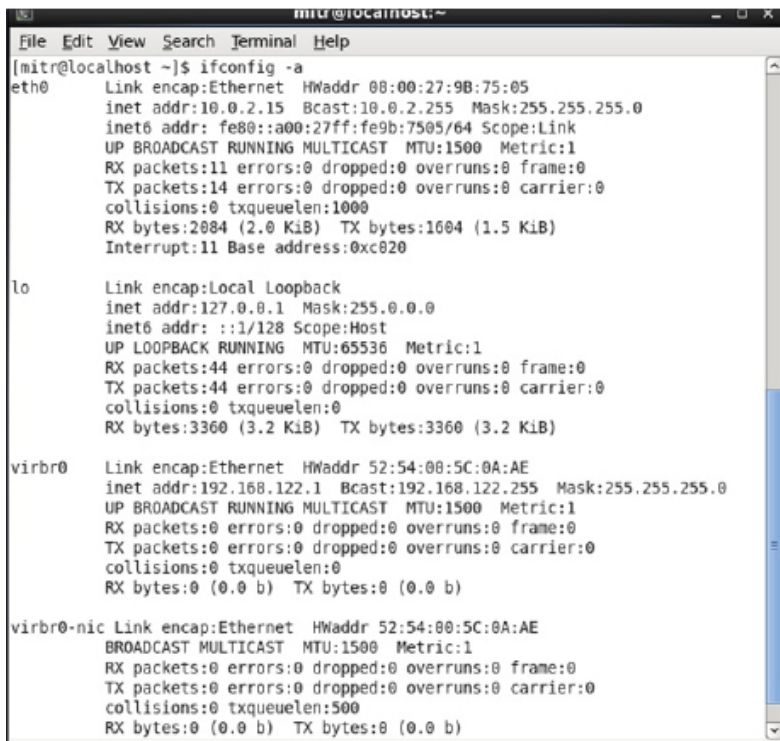


Основы системного администрирования и сетевых технологий

УРОК №11

Получение сведений о текущих сетевых настройках



```

[mitr@localhost ~]$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:9B:75:05
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe9b:7505/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2084 (2.0 KiB)  TX bytes:1604 (1.5 KiB)
          Interrupt:11 Base address:0xc020

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3360 (3.2 KiB)  TX bytes:3360 (3.2 KiB)

virbr0    Link encap:Ethernet  HWaddr 52:54:00:5C:0A:AE
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

virbr0-nic Link encap:Ethernet  HWaddr 52:54:00:5C:0A:AE
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
  
```

Рисунок 1.
Пример вывода утилиты ifconfig

Утилита ifconfig (interface configuration) используется для настройки любых сетевых интерфейсов, установленных на сетевом узле. Помимо настройки интерфейса с ее помощью можно получить сведения об интерфейсе. На рис. 1 приводится пример вывода утилиты ifconfig -a. Опция -a позволяет получить сведения и о соединениях, которые не работают. Часть данных ifconfig получает при обращении с помощью системного вызова к открытому сетевому сокету, а некоторые считывает из /proc.

Сеть начинается в месте подключения к среде передачи данных, то есть на сетевом интерфейсе. Название сетевого интерфейса состоит из его типа и порядкового номера, который определяется тем, каким по счету его распознало ядро. Все сетевые интерфейсы Ethernet в Linux называются ethно-номер, начиная с eth0.

Как видно из рис. 1, интерфейс eth0 является интерфейсом инкапсуляции линии связи (Link encap) типа Ethernet. Аппаратный адрес (HWaddr) – это MAC-адрес (Media Access Control – управление доступом к среде, состоит из 6 байтов, которые записывают в шестнадцатеричной системе счисления; каждая Ethernet-карта имеет собственный уникальный MAC-адрес, поэтому он используется для определения отправителя и получателя в рамках одной среды Ethernet. Если идентификатор получателя неизвестен, то используется аппаратный широковещательный адрес FF:FF:FF:FF:FF:FF. Сетевая карта, получив широковещательный фрейм или фрейм, MAC-адрес получателя в котором совпадает с ее MAC-адресом, отправляет его на обработку системе), inet addr – ip4-адрес, Bcast – широковещательный адрес, Mask

Получение сведений о текущих сетевых настройках

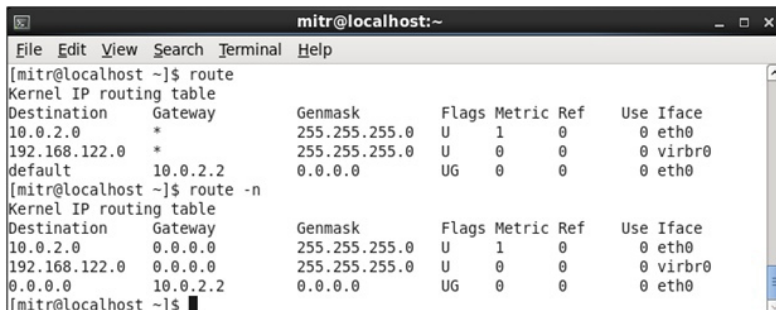
– маска сети, `inet6` – `ip6`-адрес. Параметр MTU (Maximum Transfer Unit) определяет наибольший размер фрейма. Большая часть остальных сведений об интерфейсе является статистической. Исключение составляет последняя строка, в которой указывается базовый аппаратный адрес и канал прерывания IRQ.

Интерфейс `lo` (`loopback`, локальный сетевой узел, который используется для организации сетевых взаимодействий компьютера с самим собой) соответствует адресу `localhost`. Любая система обладает интерфейсом `localhost`, которому соответствует адрес `127.0.0.1`.

Когда компьютер с некоторым IP-адресом решает отправить пакет другому компьютеру, он выясняет, принадлежит ли адресат той же локальной сети, что и отправитель. Для этого на IP-адрес получателя накладывается сетевая маска и вычисляется адрес сети, которой принадлежит получатель. Если этот адрес совпадает с адресом сети отправителя, значит оба находятся в одной локальной сети. Это в свою очередь означает, что аппаратный адрес получателя должен быть известен отправителю. MAC-адреса компьютеров локальной сети хранятся в специальной таблице ядра, называемой таблицей ARP. Просмотреть содержимое этой таблицы можно с помощью команды `arp -a`.

ARP-таблица отражает соответствие между IP- и MAC-адресами. Это динамическая таблица, устаревшие соответствия из нее удаляются. Для установления соответствия между адресами сетевого и интерфейсного уровня используется протокол ARP (Address Resolution Protocol – протокол разрешения адресов). Для преобразования IP в MAC он работает

Получение сведений о текущих сетевых настройках



```
mitr@localhost:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.2.0 * 255.255.255.0 U 1 0 0 eth0
192.168.122.0 * 255.255.255.0 U 0 0 0 virbr0
default 10.0.2.2 0.0.0.0 UG 0 0 0 eth0
mitr@localhost:~$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.2.0 0.0.0.0 255.255.255.0 U 1 0 0 eth0
192.168.122.0 0.0.0.0 255.255.255.0 U 0 0 0 virbr0
0.0.0.0 10.0.2.2 0.0.0.0 UG 0 0 0 eth0
mitr@localhost:~$
```

Рисунок 2.
Вывод команды route
без опций и с опцией -n

следующим образом: отправляется широковещательный фрейм типа «ARP-запрос», содержащий IP-адрес получателя. Компьютер, адрес которого указан в запросе, возвращает отправителю пустой фрейм типа «ARP-ответ», в поле «отправитель» которого указан MAC-адрес отправителя. На основании полученных данных заполняется ARP-таблица.

В том случае, когда компьютер, которому надо отправить сообщение, не находится в локальной сети компьютера-отправителя, то пакет нужно отправить какому-то абоненту локальной сети, чтобы он перенаправил его дальше. Этот абонент, маршрутизатор, подключен к нескольким сетям и пересылает пакеты между ними по определенным правилам. Сведения о таблице маршрутизации могут быть получены с помощью утилиты route. По умолчанию route отображает информацию с указанием символьных имен, однако в большинстве случаев IP-адреса оказываются более информативными, поэтому часто route используется с опцией -n. Вывод команды route без опций и с опцией -n показан на рис. 2. Это вывод для компьютера, который не выполняет функции маршрутизатора. Аналогичный вывод маршрутизатора был бы значительно сложнее.

Поле «Destination» - адрес сети назначения.

Поле «Gateway» - адрес следующего маршрутизатора.

Поле «Flags» описывает состояние маршрута. Для описания состояния используются следующие символы:

- U – маршрут активен и работоспособен (Up);

- H – признак специфического маршрута к определенному хосту. Маршрут ко всей сети, к которой принадлежит данный

Получение сведений о текущих сетевых настройках

хост, может отличаться от данного маршрута;

- G – означает, что маршрут пакета проходит через промежуточный маршрутизатор (gateway). Отсутствие этого флага отмечает непосредственно подключенную сеть;
- D – означает, что маршрут получен из сообщения Redirect (перенаправление) протокола ICMP. Этот флаг может присутствовать только в таблице маршрутизации конечного узла. Признак означает, что конечный узел в какой-то предыдущей передаче пакета выбрал не самый рациональный следующий маршрутизатор на пути к данной сети, и этот маршрутизатор с помощью протокола ICMP сообщил, что все последующие пакеты к данной сети нужно отправлять через другой следующий маршрутизатор. Протокол ICMP может посылать сообщения только узлу-отправителю, поэтому у промежуточного маршрутизатора этот признак встретиться не может. Признак никак не влияет на процесс маршрутизации, он только указывает администратору на источник появления записи;
- ! (Reject - отказ) – обозначает маршрут, через который не проходят пакеты. Если посылается пакет узлу или сети, которые помечены !, то будет получено сообщение, что маршрут не активен (Down). Это не означает, что по этому маршруту пакеты не могут передаваться в обратном направлении.

Поле «Ref» показывает, сколько раз на данный маршрут ссылались при продвижении пакетов.

Поле «Use» отражает количество байтов, переданных по данному маршруту.

Получение сведений о текущих сетевых настройках

Сеть 0.0.0.0 – ни содержит ни одного бита на сетевую маску, поэтому ей принадлежат любые адреса. Такая запись в таблице маршрутизации называется «маршрут по умолчанию». Если маршрут по умолчанию не задан, то попытка связаться с удаленным компьютером может закончиться ошибкой «No route to host», поскольку система не сможет определить, кому переслать пакет.

Получив IP-пакет, система последовательно сравнивает поле назначение пакета с записями в таблице маршрутизации. Если сеть адресата совпадает с сетью из таблицы маршрутизации, то пакет пересылается по адресу, указанному в поле «Gateway». Этот адрес используется вместо поля адресата и поиск возобновляется с начала таблицы. Если поле «Gateway» - нулевое, значит абонент находится в локальной сети и пакет надо передать на уровень ниже. Если ни одна сеть не подходит, выдается сообщение об ошибке.

Протокол ICMP

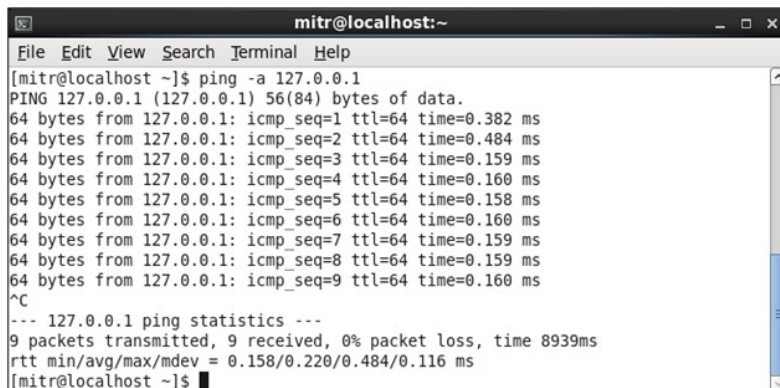
Протокол ICMP (Internet Control Message Protocol) предназначен для передачи служебных сообщений. В отличие от протоколов TCP и UDP пакеты протокола ICMP не передаются через сеть в составе пакетов IP. Протокол ICMP обладает собственным заголовком пакета. Этим протоколом пользуются, в частности, известные утилиты ping и traceroute. Пакеты ICMP могут быть нескольких типов, пакеты некоторых типов могут нести коды нескольких сообщений. В табл. 1 приведены некоторые типы ICMP пакетов и коды сообщений.

Таблица 1. Некоторые типы ICMP пакетов и коды сообщений

Тип пакета ICMP	Описание	Коды сообщений
0	Echo Reply Message (эхо-ответ)	0
3	Destination Unreachable Message (Приемник недоступен)	0 – сеть недоступна; 1- сетевой узел недоступен; 2 – протокол недоступен; 3 – порт недоступен; 4 – необходима фрагментация и флаг DF (Don't Fragment – не фрагментировать) установлен; 5 – исходный маршрут не работает.
4	Source Quench Message (сообщение об отключении источника при перегрузке с предварительным возвратом сообщения)	0

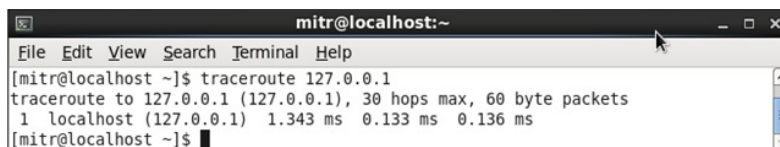
Тип пакета ICMP	Описание	Коды сообщений
5	Redirect Message (сообщение о перенаправлении)	0 – перенаправление дейтаграммы для сети; 1 - перенаправление дейтаграммы для сетевого узла; 2 – перенаправление дейтаграммы для типа службы и сети; 3 - перенаправление дейтаграммы для типа службы и сетевого узла
8	Echo Message (эхо-запрос)	0
11	Time Exceeded Message (время истекло)	0 – при передаче сообщения истекло время жизни (time to live); 1 – истекло время формирования сообщения из фрагментов (fragment reassembly time)
12	Parameter Problem Message (проблема с параметром)	0 – проблема, связанная с параметром
13	Timestamp Message (отметка времени)	0
14	Timestamp Reply Message (ответ на отметку времени)	0
15	Information Request Message (запрос информации)	0
16	Information Reply Message (ответ на запрос информации)	0

Протокол ICMP



```
mitr@localhost:~$ ping -a 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.382 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.484 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.159 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.160 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.158 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.160 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.159 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.159 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.160 ms
^C
--- 127.0.0.1 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8939ms
rtt min/avg/max/mdev = 0.158/0.220/0.484/0.116 ms
mitr@localhost ~$
```

Рисунок 3.
Использование ping



```
mitr@localhost:~$ traceroute 127.0.0.1
traceroute to 127.0.0.1 (127.0.0.1), 30 hops max, 60 byte packets
 1 localhost (127.0.0.1) 1.343 ms 0.133 ms 0.136 ms
mitr@localhost ~$
```

Рисунок 4.
Использование traceroute

Для проверки доступности сетевого ресурса можно использовать команду `ping`, имеющую следующий синтаксис:

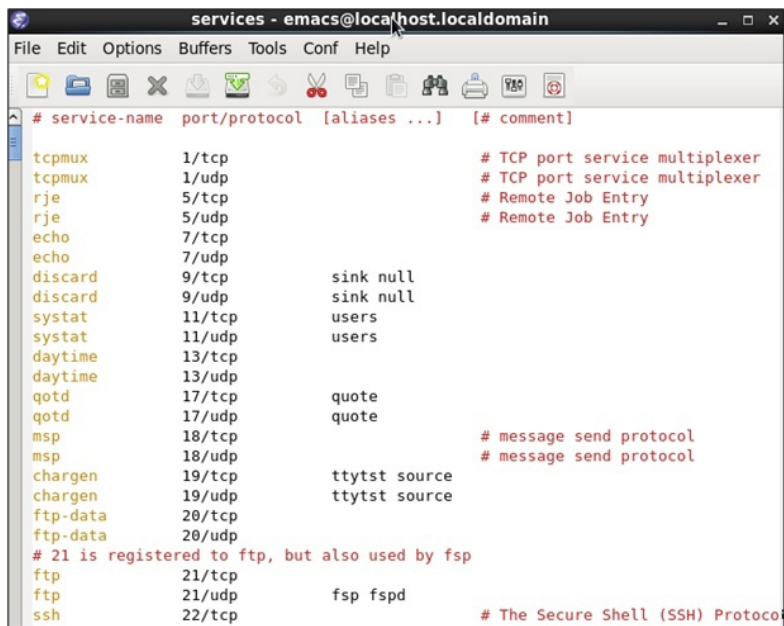
ping [опции] адрес_назначения

Все опции команды `ping` можно просмотреть в справочной системе. Наиболее часто используется опция `-a`. Пример использования `ping` приведен на рис. 3.

Утилита `traceroute` выводит список узлов, через которые проходит пакет по пути к адресату. Это приблизительный список. Дело в том, что первым трем пакетам (по умолчанию `traceroute` шлет пакеты по три) в поле TTL (Time To Live – время жизни) устанавливается значение 1. Каждый маршрутизатор должен уменьшать это значение на 1, и если оно обнулиться, то передать отправителю ICMP-пакет о том, что время жизни закончилось, а адресат пакета еще не найден. Таким образом, на первую серию пакетов отреагирует первый маршрутизатор, и `traceroute` выдаст первую строку маршрута. Второй пакет (вторая серия из трех пакетов) посылается с TTL=2 и, если за две пересылки адресат не достигнут, об этом сообщит второй маршрутизатор. Процесс продолжается до тех пор, пока очередной пакет не достигнет места назначения. Строго говоря, неизвестно, каким маршрутом шла очередная группа пакетов, потому что с тех пор, как посылалась предыдущая группа, какой-нибудь из промежуточных маршрутизаторов мог послать пакеты и другим путем.

На рис. 4 показан пример использования `traceroute`.

Протоколы транспортного уровня



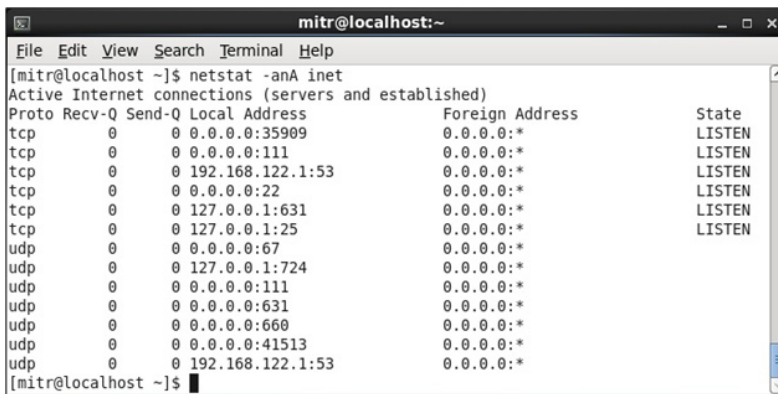
```
# service-name port/protocol [aliases ...] [# comment]
tcpmux      1/tcp          # TCP port service multiplexer
tcpmux      1/udp          # TCP port service multiplexer
rje         5/tcp          # Remote Job Entry
rje         5/udp          # Remote Job Entry
echo        7/tcp
echo        7/udp
discard     9/tcp          sink null
discard     9/udp          sink null
systat      11/tcp         users
systat      11/udp         users
daytime     13/tcp
daytime     13/udp
qotd        17/tcp          quote
qotd        17/udp          quote
msp         18/tcp          # message send protocol
msp         18/udp          # message send protocol
chargen     19/tcp          ttytst source
chargen     19/udp          ttytst source
ftp-data    20/tcp
ftp-data    20/udp
# 21 is registered to ftp, but also used by fsp
ftp         21/tcp
ftp         21/udp          fsp fspd
ssh         22/tcp          # The Secure Shell (SSH) Protocol
```

Рисунок 5.
Фрагмент файла /etc/services

Если задачей уровня межсетевого взаимодействия, к которому относится протокол IP, является передача данных между любой парой сетевых интерфейсов в составной сети, то задача транспортного уровня, которую решают протоколы TCP и UDP, заключается в передаче данных между любой парой прикладных процессов, выполняющихся в сети. После того, как пакет средствами протокола IP доставлен на сетевой интерфейс компьютера-получателя, данные необходимо направить конкретному процессу-получателю. Каждый компьютер может выполнять несколько процессов, а каждый прикладной процесс может иметь несколько точек входа, выступающих в качестве адресов назначения для пакетов данных.

Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. В терминологии TCP/IP такие очереди называются *портами*. Порт однозначно определяет приложение в пределах компьютера. Существуют два способа присвоения порта приложению – централизованный и локальный. За каждым из способов закреплен свой диапазон номеров портов: для централизованного - от 0 до 1023, для локального – от 1024 до 65535. Если процессы представляют собой популярные общедоступные службы, такие как FTP, HTTP, DNS и др., то за ними закрепляются стандартные присвоенные номера, часто называемые хорошо известными номерами. Хорошо известные номера и портов и названия соответствующих служб хранятся в файле /etc/services (рис. 5)

Протоколы транспортного уровня



```
mitr@localhost:~$ netstat -anA inet
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 0.0.0.0:35909           0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp      0      0 192.168.122.1:53        0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
udp      0      0 0.0.0.0:67              0.0.0.0:*
udp      0      0 127.0.0.1:724           0.0.0.0:*
udp      0      0 0.0.0.0:111             0.0.0.0:*
udp      0      0 0.0.0.0:631             0.0.0.0:*
udp      0      0 0.0.0.0:660             0.0.0.0:*
udp      0      0 0.0.0.0:41513           0.0.0.0:*
udp      0      0 192.168.122.1:53        0.0.0.0:*
mitr@localhost ~$
```

Рисунок 6.
Вывод команды `netstat -anA inet`

В файл `/etc/services` можно добавлять новые службы и уникальные номера портов, однако не следует в этот файл добавлять службы, которые в нем уже существуют. Относительно номеров портов в Linux нужно сделать важное замечание – порты с номерами ниже 1024 рассматриваются как привилегированные. Только root может выполнять связывание служб с этими номерами.

Получить список установленных на компьютере соединений, а также служб-обработчиков можно с помощью команды `netstat` (рис. 6).

По умолчанию `netstat` выводит только список открытых соединений. Если при вызове добавить опцию `-a`, то будут показаны все соединения.

Первая строка, отображаемая утилитой `netstat`, сообщает о том, какие сведения отображаются в данный момент утилитой. В приведенном примере в первой строке указано «*Active Internet connections (servers and established)*», то есть активные соединения с Internet. Термин «server» в данном случае означает программы, открывшие порт и ожидающие поступления запросов на соединение. Такие программы слушают в очереди ожидания, однако они еще не обмениваются данными с клиентом.

Вторая строка – заголовок для раздела вывода `netstat`.

«Proto» - протокол. Часто в этом столбце отображаются `tcp` и `udp`, однако могут быть и другие протоколы, например `raw`. В столбцах «Recv-Q» (Receive Queue – очередь приема) и «Send-Q» (Send Queue – очередь на передачу) отображается соответственно количество пакетов, принятых из сети, но не скопированных пользовательской программой, подключенной

Протоколы транспортного уровня

к сокету, и количество байтов, прием которых не подтвержден удаленной системой. После того, как сокет TCP закрывается, как правило, удаленный узел не подтверждает прием одного пакета.

В столбцах локальный (Local) и удаленный (Foreign) IP-адреса указываются адреса, между которыми устанавливается соединение. Адреса указываются в формате *адрес:порт*.

Протокол TCP подключает сервер к клиенту таким образом, что данные посылаются по одному порту, а принимаются по другому. Порт возврата жестко не задан; это может быть любой незанятый порт с номером выше 1024.

В процессе первого соединения клиент сообщает серверу номер выбранного порта и ожидает поступления на этот порт пакета SYN-ASC от сервера. Если сервер не может использовать этот порт, потому что он занят, клиенту передается сообщение ICMP Type 3 Code 3 «Port Unreachable» - порт недоступен. После этого клиент случайным образом выбирает другой порт. Когда клиент через выбранный им порт принимает от сервера пакет SYN, он возвращает серверу пакет ACK. После этого соединение считается установленным и начинается передача данных.

Когда соединение разрывается, выполняется похожая последовательность процедур.

Протоколы транспортного уровня

```

mitr@localhost:~$ netstat -anA inet -e
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User        Inode
tcp        0      0 0.0.0.0:35989          0.0.0.0:*               LISTEN      29          12310
tcp        0      0 0.0.0.0:111            0.0.0.0:*               LISTEN      0           11990
tcp        0      0 192.168.122.1:53       0.0.0.0:*               LISTEN      0           14379
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      0           13031
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      0           12438
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      0           13241
udp        0      0 0.0.0.0:67             0.0.0.0:*               0           14371
udp        0      0 127.0.0.1:724         0.0.0.0:*               0           12293
udp        0      0 0.0.0.0:111           0.0.0.0:*               0           11987
udp        0      0 0.0.0.0:631           0.0.0.0:*               0           12441
udp        0      0 0.0.0.0:600            0.0.0.0:*               0           11989
udp        0      0 0.0.0.0:41513          0.0.0.0:*               29          12306
udp        0      0 192.168.122.1:53       0.0.0.0:*               0           14380
  
```

Рисунок 7.

Вывод команды `netstat -anA inet -e`

```

File Edit View Search Terminal Help
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type        State
unix   19      [ ]        DGRAM
unix   2       [ ]        DGRAM
unix   2       [ ]        DGRAM
unix   3       [ ]        STREAM     CONNECTED
unix   3       [ ]        STREAM     CONNECTED
  
```

Рисунок 8.

Вывод команды `netstat -unix`

В столбце «State» (состояние) указывается состояние соединения. В столбе могут отображаться следующие состояния:

- ESTABLISHED;
- SYN_SENT;
- SYN_RECV;
- FIN_WAIT1;
- FIN_WAIT2;
- TIME_WAIT;
- CLOSED;
- CLOSE_WAIT;
- LAST_ACK;
- LISTEN;
- CLOSING;
- UNKNOWN.

Понятие состояния для протокола UDP не имеет смысла, Если к команде `netstat` добавить опцию `-e`, то к выводу команды добавляется столбец «User» (пользователь), в котором будет указываться UID пользователя, от имени которого запущен процесс, создавший соединение (рис. 7).

Для того, чтобы просмотреть активные доменные сокеты UNIX, используется команда `netstat -unix` (рис. 8).

В выводе присутствуют заголовки:

- Proto – протокол; для сокетов Unix, единственным допустимым значением является `unix`.
- RefCnt – счетчик ссылок; указывается количество подсоединенных процессов.

Протоколы транспортного уровня

- Flags – набор флагов; как правило, это поле пусто, однако, если поле RefCnt равно 0 и соответствующие процессы ожидают поступление запросов на соединение, поле флагов может быть равно ACC (SO_ACCEPTION - принятие), что означает, что сокет готов к приему запросов на соединение. В некоторых ситуациях могут возникать и другие флаги, например W (S)_WAITDATA – ожидание данных) и N (SO_NOSPACE – нет места).
- Type – тип сокета. Как правило, стоит метка STREAM (сокет с созданием соединения), однако, могут быть и другие метки: DGRAM (сокет без создания соединения), RAW (сокет передачи данных без транспортного протокола), RDM (Reliably Delivered Message – надежно передаваемое сообщение), SEQPACKED (Sequential Packet – последовательно передаваемый пакет), PACKET (пакет простого доступа к интерфейсу), UNKNOWN (неизвестный тип сокета для будущих усовершенствований).
- State – состояние сокета; могут использоваться следующие метки: FREE (сокет свободен), LISTENING (ожидает поступления запроса), CONNECTING (соединение устанавливается), CONNECTED (соединение установлено), DISCONNECTING (соединение разрывается), UNKNOWN (неизвестное состояние). Поле может быть и пустым. При нормальном функционировании системы сокет не может находиться в состоянии UNKNOWN.
- I-Node – номер дескриптора I-Node, соответствующий соединению. I-Node существует в каталоге /rproc только в том случае, если соединение используется.
- Path – процесс, подключенный к сокету.

Основы системного администрирования и сетевых технологий
