name net-static ifname enp3s0 ip4 192.168.0.10/24 gw4 192.168.0.254
```

Вы также можете указать IPv6-адрес и шлюз для устройства. Ниже приведен пример.

```
# nmcli con add type ethernet con-name test-lab ifname enp3s0 ip4 192.168.0.10/24 gw4 192.168.0.254 ip6 abbe::**** gw6 2001:***:*
Connection 'net-static' (63aa2036-8665-f54d-9a92-c3035bad03f7) successfully added.
```

NetworkManager устанавливает значение **manual** для внутреннего параметра **ipv4.method**, значение **yes** для параметра **connection.autoconnect** и записывает эту настройку в файл **/etc/sysconfig/network-scripts/ifcfg-my-office**. В файле для **BOOTPROTO** установлено значение **none**, а для **ONBOOT** — значение **yes**.

Выполните следующую команду от имени пользователя **root**, чтобы задать IPv4-адреса двух DNS-серверов:

```
# nmcli con mod net-static ipv4.dns "*.*.*.*.*.*)"

```

Выполните следующую команду от имени пользователя **root**, чтобы задать IPv6-адреса двух DNS-серверов:

```
# nmcli con mod net-static ipv6.dns "2001:4860:4860::**** 2001:4860:4860::****"
```

Активация сетевого подключения и проверка состояния подключения устройства

Выполните следующую команду от имени пользователя **root**, чтобы активировать сетевое подключение:

```
# nmcli con up net-static ifname enp3s0
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
```

Выполните следующую команду, чтобы проверить состояние подключения устройств:

\$ nmcli device status

```
DEVICE      TYPE STATE      CONNECTION
enp4s0      ethernet connected enp4s0
enp3s0      ethernet connected net-static
virbr0      bridge connected virbr0
lo          loopback unmanaged --
virbr0-nic  tun      unmanaged --
```

Выполните следующую команду, чтобы просмотреть сведения о подключении (используйте параметры **-p** и **-pretty**, чтобы добавить заголовок и сегмент в выходные данные):

\$ nmcli -p con show net-static

```
=====
====
Connection profile details (net-static )
=====
====
connection.id:          net-static
connection.uuid:        b9f18801-6084-4aee-af28-c8f0598ff5e1
connection.stable-id:   --
connection.type:        802-3-ethernet
connection.interface-name: enp3s0
connection.autoconnect: yes
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (default)
connection.multi-connect: 0 (default)
connection.auth-retries: -1
connection.timestamp:   1578988781
connection.read-only:   no
connection.permissions: --
connection.zone:        --
connection.master:      --
connection.slave-type:   --
connection.autoconnect-slaves: -1 (default)
connection.secondaries:  --
connection.gateway-ping-timeout: 0
connection.metered:     unknown
connection.lldp:         default
connection.mdns:        -1 (default)
connection.llmnr:       -1 (default)
```

Добавление подключения Wi-Fi

Вы можете добавить подключение Wi-Fi одним из следующих способов.

Способ 1. Подключение к сети Wi-Fi с использованием сетевого порта.

Подключитесь к сети Wi-Fi, указанной с помощью SSID или BSSID. Выполните следующую команду, чтобы найти подходящее подключение или создать его, а затем активируйте это подключение на устройстве.

```
$ nmcli device wifi connect "$SSID" password "$PASSWORD" ifname "$IFNAME"
$ nmcli --ask device wifi connect "$SSID"
```

Способ 2. Подключение к сети Wi-Fi с помощью файла конфигурации.

1. Выполните следующую команду, чтобы проверить доступные точки доступа Wi-Fi:

```
$ nmcli dev wifi list
```

2. Выполните следующую команду, чтобы создать конфигурацию статического IP-адреса, которая разрешает подключения Wi-Fi, автоматически выделяемые DNS:

```
$ nmcli con add con-name Wifi ifname wlan0 type wifi ssid MyWifi ip4 192.168.100.101/24
gw4 192.168.100.1
```

3. Выполните следующую команду, чтобы установить пароль WPA2, например **answer**:

```
$ nmcli con modify Wifi wifi-sec.key-mgmt wpa-psk
$ nmcli con modify Wifi wifi-sec.psk answer
```

4. Выполните следующую команду, чтобы изменить состояние Wi-Fi:

```
$ nmcli radio wifi [ on | off ]
```

Изменение атрибутов

Выполните следующую команду, чтобы проверить определенный атрибут, например mtu:

```
$ nmcli connection show id 'Wifi ' | grep mtu
802-11-wireless.mtu: auto
```

Выполните следующую команду, чтобы изменить атрибут:

```
$ nmcli connection modify id 'Wifi ' 802-11-wireless.mtu 1350
```

Выполните следующую команду, чтобы подтвердить изменение:

```
$ nmcli connection show id 'Wifi ' | grep mtu
802-11-wireless.mtu: 1350
```

Настройка статического маршрута

- Выполните команду `nmcli`, чтобы настроить статический маршрут для сетевого подключения:

```
$ nmcli connection modify enp3s0 +ipv4.routes "192.168.122.0/24 10.10.10.1"
```

- Выполните следующую команду, чтобы настроить статический маршрут с помощью редактора:

```
$ nmcli con edit type ethernet con-name enp3s0
===| nmcli interactive connection editor |===
Adding a new '802-3-ethernet' connection
Type 'help' or '?' for available commands.
Type 'describe [<setting>.<prop>]' for detailed property description.
You may edit the following settings: connection, 802-3-ethernet (ethernet),
802-1x, ipv4, ipv6, dcb
nmcli> set ipv4.routes 192.168.122.0/24 10.10.10.1
nmcli>
nmcli> save persistent
Saving the connection with 'autoconnect=yes'. That might result in an immediate
activation of the connection.
Do you still want to save? [yes] yes
Connection 'enp3s0' (1464ddb4-102a-4e79-874a-0a42e15cc3c0) successfully
saved.
nmcli> quit
```

Использование команды `ip`



ПРИМЕЧАНИЕ.

Конфигурация сети, настроенная с помощью команды `ip`, вступает в силу немедленно, однако она будет потеряна после перезагрузки системы.

Настройка IP-адресов

Выполните команду `ip`, чтобы настроить IP-адрес для интерфейса. Команда имеет следующий формат, где *interface-name* обозначает имя сетевой карты.

```
ip addr [ add | del ] address dev interface-name
```

Настройка статического IP-адреса

Выполните следующую команду от имени пользователя **root**, чтобы настроить IP-адрес:

```
# ip address add 192.168.0.10/24 dev enp3s0
```

Выполните следующую команду от имени пользователя **root**, чтобы просмотреть результат настройки:

```
# ip addr show dev enp3s0
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:aa:ad:4a brd ff:ff:ff:ff:ff:ff
    inet 192.168.202.248/16 brd 192.168.255.255 scope global dynamic noprefixroute
enp3s0
    valid_lft 9547sec preferred_lft 9547sec
    inet 192.168.0.10/24 scope global enp3s0
    valid_lft forever preferred_lft forever
    inet6 fe80::32e8:cc22:9db2:f4d4/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Настройка нескольких IP-адресов

Команда **ip** позволяет назначить интерфейсу несколько IP-адресов. Чтобы назначить IP-адреса интерфейсу, можно выполнить команду **ip** несколько раз от имени пользователя **root**. Ниже приведен пример.

```
# ip address add 192.168.2.223/24 dev enp4s0
# ip address add 192.168.4.223/24 dev enp4s0
# ip addr
```

```
3: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:aa:da:e2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.203.12/16 brd 192.168.255.255 scope global dynamic noprefixroute
enp4s0
    valid_lft 8389sec preferred_lft 8389sec
    inet 192.168.2.223/24 scope global enp4s0
    valid_lft forever preferred_lft forever
    inet 192.168.4.223/24 scope global enp4s0
    valid_lft forever preferred_lft forever
    inet6 fe80::1eef:5e24:4b67:f07f/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Настройка статического маршрута

Чтобы добавить статический маршрут в таблицу маршрутизации, выполните команду **ip route add**. Чтобы удалить маршрут, выполните команду **ip route del**. Ниже показан общий формат команды **ip route**:

```
ip route [ add | del | change | append | replace ] destination-address
```

Чтобы отобразить текущую таблицу IP-маршрутизации, выполните команду **ip route** от имени пользователя **root**. Ниже приведен пример.

```
# ip route
```

```
default via 192.168.0.1 dev enp3s0 proto dhcp metric 100
default via 192.168.0.1 dev enp4s0 proto dhcp metric 101
192.168.0.0/16 dev enp3s0 proto kernel scope link src 192.168.202.248 metric 100
192.168.0.0/16 dev enp4s0 proto kernel scope link src 192.168.203.12 metric 101
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
```

Чтобы добавить статический маршрут к адресу хоста, выполните следующую команду от имени пользователя **root**:

```
ip route add 192.168.2.1 via 10.0.0.1 [dev interface-name]
```

В предыдущей команде **192.168.2.1** — это IP-адрес в десятично-точечной нотации, **10.0.0.1** — следующий hop, а *interface-name* — наименование сетевого интерфейса для данного маршрута. Чтобы добавить для сети статический маршрут, то есть IP-адрес, представляющий диапазон IP-адресов, выполните следующую команду от имени пользователя **root**:

```
ip route add 192.168.2.0/24 via 10.0.0.1 [dev interface-name]
```

В предыдущей команде **192.168.2.1** — это IP-адрес целевой сети, *10.0.0.1* — префикс сети, а *interface-name* — имя сетевой карты.

Настройка сети с помощью файла `ifcfg`

ПРИМЕЧАНИЕ.

Настройка сети в файле `ifcfg` не вступает в силу немедленно. После изменения файла (например, `ifcfg-enp3s0`) нужно выполнить команду `nmcli con reload; nmcli con up enp3s0` от имени пользователя **root**, чтобы перезагрузить файл конфигурации и активировать подключение для вступления изменения в силу.

Настройка статической адресации сети

Ниже сетевой интерфейс **enp4s0** используется в качестве примера для описания того, как настроить статическую сеть, изменив файл **ifcfg** от имени пользователя **root**. Файл **ifcfg-enp4s0** создается в каталоге **/etc/sysconfig/network-scripts/**. Измените следующие параметры в этом файле:

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.0.10
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp4s0static
UUID=08c3a30e-c5e2-4d7b-831f-26c3cdc29293
DEVICE=enp4s0
ONBOOT=yes
```

Настройка динамической адресации сети

Ниже сетевой интерфейс **em1** используется в качестве примера для описания того, как настроить динамическую сеть, изменив файл **ifcfg**. Файл **ifcfg-em1** создается в каталоге **/etc/sysconfig/network-scripts/**. Измените следующие параметры в этом файле:

```
DEVICE=em1
BOOTPROTO=dhcp
ONBOOT=yes
```

Чтобы настроить интерфейс для отправки разных имен хостов на DHCP-сервер, добавьте в файл **ifcfg** следующее:

```
DHCP_HOSTNAME=hostname
```

Чтобы настроить интерфейс для игнорирования маршрутов, отправляемых сервером DHCP, в целях запрета обновления сетевыми сервисами файла **/etc/resolv.conf** с использованием DNS-сервера, полученного от DHCP-сервера, добавьте в файл **ifcfg** следующее:

```
PEERDNS=no
```

Чтобы настроить интерфейс для использования определенного DNS-сервера, установите для параметра **PEERDNS** значение **no** и добавьте в файл **ifcfg** следующее:

DNS1=ip-address

DNS2=ip-address

ip-address — это IP-адрес DNS-сервера. Это позволяет сетевому сервису обновлять файл **/etc/resolv.conf**, используя указанный DNS-сервер.

Конфигурация сетевого шлюза по умолчанию

При определении шлюза по умолчанию проанализируйте файл **/etc/sysconfig/network**, а затем файл **ifcfg** и используйте последнее считанное значение **GATEWAY** в качестве маршрута по умолчанию в таблице маршрутизации.

Когда NetworkManager используется для управления хостами в динамической сетевой среде, рекомендуется настроить шлюз по умолчанию для назначения DHCP.

Настройка имени хоста

Введение

Существует три типа имен хостов: **static**, **transient** и **pretty**.

- **static**: статическое имя хоста, которое может быть задано пользователями и сохранено в файле **/etc/hostname**.
- **transient**: динамическое имя хоста, поддерживаемое ядром. Начальное значение — статическое имя хоста. Значение по умолчанию — **localhost**. Это значение можно изменить, когда работает сервер mDNS или DHCP.
- **pretty**: гибкое имя хоста, которое можно задавать в любой форме (включая специальные символы и пробелы). На статические и динамические имена хостов распространяются общие ограничения для доменных имен.

ПРИМЕЧАНИЕ.

Статические и динамические имена хостов могут содержать только буквы (a–z и A–Z), цифры (0–9), дефисы (-), символы подчеркивания (_) и точки (.). Имена хостов не могут начинаться или заканчиваться

точкой (.), а также содержать две последовательные точки (.). Имя хоста может содержать не более 64 символов.

Настройка имени хоста с помощью команды `hostnamectl`

Просмотр всех имен хостов

Выполните следующую команду, чтобы просмотреть текущее имя хоста:

```
$ hostnamectl status
```

ПРИМЕЧАНИЕ.

Если в команде не указан параметр, по умолчанию используется параметр **status**.

Задание всех имен хостов

Выполните следующую команду от имени пользователя **root**, чтобы задать все имена хостов:

```
# hostnamectl set-hostname name
```

Задание конкретного имени хоста

Выполните следующую команду от имени пользователя **root**, чтобы задать конкретное имя хоста:

```
# hostnamectl set-hostname name [option...]
```

В качестве `option` можно указать один или несколько из параметров **–pretty**, **–static** и **–transient**.

Если **–static** или **–transient** используется вместе с **–pretty**, имена хостов типа **static** или **transient** будут упрощены до имен хостов типа **pretty** с заменой пробелов дефисами (-) и удалением специальных символов.

При задании имени хоста типа **pretty** используйте кавычки, если имя хоста содержит пробелы или одиночные кавычки. Ниже приведен пример.

```
# hostnamectl set-hostname "Stephen's notebook" --pretty
```

Сброс определенного имени хоста на значение по умолчанию

Чтобы выполнить сброс конкретного имени хоста и восстановить его в формате по умолчанию, выполните следующую команду от имени пользователя **root**:

```
# hostnamectl set-hostname "" [option...]
```

В предыдущей команде "" — это пустая строка символов, а в качестве *option* можно указать один или несколько из параметров **-pretty**, **-static** и **-transient**.

Удаленное изменение имени хоста

Чтобы изменить имя хоста в удаленной системе, выполните команду **hostnamectl** с параметром **-H** или **-host** от имени пользователя **root**.

```
# hostnamectl set-hostname -H [username]@hostname new_hostname
```

В предыдущей команде *hostname* обозначает имя настраиваемого удаленного хоста, *username* — определенное пользователем имя, а *new_hostname* — новое имя хоста. Команда **hostnamectl** используется для подключения к удаленной системе по протоколу SSH.

Настройка имени хоста с помощью команды nmcli

Чтобы запросить статическое имя хоста, выполните следующую команду:

```
$ nmcli general hostname
```

Чтобы присвоить статическому хосту имя **host-server**, выполните следующую команду от имени пользователя **root**:

```
# nmcli general hostname host-server
```

Чтобы система могла обнаружить изменение статического имени хоста, выполните следующую команду от имени пользователя **root** для перезагрузки сервиса `hostnamed`:

```
# systemctl restart systemd-hostnamed
```

Настройка объединения сетевых интерфейсов

Есть несколько режимов работы:

Active-backup — режим резервирования. В этом режиме один сетевой интерфейс активный, а второй в режиме ожидания. Когда основной выходит из строя, второй принимает работу на себя. Таким образом обеспечивается стабильный доступ.

Balance-rr – пакеты отправляются последовательно. Этот режим применяется для балансировки нагрузки и отказоустойчивости.

802.3ad – при таком режиме задействуются сразу все сетевые карты. (этот режим должен поддерживаться и на коммутаторе)

Balance-xor – режим применяется для отказоустойчивости и балансировки. Одна и та же сетевая карта передает пакеты одним и тем же получателям

Broadcast – этот режим передает все пакеты на все сетевые карты.

Balance-tlb – режим адаптивной балансировки передачи. Исходящий трафик распределяется в зависимости от загруженности каждой сетевой карты (определяется скоростью загрузки). Не требует дополнительной настройки на коммутаторе. Входящий трафик приходит на текущую сетевую карту. Если она выходит из строя, то другая сетевая карта берёт себе MAC адрес вышедшей из строя карты.

Balance-alb режим адаптивной балансировки нагрузки. Включает в себя политику balance-tlb плюс осуществляет балансировку входящего трафика. Не требует дополнительной настройки на коммутаторе. Балансировка входящего трафика достигается путём ARP переговоров. Драйвер bonding перехватывает ARP ответы, отправляемые с локальных сетевых карт наружу, и переписывает MAC адрес источника на один из уникальных MAC адресов сетевой карты, участвующей в объединении. Таким образом различные пиры используют различные MAC адреса сервера. Балансировка входящего трафика распределяется последовательно (round-robin) между интерфейсами.

Выполнение команды nmcli

В качестве примера будем использовать вариант BOND в режиме Active-Passive.

- Чтобы создать объединяющую связь (также называемую объединяющим логическим интерфейсом) с именем **mybond0**, выполните следующую команду:

```
$ nmcli con add type bond con-name mybond0 ifname mybond0 mode active-backup
```

- Чтобы добавить ведомый интерфейс, выполните следующую команду:

```
$ nmcli con add type bond-slave ifname enp3s0 master mybond0
```

Чтобы добавить еще один ведомый интерфейс, повторите предыдущую команду с новым именем интерфейса:

```
$ nmcli con add type bond-slave ifname enp4s0 master mybond0
Connection 'bond-slave-enp4s0' (05e56afc-b953-41a9-b3f9-0791eb49f7d3)
successfully added.
```

- Для включения объединяющей связи выполните следующую команду, чтобы сначала включить ведомый интерфейс:

```
$ nmcli con up bond-slave-enp3s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/14)
```

```
$ nmcli con up bond-slave-enp4s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/15)
```

Затем выполните следующую команду, чтобы включить объединяющую связь:

```
$ nmcli con up mybond0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/16)
```

Настройка объединения сетевых интерфейсов с помощью командной строки

Проверка установки модуля ядра объединения интерфейсов

Модуль ядра, отвечающий за объединение интерфейсов, загружается по умолчанию. Чтобы загрузить этот модуль, выполните следующую команду от имени пользователя **root**:

```
# modprobe --first-time bonding
```

Выполните следующую команду от имени пользователя **root**, чтобы отобразить информацию о модуле:

```
# modinfo bonding
```

Для получения сведений о дополнительных командах выполните команду `modprobe -help` от имени пользователя **root**.

Создание интерфейса объединения каналов

Чтобы создать интерфейс объединения каналов, можно создать файл с именем **ifcfg-bondN** в каталоге `/etc/sysconfig/network-scripts/` от имени пользователя **root** (заменив N фактическим номером интерфейса, например 0).

Запишите соответствующее содержимое в файл конфигурации в соответствии с типом объединяемого интерфейса, например сетевой интерфейс. Пример файла конфигурации интерфейса выглядит следующим образом.

```
DEVICE=bond0  
NAME=bond0  
TYPE=Bond  
BONDING_MASTER=yes  
IPADDR=192.168.1.1  
PREFIX=24  
ONBOOT=yes  
BOOTPROTO=none  
BONDING_OPTS="bonding parameters separated by spaces"
```

Создание ведомого интерфейса

После создания интерфейса объединения каналов необходимо добавить инструкции **MASTER** и **SLAVE** в файл конфигурации ведомого интерфейса.

Например, для объединения двух сетевых интерфейсов `enp3s0` и `enp4s0` в канальном режиме файлы конфигурации имеют следующий вид.

```
TYPE=Ethernet  
NAME=bond-slave-enp3s0  
UUID=3b7601d1-b373-4fdf-a996-9d267d1cac40  
DEVICE=enp3s0  
ONBOOT=yes  
MASTER=bond0  
SLAVE=yes
```

```
TYPE=Ethernet  
NAME=bond-slave-enp4s0  
UUID=00f0482c-824f-478f-9479-abf947f01c4a
```

```
DEVICE=enp4s0
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

Активация объединения каналов

Чтобы активировать объединение каналов, необходимо включить все ведомые интерфейсы. Выполните следующую команду от имени пользователя **root**:

```
# ifup enp3s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/7)
```

```
# ifup enp4s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/8)
```



ПРИМЕЧАНИЕ.

Если интерфейс находится в состоянии **up**, выполните команду **ifdown enp3s0**, чтобы изменить состояние на **down**. В этой команде **enp3s0** обозначает фактическое имя сетевой карты.

После этого включите все ведомые интерфейсы, чтобы включить объединение (не устанавливайте их в состояние **Down**).

Чтобы NetworkManager мог обнаруживать изменения, внесенные системой, выполните следующую команду от имени пользователя **root** после каждого изменения:

```
# nmcli con load /etc/sysconfig/network-scripts/ifcfg-device
```

Выполните следующую команду от имени пользователя **root**, чтобы проверить состояние объединенного интерфейса:

```
# ip link show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
```

```
mode DEFAULT group default qlen 1000
    link/ether 52:54:00:aa:ad:4a brd ff:ff:ff:ff:ff:ff
3: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
    link/ether 52:54:00:aa:da:e2 brd ff:ff:ff:ff:ff:ff
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
mode DEFAULT group default qlen 1000
    link/ether 86:a1:10:fb:ef:07 brd ff:ff:ff:ff:ff:ff
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN
mode DEFAULT group default qlen 1000
    link/ether 52:54:00:29:35:4c brd ff:ff:ff:ff:ff:ff
```

Создание нескольких объединяющих связей

Система создает интерфейс объединения каналов для каждой объединяющей связи, включая инструкцию **BONDING_OPTS**. Такой метод настройки позволяет нескольким объединенным устройствам использовать разные конфигурации. Выполните следующие операции, чтобы создать несколько интерфейсов объединения каналов:

- Создайте несколько файлов **ifcfg-bondN**, содержащих инструкцию **BONDING_OPTS**, чтобы сетевые скрипты могли создавать требуемые интерфейсы объединения.
- Создайте или отредактируйте существующий файл конфигурации интерфейса, который необходимо объединить, и добавьте инструкцию **SLAVE**.
- Используйте инструкцию **MASTER**, чтобы назначить интерфейс, который необходимо объединить, т. е. ведомый интерфейс, интерфейсу объединения каналов.

Ниже приведен пример файла конфигурации интерфейса объединения каналов.

```
DEVICE=bondN
NAME=bondN
TYPE=Bond
BONDING_MASTER=yes
IPADDR=192.168.1.1
PREFIX=24
ONBOOT=yes
BOOTPROTO=none
BONDING_OPTS="bonding parameters separated by spaces"
```

В этом примере замените N номером объединенного интерфейса. Например, для создания двух интерфейсов нужно создать два файла конфигурации **ifcfg-bond0** и **ifcfg-bond1** с правильными IP-адресами.

Отличия IPv6 от IPv4

Ограничения

- `chrony` поддерживает глобальные адреса, но не адреса, предназначенные только для коммуникаций в пределах одного сегмента местной сети или магистральной линии (адрес `link-local`).
- `Firefox` поддерживает доступ к глобальному адресу по протоколу HTTP или HTTPS, но не поддерживает доступ к адресу `link-local`.

Описание конфигурации

Настройка MTU устройства интерфейса

Обзор

В сценарии IPv6 минимальное значение MTU всего пути маршрутизации используется в качестве значения PMTU текущего канала. На основе значения PMTU сторона-источник определяет, следует ли фрагментировать пакеты. Другим устройствам на всем пути не требуется фрагментировать пакеты. Это снижает нагрузку на промежуточные устройства маршрутизации. Минимальное значение PMTU IPv6 — 1280.

Настройка MTU устройства интерфейса

Если для MTU интерфейса с настроенным IPv6-адресом установлено значение меньше **1280** (минимальное значение PMTU IPv6), IPv6-адрес интерфейса будет удален, а добавить его снова будет невозможно. Поэтому в сценариях IPv6 значение MTU устройства интерфейса должно быть больше или равно 1280. Выполните следующие команды от имени пользователя **root**, чтобы просмотреть подробные сведения.

```
# ip addr show enp3s0
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
    inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic
enp3s0
    valid_lft 38663sec preferred_lft 38663sec
    inet6 2001:222::2/64 scope global
    valid_lft forever preferred_lft forever
```



```
# ip link set dev enp3s0 mtu 1200
# ip addr show enp3s0
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1200 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
    inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic
enp3s0
    valid_lft 38642sec preferred_lft 38642sec

# ip addr add 2001:222::2/64 dev enp3s0
RTNETLINK answers: No buffer space available

# ip link set dev enp3s0 mtu 1500
# ip addr show enp3s0
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
    inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic
enp3s0
    valid_lft 38538sec preferred_lft 38538sec

# ip addr add 2001:222::2/64 dev enp3s0
# ip addr show enp3s0
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
    inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic
enp3s0
    valid_lft 38531sec preferred_lft 38531sec
    inet6 2001:222::2/64 scope global
    valid_lft forever preferred_lft forever
```

Автонастройка IPv6-адреса с отслеживанием состояния

Обзор

Адреса IPv6 и IPv4 можно получить через протокол DHCP в качестве пользователя **root**. Существуют методы настройки IPv6-адреса: автонастройка без отслеживания состояния и автонастройка с отслеживанием состояния.

- Автонастройка без отслеживания состояния

DHCP-сервер для управления не требуется. Устройство получает префикс сети в соответствии с объявлением маршрутизатора (Router Advertisement, RA), либо префиксу адреса link-local назначается фиксированное значение fe80::. Идентификатор

интерфейса получается автоматически на основе значения IPV6_ADDR_GEN_MODE в файле ifcfg.

1. Если IPV6_ADDR_GEN_MODE имеет значение stable-privacy, устройство определяет случайный идентификатор интерфейса на основе сведений об устройстве и сетевой среде.
 2. Если IPV6_ADDR_GEN_MODE имеет значение EUI64, устройство определяет идентификатор интерфейса на основе MAC-адреса устройства.
- Автонастройка с отслеживанием состояния: DHCP-сервер управляет IPv6-адресами и арендует их на DHCPv6-сервере по протоколу DHCPv6.

При автоматической настройке с отслеживанием состояния DHCPv6-сервер может классифицировать клиентов по классу поставщика, настроенному на клиентах, и назначать разным типам клиентов IPv6-адреса из разных сегментов. В сценариях IPv4 клиент может использовать параметр -V команды dhclient, чтобы задать поле vendor-class-identifier. DHCP-сервер классифицирует клиентов на основе поля vendor-class-identifier в файле конфигурации. Если тот же метод применяется для классификации клиентов в сценариях IPv6, такая классификация не работает.

```
dhclient -6 <interface> -V <vendor-class-identifier string> <interface>
```

Это связано с тем, что DHCPv6 сильно отличается от DHCP. vendor-class-option в DHCPv6 заменяет vendor-class-identifier в DHCP. Однако для параметра -V команды dhclient нельзя задать vendor-class-option.

Установка класса поставщика для dhclient при автонастройке IPv6-адреса с отслеживанием состояния

- Добавьте на клиенте параметр класса поставщика с помощью файла конфигурации.

Файл конфигурации клиента (/etc/dhcp/dhclient6.conf):
расположение файла можно настроить. Вам нужно указать файл конфигурации, используя параметр -cf команды dhclient.

```
option dhcp6.vendor-class code 16 = {integer 32, integer 16, string};
interface "enp3s0" {
    send dhcp6.vendor-class <Enterprise-ID number> <vendor class string length>
```

```
<vendor class string>;
}
```

ПРИМЕЧАНИЕ.

- \<Enterprise-ID number\>: 32-значное целое число, указывающее идентификатор предприятия. Предприятие регистрируется через IANA.
- \<vendor class string length\>: 16-значное целое число, указывающее длину строки класса поставщика.
- \<vendor class string\>: строка символов устанавливаемого класса поставщика, например HWHW.

На клиенте:

```
dhclient -6 <interface> -cf /etc/dhcp/dhclient6.conf
```

- Файл конфигурации DHCPv6-сервера (/etc/dhcp/dhcpd6.conf) нужно указать с помощью параметра -cf команды dhcpd.

```
option dhcp6.vendor-class code 16 = {integer 32, integer 16, string};
subnet6 fc00:4:12:ffff::/64 {
    class "hw" {
        match if substring ( option dhcp6.vendor-class, 6, 10 ) = "HWHW";
    }
    pool6 {
        allow members of "hw";
        range6 fc00:4:12:ffff::ff10 fc00:4:12:ffff::ff20;
    }
    pool6 {
        allow unknown clients;
        range6 fc00:4:12:ffff::100 fc00:4:12:ffff::120;
    }
}
```

ПРИМЕЧАНИЕ.

Начальная позиция подстроки (option dhcp6.vendor-class, 6, 10) равна 6, так как подстрока содержит <Enterprise-ID number\>

длиной четыре байта и `<string length\>` длиной два байта. Конечная позиция подстроки равна $6 + \langle \text{vendor class string length} \rangle$. В этом примере строкой класса поставщика является HWW, а длина строки равна 4. Таким образом, конечной позицией подстроки является $6 + 4 = 10$. Вы можете указывать `<vendor class string\>` и `<vendor class string length\>` в соответствии с предъявляемыми потребностями.

На сервере:

```
dhcpd -6 -cf /etc/dhcp/dhcpd6.conf <interface>
```

Ядро с поддержкой системных вызовов, связанных с сокетами

Обзор

Длина IPv6-адреса увеличена до 128 бит, что указывает на наличие достаточного количества IPv6-адресов для выделения. Заголовок IPv6-адреса стал проще по сравнению с заголовком IPv4-адреса, кроме того, улучшена функция автоматической настройки IPv6. IPv6-адреса подразделяются на следующие виды: для одноадресной передачи (unicast), многоадресной передачи (multicast) и произвольной передачи (anycast). Типовые адреса для одноадресной передачи включают в себя адреса link-local, уникальные локальные адреса и глобальные адреса. Поскольку существует достаточное количество глобальных IPv6-адресов, уникальные локальные адреса не используются (ранее они назывались локальными адресами для сетевого узла или адресами site-local, но от них отказались в 2004 году). В настоящее время основными адресами для одноадресной передачи являются адрес link-local и глобальный адрес. Текущее ядро поддерживает системный вызов сокетов. Адрес link-local и глобальный адрес, использующие адреса для одноадресной передачи, различаются.

Различия между адресом link-local и глобальным адресом во время вызова сокета

Документ RFC 2553 “Основные расширения интерфейса сокетов для IPv6” (Basic Socket Interface Extensions for IPv6) определяет структуру данных sockaddr_in6 следующим образом.

```
struct sockaddr_in6 {
    uint8_t      sin6_len;    /* length of this struct */
    sa_family_t  sin6_family; /* AF_INET6 */
    in_port_t    sin6_port;   /* transport layer port # */
```

```
uint32_t      sin6_flowinfo; /* IPv6 flow information */
struct in6_addr sin6_addr;   /* IPv6 address */
uint32_t      sin6_scope_id; /* set of interfaces for a scope */
};
```

ПРИМЕЧАНИЕ.

sin6_scope_id: 32-разрядное целое число. Для адреса link-local это значение идентифицирует индекс указанного интерфейса. Для link-range sin6_addr это значение идентифицирует индекс указанного интерфейса. Для site-range sin6_addr это значение используется в качестве идентификатора сайта (использование адреса site-local было прекращено).

Когда для связи с сокетом используется адрес link-local, требуется указать индекс интерфейса, соответствующий этому адресу, при формировании адреса назначения. Для преобразования имени интерфейса в номер индекса интерфейса обычно можно использовать функцию if_nametoindex. Ниже приведены подробные сведения.

```
int port = 1234;
int sk_fd;
int iff_index = 0;
char iff_name[100] = "enp3s0";
char * ll_addr[100] = "fe80::123:456:789";
struct sockaddr_in6 server_addr;

memset(&server_addr,0,sizeof(struct sockaddr_in6));
iff_index=if_nametoindex(iff_name);

server_addr.sin6_family=AF_INET6;
server_addr.sin6_port=htons(port);
server_addr.sin6_scope_id=iff_index;
inet_pton(AF_INET6, ll_addr, &(server_addr.sin6_addr));

sk_fd=socket(AF_INET6, SOCK_STREAM, IPPROTO_TCP);
connect(sk_fd, (struct sockaddr *)&server_addr, sizeof(struct sockaddr_in6));
```

Настройка устойчивости процесса управляющей программы dhclient для IPv4-адресов

Обзор

Если при использовании сервиса NetworkManager для управления сетевыми сервисами файл конфигурации `ifcfg-<interface-name\>` интерфейса настроен для получения IP-адреса в режиме DHCP, сервис NetworkManager запускает процесс управляющей программы `dhclient` для получения IP-адреса с DHCP-сервера.

`dhclient` предоставляет параметр `-1`, позволяющий определить, пытается ли процесс `dhclient` устойчивым образом запрашивать IP-адрес либо завершает работу после истечения времени ожидания запроса до получения ответа от DHCP-сервера. Для процесса управляющей программы `dhclient` для IPv4-адресов можно задать `PERSISTENT_DHCLIENT` в файле конфигурации `ifcfg-<interface-name\>`, чтобы определить, следует ли устанавливать свойство устойчивости процесса `dhclient` для IPv4-адресов.

Ограничения

1. Если текущий процесс `dhclient` остановлен, сетевой сервис не может запустить его автоматически. Поэтому требуется обеспечить надежность работы.
2. Если параметр `PERSISTENT_DHCLIENT` настроен, убедитесь, что соответствующий DHCP-сервер существует. Если при запуске сетевого сервиса DHCP-сервер недоступен, а процесс `dhclient` постоянно пытается отправлять пакеты запросов, но не получает никакого ответа, сетевой сервис приостанавливается до истечения времени ожидания сетевого сервиса. Сетевой сервис запускает процессы `dhclient` для IPv4-адресов у нескольких сетевых карт в последовательном режиме. Если для сетевой карты настроена устойчивость, однако DHCP-сервер не готов, сетевой сервис будет приостановлен при получении IPv4-адреса для этой сетевой карты. В результате сетевая карта не может получить IPv4- или IPv6-адрес.

Описанные выше ограничения применяются к особым сценариям. Вам требуется обеспечить надежность работы.

Различия в конфигурации между IPv4 DHCP и IPv6 DHCPv6

Вы можете настроить параметр `ifcfg-<interface-name\>` в интерфейсе, чтобы включить IPv4 и IPv6 для динамического получения IP-адресов с использованием DHCP или DHCPv6. Конфигурация выглядит следующим образом.

BOOTPROTO=none|bootp|dhcp

DHCPV6C=yes|no

PERSISTENT_DHCLIENT=yes|no|1|0

- BOOTPROTO: **none** означает, что адрес IPv4 настроен статически. bootp|dhcp позволяет DHCP-клиенту динамически получать IPv4-адрес.
- DHCPV6C: значение **no** указывает, что IPv6-адрес настроен статическим образом, а значение **yes** указывает, что DHCPv6 dhclient может получить IPv6-адрес динамически.
- PERSISTENT_DHCLIENT: значение **no|0** указывает, что процесс IPv4 dhclient настроен как неустойчивый. Если dhclient отправляет пакет запроса на DHCP-сервер и не получает никакого ответа, dhclient завершает работу через определенный период времени с кодом выхода 2. Значение **yes|1** указывает, что процесс IPv4 dhclient настроен для устойчивой работы. Процесс dhclient неоднократно отправляет пакеты запросов на DHCP-сервер. **Если параметр PERSISTENT_DHCLIENT не настроен, для IPv4 dhclient по умолчанию установлено значение yes|1.**



ПРИМЕЧАНИЕ.

Настройка PERSISTENT_DHCLIENT применяется только для IPv4 и не применяется для процессов dhclient -6, связанных с IPv6. По умолчанию настройка устойчивости не выполняется для IPv6.

Различия между конфигурациями IPv4 и IPv6 при использовании команды iproute

Обзор

IPv4 и IPv6 — это два разных стандарта протокола. Поэтому команды iproute используются по-разному. В этом разделе описаны различия между командами для IPv4 и IPv6 в пакете iproute.

Для выполнения команд iproute вам нужны права root.

Жизненный цикл IPv6-адреса

Состояние IPv6

Описание

tentative

Временное состояние: недавно добавленный адрес все еще находится в процессе проверки дублирования адресов (DAD).

preferred

Предпочтительное состояние: процесс DAD завершен, однако пакет NA не получен, что указывает на отсутствие конфликта для этого адреса.

deprecated

Состояние устаревания: адрес имеет срок действия (`valid_lft` или `preferred_lft`). После истечения срока `preferred_lft` адрес переходит в состояние устаревания.

Адрес в таком состоянии не подходит для создания подключения, однако исходное подключение использовать можно.

invalid

Недействительное состояние: если после истечения срока `preferred_lft` продление аренды завершается с ошибкой, адрес переходит в недействительное состояние после истечения срока `valid_lft`, что указывает на невозможность повторного использования адреса.

ПРИМЕЧАНИЕ.

- `preferred_lft`: предпочтительное время существования. Срок действия `preferred_lft` адреса не истек, и его можно использовать для обычного взаимодействия. Если существует несколько адресов в состоянии `preferred`, адрес выбирается на основе механизма ядра.
- `valid_lft`: действительное время существования. Адрес нельзя использовать для создания подключений в течение периода `[preferred_lft, valid_lft]`. Существующие подключения продолжают действовать.

Команда ip link

Команды имеют следующий формат:

```
ip link set IFNAME mtu MTU
```


Минимальный PMTU для IPv6 равен 1280. Если задано значение MTU меньше 1280, IPv6-адреса будут потеряны. Другие устройства не могут проверять связь с IPv6-адресом.

Команда ip addr

1. Команды имеют следующий формат:

```
ip [-6] addr add IFADDR dev IFNAME
```

Вы можете добавить параметр -6 или не добавлять IPv6-адрес. Команда ip addr определяет принадлежность адреса к IPv4 или IPv6 в зависимости от его типа.

Если указан параметр -6, а IFADDR является IPv4-адресом, возвращается сообщение об ошибке.

2. Команды имеют следующий формат:

```
ip [-6] addr add IFADDR dev IFNAME [home|nodad]
```

[home|nodad] действителен только для IPv6-адресов.

- home: указывает домашний адрес, определенный в документе RFC 6275. (Этот адрес получается мобильным узлом из домашнего канала и является постоянным адресом мобильного узла. Если мобильный узел остается в одном домашнем канале, взаимодействие между различными объектами осуществляется нормальным образом.)
- nodad: указывает, что DAD не выполняется при добавлении этого IPv6-адреса. (RFC 4862) Если для нескольких интерфейсов на устройстве настроен один и тот же IPv6-адрес с помощью параметра nodad, этот IPv6-адрес используется в порядке следования интерфейсов. Нельзя добавить к одному интерфейсу IPv6-адрес как с параметром nodad, так и без него, потому что эти два IP-адреса одинаковы. В противном случае отображается сообщение "RTNETLINK answers: File exists" (Ответ RNETLINK: файл существует).

3. Команды имеют следующий формат:

```
ip [-6] addr del IFADDR dev IFNAME
```

Вы можете добавить параметр -6 или не удалять IPv6-адрес. Команда ip addr del определяет, используется ли адрес IPv4 или IPv6, в зависимости от типа адреса.

4. Команды имеют следующий формат:

```
ip [-6] addr show dev IFNAME  
[tentative|tentative|deprecated|-deprecated|dadfailed|-dadfailed|temporary]
```

- Если параметр -6 не указан, отображаются как IPv4-адреса, так и IPv6-адреса. Если параметр -6 указан, отображаются только IPv6-адреса.

-

[tentative|tentative|deprecated|-deprecated|dadfailed|-dadfailed|temporary]. Эти параметры предназначены только для IPv6. Вы можете фильтровать и просматривать адреса на основе состояния IPv6-адреса.

1. tentative (только для IPv6): перечисляет только те адреса, которые не прошли проверку дублирования адресов (DAD).
2. -tentative (только для IPv6): перечисляет только адреса, которые не проходят процесс DAD.
3. deprecated (только для IPv6): перечисляет только устаревшие адреса.
4. -deprecated (только для IPv6): перечисляет только те адреса, которые не являются устаревшими.
5. dadfailed (только для IPv6): перечисляет только те адреса, которые не проходят проверку DAD.
6. -dadfailed (только для IPv6): перечисляет только те адреса, которые не приводят к ошибкам проверки DAD.
7. temporary (только для IPv6): перечисляет только временные адреса.

Command ip route

1. Команды имеют следующий формат:

```
ip [-6] route add ROUTE [mtu lock MTU]
```

- Параметр -6: вы можете как добавить, так и не добавлять параметр -6 при добавлении маршрута IPv6. Команда ip route определяет, используется ли адрес IPv4 или IPv6, в зависимости от типа адреса.
- mtu lock MTU: specifies the MTU of the locked route. Если MTU не заблокирован, значение MTU может быть изменено ядром во время процесса PMTUD. Если MTU заблокирован, попытка выполнения PMTUD не предпринимается. Бит DF не

устанавливается ни для одного из пакетов IPv4, а пакеты IPv6 сегментируются на основе MTU.

2. Команды имеют следующий формат:

```
ip [-6] route del ROUTE
```

Вы можете выбрать, нужно ли добавлять параметр -6, при удалении маршрута IPv6. Команда ip route определяет, используется ли адрес IPv4 или IPv6, в зависимости от типа адреса.

Команда ip rule

1. Команды имеют следующий формат:

```
ip [-6] rule list
```

Параметр -6: если параметр -6 установлен, выводятся маршруты на основе политики IPv6. Если параметр -6 не установлен, выводятся маршруты на основе политики IPv4. Поэтому вам нужно настроить параметр -6 в соответствии с конкретным типом протокола.

2. Команды имеют следующий формат:

```
ip [-6] rule [add|del] [from|to] ADDR table TABLE pref PREF
```

Параметр -6: этот параметр необходимо настроить для записей маршрутизации на основе политики IPv6. В противном случае отображается сообщение об ошибке "Error: Invalid source address." (Неверный исходный адрес). Соответственно, параметр -6 не может быть установлен для записей маршрутизации на основе политики IPv4. В противном случае отображается сообщение об ошибке "Error: Invalid source address." (Неверный исходный адрес).

Различия в конфигурации сервиса NetworkManager

Обзор

Сервис NetworkManager использует определение логического интерфейса ifup/ifdown для расширенной настройки сети. Большинство параметров задаются в файлах конфигурации /etc/sysconfig/network и /etc/sysconfig/network-scripts/ifcfg-<interface-name>. Первый отвечает за глобальную настройку, а второй — за настройку указанной сетевой карты. Когда две эти настройки конфликтуют, применяется настройка из второго файла.

Различия в конфигурации

Различия конфигурации в /etc/sysconfig/network:

IPv4	IPv6	Описание
Н/Д	IPV6FORWARDING=yes no	Переадресация IPv6. По умолчанию переадресация пакетов IPv6 не осуществляется.
Н/Д	IPV6_AUTOCONF=yes no	Если переадресация IPv6 включена, используется значение no. В противном случае используется значение yes.
Н/Д	IPV6_ROUTER=yes no	Если переадресация IPv6 включена, используется значение yes. В противном случае используется значение no.
Н/Д	IPV6_AUTOTUNNEL=yes no	Указывает на автоматический туннельный режим. Значение по умолчанию – no.
GATEWAY	IPV6_DEFAULTGW=<IPv6 address[%interface]> (необязательно)	Указывает шлюз по умолчанию в формате IPv6.
Н/Д	IPV6_DEFAULTDEV=<interface> (необязательно)	Указывает сетевую карту переадресации по умолчанию.
Н/Д	IPV6_RADVD_PIDFILE=<pid-file> (необязательно)	Путь по умолчанию для ipv6_radvd_pid – /var/run/radvd/radvd.pid.

Н/Д	IPV6_RADVD_TRIGGER_ACTION =start stop reload restart SIGHU P (необязательно)	Действие триггера radvd по умолчанию.
-----	--	---------------------------------------

Различия в /etc/sysconfig/network-scripts/ifcfg-<interface-name\>:

IPv4	IPv6	Описание
IPADDRn	IPV6ADDR=<IPv6 address>[/<prefix length>]	Указывает IP-адрес.
PREFIXn	Н/Д	Префикс сети, псевдоним сети и PPP недопустимы. Приоритет выше, чем у NETMASK.
NETMASKn	Н/Д	Указывает маску подсети. Используется только для псевдонима и PPP.
GATEWAY	IPV6_DEFAULTGW=<IPv6 address[%interface]> (необязательно)	Шлюз по умолчанию.
MTU	IPV6_MTU=<MTU of link> (необязательно)	MTU по умолчанию.

IPV4_FAILURE_FATAL=yes no	IPV6_FAILURE_FATAL	Значение по умолчанию – no. Если для этого параметра задано значение yes, ifup-eth завершает работу при сбое dhclient.
Н/Д	IPV6_PRIVACY=rfc3041	Отключено по умолчанию.
Н/Д	IPV6INIT=yes no	IPv6 включен по умолчанию.
Н/Д	IPV6FORWARDING=yes no	Эта функция по умолчанию отключена, а ее использование прекращено.

Часто задаваемые вопросы

iscsi-initiator-utils не поддерживает IPv6-адрес fe80

Симптом

Когда клиент использует IPv6-адрес для входа на iSCSI-сервер, выполните команду `iscsiadm -m node -p ipv6address -l`. Если используется глобальный адрес, замените `ipv6address` в примере команды глобальным адресом. Однако адрес `link-local` (IPv6-адрес, начинающийся с `fe80`) использовать нельзя, так как текущий механизм `iscsi-initiator-utils` не поддерживает вход на iSCSI-сервер с использованием адреса `link-local`.

Возможная причина

Если вы входите в систему, используя формат `iscsiadm -m node -p fe80::xxxx -l`, возвращается ошибка времени ожидания входа. Это связано с тем, что при использовании адреса `link-local` необходимо указать интерфейс. В противном случае функция `iscsi_io_tcp_connect` не может вызвать функцию подключения и выдается стандартный код ошибки 22.

Если вы используете для входа в систему формат `iscsiadm -m node -p fe80::xxxx%enp3s0 -l`, функция `iscsi_addr_match` сравнит адрес `fe80::xxxx%enp3s0` с адресом `fe80::xxxx` в информации об узле, возвращаемой сервером. Результат сравнения не совпадает, что приводит к ошибке входа.

Поэтому **текущий механизм `iscsi-initiator-utils` не поддерживает вход на сервер iSCSI с использованием адреса `link-local`.**

Потеря IPv6-адреса после отключения сетевой карты

Симптом

Выполните команду `ip link down+up NIC` или `ifconfig down+up NIC`, чтобы отключить сетевую карту, а затем включить ее для подключения к сети. Проверьте IP-адрес, настроенный на сетевой карте. Оказывается, что IPv4-адрес не теряется, а настроенный IPv6-адрес теряется.

Возможная причина

Согласно логической схеме обработки в ядре, если сетевая карта переходит в отключенное состояние, все адреса IPv4 и IPv6 будут сброшены. После того как сетевая карта переходит в рабочее состояние, IPv4-адрес автоматически восстанавливается, а также восстанавливается автоматически настроенный `link-local` адрес IPv6 на сетевой карте. Однако другие IPv6-адреса по умолчанию теряются. Чтобы сохранить эти IPv6-адреса, выполните команду **`sysctl -w net.ipv6.conf.<имя сетевой карты>.keep_addr_on_down=1`**.

Долгое добавление или удаление IPv6-адреса для интерфейса объединения с несколькими IPv6-адресами

Симптом

Когда пользователи выполняют следующую команду для добавления или удаления (включая сброс) IPv6-адреса, время ожидания линейно возрастает вместе с количеством IPv6-адресов, настроенных в интерфейсе объединения. **X** — младшие 16 битов, которые изменяются динамически. Например, для добавления 3000 IPv6-адресов или их удаления из интерфейса объединения, который уже имеет четыре физических сетевых карты, использующих один поток, требуется около пяти минут, тогда как у одной физической сетевой карты это занимает менее 10 секунд.

```
ip a add/del 192:168::18:X/64 dev DEVICE
```

Возможная причина

Когда IPv6-адрес добавляется к интерфейсу объединения, создается IPv6-адрес многоадресной передачи, который синхронизируется со всеми физическими сетевыми картами. Требуемое время возрастает с увеличением количества IPv6-адресов. В итоге это занимает слишком много времени.

Решение

IPv6-адрес для многоадресной передачи создается путем объединения младших 24 битов IPv6-адреса и 33-33-ff. Если адресов для многоадресной передачи слишком много, добавление или удаление адреса занимает много времени. Если таких адресов немного, требуемое время не увеличивается.

Рекомендуется сделать так, чтобы младшие 24 бита IPv6-адреса совпадали со старшими 24 битами IPv6-адреса. Таким образом, одна сетевая карта может взаимодействовать с внешними устройствами, используя только один IP-адрес в сегменте сети.

Задержка при передаче журнала rsyslog в сценарии, где используются как IPv4-адреса, так и IPv6-адреса

Симптом

Если в файле конфигурации клиента rsyslog настроены как IPv4-адреса, так и IPv6-адреса, а конфигурации портов одинаковы, существует вероятность задержки вывода журнала при сборе журналов сервером.

Возможная причина

Задержка вызвана механизмом буферных очередей rsyslog. По умолчанию rsyslog записывает данные в файл только тогда, когда количество буферных очередей достигает заданного значения.

Решение

Вы можете отключить механизм буферных очередей, настроив прямой режим (Direct) от имени пользователя **root**. Добавьте следующую информацию в начало нового файла конфигурации удаленной передачи, находящегося в каталоге /etc/rsyslog.d на сервере удаленной передачи rsyslog:

```
$ActionQueueType Direct
$MainMsgQueueType Direct
```

 **ПРИМЕЧАНИЕ.**

- В прямом режиме размер очереди уменьшается на 1. Поэтому для вывода следующего журнала в очереди резервируется один журнал.
- Прямой режим снижает производительность rsyslog на сервере.

Управление сервисами

В этом разделе описано использование systemd для управления операционной системой и сервисами.

Введение в systemd

systemd – это менеджер системы и сервисов в операционной системе GNU/Linux. Он обеспечивает обратную совместимость со скриптами инициализации SysV и LSB и поддерживает такие функции, как активация сервисов на основе сокетов и D-Bus, активация демонов по запросу, моментальные снимки состояния системы и управление точками монтирования и автоматического монтирования. За счет systemd можно усовершенствовать логику управления и параллелизацию сервисов.

Компоненты systemd

В systemd содержатся служебные модули, необходимые для управления всеми службами в составе ОС.

Компоненты классифицируются по представляемому ими типу ресурсов и определяются в файлах конфигурации компонентов. Например, компонент avahi.service представляет демон Avahi и определен в файле **avahi.service**. В [таблице 1](#) перечислены доступные типы компонентов systemd.

Таблица 1. Доступные типы компонентов systemd

Тип компонента	Расширение файла	Описание
Компонент сервиса	*.service	Системный сервис.
Целевой компонент	*.target	Группа компонентов systemd.
Компонент автоматического монтирования	*.automount	Точка автоматического монтирования файловой системы.
Компонент устройства	*.device	Файл устройства, распознаваемого ядром.
Компонент монтирования	*.mount	Точка монтирования файловой системы.
Компонент пути	*.path	Файл или каталог в файловой системе.
Компонент области	*.scope	Внешне создаваемый процесс.
Компонент среза	*.slice	Группа иерархически упорядоченных компонентов, управляющих системными процессами.
Компонент сокета	*.socket	Сокет межпроцессного взаимодействия.

Компонент подкачки	*.swap	Устройство или файл подкачки.
Компонент таймера	*.timer	Таймер systemd.

Все доступные типы компонентов systemd располагаются в одном из каталогов, приведенных в [таблице 2](#).

Таблица 2. Расположение доступных компонентов systemd

Каталог	Описание
/usr/lib/systemd/system/	Компоненты systemd, распространяемые с установленными пакетами RPM.
/run/systemd/system/	Компоненты systemd, создаваемые во время выполнения.
/etc/systemd/system/	Компоненты systemd, создаваемые и контролируемые системным администратором.

Возможности

Быстрая активация

Параллелизация в `systemd` более агрессивная, чем в `UpStart`. Активация на основе сокетов и `D-Bus` уменьшает время загрузки операционной системы.

Чтобы ускорить загрузку системы, `systemd` пытается сделать следующее:

- активировать только необходимые процессы;
- активировать как можно больше процессов параллельно.

Активация по требованию

При инициализации `SysVinit` активируются все возможные процессы фоновых сервисов, которые в принципе могут использоваться. Пользователи могут войти в систему только после активации всех этих сервисных процессов прописанных в выбранном `Runlevel`. Недостатки `SysVinit` очевидны: медленная загрузка системы.

Некоторые сервисы могут практически никогда не использоваться во время работы системы. Например, сервисы печати `CUPS` на большинстве серверов используются редко. На многих серверах редко используется `SSH`. Тратить время на запуск этих сервисов и системных ресурсов бессмысленно.

Активация `systemd` возможна только при запросе сервиса. Если запрос завершен, `systemd` останавливает работу.

Управление жизненным циклом сервисов с помощью `Cgroups`

Важной ролью системы инициализации является отслеживание и управление жизненным циклом сервисов. Она может запускать и останавливать сервис. Однако закодировать систему инициализации для остановки сервисов гораздо сложнее, чем можно подумать.

Сервисные процессы часто работают в фоновом режиме как демоны и иногда дважды разветвляются. Для `UpStart` необходима корректная настройка строфы `execst` в файле конфигурации. В противном случае `UpStart` не сможет узнать ИД процесса (`PID`) демона, подсчитав количество ответвлений.

Контрольные группы (`Cgroups`), давно используемые для управления квотами системных ресурсов, упрощают эти задачи. Их простота во многом обусловлена пользовательским интерфейсом, аналогичным файловой системе. Когда родительский сервис создает дочерний, последний наследует все атрибуты группы `Cgroup`, в которую входит первый. Это означает, что все необходимые сервисы помещаются в одну и ту же `Cgroup`. Для `systemd` достаточно просто пройти по контрольной группе, чтобы найти `PID` всех нужных сервисов, а затем остановить их один за другим.

Управление точками монтирования и автоматического монтирования

В традиционных системах Linux пользователи могут поддерживать фиксированные точки монтирования файловой системы с помощью файла **/etc/fstab**. Эти точки автоматически монтируются при запуске системы. После запуска они становятся доступными. Эти точки монтирования являются файловыми системами, которые критически важны для работы всей системы, например каталогом **HOME**. Как и SysVinit, systemd контролирует эти точки, чтобы они автоматически монтировались при запуске системы. И systemd также обеспечивает совместимость с файлом **/etc/fstab**. Вы по-прежнему можете использовать его для управления точками монтирования.

Бывают случаи, когда необходимы монтирование или размонтирование по требованию. Например, временная точка монтирования необходима для доступа к содержимому на DVD-диске. А когда оно больше не нужно, точка монтирования удаляется (с помощью команды **umount**) для экономии ресурсов. Традиционно это делается с помощью сервиса autofs.

При использовании systemd возможно автоматическое монтирование без необходимости установки autofs.

Управление транзакционными зависимостями

Загрузка системы включает множество отдельных сервисов, некоторые из которых могут быть зависимыми друг от друга. Например, демон сетевой файловой системы (NFS) может запуститься только после активации сетевого подключения. В systemd возможен параллельный запуск многих зависимых заданий, но не всех из них. Возвращаясь к примеру с NFS, невозможно смонтировать NFS и активировать сеть одновременно. Прежде чем запускать сервис, systemd вычисляет его зависимости, создает временную транзакцию и проверяет, что эта транзакция непротиворечива (все необходимые сервисы могут быть активированы независимо друг от друга).

Совместимость со скриптами SysVinit

Как и в UpStart, в systemd вводятся новые методы настройки и предъявляются новые требования к разработке приложений. Если вы хотите заменить текущую систему инициализации на systemd, то нужно обеспечить совместимость systemd с существующей программой. В любом дистрибутиве Linux изменить весь код сервисов за короткое время для использования systemd будет непросто.

В systemd доступны функции, совместимые со скриптами инициализации SysVinit и LSB. Изменять имеющиеся в системе сервисы и процессы не

требуется. Это уменьшает цену переноса системы на systemd, позволяя пользователям заменить на systemd имеющуюся систему инициализации.

Моментальные снимки и восстановление состояния системы

Запускать systemd можно по запросу. Из-за этого рабочее состояние системы изменяется динамически и вы не можете знать, какие конкретно сервисы в системе сейчас выполняются. Моментальные снимки systemd позволяют сохранять и восстанавливать текущее состояние работы системы.

Например, если в системе запущены сервисы А и В, вы можете выполнить команду **systemd**, чтобы создать моментальный снимок ее текущего состояния. Затем остановите процесс А или внесите в систему любое другое изменение, например запустите процесс С. После этого выполните команду systemd для восстановления из снимка, чтобы восстановить систему до точки, в которой был сделан снимок, то есть когда работали только сервисы А и В. Возможный сценарий применения – отладка. Например, если на сервере возникает исключение, пользователь может сохранить текущее состояние в виде моментального снимка для целей отладки и выполнить любую операцию, например остановить сервис. После завершения отладки происходит восстановление из снимка.

Управление системными сервисами

В systemd доступна команда systemctl для запуска, остановки, перезапуска, просмотра, включения и отключения системных сервисов.

Сравнение команд SysVinit и systemd

Функции команды **systemctl** из **systemd** подобны функциям команды **SysVinit**. Обратите внимание, что в этой версии поддерживаются команды **service** и **chkconfig**. Подробнее см. [таблицу 3](#). Рекомендуется использовать команду **systemctl** для управления системными сервисами.

Таблица 3. Сравнение команд SysVinit и systemd

Команда SysVinit	Команда systemd	Описание

service network start	systemctl start network.service	Запускает сервис.
service network stop	systemctl stop network.service	Останавливает сервис.
service network restart	systemctl restart network.service	Перезапускает сервис.
service network reload	systemctl reload network.service	Перезагружает файл конфигурации без прерывания операции.
service network condrestart	systemctl condrestart network.service	Перезапускает сервис, только если он работает.
service network status	systemctl status network.service	Проверяет состояние работы сервиса.
chkconfig network on	systemctl enable network.service	Включает сервис, когда наступает время его активации или выполняется условие триггера для его включения.
chkconfig network off	systemctl disable network.service	Отключает сервис, когда наступает время его активации или выполняется условие триггера для его отключения.

chkconfig network	systemctl is-enabled network.service	Проверяет, включен ли сервис.
chkconfig --list	systemctl list-unit-files --type=service	Выводит список всех сервисов на каждом уровне выполнения и проверяет, включены ли они.
chkconfig network --list	ls /etc/systemd/system/*.wa nts/network.service	Перечисляет уровни выполнения, на которых включен сервис, и те, на которых он отключен.
chkconfig network --add	systemctl daemon-reload	Используется, когда нужно создать сервисный файл или изменить настройки.

Перечисление сервисов

Чтобы вывести список всех загруженных в данный момент сервисов, выполните следующую команду:

```
systemctl list-units --type service
```

Чтобы вывести список всех сервисов вне зависимости от того, загружены они или нет, выполните следующую команду (с параметром all):

```
systemctl list-units --type service --all
```

Пример списка всех загруженных на данный момент сервисов:

```
$ systemctl list-units --type service
UNIT                                LOAD ACTIVE SUB    JOB DESCRIPTION
atd.service                         loaded active running Deferred execution scheduler
auditd.service                      loaded active running Security Auditing Service
avahi-daemon.service                loaded active running Avahi mDNS/DNS-SD Stack
chronyd.service                     loaded active running NTP client/server
crond.service                       loaded active running Command Scheduler
dbus.service                         loaded active running D-Bus System Message Bus
```



```
dracut-shutdown.service    loaded active exited      Restore /run/initramfs on
shutdown
firewalld.service         loaded active running      firewalld - dynamic firewall daemon
getty@tty1.service        loaded active running      Getty on tty1
gssproxy.service          loaded active running      GSSAPI Proxy Daemon
irqbalance.service        loaded active running      irqbalance daemon
iscsid.service            loaded activating start    start Open-iSCSI
```

Отображение состояния сервисов

Чтобы отобразить состояние сервиса, выполните следующую команду:

```
systemctl status name.service
```

В [таблице 4](#) описаны параметры в выходных данных команды.

Таблица 4. Выходные параметры

Параметр	Описание
Loaded	Информация о том, загружен ли сервис, абсолютный путь к файлу сервиса и запись о том, включен ли он.
Active	Информация о том, запущен ли сервис, и метка времени.
Main PID	PID сервиса.
CGroup	Дополнительная информация о связанных контрольных группах.

Чтобы проверить, запущен ли конкретный сервис, выполните следующую команду:

```
systemctl is-active name.service
```

Вывод команды **is-active** выглядит следующим образом:

Таблица 5. Вывод команды is-active

Состояние	Описание
active(running) active(exited)	сервис в системе работает. Сервис, штатно завершающий работу после срабатывания только один раз. В настоящее время никакая программа в системе не запущена. Например, функция quotaon выполняется только при запуске или монтировании программы.
active(waiting)	Программа ожидает других событий, чтобы продолжить работу. Например, сервис очереди печати запускается, но ему необходимо получить что-то в очередь (задания на печать), чтобы затем пробудить сервис принтера, который выполнит следующую функцию печати.
inactive	Сервис не запущен.

Точно так же, чтобы определить, включен ли конкретный сервис, выполните следующую команду:

```
systemctl is-enabled name.service
```

Вывод команды **is-enabled** выглядит следующим образом:

Таблица 6. Вывод команды is-enabled

Состояние	Описание
"enabled"	Сервис постоянно включен через псевдоним Alias=, .wants/ или символическую ссылку .requires/ в каталоге /etc/systemd/system/.
"enabled-runtime"	Сервис временно включен через псевдоним Alias=, .wants/ или символическую ссылку .requires/ в каталоге /run/systemd/system/.

"linked"	Хотя файл компонента не находится в стандартном каталоге компонентов, на него имеются символичные ссылки в постоянном каталоге <code>/etc/systemd/system/</code> .
"linked-runtime"	Хотя файл компонента не находится в стандартном каталоге компонентов, на него имеются символичные ссылки во временном каталоге <code>/run/systemd/system/</code> .
"masked"	Сервис постоянно замаскирован в каталоге <code>/etc/systemd/system/</code> (символическая ссылка на <code>/dev/null</code>). Из-за этого операция <code>start</code> не срабатывает.
"masked-runtime"	Сервис временно замаскирован в каталоге <code>/run/systemd/system/</code> (символическая ссылка на <code>/dev/null</code>). Из-за этого операция <code>start</code> не срабатывает.
"static"	Сервис не включен. В разделе <code>[Install]</code> файла компонента нет доступного параметра для команды <code>enable</code> .
"indirect"	Сервис не включен. Однако список значений для параметра <code>Also=</code> в разделе <code>[Install]</code> файла компонента не пустой (то есть некоторые компоненты в списке могли быть включены) либо файл компонента имеет символическую ссылку-псевдоним, которая не входит в список <code>Also=</code> . Для компонента шаблона это означает, что включен экземпляр, отличный от <code>DefaultInstance=</code> .
"disabled"	Сервис не включен. Однако раздел <code>[Install]</code> файла компонента содержит параметры, доступные для команды <code>enable</code> .
"generated"	Файл компонента создается генератором автоматически. Сгенерированный файл не может быть включен напрямую, но неявно включается генератором.

"transient"	Файл компонента создается динамически и на время API-интерфейсом runtime (среды выполнения). Включение временного компонента невозможно.
"bad"	Файл компонента неверный, или возникают другие ошибки. Команда is-enabled не возвращает это состояние, а выдает сообщение об ошибке. Этот компонент может быть отображен командой list-unit-files.

Например, чтобы отобразить состояние `gdm.service`, выполните команду **`systemctl status gdm.service`**.

```
# systemctl status gdm.service
gdm.service - GNOME Display Manager  Loaded: loaded
(/usr/lib/systemd/system/gdm.service; enabled)  Active: active (running) since Thu
2013-10-17 17:31:23 CEST; 5min ago
Main PID: 1029 (gdm)
CGroup: /system.slice/gdm.service
├─1029 /usr/sbin/gdm
├─1037 /usr/libexec/gdm-simple-slave --display-id /org/gno...
└─1047 /usr/bin/Xorg :0 -background none -verbose -auth /r...Oct 17 17:31:23
localhost systemd[1]: Started GNOME Display Manager.
```

Запуск сервиса

Чтобы запустить сервис, выполните от имени пользователя **root** следующую команду:

```
systemctl start name.service
```

Например, чтобы запустить сервис `httpd`, выполните следующую команду:

```
# systemctl start httpd.service
```

Остановка сервиса

Чтобы остановить сервис, выполните от имени пользователя **root** следующую команду:

```
systemctl stop name.service
```

Например, чтобы остановить сервис `Bluetooth`, выполните следующую команду:

```
# systemctl stop bluetooth.service
```

Перезапуск сервиса

Чтобы перезапустить сервис, выполните от имени пользователя **root** следующую команду:

```
systemctl restart name.service
```

Эта команда останавливает выбранный сервис в текущем сеансе и сразу же запускает его снова. Если выбранный сервис не выполняется, эта команда также запускает его.

Например, чтобы перезапустить сервис Bluetooth, выполните следующую команду:

```
# systemctl restart bluetooth.service
```

Включение сервиса в автозагрузку

Чтобы настроить автоматический запуск сервиса во время загрузки системы, выполните следующую команду от имени пользователя **root**:

```
systemctl enable name.service
```

Например, чтобы настроить при загрузке системы автоматический запуск сервиса httpd, выполните следующую команду:

```
# systemctl enable httpd.service  
ln -s '/usr/lib/systemd/system/httpd.service'  
'/etc/systemd/system/multi-user.target.wants/httpd.service'
```

Исключение сервиса из автозагрузки

Чтобы сервис не запускался во время загрузки системы автоматически, от имени пользователя **root** выполните следующую команду:

```
systemctl disable name.service
```

Например, чтобы предотвратить при загрузке системы автоматический запуск сервиса Bluetooth, выполните следующую команду:

```
# systemctl disable bluetooth.service  
Removed /etc/systemd/system/bluetooth.target.wants/bluetooth.service.  
Removed /etc/systemd/system/dbus-org.bluez.service.
```

Изменение уровня выполнения

Уровни выполнения и цели

В systemd понятие уровней выполнения заменено на “цели” для большей гибкости. Например, вы можете наследовать существующую цель и преобразовать ее в цель для своих нужд, добавив другие сервисы. В [таблице 7](#) представлен полный список уровней выполнения и соответствующих им целей systemd.

Таблица 7. Соответствие целей и уровней выполнения

Уровень выполнения	Цель systemd	Описание
0	runlevel0.target, poweroff.target	Выключение операционной системы
1, s, single	runlevel1.target, rescue.target	Операционная система работает в однопользовательском режиме.
2, 4	runlevel2.target, runlevel4.target, multi-user.target	Операционная система находится на уровне выполнения, который задан пользователем или относится к конкретному домену (по умолчанию эквивалентно уровню выполнения 3).
3	runlevel3.target, multi-user.target	Операционная система находится в неграфическом многопользовательском режиме и доступна из нескольких консолей или сетей.
5	runlevel5.target, graphical.target	Операционная система находится в графическом многопользовательском

		режиме. Все сервисы, работающие на уровне 3, доступны для входа через графический интерфейс.
6	runlevel6.target, reboot.target	Операционная система перезагружается.
emergency	emergency.target	Аварийная оболочка.

Просмотр цели запуска по умолчанию

Выполните следующую команду, чтобы просмотреть цель запуска системы по умолчанию:

```
systemctl get-default
```

Просмотр всех целей запуска

Выполните следующую команду, чтобы просмотреть все цели запуска системы:

```
systemctl list-units --type=target
```

Изменение цели по умолчанию

Чтобы изменить цель по умолчанию, выполните от имени пользователя **root** следующую команду:

```
systemctl set-default name.target
```

Изменение текущей цели

Чтобы изменить текущую цель, выполните от имени пользователя **root** следующую команду:

```
systemctl isolate name.target
```

Переход в режим восстановления

Чтобы перевести операционную систему в режим восстановления, выполните следующую команду от имени пользователя **root**:

```
systemctl rescue
```

Эта команда подобна команде **systemctl isolate rescue.target**. После выполнения команды на последовательный порт выводится следующая информация.

You are in rescue mode. After logging in, type "journalctl -xb" to view system logs, "systemctl reboot" to reboot, "systemctl default" or "exit" to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):

ПРИМЕЧАНИЕ.

Чтобы перейти из режима восстановления в нормальный режим работы, необходимо перезагрузить систему.

Переход в аварийный режим

Чтобы перевести операционную систему в аварийный режим, выполните следующую команду от имени пользователя **root**:

```
systemctl emergency
```

Эта команда подобна команде **systemctl isolate emergency.target**. После выполнения команды на последовательный порт выводится следующая информация.

You are in emergency mode. After logging in, type "journalctl -xb" to view system logs, "systemctl reboot" to reboot, "systemctl default" or "exit" to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):

ПРИМЕЧАНИЕ.

Чтобы перейти из аварийного режима в нормальный режим работы, необходимо перезагрузить систему.

Завершение работы, приостановка и гибернация операционной системы

Команда systemctl

Вместо старых команд управления системой Linux в systemd используется команда systemctl для завершения работы, перезапуска, приостановки и гибернации операционной системы. Хотя предыдущие команды управления системой Linux по-прежнему доступны в systemd из соображений совместимости, рекомендуется использовать **systemctl**, когда это возможно. Соответствие между ними показано в [таблице 8](#).

Таблица 8. Соответствие systemctl старым командам управления системой Linux

Команда управления Linux	Команда systemctl	Описание
halt	systemctl halt	Завершает работу операционной системы.
poweroff	systemctl poweroff	Выключает операционную систему.
reboot	systemctl reboot	Перезагружает операционную систему.

Завершение работы операционной системы

Чтобы завершить работу операционной системы и выключить систему, выполните от имени пользователя **root** следующую команду:

```
systemctl poweroff
```

Чтобы завершить работу операционной системы, не выключая ее, от имени пользователя **root** выполните следующую команду:

```
systemctl halt
```

По умолчанию при выполнении любой из этих команд systemd отправляет информационное сообщение всем пользователям, вошедшим в систему.

Чтобы это сообщение systemd не отправлялось, выполните команду с параметром **-no-wall**. Команда выглядит следующим образом:

```
systemctl --no-wall poweroff
```

Перезагрузка операционной системы

Чтобы перезагрузить операционную систему, от имени пользователя **root** выполните следующую команду:

```
systemctl reboot
```

По умолчанию при выполнении любой из этих команд systemd отправляет информационное сообщение всем пользователям, вошедшим в систему. Чтобы это сообщение systemd не отправлялось, выполните команду с параметром **-no-wall**. Команда выглядит следующим образом:

```
systemctl --no-wall reboot
```

Приостановка операционной системы

Чтобы приостановить работу операционной системы, выполните от имени пользователя **root** следующую команду:

```
systemctl suspend
```

Гибернация операционной системы

Чтобы перевести операционную систему в режим гибернации, от имени пользователя **root** выполните следующую команду:

```
systemctl hibernate
```

Чтобы приостановить работу операционной системы и перевести ее в режим гибернации, выполните от имени пользователя **root** следующую команду:

```
systemctl hybrid-sleep
```