

Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)**

Ю.С. Белов, Е.А. Черепков

СЕТЕВЫЕ ИНТЕРФЕЙСЫ В LINUX

Методические указания к выполнению лабораторной работы
по курсу «Операционные системы»

Калуга – 2018

УДК 004.62
ББК 32.972.1
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий».

Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № 3 от «24» октября 2018 г.

Зав. кафедрой ИУ4-КФ

 к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ИУ-КФ протокол № 3 от «29» сентября 2018 г.


Председатель методической
комиссии факультета ИУ-КФ

 к.т.н., доцент М.Ю. Адкин

- Методической комиссией

КФ МГТУ им.Н.Э. Баумана протокол № 2 от «6» мая 2018 г.

Председатель методической комиссии
КФ МГТУ им.Н.Э. Баумана

 д.э.н., профессор О.Л. Перерва

Рецензент:

к.т.н., доцент кафедры ИУ3-КФ

 А.В. Фиошин

Авторы

к.ф.-м.н., доцент кафедры ИУ4-КФ
ассистент кафедры ИУ4-КФ

 Ю.С. Белов
 Е.А. Черепков

Аннотация

Методические указания к выполнению лабораторной работы по курсу «Операционные системы» содержат общие сведения о протоколах стека TCP/IP, адресации в сетях и конфигурировании сетевых интерфейсов в ОС семейства Linux.

Предназначены для студентов 3-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2018 г.
© Ю.С. Белов, Е.А. Черепков, 2018 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ	6
СЕТИ ТСР/Р	12
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ	35
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ	36
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ	37
ОСНОВНАЯ ЛИТЕРАТУРА.....	38
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА	38

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Операционные системы» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 3-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание команд для настройки сетевого интерфейса, и работы с сетью в ОС Linux.

Методические указания составлены для ознакомления студентов с работой с сетевыми интерфейсами, файлами конфигурации, командами проверки сетевого интерфейса. Для выполнения лабораторной работы студенту необходимы минимальные навыки программирования и знания об операционной системе Linux.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является приобретение практических навыков по настройке сетевого интерфейса в ОС Linux.

Основными задачами выполнения лабораторной работы являются:

1. Узнать, что такое IP-адрес и маска сети.
2. Получить навыки работы с командами для настройки сетевого интерфейса в ОС Linux.

Результатами работы являются:

1. Демонстрация выполнения команд по настройке сетевого адаптера.
2. Подготовленный отчет.

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Операционная система Linux изначально разрабатывались для работы с сетями, использующими протоколы TCP/IP. Именно эти протоколы применяются в глобальной сети Internet и во многих локальных сетях.

Под аббревиатурой TCP/IP понимается большая группа протоколов, предназначенных для обеспечения устойчивой связи между компьютерами с разными операционными системами и аппаратными средствами. Эти протоколы были разработаны в 70-е годы в рамках специального проекта, осуществленного Управлением перспективными исследованиями и разработками Министерства обороны США (DAPRA).

Целью проекта было получение надежной системы связи между университетами и научными центрами. Первые протоколы разрабатывались для применения в системах UNIX, причем основные научные исследования проводились в университете города Беркли (штат Калифорния). Операционная система Linux во многом выигрывает благодаря ориентации на использование протоколов UNIX. В настоящее время усовершенствование протоколов TCP/IP проходит под управлением группы Internet Engineering Task Force (IETF), которая, в свою очередь, контролируется организацией Internet Society (ISOC). ISOC контролирует несколько групп, ответственных за различные разработки в области Internet, таких как, например, агентство Internet Assigned Numbers Authority (IANA). Эта группа разрабатывает систему адресации в Internet ([табл. 1](#)). Через несколько лет после начала работы стандарты и документация по TCP/IP были выпущены в виде документов Requests/or Comments (RFC). С последней версией этой разработки можно ознакомиться на Web-узле IETF по адресу www.ietf.org.

Таблица 1. Группы по разработке протоколов TCP/IP

Группа	Название	Описание
ISOC	Internet Society	Профессиональная организация Экспертов Internet, регулирующая политику сетевой обработки.
IESG	The Internet Engineering Steering Group	Группа технического контроля за деятельностью IETF и принятием стандартов Internet
IANA	Internet Assigned Numbers Authority	Группа, отвечающая за присвоение адресов Internet (IP-адреса).
IAB	Internet Architecture Board	Группа по общим вопросам архитектуры Internet, осуществляет общее руководство и определяет направления деятельности IETF.
IETF	Internet Engineering Task Force	Группа по разработке протоколов, стандартов и проектированию Internet;

Набор протоколов TCP/IP реализуется в виде многих программ, каждая из которых выполняет в сети определенную задачу. Основными являются три протокола: протокол управления передачей (*Transmission Control Protocol — TCP*), обеспечивающий отправку и прием сообщений; протокол передачи данных (*Internet Protocol — IP*), отвечающий за процесс передачи данных от одного компьютера к другому, и протокол пользовательских дейтаграмм (*User Datagram Protocol -UDP*), поддерживающий отправку и прием пакетов. Протокол IP является базовым и обеспечивает работу других протоколов. Он поддерживает передачу сообщений, обрабатывает пакеты данных вместе с информацией об отправителе и получателе каждого пакета. Протокол TCP создан для обработки связанных сообщений или данных. При пересылке данных он разбивает всю информацию на отдельные пакеты, располагая пакеты в определенном порядке. При

приеме данных протокол TCP выполняет проверку полученных пакетов и сортировку их в том порядке, в котором они были переданы, с целью восстановления исходного сообщения.

Протокол UDP функционирует на более высоком уровне. Как и TCP, UDP также разбивает данные на пакеты, но не следит за их порядком. Набор TCP/IP рассчитан на создание стабильной и надежной связи, когда все переданные данные доходят до адресата и собираются в исходном порядке. Протокол UDP предназначен только для передачи данных любым способом и не гарантирует, что пакеты будут получены и размещены в необходимом порядке. Протокол UDP часто используется при передаче больших объемов данных, для которых потеря нескольких пакетов не имеет большого значения (например, это могут быть промежуточные изображения, видео и баннеры, которые отображаются в Internet).

Другие протоколы предназначены для различных сетевых и пользовательских служб. Служба доменных имен (*Domain Name Service— DNS*) обеспечивает преобразование адресов. Протокол *FTP* (*File Transfer Protocol*) обеспечивает передачу файлов, а протокол *NFS* (*Network File System*) предоставляет доступ к удаленным файловым системам. В табл. 2 перечислены различные протоколы, образующие набор протоколов TCP/IP. Эти протоколы определяют применение TCP либо UDP для отправки и приема пакетов, в то время как протокол IP выполняет фактическую передачу пакетов.

Таблица 2. Набор протоколов TCP/IP

Транспорт	Описание
TCP	Transmission Control Protocol (протокол управления передачей); обеспечивает непосредственное взаимодействие между сетевыми устройствами

Продолжение таблицы 2

Транспорт	Описание
UDP	User Datagram Protocol (протокол пользовательских Дейтаграмм)
IP	Internet Protocol (протокол Internet); передает данные
ICMP	Internet Control Message Protocol (протокол управления сообщениями Internet)
Маршрутизация	Описание
RIP	Routing Information Protocol (протокол маршрутизации информации); определяет маршрут ,
OSPF	Open Shortest Path First (поиск кратчайшего пути); определяет кратчайший маршрут
Сетевые адреса	Описание
ARP	Address Resolution Protocol (протокол утверждения адресов); определяет уникальные IP-адреса систем
DNS	Domain Name Service (Служба доменных имен); преобразует имена хостов в IP-адреса ;
RARP	Reverse Address Resolution Protocol (протокол преобразования обратных адресов); определяет адреса систем
Пользовательские службы	Описание
FTP	File Transfer Protocol (протокол передачи файлов); передает файлы из одной системы в другую, используя TCP

Продолжение таблицы 2

Пользовательские службы	Описание
TFTP	Trivial File Transfer Protocol (простой протокол передачи файлов); обеспечивает передачу файлов с помощью UDP
TELNET	Обеспечивает удаленную регистрацию в другой системе сети
SMTP	Simple Mail Transfer Protocol (простой протокол передачи почты); обеспечивает передачу электронной почты между системами
RPC	Remote Procedure Call (вызов удаленных процедур); обеспечивает взаимодействие между программами в удаленных системах
Шлюз	Описание
EGP	Exterior Gateway Protocol (протокол внешнего, шлюза); поддерживает маршрутизацию для внешних сетей
GGP	Gateway-to-Gateway Protocol (межшлюзовой протокол); поддерживает маршрутизацию между шлюзами Internet
IGP	Interior Gateway Protocol (протокол внутреннего шлюза); поддерживает маршрутизацию для внутренних сетей
Сетевые службы	Описание
NFS	Network File System (сетевая файловая система); обеспечивает монтирование файловых систем на удаленных компьютерах

Продолжение таблицы 2

Сетевые службы	Описание
NIS	Network Information Service (сетевая информационная служба); поддерживает в сети пользовательские учетные записи
BOOTP	Boot Protocol (протокол загрузки); запускает систему, используя загрузочную информацию, с сетевого сервера
SNMP	Simple Network Management Protocol (простой протокол сетевого управления); поддерживает передачу сообщений о конфигурации TCP/IP .
DHCP	Dynamic Host Configuration Protocol (протокол динамического конфигурирования хоста); автоматически предоставляет информацию для конфигурирования хост-систем

В сети TCP/IP сообщения разбиваются на небольшие компоненты, именуемые дейтаграммами, которые передаются посредством многочисленных взаимосвязанных маршрутов и затем доставляются на целевые компьютеры. После получения дейтаграммы упорядочиваются, и таким образом происходит восстановление исходного сообщения. Следует отметить, что дейтаграммы также могут разбиваться на пакеты, имеющие меньшие размеры. *Пакет* представляет собой физическую единицу сообщения, которая фактически передается по сети. Пересылка сообщения, состоящего из небольших компонентов, осуществляется намного быстрее и надежнее, чем в случае одной большой посылки. Например, если один из небольших составляющих компонентов будет поврежден или утерян, понадобится повторная пересылка только этого компонента. Если же будет утеряна или повреждена часть большой посылки, понадобится повторная передача всей посылки.

СЕТИ TCP/IP

Конфигурация сети TCP/IP в системе Linux осуществляется с помощью набора файлов-конфигураций сети. Для редактирования многих из этих файлов используются административные программы, такие как **Linuxconf** или [netcfg](#). Можно также воспользоваться специализированными программами, в частности **netstat**, **ifconfig** и **route**. Некоторые файлы конфигурации легко модифицируются с помощью текстового редактора.

Сети TCP/IP могут конфигурироваться и управляться с помощью следующих утилит: [ifconfig](#), **route** и **netstat**. Утилита **ifconfig** запускается с рабочего стола пользователя root и позволяет полностью сконфигурировать сетевые интерфейсы, добавляя новые и изменяя существующие модули. Утилиты **ifconfig** и **route** являются программами низкого уровня, для эффективного использования которых требуются более глубокие знания о сети. С помощью утилиты **netstat** можно получить информацию о статусе сетевых соединений.

Системы адресации IPv4 и IPv6

Традиционно адрес TCP/IP состоит из четырех сегментов, представляющих собой числа; сегменты отделяются друг от друга точками. Такая структура именуется *IP-адресом*. По сути, IP-адрес является 32-разрядным целым числом, отдельные разряды которого идентифицируют сеть и хост (любое устройство, подключенное к сети по протоколу TCP/IP; узловой компьютер). Данная форма IP-адресации соответствует протоколу Internet версии 4, известному как IPv4. Этот вид IP-адресации по-прежнему широко используется.

В настоящее время появилась новая версия протокола IP, именуемая протоколом Internet версии 6 (IPv6). Эта версия постепенно заменяет IPv4. Протокол IPv6 увеличивает количество возможных IP-адресов благодаря поддержке 128-разрядных адресов. Он полностью совместим с системами, использующими протокол IPv4. Адреса IPv6 выглядят по-другому. В данном случае используется набор из восьми 16-разрядных

сегментов, разделенными двоеточием. Каждый сегмент представлен в виде шестнадцатеричного числа. Пример такого адреса приведен ниже:

FEDC:0:0:200C:800:BA98:7654:3210

IPv6 позволяет за счет более простых заголовков повысить скорость соединения, а также поддерживает возможности шифрования и идентификации. При этом количество возможных адресов может достигать 2 в степени 128 (для сравнения: IPv4 поддерживает 4,2 млрд. адресов).

Адресация в сетях TCP/IP

Адрес в сетях TCP/IP, работающих по протоколу IPv4, состоит из четырех групп чисел, разделенных точками. Этот тип адреса широко используется до сих пор, и именно его называют *IP-адресом*. Одна часть IP-адреса применяется для получения адреса сети, другая — для идентификации конкретного интерфейсного устройства в данной сети. Следует помнить, что IP-адреса назначаются интерфейсным устройствам (сетевым картам Ethernet или модемам), но не хост-компьютерам. Обычно компьютер имеет только один интерфейс, по IP-адресу которого и осуществляется доступ к этому компьютеру. Таким образом, можно считать, что IP-адрес играет роль идентификатора хост-компьютера в сети. Поэтому IP-адрес часто называют *адресом хоста*.

Однако никто не запрещает любому хосту иметь несколько интерфейсных устройств, обладающих собственными IP-адресами. Это удобно в случае, когда компьютер служит в качестве шлюза или брандмауэра между локальной сетью и Internet. При наличии двух сетевых плат одна из них обычно обеспечивает подключение к локальной сети, а вторая — к Internet. Каждое интерфейсное устройство (карта Ethernet) имеет свой собственный IP-адрес. Например, программа Linuxconf позволяет назначить IP-адреса четырем сетевым картам. Если для подключения к провайдеру Internet используется модем, можно установить соединение по протоколу PPP, которому также присваивается IP-адрес (обычно назначаемый провайдером в

динамическом режиме). Это необходимо, учитывая, если планируется применять Linux для организации локальной или домашней сети и использовать компьютер с Linux в качестве шлюза для подключения к Internet.

Адрес сети

В IP-адресе выделяют две части: одна из них идентифицирует сеть, вторая — определяет конкретный хост. Адрес сети идентифицирует сеть, частью которой является конкретный хост. Существует два метода реализации составляющих частей IP-адреса: традиционная IP-адресация с использованием классов и адресация, основанная на бесклассовой междоменной маршрутизации (*Classless Interdomain Routing* — *CIDR*). IP-адресация на базе классов включает официально назначаемые части адреса для сети и хоста, в то время как адресация CIDR позволяет определять части адреса динамически, используя маску сети.

IP-адресация, основанная на классах

Изначально [IP-адреса](#) были организованы в соответствии с классами. В зависимости от размера все входящие в Internet сети разбиты на три класса: -А, В и С. Сети класса А используют первый сегмент для указания адреса сети, а оставшиеся три — для определения адреса хоста. Благодаря этому к одной сети можно подключить большое число компьютеров. Обратная картина наблюдается в сетях класса С. Здесь для идентификации сети применяются старшие три сегмента, а для идентификации хоста служит один, последний сегмент. Наличие классов сетей позволяет сформировать уникальный адрес, который может применяться для идентификации любого интерфейсного устройства, входящего в состав сети TCP/IP. Например, в IP-адресе 192.168.1.72 адрес сети имеет значение 192.168.1, а адрес хоста (интерфейса) равен 72. Интерфейсное устройство представляет собой часть сети, адрес которой 192.168.1.0.

В сети класса С три старших сегмента представляют адрес сети. Адрес сети делится на три части, каждая из которых идентифицирует подсеть. Глобальная сеть Internet делится на подсети, которые ранжируются в порядке уменьшения размеров. Последнее (правое) число в адресе идентифицирует отдельный, компьютер, называемый *хостом*. Internet можно представить в виде совокупности сетей, которые, в свою очередь, включают меньшие по размеру подсети. Так, адрес Internet 192.168.187.4 определяет, что четвертый компьютер подключен к сети, идентифицируемой номером 187. Сеть с номером 187, в свою очередь, является подсетью большей сети, номер которой равен 168. Эта большая сеть является подсетью сети с номером 192.

Маска сети

Для того чтобы отделить адрес сети от адреса хоста, используется маска сети. IP-адрес представляет собой 32-разрядное двоичное число, одни разряды которого идентифицируют сеть, а другие — хост. Маска сети также является 32-разрядным числом, в котором разряды, соответствующие адресу сети, имеют значение 1. Остальные разряды этого числа равны 0 (рис. 1). Таким образом, в маске для стандартного IP-адреса, основанного на классах, все байты сетевой части имеют значения 255, а байты, соответствующие идентификатору хоста, равны 0. В результате все двоичные разряды маски, отвечающие адресу сети, будут содержать значение 1. Например, для адреса 192.168.1.72 выбирается маска сети 255.255.255.0. Сетевой части адреса (192.168.1) отвечает значение 255.255.255; а хостовой части (72) — значение 0.

Если вы знакомы с булевой алгеброй, то знаете, что в результате поразрядного логического умножения (AND) маски сети и полного адреса разряды, соответствующие номеру хоста, получают значения 0. Таким образом, маска сети позволяет выделить сетевую часть адреса. Адрес можно представить в виде 4-байтового целого числа, где каждый байт соответствует сегменту адреса. При использовании адресов класса С три сетевых сегмента соответствуют трем старшим байтам, а сегмент хоста — младшему. Маска сети предназначена для маскирования

хостовой части адреса и выделения сетевых сегментов. В маске для сети класса С всем разрядам старших трех байтов присвоены единицы, а младший байт имеет нулевое значение. В этом случае младший байт маски маскирует хостовую часть адреса, а первые три байта, все разряды которых установлены в 1, позволяют выделить сетевую часть адреса. На рис. 1 проиллюстрировано, как выполняется поразрядное логическое умножение (AND) маски сети и адреса 192.168.1.4. В данном случае идет речь об адресе класса С и маске, которая состоит из 24 единиц (старшие три байта) и 8 нулей (младший байт). После логического умножения значений этой маски с адресом 192.168.1.4 остается сетевая часть адреса (192.168.1), а хостовая часть адреса (4) маскируется. В результате адрес 192.168.1.0 играет роль идентификатора сети.

Class-based Addressing				
IP Address 192.168.1.4				
	Network			Host
binary	11000000	10101000	00000001	00000100
numeric	192	168	1	4
Netmask 255.255.255.0				
binary	11111111	11111111	11111111	00000000
numeric	255	255	255	000
Network Address 192.168.1.0				
binary	11000000	10101000	00000001	00000000
numeric	192	168	1	0
Netmask Operation				
IP Address	11000000	10101000	00000001	00000100
Netmask	11111111	11111111	11111111	00000000
Net Address	11000000	10101000	00000001	00000000

Рис. 1. Операции с масками сети для адресов, основанных на классах

IP-адресация в формате CIDR

В настоящее время IP-адреса, основанные на [классах](#), вытесняются адресами с форматом CIDR (Classless Interdomain Routing). Этот формат предназначен для использования в сетях, которые по своему размеру занимают промежуточное положение между сетями класса С и сетями, где число хостов находится в диапазоне от 256 до 65534. В IP-адресах сетей класса С для идентификации хостов используется только один сегмент (8-разрядное двоичное число). Это означает, что в таких сетях количество адресов хостов не может быть больше 256. В IP-адресах сетей класса В для идентификации хостов применяются два сегмента, то есть 16-разрядное двоичное число, максимальное значение которого составляет 65534.

В схеме, основанной на классах, “единицей перераспределения” значений между адресом хоста и адресом сети служит 1 сегмент (1 байт).

С помощью же схемы адресации CIDR можно поразрядно определять размер хостовой и сетевой частей адреса. Например, воспользовавшись данной схемой, вы можете расширить сегмент хоста с 8 до 9 разрядов вместо того, чтобы переходить к 16-разрядным адресам класса В (два сегмента).

Синтаксис CIDR позволяет интегрировать в адрес информацию о маске сети (маска сети применяется к IP-адресу с целью выделения сетевой части адреса). Число разрядов, образующих адрес сети, просто указывается в конце IP-адреса (после символа “/”). Например, форма CIDR IP-адреса 192.168.187.4 для сети класса С имеет следующий вид

192.168.187.4 /24

Преимущество схемы адресации CIDR проявляется в том, что вместо трех сегментов для адресации сети можно применять любое число разрядов (конечно, в рамках допустимого). Так, можно создавать сети с адресами в 14, 22 или даже 25 разрядов. Для адреса хоста используются оставшиеся разряды. Например, если в IP-адресе сетевая часть занимает 21 разряд, то для идентификации хостов применяется 11 разрядов (диапазон чисел от 0 до 2047).

На рис. 2 показан пример адреса CIDR и его сетевой маски. IP-адрес 192.168.1.6 с маской сети в 22 разряда записывается как 192.168.1.6/22. Адрес сети занимает старшие 22 разряда IP-адреса, а оставшиеся 10 разрядов применяются для идентификации хостов.

CIDR Addressing				
IP Address 192.168.4.6/22				
	Network			Host
binary	11000000	10101000	00000100	00000110
numeric	192	168	4	6
Netmask 255.255.252.0 22 bits				
binary	11111111	11111111	11111000	00000000
numeric	255	255	252	000

Рис. 2. Схема адресации CIDR

В любом стандартном IP-адресе класса С адрес сети занимает старшие три сегмента (24 разряда). Если требуется создать сеть, включающую до 512 хостов, адрес сети будет занимать 23 разряда, а адрес хоста — 9 разрядов (от 0 до 511). Форма IP-адреса при этом остается той же, то есть он по-прежнему будет состоять из четырех разрядных сегментов. Однако число, относящееся к третьему сегменту, в данном случае будет использовано для идентификации как сети, так и хоста. Иначе говоря, в нем для сетевого адреса выделяются старшие 7 разрядов, а для хост-адреса — младший разряд. Так, в следующем примере третье число, 145, хранит завершающую часть сетевого адреса и служит в качестве начала хост-адреса.

Получение IP-адреса

IP-адреса официально выделяются комитетом IANA, который управляет всеми аспектами адресации в Internet (www.iana.org). Комитет IANA контролирует регистры Internet (PI), с помощью которых поддерживаются адреса Internet на региональном и локальном уровнях. Регистром Internet для Америки является *American Registry for Internet Numbers (ARIN)*, Web-узел которого находится по адресу

www.arin.net. Функция выделения адресов для пользователей Internet осуществляется провайдерами услуг Internet. Любой пользователь может получить адрес у провайдера либо в случае, если его локальная сеть уже подключена к Internet, обратиться к сетевому администратору. Если вы воспользуетесь услугами провайдера, то сможете получить временный адрес из пула доступных для этой цели адресов.

Некоторые адреса являются зарезервированными. Так, числа 127, 0 и 255 не могут быть частью официального IP-адреса. Число 127 применяется в качестве сетевого адреса интерфейса обратной связи (loopback) в системе. Этот интерфейс обеспечивает связь между пользователями одной системы без обращения к сети. В данном случае адрес сети может иметь вид 127.0.0.0, а IP-адрес — 127.0.0.1. При использовании IP-адресации, основанной на классах, число 255 является специальным [широковещательным](#) идентификатором, который служит для трансляции сообщений на все узлы сети. При указании числа 255 в качестве части любого IP-адреса опрашиваются все узлы, подключенные на этом уровне. Например, адрес 192.168.255.255 применяется для отправки широковещательных сообщений всем компьютерам сети 192.168, всем подсетям этой сети, а также всем хостам. При использовании адреса 192.168.187.255 широковещательные сообщения посылаются каждому компьютеру локальной сети. Если в сетевой части адреса указаны нули, хостовая часть адреса служит ссылкой на компьютер в составе локальной сети. Например, адрес 0.0.0.6 относится к шестому компьютеру в локальной сети. Если требуется отправить широковещательное сообщение всем компьютерам локальной сети, можно воспользоваться адресом 0.0.0.255.

Для локальных сетей, не подключенных к Internet, зарезервирован специальный набор адресов (RFC 1918). Это адреса, которые начинаются со специального сетевого номера 192.168 (для сетей класса C), как в приведенных выше примерах. Если локальная сеть служит для поддержки домашнего офиса или небольшого предприятия, эти IP-адреса можно использовать для локальных машин. Для этих машин

можно применять IP-адреса, которые начинаются с номера 192.168.1.1. Максимальное значение для хостового сегмента составляет 256. Например, если в состав локальной сети включены три компьютера, им можно назначить IP-адреса 192.168.1.1, 192.168.1.2 и 192.168.1.3. На локальных компьютерах можно устанавливать службы Internet: FTP, Web и электронной почты. Все они применяют те же протоколы TCP/IP, что и Internet. Например, при помощи FTP можно передавать файлы между компьютерами сети, Установив службу электронной почты, вы сможете пересылать сообщения между компьютерами, а при наличии Web-браузера— получать доступ к локальным Web-узлам, Если один из компьютеров требуется подключить к Internet или другой сети следует организовать на нем шлюз. Шлюзу обычно присваивается адрес 192.168.1.1. Для подключения локальных компьютеров к Internet через шлюз используется метод, называемый *IP-маскировкой*.

Существуют также зарезервированные адреса для локальных сетей классов А и В, не подключенных к Internet. Эти адреса перечислены в табл. 3. В этом случае, значения хостовых сегментов адреса также находятся в диапазоне от 0 до 255. Например, диапазон сетевых адресов класса В простирается от 172.16.0.0 до 172.31.255.255. Это позволяет адресовать до 32356 хостов. Диапазон IP-адресов сетей класса С охватывает адреса от 192.168.0.0 до 192.168.255.255, Благодаря этому можно устанавливать до 256 подсетей, каждая из которых может включать до 256 хостов. Сетевой адрес 127.0.0.0 зарезервирован для интерфейса обратной связи. Этот интерфейс предназначен для осуществления внутренней коммуникации и обеспечивает обмен сообщениями между пользователями одной системы.

Таблица 3. IP-адреса для локальных сетей, не подключенных к Internet

Частный сетевой адрес	Класс сети
10.0.0.0	Сеть класса А
от 172.16.0.0 до 172.31.255.255	Сеть класса В
192.168.0.0	Сеть класса С
127.0.0.0	Адрес для обратной связи

Широковещательный адрес

Благодаря наличию широковещательного адреса система может разослать одно и то же сообщение сразу всем компьютерам сети. При использовании IP-адресации, основанной на классах, широковещательный адрес легко определяется на основе адреса хоста: у широковещательного адреса хостовая часть равна 255. Сетевая часть адреса при этом не изменяется. Таким образом, если IP-адрес равен 192.168.1.72, то широковещательный адрес будет иметь вид 192.168.1.255. При использовании IP-адресации в форме CIDR нужно учитывать количество, разрядов в маске сети, а оставшимся разрядам присвоить значение 1 (рис. 1). Например, IP-адресу 192.168.4.6/22 соответствует широковещательный адрес 192.168.7.255/22. В этом случае старшие 22 разряда представляют адрес сети, а последние 10 разрядов являются хостовой частью, которой и присвоено широковещательное значение (все единицы).

Адрес шлюза

В некоторых сетях для подключения к другим сетям в качестве шлюза используется отдельный компьютер. Все соединения данной, сети с другими сетями производятся через него. В большинстве локальных сетей для подключения к Internet также применяются шлюзы. Если вы работаете в сети такого типа, необходимо задать адрес шлюза. Если же ваша сеть не подключена к Internet или вы работаете в изолированной системе, или соединяетесь с Internet-провайдером по коммутируемым линиям связи, адрес шлюза может вам не

понадобиться. Адрес шлюза является адресом хост-системы, на которой установлено соответствующее программное обеспечение. Во многих сетях ее идентификационный номер равен 1. Таким образом, адрес шлюза сети с адресом 192.168.1 равен 192.168.1.1. Однако это справедливо не для всех сетей. Для того чтобы узнать адрес вашего шлюза, обратитесь к системному администратору.

Файлы конфигурации TCP/IP

Перечень файлов конфигурации, находящихся в каталоге /etc (табл. 4); используется для установки и управления сетью TCP/IP. Эти файлы определяют такую сетевую информацию, как имена хостов и доменов, IP-адреса, а, также параметры интерфейса. Здесь также вводятся IP-адреса и доменные имена других хостов Internet, к которым требуется получить доступ. Если сеть была сконфигурирована во время, инсталляций, описанные выше сведения уже находятся в файлах конфигурации. Удобными средствами для ввода данных в файлы конфигурации являются программы netcfg, Linuxconf и netconfig.

Файл/etc/hosts (имена хостов)

Не зная уникального IP-адреса, который присвоен компьютеру в сети TCP/IP, компьютер найти нельзя. Поскольку IP-адреса трудны для запоминания и работы, вместо них используют доменные имена. Каждому IP-адресу ставится в соответствие доменное имя. Система преобразует доменное имя, по которому пользователь обращается к определенному компьютеру, в соответствующий IP-адрес, используемый для установления соединения с таким компьютером.

Вначале список имен и IP-адресов хостов велся на всех компьютерах сети. Такой список до сих пор хранится в файле /etc/hosts. Получив от пользователя доменное имя, система ищет в файле **hosts** соответствующий адрес. За ведение этого списка отвечает системный администратор. Вследствие стремительного роста Internet и появления все новых и новых очень больших сетей функции преобразования доменных имен в IP-адреса были переданы серверам доменных имен.

Тем не менее файл **hosts** продолжает использоваться для хранения доменных имен и IP-адресов хостов, соединения с которыми устанавливаются наиболее часто. Перед тем как обращаться к серверу имен; система всегда будет обращаться к файлу **hosts** и искать в нем IP-адрес для заданного ей доменного имени.

Каждая запись в файле **hosts** состоит из IP-адреса, пробела и доменного имени. В одной строке с записью можно ввести комментарий, который всегда предваряется символом “#”. В файле **hosts** уже имеется запись для локального компьютера Localhost с IP-адресом 127.0.0.1. Localhost — это специальный зарезервированный IP-адрес 127.0.0.1, который позволяет пользователям вашей системы связываться друг с другом в локальном режиме

```
/etc/hosts
127.0.0.1 turtle, trek.com localhost
199.35.209.72 turtle.trek.com
204.32.168.56 pangpl.train.com
202.211.234.1 rose.berkeley.edu
```

Файл **/etc/networks** (имена сетей)

В файле **/etc/networks** хранятся доменные имена и IP-адреса сетей, с которыми у вашей системы есть соединение, а не доменные имена/конкретных компьютеров.

Локальные сети имеют сокращенные IP-адреса. В зависимости от типа сети ее IP-адрес может состоять из одного, двух или трех чисел. Адрес сети для локального компьютера — 127.0.0.0.

IP-адреса записываются в файле **/etc/networks** вместе с соответствующими им доменными именами сетей. IP-адрес состоит из сетевой и интерфейсной (хостовой) частей. Сетевая часть — это адрес сети, который хранится в файле **networks**. В данном файле всегда будет присутствовать ,отдельная запись для сетевой части IP-адреса вашего компьютера.

```
/etc/networks  
loopback 127.0.0.0  
trek.com 192.168.1.0
```

Файл/etc/HOSTNAME

В файле **/etc/HOSTNAME** содержится имя хоста, назначенное вашей системе. Чтобы изменить это имя, файл нужно отредактировать. Эту задачу можно решить с помощью программы **netcfg**, которая позволяет изменить имя хоста и помещает новое имя в файл **/etc/HOSTNAME**. Имя хоста можно узнать не только путем вывода на экран содержимого этого файла, но и с помощью команды **hostname**:

```
$ hostname  
turtle.trek.com
```

Файл/etc/services

Файл **/etc/services** содержит перечень сетевых служб, доступных в системе (например, FTP или telnet), а также информацию о том, с каким портом связана каждая служба. Это позволяет определить порт, используемый Web- или FTP-сервером. Службе можно присвоить псевдоним, он указывается после номера порта. После этого вы сможете обращаться к службе, применяя этот псевдоним.

Файл /etc/protocols

В файле **/etc/protocols** находится список протоколов TCP/IP, поддерживаемых системой.

Файл /etc/sysconfig/network

Файл **/etc/sysconfig/network** содержит системные определения сетевой конфигурации. Это определения доменного имени, шлюза, имени хоста и некоторые другие.

Таблица 4. Адреса и файлы конфигурации TCP/IP

Адрес	Описание
Адрес хоста	IP-адрес системы; включает сетевую часть, идентифицирующую сеть, а также хостовую часть, которая идентифицирует хост
Адрес сети	IP-адрес сети
<u>Широковещательный адрес</u>	IP-адрес, предназначенный для рассылки сообщений всем хостам сети
<u>Адрес шлюза</u>	IP-адрес системного шлюза в случае, если имеется один шлюз (обычно сетевая часть IP-адреса хоста, где хостовой части присвоено значение 1)
Адреса сервера доменных имен	IP-адреса серверов доменных имен, используемых сетью
<u>Маска сети</u>	Применяется для определения сетевой и хостовой частей IP-адреса
/etc/hosts	Содержит имена хостов и соответствующие им IP-адреса
/etc/networks	Устанавливает соответствие между доменными именами и адресами сетей
/etc/host.conf	Параметры программы-распознавателя
/etc/nsswitch.conf	Параметры программы-распознавателя
/etc/hosts	Содержит список доменных имен удаленных хостов с Соответствующими IP-адресами
/etc/resolv.conf	Включает перечень имен и IP-адресов серверов DNS (nameserver), а также доменных имен, соответствующих удаленным хостам (search)

Продолжение таблицы 4

Адрес	Описание
/etc/protocols	Содержит перечень протоколов, доступных в системе
/etc/services	Содержит перечень доступных сетевых служб, таких как FTP и telnet, а также используемых ими портов
/etc/sysconflg/network	Хранит информацию о конфигурации сети
/etc/HOSTNAME	Хранит имя хоста

Служба доменных имен (DNS)

Каждый компьютер, подключенный к сети TCP/IP (например, к Internet), идентифицируется своим IP-адресом. [IP-адрес](#) представляет собой комбинацию из четырех чисел, определяющих конкретную сеть и конкретный компьютер (хост) в этой сети. IP-адреса очень трудно запоминать, поэтому для идентификации компьютера вместо IP-адреса можно использовать доменное имя. Доменное имя состоит из двух частей — имени хоста и имени домена. Имя хоста — это собственно имя компьютера, а домен обозначает сеть, частью которой этот компьютер является. Имена доменов, используемые в США, обычно имеют расширения, обозначающие тип сети. Например, для учебных заведений используется расширение .edu, а для коммерческих организаций — расширение .com. Международные домены обычно имеют расширения, которые обозначают страну, в которой они расположены, например .du для Германии и .ai для Австралии. Комбинация имени хоста, имени домена и расширения представляет собой уникальное имя, по Которому можно обращаться к компьютеру. Домен, в свою очередь, иногда разбивается на поддомены.

Вы знаете, что даже если компьютер имеет имя хоста, в сети его можно идентифицировать только по IP-адресу. Обратиться к компьютеру в сети по имени хоста можно, но это предполагает поиск соответствующего IP-адреса в базе данных. Сеть использует для

доступа к компьютеру не имя хоста, а IP-адрес. До появления очень больших сетей ТСП/IP, в частности Internet, на каждом компьютере сети хранился файл с перечнем доменных имен и IP-адресов всех компьютеров, включенных в сеть. При обращении по имени хоста компьютер искал это имя в данном файле и находил соответствующий ему IP-адрес. Такой метод, можно применять и сейчас в отношении удаленных систем, соединения с которыми устанавливаются чаще всего.

По мере роста сетей ситуация изменилась. Ведение отдельного списка всех доменных имен и IP-адресов на каждом компьютере стало нецелесообразным, а в случае с Internet просто невозможным. Чтобы обеспечивать преобразование доменных адресов в IP-адреса, были разработаны и установлены на отдельно выделенные серверы базы данных, содержащие доменные имена и соответствующие им IP-адреса. При необходимости найти IP-адрес по доменному имени на сервер имен посылается соответствующий запрос. Сервер имен ищет IP-адрес и посылает его обратно. В крупной сети может быть несколько серверов имен, обслуживающих различные ее части. Если какой-либо сервер имен не может найти необходимый IP-адрес, он посылает запрос на другой сервер. Серверы имен могут предоставлять и такую информацию, как название организации, которой принадлежит искомый компьютер, ее адрес и даже фамилию лица, обслуживающего компьютер.

Если для администрируемой вами сети необходимо создать сервер имен, то Linux-систему можно сконфигурировать таким образом, что она будет выполнять эту функцию. Для этого нужно запустить демон сервера имен, который будет ожидать поступления запросов на преобразование имен. Такой демон применяет несколько файлов конфигурации, позволяющих ему отвечать на подобные запросы: Программным обеспечением сервера имен, используемым в Linux-системах, является пакет Berkeley Internet Name Domain (BIND), распространяемый организацией Internet Software Consortium (www.isc.org).

Запросы на серверы имен посылают особые программы, которые называют распознавателями (resolver). Распознаватель — это программа, предназначенная для получения адресов с серверов имен. Чтобы пользоваться доменными именами, вам придется сконфигурировать распознаватель. Конфигурация локального распознавателя задается в файлах **/etc/hostconf** и **/etc/resolv.conf**. Вместо файла **/etc/hostconf** можно воспользоваться файлом — **/etc/nsswitch**.

Сетевые интерфейсы: команда ifconfig

Подключение системы к сети осуществляется через аппаратный интерфейс, в частности, через, карту Ethernet или модем. Данные, проходящие через этот интерфейс, направляются в сеть. Сначала при помощи команды **ifconfig** производится конфигурирование сетевого интерфейса, а затем посредством команды **route** обеспечивается маршрутизация. Если для конфигурирования интерфейса используется какое-либо средство конфигурирования сети, к примеру **netcfg**, **Linuxconf** или **YaST**, вам не нужно выполнять команду **ifconfig**. Однако при необходимости этими командами, можно конфигурировать интерфейсы напрямую. При каждом запуске системы должны быть заданы сетевые интерфейсы и обеспечена маршрутизация. Это может осуществляться автоматически во время загрузки при помощи команд **ifconfig** и **route**, которые должны быть заданы для каждого интерфейса в файле конфигурации **/etc/rc.d/init.d/network**, выполняемом при каждом запуске системы. Если вы вручную добавляете новые интерфейсы, то вам следует создать сценарий, обеспечивающий выполнение команды **ifconfig**.

В системе Red Hat сетевой интерфейс запускается с помощью сценария **network**, расположенного в каталоге **/etc/rc.d/init.d**. Можно вручную отключить и перезагрузить сетевой интерфейс, используя данный сценарий и опции **start** или **stop**. Следующие команды отключают, а затем запускают сетевой интерфейс.

```
/etc/rc.d/init.d/network stop /etc/rc.d/init.d/networkstart.
```

Для проверки работоспособности интерфейса воспользуйтесь командой `ping`, указав IP-адрес системы в сети, например IP-адрес шлюзового компьютера. Эта команда будет повторяться непрерывно, для ее отмены необходимо нажать комбинацию клавиш [Ctrl+C].

```
ping 192.168.1.42
```

Команда `ifconfig`

В качестве аргументов команда `ifconfig` использует имя интерфейса и IP-адрес. Кроме того, она имеет ряд опций. Команда `ifconfig` предназначена для присвоения заданному сетевому интерфейсу указанного IP-адреса. Таким образом, она сообщает вашей системе о том, что данный интерфейс существует и что она обращается к нему по указанному IP-адресу. Кроме того, можно указать, чем является IP-адрес — адресом хоста или адресом сети. Вместо IP-адреса можно использовать доменное имя, при условии, что оно вместе с IP-адресом указано в файле `/etc/hosts`. Команда `ifconfig` имеет следующий синтаксис:

```
# ifconfig интерфейс -хост_сеть_флаг адреса опции
```

Флаг `-хост_сеть_флаг` может принимать одно из двух значений: `-host` или `-net`. Флаг `-host` свидетельствует о том, что данный IP-адрес является адресом хоста, а `-net` означает, что данный IP-адрес является адресом сети. По умолчанию принимается флаг `-host`. У команды `ifconfig` есть несколько опций, которые задают различные характеристики интерфейса, например максимальное число байтов, которые он может передать за один раз (`mtu`), широковещательный адрес, и т. д. Опция `up` активизирует интерфейс, а опция `down` деактивизирует его. В следующем примере команда `ifconfig` используется для конфигурирования интерфейса Ethernet.

```
$ ifconfig eth0 204.32.168.56
```

Для такой простой конфигурации, как эта, `ifconfig` автоматически создает стандартный широковещательный адрес и маску сети. Стандартный Широковещательный адрес — это адрес, в котором номер хоста равен 255. Напомним, что стандартная маска сети — 255.255.255.0. Если же вы подключены к сети с другой маской сети и нестандартным широковещательным адресом, их необходимо указать в командной строке `ifconfig`. Широковещательный адрес задается в опции `broadcast`, а маска сети — в опции `netmask`. Давайте рассмотрим основные из них:

- `up` — включить интерфейс;
- `down` — выключить интерфейс;
- `arp` — включить или выключить использование протокола ARP для интерфейса;
- `promisc` — включить или выключить неразборчивый режим для интерфейса;
- `allmulti` — включить или выключить режим `multicast`;
- `metric` — изменить параметр `metric`;
- `mtu` — изменить максимальный размер пакета;
- `netmask` — установить маску сети;
- `add` — добавить `ip` адрес для интерфейса;
- `del` — удалить `ip` адрес интерфейса;
- `media` — установить тип внешнего протокола;
- `broadcast` — установить широковещательный адрес или отключить эту функцию;
- `hw` — установить MAC адрес для интерфейса;
- `txqueuelen` — размер очереди интерфейса;

Параметры и адрес необязательны и зависят от используемой команды. А опции влияют на поведение утилиты более глобально. Опций всего несколько, рассмотрим их:

- a — применять команду ко всем интерфейсам, например, полезно, если вы хотите отключить или включить все сетевые интерфейсы;
- s — вывести краткий список интерфейсов.

В следующем примере, `ifconfig` задает маску сети и широковещательный адрес.

```
$ ifconfig eth0 204.32.168.56 broadcast 204.128.244.127 netmask 255.255.255.0
```

После того как интерфейс сконфигурирован, для его активизации можно использовать команду `ifconfig` с параметром **up**, а для деактивизации — команду `ifconfig` с параметром **down**.

```
$ ifconfig eth0 up
```

Контроль за состоянием сети: программа `ping`

Программа `ping` позволяет проверить наличие фактического доступа к другому хосту вашей сети. Эта программа посылает запрос на хост и ожидает ответа. Если ответ приходит, то сообщение о нем появляется на экране монитора. Программа `ping` будет непрерывно посылать такие запросы до тех пор, пока вы не остановите ее при помощи команды **break** или сочетания клавиш [Ctrl+C]. На экране станут отображаться бегущие одно за другим сообщения до тех пор, пока программа не будет остановлена. Если программа `ping` не получит ответа от хоста, она выведет сообщение о том, что хост недоступен, то есть сетевое соединение не работает. Причиной может быть неработающий интерфейс, ошибка в конфигурации или просто плохой физический контакт. Программа `ping` использует протокол Internet Control Message Protocol (ICMP). Этот протокол может блокироваться в сетях (из соображений безопасности), в результате чего команда `ping` просто не будет работать.

Запускается программа `ping` командой `ping` с указанием имени хоста.

```
$ ping my.comp.ru
```

Программа netstat позволяет получить в реальном масштабе времени информацию о состоянии сетевых соединений, а также статистические данные и таблицу маршрутизации. У этой программы есть несколько опций, с помощью которых можно задавать вид получаемой информации.

Команда traceroute

Утилита ping позволяет только определить наличие проблемы, что узел не отвечает, но как узнать где обрывается соединение? Для этого применяется утилита traceroute.

Как работает traceroute?

Вы, наверное, уже знаете, что вся информация в сети передается в виде пакетов. Поток данных разбивается специальным программным обеспечением на небольшие пакеты и передается через сеть интернет на целевой узел, а там собирается обратно.

Каждый пакет проходит на своем пути определенное количество узлов, пока достигнет своей цели. Причем, каждый пакет имеет свое время жизни. Это количество узлов, которые может пройти пакет перед тем, как он будет уничтожен. Этот параметр записывается в заголовке TTL, каждый маршрутизатор, через который будет проходить пакет уменьшает его на единицу. При TTL=0 пакет уничтожается, а отправителю отсылается сообщение Time Exceeded.

Команда traceroute linux использует UDP пакеты. Она отправляет пакет с TTL=1 и смотрит адрес ответившего узла, дальше TTL=2, TTL=3 и так пока не достигнет цели. Каждый раз отправляется по три пакета и для каждого из них измеряется время прохождения. Пакет отправляется на случайный порт, который, скорее всего, не занят. Когда утилита traceroute получает сообщение от целевого узла о том, что порт недоступен трассировка считается завершенной.

Утилита traceroute

Перед тем как перейти к примерам работы с утилитой давайте рассмотрим ее синтаксис и основные опции. Синтаксис вызова очень прост:

```
# traceroute опции адрес_узла
```

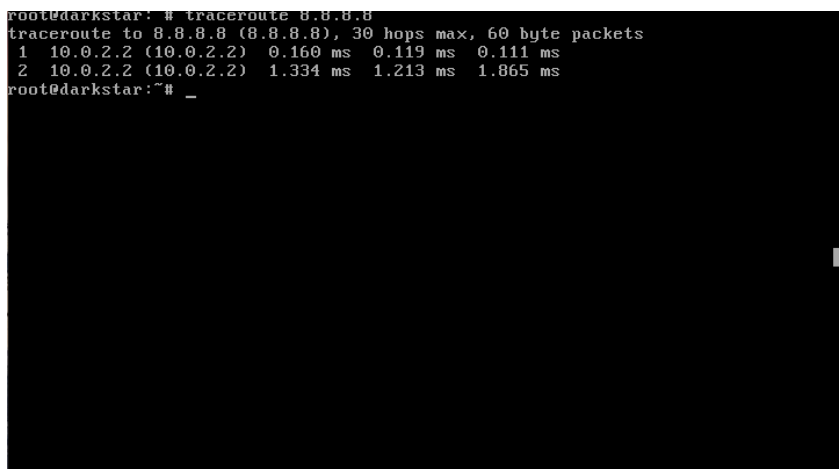
В качестве адреса может использоваться ip адрес или доменное имя. Рассмотрим основные опции:

- 4 или -6 — использовать ipv4 или ipv6 протокол;
- I — использовать ICMP пакеты вместо UDP;
- T — использовать TCP пакеты вместо UDP;
- F — не фрагментировать пакеты;
- f — указать TTL с которого нужно начать;
- g — передавать пакет через указанный шлюз;
- i — передавать пакет через указанный интерфейс;
- m — максимальное количество узлов, через которые пройдет пакет;
- q — количество пакетов, отправляемых за раз, по умолчанию три;
- n — не узнавать доменные имена;
- p — указать порт вместо порта по умолчанию;
- w — установить время ожидания ответа от узла, по умолчанию полсекунды;
- r — использовать другой роутер вместо того, что указанный в таблице маршрутизации;
- z — минимальный интервал между пакетами;
- U — использовать UDP с увеличением номера порта;
- UL — использовать протокол UDPLITE;
- D — использовать протокол DCCP;
- mtu — указать размер пакета;
- P — протокол, доступны такие значения: raw, dccp, udplite, udp, tcpconn, tcp, icmp.

Это не все опции утилиты, но все основные, которыми вы будете пользоваться. Далее перейдем практике того, как выполняется трассировка сети Linux.

Например, выполним трассировку до сервера 8.8.8.8:

```
$ traceroute 8.8.8.8
```



```
root@darkstar:~# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.160 ms  0.119 ms  0.111 ms
 2  10.0.2.2 (10.0.2.2)  1.334 ms  1.213 ms  1.065 ms
root@darkstar:~# _
```

Рис. 3. Трассировка до сервера 8.8.8.8

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Ознакомиться с видами и классами адресации, принципами построения IP адресов. Изучить файлы конфигурации TCP/IP. Научиться настраивать сетевой интерфейс в операционной системе Linux. Научиться пользоваться командами для настройки и проверки сети. Для выполнения работы выполнить следующие шаги:

1. Проверить конфигурацию сетевого адаптера
2. При необходимости удалить IP адрес
3. Настроить сетевой адаптер, присвоив ему IP адрес
4. Задать имя хоста
5. Задать маску сети
6. Задать широковещательный адрес
7. Активизировать (запустить) сетевой интерфейс
8. Проверить работоспособность сетевого интерфейса (проверить доступность других машин в локальной сети)
9. Настроить шлюз для выхода сеть
10. Проверить доступность машин в сети интернет

Для проверки настройки сети использовать утилиты ping и traceroute.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите группы по разработке протоколов TCP/IP.
2. Назовите основные протоколы TCP/IP.
3. Перечислите протоколы, образующие набор протоколов TCP/IP.
4. Опишите понятие пакет.
5. Дайте определение понятию IP-адрес и назовите для чего он необходим.
6. Перечислите отличия систем адресации IPv4 и IPv6.
7. Опишите понятия адрес хоста и адрес сети. Приведите пример.
8. Назовите классы IP адресов.
9. Раскройте понятие маски сети.
10. Опишите понятие адреса с форматом CIDR.
11. Приведите схему адресации CIDR
12. Назовите способ получения IP адреса.
13. Назовите зарезервированные IP адреса.
14. Назовите назначения широковещательного адреса и адреса шлюза.
15. Перечислите файлы конфигурации TCP/IP. Назовите параметры, которые они определяют.
16. Опишите структуру /etc/networks. Приведите пример содержимого.
17. Раскройте понятие DNS.
18. Предложите вариант применения команды ifconfig для назначения сетевому интерфейсу IP адреса.
19. Предложите вариант применения команды ifconfig для задания маски сети и широковещательного адреса.
20. Приведите пример команды для включения и отключения сетевого интерфейса.
21. Приведите пример применения команды ping

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы, выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон [Электронный ресурс] / Н. Вирт, Ю. Гуткнехт ; пер.с англ. Борисов Е.В., Чернышов Л.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2012. — 560 с. — Режим доступа: <https://e.lanbook.com/book/39992>

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

2. Крищенко, В.А. Сервисы Windows [Электронный ресурс] : учебное пособие / В.А. Крищенко, Н.Ю. Рязанова. — Электрон. дан. — Москва : МГТУ им. Н.Э. Баумана, 2011. — 47 с. — Режим доступа: <https://e.lanbook.com/book/52416..>

3. Войтов, Н.М. Администрирование ОС Red Hat Enterprise Linux. Учебный курс [Электронный ресурс] : учебное пособие / Н.М. Войтов. — Электрон. дан. — Москва : ДМК Пресс, 2011. — 192 с. — Режим доступа: <https://e.lanbook.com/book/1081>

4. Стащук, П.В. Администрирование и безопасность рабочих станций под управлением Mandriva Linux: лабораторный практикум [Электронный ресурс] : учебно-методическое пособие / П.В. Стащук. — Электрон. дан. — Москва : ФЛИНТА, 2015. — 182 с. — Режим доступа: <https://e.lanbook.com/book/70397>

5. Снейдер, Й. Эффективное программирование TCP/IP [Электронный ресурс] : учебное пособие / Й. Снейдер. — Электрон. дан. — Москва : ДМК Пресс, 2009. — 320 с. — Режим доступа: <https://e.lanbook.com/book/1272>

Электронные ресурсы:

6. Научная электронная библиотека <http://eLIBRARY.RU>
7. Электронно-библиотечная система <http://e.lanbook.com>
8. Losst - Linux Open Source Software Technologies <https://losst.ru>