

ifconfig(1M)

НАЗВАНИЕ

ifconfig - конфигурирование параметров сетевого интерфейса

СИНТАКСИС

```
/sbin/ifconfig интерфейс
[семейство_адресов] [адрес [/длина_префикса] [целевой_адрес]]
[addif адрес [/длина_префикса]] [removeif адрес [/длина_префикса]]
[arp | -arp] [auth_algs алгоритм_аутентификации]
[encr_algs алгоритм_шифрования] [encr_auth_algs алгоритм_аутентификации]
[auto-revarp] [broadcast адрес] [deprecated | -deprecated] [destination целевой_адрес]
[[failover] | [-failover]] [group [[имя] | ""]] [index {if_index}] [metric n]
[modlist] [modinsert имя_модуля@позиция] [modremove имя_модуля@позиция]
[mtu n] [netmask маска]
[plumb] [unplumb] [private | -private] [nud | -nud]
[set [адрес] [/маска_сети]]
[[standby] | [-standby]] [subnet адрес_подсети]
[tdst целевой_адрес_тоннеля] [tsrc исходный_адрес_тоннеля]
[trailers | -trailers] [up] [down] [xmit | -xmit]

/usr/sbin/ifconfig интерфейс
[семейство_адресов] [адрес [/длина_префикса] [целевой_адрес]]
[addif адрес [/длина_префикса]] [removeif адрес [/длина_префикса]]
[arp | -arp] [auth_algs алгоритм_аутентификации]
[encr_algs алгоритм_шифрования] [encr_auth_algs алгоритм_аутентификации]
[auto-revarp] [broadcast адрес] [deprecated | -deprecated] [destination целевой_адрес]
[[failover] | [-failover]] [group [[имя] | ""]] [index {if_index}] [metric n]
[modlist] [modinsert имя_модуля@позиция] [modremove имя_модуля@позиция]
[mtu n] [netmask маска]
[plumb] [unplumb] [private | -private] [nud | -nud]
[set [адрес] [/маска_сети]]
[[standby] | [-standby]] [subnet адрес_подсети]
[tdst целевой_адрес_тоннеля] [tsrc исходный_адрес_тоннеля]
[trailers | -trailers] [up] [down] [xmit | -xmit]

/sbin/ifconfig интерфейс
{auto-dhcp | dhcp} [primary] [wait секунд]
drop | extend | inform | ping | release | start | status

/usr/sbin/ifconfig интерфейс
{auto-dhcp | dhcp} [primary] [wait секунд]
drop | extend | inform | ping | release | start | status
```

ОПИСАНИЕ

Команда **ifconfig** используется для присвоения адреса сетевому интерфейсу и для конфигурирования параметров сетевого интерфейса. Команду **ifconfig** необходимо использовать при загрузке для задания сетевого адреса каждого из имеющихся на машине интерфейсов; ее можно также использовать в дальнейшем для переопределения адреса или других параметров работы интерфейса. Если опции не указаны, команда **ifconfig** выдает текущую конфигурацию сетевого интерфейса. Если семейство адресов задано, команда **ifconfig** выдает только информацию, связанную с соответствующим семейством адресов. Только суперпользователь может изменять конфигурацию сетевого интерфейса. Если опции указаны в фигурных скобках ({}), одну из них надо обязательно указать.

Две версии команды **ifconfig**, **/sbin/ifconfig** и **/usr/sbin/ifconfig**, работают по разному со службами имен. Порядок разрешения имен утилитой **/sbin/ifconfig** при загрузке системы фиксирован и не может быть изменен. Напротив, изменение файла конфигурации **/etc/nsswitch.conf** может повлиять на работу программы

/usr/sbin/ifconfig. Системный администратор может сконфигурировать источники информации и порядок их просмотра за счет переключения службы имен. Подробнее об этом см. на странице справочного руководства **nsswitch.conf(4)**.

Конфигурирование DHCP

Третья и четвертая форма этой команды используются для конфигурирования интерфейса по протоколу Dynamic Host Configuration Protocol (DHCP). Конфигурирование по DHCP возможно только для интерфейсов, для которых заданы адреса семейства **inet**. В этом режиме команда **ifconfig** используется для управления работой **dhcpcagent(1M)**, клиентского демона DHCP. После того, как интерфейс передан под управление DHCP с помощью опции **start**, команда **ifconfig**, при обычной работе, не должна использоваться для изменения адреса или характеристик интерфейса. Если адрес интерфейса, работающего под управлением DHCP, изменен, программа **dhcpcagent** перестанет управлять им.

ОПЦИИ

Поддерживаются следующие опции:

addif адрес

Создает следующий не используемый логический интерфейс на указанном физическом **интерфейсе**.

arp

Разрешает использование протокола разрешения адреса (Address Resolution Protocol - ARP) при сопоставлении адресов уровня сети и адресов уровня связи (этот протокол разрешается использовать по умолчанию). Этот протокол сейчас реализует сопоставление между адресами IPv4 и адресами Ethernet.

-arp

Запрещает использование протокола ARP.

auth_algs алгоритм_аутентификации

При туннелировании включает аутентификацию IPsec с указанным **алгоритмом**. Алгоритм можно задавать по номеру или по имени. Можно также указать **any**, не задавая предпочтительный алгоритм. Все свойства туннеля IPsec необходимо задать в той же командной строке. Для отключения защиты туннеля укажите алгоритм аутентификации **none**.

auto-dhcp

Использовать протокол DHCP для автоматического получения адреса для этого **интерфейса**. У этой опции есть синоним, **dhcp**.

primary

Задаёт **интерфейс** как основной. Интерфейс определяется как предпочтительный для получения данных о конфигурации клиента. Только один интерфейс может быть основным в каждый момент времени. Если другой интерфейс в дальнейшем назначается основным, он заменяет в этом качестве предыдущий. Назначение интерфейса основным не имеет особого значения после загрузки, поскольку многие программы уже запущены и были сконфигурированы на основе данных, прочитанных из прежнего основного интерфейса.

wait секунд

Команда **ifconfig** будет ждать пока не будет завершено действие или пока не пройдет указанное количество **секунд**. Если количество секунд не задано и действие относится к тем, которые не могут завершиться немедленно, **ifconfig** будет ждать завершения запрошенного действия 30 секунд. Можно также указать значение **forever**, т.е. ждать завершения бесконечно.

drop

Выводит указанный **интерфейс** из-под контроля протокола DHCP. Кроме того, устанавливает нулевой IP-адрес и помечает интерфейс как отключенный, **down**.

extend

Пытается продлить аренду **интерфейсом** адреса IPv4. Это не обязательно делать, поскольку агент автоматически продлит аренду перед завершением ее срока.

inform

Получает параметры конфигурации сети по протоколу DHCP, не арендуя IP-адрес. Это полезно в ситуациях, когда IP-адрес получается по другим механизмам, помимо DHCP.

ping

Проверяет, что указанный **интерфейс** находится под контролем протокола DHCP, т.е. этот интерфейс управляется агентом DHCP и работает нормально. Статус выхода 0 означает успешную проверку. Эта команда не имеет смысла, если указанный **интерфейс** представляет несколько интерфейсов.

release

Освобождает адрес IPv4 **интерфейса** и помечает интерфейс как отключенный, **down**.

start

Запускает DHCP на **интерфейсе**.

status

Выдает информацию о состоянии конфигурации DHCP для указанного **интерфейса**.

auto-revarp

Использовать протокол обратного разрешения адреса (Reverse Address Resolution Protocol - RARP) для автоматического получения адреса для данного **интерфейса**.

broadcast адрес

Только для протокола IPv4. Задаёт адрес, который будет использоваться для широковещательной отправки пакетов в сеть. Стандартным широковещательным адресом является адрес, в котором часть, соответствующая хосту, состоит из одних битов 1. Знак "+" (плюс) в качестве значения широковещательного адреса вызывает сброс широковещательного адреса в стандартный соответствующий (возможно новому) адресу интерфейса и маске сети. Аргументы команды **ifconfig** интерпретируются слева направо. Поэтому команды

```
example% ifconfig -a netmask + broadcast +
```

и

```
example% ifconfig -a broadcast + netmask +
```

могут привести к установке различных широковещательных адресов для интерфейсов.

deprecated

Помечает **адрес** как *нежелательный*. Адреса, помеченные как нежелательные, будут использоваться в качестве исходного адреса для исходящих пакетов только если для этого **интерфейса** нет других доступных адресов или приложение было явно привязано к соответствующему адресу. При выдаче информации о состоянии **интерфейса** среди флагов будет указано значение **DEPRECATED**.

-deprecated

Помечает адрес как не являющийся нежелательным.

destination целевой_адрес

Устанавливает целевой адрес для двухточечного (PPP) интерфейса.

dhcp

Эта опция является псевдонимом [**auto-dhcp**](#).

down

Помечает **интерфейс** как отключенный (**down**). Когда интерфейс помечен как отключенный, система не пытается передавать через него сообщения. По возможности, интерфейс сбрасывается, чтобы отключить также и прием. Это действие не исключает автоматически маршруты, использующие этот интерфейс.

encr_auth_algs алгоритм_аутентификации

Для туннеля включает протокол IPsec ESP с указанным алгоритмом аутентификации. Алгоритм можно задать по номеру или по имени, в том числе, можно задавать имена **any** или **none**, свидетельствующие об отсутствии предпочтительного алгоритма. Если при использовании ESP алгоритм шифрования задается, а алгоритм аутентификации - нет, стандартным значением алгоритма аутентификации ESP будет **any**.

encr_algs алгоритм_шифрования

Для туннеля включает протокол IPsec ESP с указанным алгоритмом шифрования. Алгоритм можно задать по номеру или по имени. Учтите, что все свойства туннеля IPsec необходимо указать в одной командной строке. Чтобы отключить защиту туннеля, необходимо указать алгоритм шифрования **none**. Если указан алгоритм аутентификации ESP, а алгоритм шифрования - нет, стандартным значением алгоритма шифрования ESP будет **null**.

-failover

Помечает адрес как не восстанавливаемый автоматически. Помеченные так адреса не восстанавливаются при сбое интерфейса. При выдаче статуса среди флагов указывается значение **NOFAILOVER**.

failover

Помечает адрес как автоматически восстанавливаемый. Соответствующий адрес будет восстанавливаться при сбое интерфейса. При выдаче статуса среди флагов значение **NOFAILOVER** не указывается.

group [имя | ""]

Добавляет интерфейс в многопоточную (multipathing) группу с указанным **именем**. Для удаления интерфейса из группы, укажите пустую строку, "".

При вызове для логического интерфейса с идентификатором ноль, среди информации о статусе выдается имя группы.

index n

Изменяет индекс **интерфейса**. В качестве значения **n** необходимо задавать индекс интерфейса (**if_index**), не используемого другим интерфейсом. Значение **if_index** будет положительным числом, уникально идентифицирующим сетевой интерфейс в системе.

metric n

Задаёт метрику маршрутизации (*routing metric*) интерфейса равной **n**; если значение не указано, используется стандартное значение 0. Метрика маршрутизации используется протоколом маршрутизации. Большие значения метрики делают маршрут менее предпочтительным; метрика учитывается как дополнительные *переходы* (hops) к целевой сети или хосту.

modinsert имя_модуля@позиция

Вставляет модуль **имя_модуля** в поток устройств в указанной **позиции**. Позиция задается относительно начала потока.

Позиция 0 означает вставку в самом начале потока.

Продолжая пример, представленный в описании опции **modlist**, следующая команда позволит вставить модуль **ipqos** после модуля **ip** и перед модулем **firewall**:

```
example% ifconfig hme0 modinsert ipqos@2
```

После этого список модулей в потоке устройства будет таким:

```
example% ifconfig hme0 modlist
0 arp
1 ip
2 ipqos
3 firewall
4 hme
```

modlist

Выдает список модулей в потоке устройства.

В следующем примере выдаются все модули в потоке устройства:

```
example% ifconfig hme0 modlist
0 arp
1 ip
2 firewall
4 hme
```

modremove имя_модуля@позиция

Удаляет модуль **имя_модуля** из указанной **позиции** в потоке устройства. Позиция задается относительно начала потока.

Продолжая пример, представленный в описании опции **modinsert**, с помощью следующей команды можно удалить модуль **firewall** из потока после уставки модуля **ipqos**:

```
example% ifconfig hme0 modremove firewall@3
```

После этого список модулей в потоке устройства будет иметь следующий вид:

```
example% ifconfig hme0 modlist
0 arp
1 ip
2 ipqos
3 hme
```

Учтите, что базовые модули IP-стека, например, модули **ip** и **tun**, удалить нельзя.

mtu n

Set the maximum transmission unit of the interface to n. For many types of networks, the mtu has an upper limit, for example, 1500 for Ethernet.

netmask маска

Только для IPv4. Указывает, какая часть адреса резервируется для деления сетей на подсети. **Маска** включает сетевую часть локального адреса и часть подсети, которая берется из поля хоста в адресе. **Маска** содержит 1 в битовых позициях 32-битового адреса, которые должны использоваться для сети и подсети, и 0 в позициях, соответствующих хосту. Маска должна содержать, как минимум, стандартную часть, соответствующую сети, а поле подсети должно идти сразу же после части сети. Маску можно задать одним из четырех способов:

1. в виде одного шестнадцатеричного числа, начинающегося с **0x**
2. в виде адреса с точками
3. как "+" (знак плюс)
4. с помощью псевдонима хоста/псевдонима сети, указанных в базе данных сетей **networks(4)**.

Если в качестве значения маски сети указан знак "+" (плюс), маска ищется в базе данных **netmasks(4)**. При этом поиске находится самая длинная соответствующая маска сети в базе данных, начиная с IPv4-адреса интерфейса и последовательно маскируя в нем младшие биты. Этот итеративный поиск гарантирует, что можно будет использовать базу данных **netmasks(4)**, когда в пределах сети используются маски подсетей переменной длины.

Если в качестве значения **netmask** передан псевдоним хоста или сети, данные **netmask** ищутся в базе данных хостов или сетей. Имена ищутся сначала с помощью функции **gethostbyname(3NSL)**. Если так имя найти не удалось, оно ищется с помощью функции **getnetbyname(3SOCKET)**. Эти функции, в свою очередь, могут использовать файл **nsswitch.conf(4)** для определения того, их каких источников получать соответствующее значение.

Для обоих семейств протоколов, **inet** и **inet6**, информацию, задаваемую маской, можно также указывать как параметр **длина_префикса**, добавляемый через косую после параметра **адрес**.

nud

Включает механизм определения недостижимости соседа на выходном (point-to-go) интерфейсе.

-nud

Отключает механизм определения недостижимости соседа на выходном (point-to-go) интерфейсе.

plumb

Открывает устройство, связанное с указанным физическим интерфейсом, и настраивает потоки, необходимые для использования протокола IP на устройстве. Если эта опция используется с именем логического интерфейса, она создает указанный логический интерфейс. Интерфейс надо отдельно создавать для использования протоколами IPv4 и IPv6. Параметр **семейство_адресов** управляет тем, применяется ли команда **ifconfig** для конфигурирования IPv4 или IPv6.

Пока к интерфейсу не будет применена опция **plumb**, он не выдается в результатах команды **ifconfig -a**.

private

Сообщает демону маршрутизации **in.routed**, что интерфейс не надо рекламировать.

-private

Задаёт не рекламируемые интерфейсы.

removeif адрес

Удаляет логический интерфейс на указанном физическом **интерфейсе**, соответствующий заданному **адресу**.

set

Устанавливает **адрес**, **длину_префикса** или оба эти параметра для интерфейса.

standby

Помечает физический интерфейс как ждущий (standby). Если интерфейс помечен как **STANDBY** и входит в многопоточную (multipathing) группу, он не будет использоваться для отправки пакетов, если только на другом интерфейсе не произойдет сбой и доступ к сети не будет перенаправлен после сбоя на этот ждущий интерфейс.

Состояние интерфейса выдается как "**STANDBY, INACTIVE**", т.е. интерфейс находится в режиме ожидания и не активен. Флаг **IFF_INACTIVE** будет сброшен, если в следствие сбоя на одном из интерфейсов той же многопоточной группы доступ пойдет через этот ждущий интерфейс. Когда работа исходного интерфейса будет восстановлена, состояние интерфейса снова станет **INACTIVE**.

-standby

Отключает режим ожидания для данного **интерфейса**.

subnet

Задаёт адрес подсети для **интерфейса**.

tdst целевой_адрес_тоннеля

Задаёт целевой адрес тоннеля. Он не должен совпадать с исходным адресом тоннеля, поскольку по такому тоннелю ни один пакет из системы не выйдет.

trailers

Этот флаг ранее вызывал нестандартную инкапсуляцию пакетов inet на определенных уровнях связи. Драйверы, поставляемые в этой версии ОС, этот флаг больше не используют. Он поддерживается для совместимости, но игнорируется.

-trailers

Отключает использование trailer-инкапсуляции на уровне связи.

tsrc исходный_адрес_тоннеля

Задаёт исходный адрес туннеля. Это будет исходный адрес для исходящего пакета, инкапсулирующего IP-заголовок. Это должен быть адрес другого интерфейса, уже сконфигурированного с помощью **ifconfig**.

unplumb

Удаляет все потоки (streams), связанные с указанным физическим **интерфейсом**, и закрывает соответствующее устройство. При указании имени логического интерфейса, логический интерфейс удаляется из системы. После выполнения этой команды, устройство с соответствующим именем больше не будет выдаваться в результатах выполнения команды **ifconfig -a**. Перед этим интерфейс должен быть отключен, **down**.

up

Помечает **интерфейс** как работающий, "**up**". Это происходит автоматически при задании первого адреса на интерфейс. Опция **up** включает интерфейс, отключенный командой **ifconfig down**, повторно инициализируя оборудование.

xmit

Включает передачу пакетов по интерфейсу. Это стандартное состояние для работающего (**up**) интерфейса.

-xmit

Отключает передачу пакетов по интерфейсу. Интерфейс будет продолжать принимать пакеты.

ОПЕРАНДЫ

Операнд **интерфейс**, а также параметры **адреса** для него, описаны далее.

интерфейс

Строка вида **имя физический-экземпляр**, например, **le0** или **ie1**; или вида **имя физический-экземпляр:логический-экземпляр**, например, **le0:1**; или вида **ip.tunN** для туннелей.

Если имя интерфейса начинается с дефиса (-), оно интерпретируется как набор опций, задающий набор интерфейсов. В этом случае среди опций должна быть опция **-a**, а остальные представленные ниже опции могут указываться в любом порядке. Если указано одно из таких имен интерфейса, указанные опции применяются ко всем соответствующим интерфейсам.

-a

Применять опции ко всем интерфейсам в системе.

-d

Применять опции ко всем отключенным (**down**) интерфейсам в системе.

-D

Применять опции ко всем интерфейсам, не находящимся под управлением протокола DHCP (Dynamic Host Configuration Protocol).

-u

Применять опции ко всем работающим (**up**) интерфейсам в системе.

-4

Применять команды ко всем интерфейсам IPv4.

-6

Применять команды ко всем интерфейсам IPv6.

семейство_адресов

Семейство адресов задается параметром **семейство_адресов**. Команда **ifconfig** в настоящее время поддерживает следующие семейства: **ether**, **inet** и **inet6**. Если семейство адресов не указано, используется **inet**.

адрес

Для семейства адресов IPv4 (**inet**) в качестве адреса может задаваться либо имя хоста, представленное в базе данных имен хостов (см. **hosts(4)**) или в карте **hosts** сетевой информационной службы (Network Information Service - NIS), либо адрес IPv4 в стандартной для Internet форме записи "через точку".

Для семейства IPv6 (**inet6**), адрес задается либо как имя хоста, указанное в базе данных имен хостов (см. **ipnodes(4)**) или в карте **ipnode** сетевой информационной службы (Network Information Service - NIS), либо как адрес IPv6 в стандартном для Internet шестнадцатеричном виде через двоеточия, **x:x:x:x:x:x**, где **x** - шестнадцатеричное число от 0 до FFFF.

Для обоих семейств адресов можно указывать Ethernet-адрес в виде **x:x:x:x:x:x**, где **x** - шестнадцатеричное число от 0 до FF.

Многие, хотя и не все, сетевые карты Ethernet имеют собственные адреса. Использование карт, не имеющих собственных адресов, описано в разделе 3.2.3(4) спецификации IEEE 802.3, где дается определение локально администрируемого адресного пространства. Группы интерфейсов должны организовываться только на картах, имеющих собственные адреса (см. раздел **ГРУППЫ ИНТЕРФЕЙСОВ**).

длина_префикса

Для семейств адресов IPv4 и IPv6 (**inet** и **inet6**), **длина_префикса** - это число от 0 до количества битов в адресе. Для **inet** в адресе 32 бита, а для **inet6** - 128. **Длина_префикса** задает количество начальных установленных битов в маске сети.

целевой_адрес

Если кроме параметра **адрес** передан параметр **целевой_адрес**, он задает адрес респондента с другой стороны двухточечной связи (PPP).

целевой_адрес_тоннеля

Адрес, который доступен или будет доступен через интерфейс, отличный от конфигурируемого тоннеля. Этот параметр указывает тоннелю, куда передавать попавшие в тоннель пакеты. Этот адрес не должен совпадать с конфигурируемым целевым (исходным? Надо проверить... - прим. переводчика) адресом тоннеля.

исходный_адрес_тоннеля

Адрес, связанный с уже сконфигурированным и запущенным (**up**) с помощью команды **ifconfig** интерфейсом.

ЛОГИЧЕСКИЕ ИНТЕРФЕЙСЫ

Стек протоколов Solaris TCP/IP позволяет связывать с одним физическим сетевым интерфейсом несколько логических интерфейсов. Это дает возможность задать несколько IP-адресов для одной машины, даже если она имеет всего один сетевой интерфейс. Физические сетевые интерфейсы имеют имена в виде **<имя драйвера><номер физического устройства>**, а логические - в виде **<имя драйвера><номер физического устройства>;<номер логического устройства>**. Физический интерфейс конфигурируется в системе с помощью команды **plumb**. Например:

```
example% ifconfig le0 plumb
```

После инициализации физического интерфейса с помощью **plumb**, можно конфигурировать связанные с ним логические интерфейсы с помощью отдельных вызовов команды **ifconfig** с опциями **plumb** или **addif**. Например, команда:

```
example% ifconfig le0:1 plumb
```

выделяет указанный логический интерфейс, связанный с физическим интерфейсом **le0**. Команда:

```
example% ifconfig le0 addif 192.9.200.1/24 up
```


выделяет следующий доступный логический номер экземпляра (logical unit number) на физическом интерфейсе **le0** и назначает ему указанный **адрес** с соответствующей **длиной_префикса**.

Для логического интерфейса можно сконфигурировать параметры (**адрес**, **длину_префикса** и т.д.), отличающиеся от параметров физического интерфейса, с которым он связан. Логические интерфейсы, связанные с одним и тем же физическим интерфейсом, тоже могут иметь разные параметры. Каждый логический интерфейс должен быть связан с существующим и работающим (**up**) физическим интерфейсом. Так, например, логический интерфейс **le0:1** можно конфигурировать после инициализации физического интерфейса **le0**.

Для удаления логического интерфейса используются опции **unplumb** или **removeif**. Например, команда:

```
example% ifconfig le0:1 down unplumb
```

удалит логический интерфейс **le0:1**.

ГРУППЫ ИНТЕРФЕЙСОВ

Если интерфейс (логический или физический) имеет IP-префикс, совпадающий с префиксом другого интерфейса, эти интерфейсы собираются в *группу интерфейсов*. IP-протокол использует группу интерфейсов для циклического выбора исходного адреса когда он не задан, и, при наличии в группе нескольких физических адресов, для распределения передаваемой на один целевой адрес (per-IP-destination) информации по различным IP-адресам. Подробнее о передаваемой на один целевой адрес информации см. на странице справочного руководства **netstat(1M)**.

Группировку можно включить с помощью команды **ndd(1M)**.

Ключевое слово **group** можно также использовать для формирования *многопоточной* (multipathing) группы. При использовании многопоточных групп возможности обычной группы интерфейсов тоже реализуются. Многопоточная группа обеспечивает также выявление сбоев и восстановления интерфейсов в группе. См. **in.mpathd(1M)** и System Administration Guide, Volume 3.

Группы интерфейсов, сформированные с помощью **ndd(1M)**, в дальнейшем перестанут поддерживаться. Поэтому рекомендуется формировать многопоточные группы с помощью ключевого слова **group**.

КОНФИГУРИРОВАНИЕ ИНТЕРФЕЙСОВ IPv6

Когда физический интерфейс IPv6 инициализируется (**plumb**) и конфигурируется как работающий (**up**) с помощью **ifconfig**, ему автоматически присваивается *локальный адрес связи* (link-local address) IPv6, в котором последние 64 бита вычисляются на основе MAC-адреса интерфейса.

```
ifconfig le0 inet6 plumb up
```

Следующий пример показывает, что локальный адрес связи имеет префикс **fe80::/10**.

```
example% ifconfig le0 inet6
le0: flags=2000841<UP,RUNNING,MULTICAST,IPv6>
mtu 1500 index 2
inet6 fe80::a00:20ff:fe8e:f3ad/10
```

Если в локальной сети существует "рекламирующий" (advertising) маршрутизатор IPv6, информирующий о префиксах, для вновь проинициализированного интерфейс IPv6 будут автоматически конфигурироваться логические интерфейсы для полученных префиксов. Например, если получена информация о префиксах **fec0:0:0:55::/64** и **3ff0:0:0:55::/64**, автоматически будут сконфигурированы следующие интерфейсы:

```
le0:1: flags=2080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6>
mtu 1500 index 2
    inet6 fec0::55:a00:20ff:fe8e:f3ad/64
le0:2: flags=2080841<UP,RUNNING,MULTICAST,ADDRCONF,IPv6>
mtu 1500 index 2
    inet6 3ff0::55:a00:20ff:fe8e:f3ad/64
```

Даже если информация о префиксах из сети не поступает, можно присвоить локальные и глобальные адреса вручную, например:

```
example% ifconfig le0 inet6 addif fec0::55:a00:20ff:fe8e:f3ad/64 up
example% ifconfig le0 inet6 addif 3ff0::55:a00:20ff:fe8e:f3ad/64 up
```

Чтобы задать стандартные значения для интерфейса **le0** во время загрузки, поместите следующие записи в файл **/etc/hostname6.le0**:

```
addif fec0::55:a00:20ff:fe8e:f3ad/64 up
addif 3ff0::55:a00:20ff:fe8e:f3ad/64 up
```

Локальные адреса связи используются только для взаимодействия в локальной сети и в других подсетях не видны.

Конфигурирование тоннелей IPv6/IPv4

Тоннельный интерфейс IPv6 поверх IPv4 позволяет посылать и получать пакеты IPv6, инкапсулированные в пакеты IPv4. Создайте тоннели на обеих сторонах соединения, указывающие друг на друга. Тоннели IPv6 поверх IPv4 требуют указания для тоннеля исходных и целевых адресов IPv4 и IPv6. ОС Solaris 8 поддерживает как автоматическое, так и ручное конфигурирование тоннелей. Для автоматически конфигурируемых тоннелей используется IPv4-совместимый адрес IPv6. Ниже продемонстрировано автоматическое конфигурирование тоннеля:

```
example% ifconfig ip.atun0 inet6 plumb
example% ifconfig ip.atun0 inet6 tsrc <IPv4-address> \
    ::<IPv4 address>/96 up
```

где первый **IPv4-address** - это IPv4-адрес интерфейса, через который будет передаваться информация тоннеля, а второй IPv4-адрес, **::<IPv4-address>**, - соответствующий IPv4-совместимый адрес IPv6.

Вот пример создания сконфигурированного тоннеля:

```
example% ifconfig ip.tun0 inet6 plumb tsrc <my-ipv4-address> \
    tdst <peer-ipv4-address> up
```

Эта команда создает сконфигурированный тоннель между локальным **my-ipv4-address** и удаленным **peer-ipv4-address** адресами с соответствующими локальными адресами связи. Для тоннелей с глобальными адресами или *локальными адресами сайта* (site-local addresses), логические интерфейсы тоннеля должны конфигурироваться в следующем виде:

```
ifconfig ip.tun0 inet6 addif <my-v6-address> <peer-v6-address> up
```

Например:

```
example% ifconfig ip.tun0 inet6 plumb tsrc 109.146.85.57 \
    tdst 109.146.85.212 up
example% ifconfig ip.tun0 inet6 addif 2::45 2::46 up
```

Чтобы получить информацию о всех работающих и сконфигурированных интерфейсах IPv6, выполните следующую команду:

```
example% ifconfig -au6
ip.tun0: flags=2200851<UP, POINTOPOINT, RUNNING, MULTICAST, NONUD, IPv6>
mtu 1480 index 3
    inet tunnel src 109.146.85.57    tunnel dst 109.146.85.212
    inet6 fe80::6d92:5539/10 --> fe80::6d92:55d4
ip.tun0:1: flags=2200851<UP, POINTOPOINT, RUNNING, MULTICAST, NONUD, IPv6>
mtu 1480 index 3
    inet6 2::45/128 --> 2::46
```

ПРИМЕРЫ

Пример 1: Использование команды ifconfig

Если работчая станция не подключена к сети Ethernet, интерфейс **le0** следует пометить как отключенный следующим образом:

```
example% ifconfig le0 down
```

Пример 2: Выдача информации об адресах

Для получения информации об адресах всех интерфейсов используется следующая команда:

```
example% ifconfig -a
```

Пример 3: Переустановка широковещательного адреса

Для переустановки широковещательных адресов всех интерфейсов после правильной установки масок сетей, используется следующая команда:

```
example% ifconfig -a broadcast +
```

Пример 4: Изменение Ethernet-адреса

Для изменения Ethernet-адреса интерфейса **le0** используется следующая команда:

```
example% ifconfig le0 ether aa:1:2:3:4:5
```

Пример 5: Конфигурирование туннеля IP-in-IP

Для конфигурирования туннеля IP-in-IP, сначала проинициализируйте его с помощью следующей команды:

```
example% ifconfig ip.tun0 plumb
```

Затем сконфигурируйте его как двухточечный (point-to-point) интерфейс, передав исходный и целевой адреса туннеля:

```
example% ifconfig ip.tun0 myaddr mydestaddr tsrc another_myaddr \
    tdst a_dest_addr up
```

Все средства защиты туннеля необходимо конфигурировать одним вызовом **ifconfig**:

```
example% ifconfig ip.tun0 encr_auth_algs md5 encr_algs 3des
```

Пример 6: Конфигурирование защиты туннеля без указания предпочтительных алгоритмов

Чтобы запросить защиту, не указывая конкретных алгоритмов, задайте значения **any**:

```
example% ifconfig ip.tun0 encr_auth_algs any encr_algs any
```

Пример 7: Отключение защиты

Чтобы отключить всю защиту, укажите имя защищенного туннеля и **none** в качестве алгоритма:

```
example% ifconfig ip.tun0 auth_algs none
```

или

```
example% ifconfig ip.tun0 encr_algs none
```

ФАЙЛЫ

/etc/netmasks

данные о масках сетей

АТРИБУТЫ

Описание следующих атрибутов см. на странице справочного руководства **attributes(5)**:

/usr/sbin

ТИП АТТРИБУТА	ЗНАЧЕНИЕ АТТРИБУТА
Доступен в пакете	SUNWcsu
Уровень стабильности опций modlist , modinsert и modremove	Развиваются

/sbin

ТИП АТТРИБУТА	ЗНАЧЕНИЕ АТТРИБУТА
Доступен в пакете	SUNWcsr
Уровень стабильности опций modlist , modinsert и modremove	Развиваются

ССЫЛКИ

dhcinfo(1), **dhcagent(1M)**, **in.mpathd(1M)**, **in.routed(1M)**, **ndd(1M)**, **netstat(1M)**, **ethers(3SOCKET)**, **gethostbyname(3NSL)**, **getnetbyname(3SOCKET)**, **hosts(4)**, **netmasks(4)**, **networks(4)**, **nsswitch.conf(4)**, **attributes(5)**, **arp(7P)**, **ipsec(7P)**, **ipseccsp(7P)**, **tun(7M)**
System Administration Guide, Volume 3

ДИАГНОСТИКА

Команда **ifconfig** выдает соответствующие сообщения в случаях, когда:

- указанный **интерфейс** не существует
- затребованный адрес неизвестен
- пользователь без соответствующих привилегий пытается изменить конфигурацию интерфейса

ПРИМЕЧАНИЯ

Рекомендуется не использовать в качестве имен хостов имена **broadcast**, **down**, **private**, **trailers**, **up** и другие, совпадающие с именами опций. Выбор любой опции **ifconfig** в качестве имени хоста может привести к странным и трудно выявляемым проблемам.

Последнее изменение: 25 октября 2000 года

Copyright (no c) - **Fuck copyright!**, 2003 В. Кравчук, [OpenXS Initiative](#), перевод на русский язык