Минобрнауки России

Федеральное государственное автономное образовательное учреждение высшего образования   
«Национальный исследовательский университет

«Московский институт электронной техники»

**САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ УЧЕБНОЙ ДИСЦИПЛИНЫ**

«Администрирование ЛВС»

Направление подготовки – 09.03.01 «Информатика и вычислительная техника»

Профиль/Программа – «Элементы и устройства микропроцессорных информационно-управляющих систем»

**ИСПОЛЬЗОВАНИЕ ВИРТУАЛИЗАЦИИ ДЛЯ ИЗУЧЕНИЯ СЕТЕВЫХ СРЕДСТВ LINUX**

Самостоятельная работа студентов по подготовке к лабораторной работе №1

2016 г.

**Задание 1. Использование виртуализации для изучения Linux**

**Содержание:**

1. Особенности использования Linux
2. Программное обеспечение *виртуализации* VirtualBox.
3. Сетевые возможности VirtualBox.

**Особенности использования Linux**

В широком смысле под термином **Linux** понимают *группу Unix-подобных операционных систем, основанных на открытом ядре Linux* (нач. 90-х годов — Линус Торвальдс, университет Хельсинки) *и различных библиотеках, утилитах и программах*. **Linux** тесно связан с понятиями **свободного программного обеспечения** (*free*) и **программного обеспечения с открытым исходным кодом** (open-source). Данные понятия подкреплены юридически соотвествующими лицензиями для *ПО* ( **GPL**, **BSD** и т.д); обычно *обозначают ПО с доступным исходным кодом* (бинарный код запрещено распространять без исходного) *и возможностью свободного использования, модификации и распространения*.

В узком смысле под термином **Linux** имеют в виду данный **дистрибутив Linux —** *вариант работоспособной Linux-системы, включающий компиляцию ядра, набор системных скриптов и конфигурационных файлов, а также различное программное обеспечение*. Условно можно сказать, что *дистрибутив* состоит из следующих компонентов: набора пакетов (форма представления *ПО* в дистрибутиве, включающая бинарный код, настройки, помощь и документацию или исходный код с настройками компиляции какого-либо вида *ПО*); *ПО* *для управления, обновления и распространения пакетов* (управление *репозиториями**ПО*, *загрузка* *ПО* из удаленных *репозиториев*, *определение* и устранение зависимостей, средства создания пакетов и т.д.); *ПО* *для инсталляции дистрибутива*.

Существует множество дистрибутивов **Linux**, различающихся основным *форматом пакетов*, функциональным назначением, набором программного обеспечения, поддержкой различных аппаратных платформ и т.д. В данном пособии используется **Debian GNU/Linux** или просто **Debian**, характеризующийся, в частности, развитой системой управления программным обеспечением и большой базой программного обеспечения, содержащегося в стандартных и дополнительных репозиториях.

Основными способами взаимодействия и управления системой, используемыми в данном пособии, являются **интерфейс командной строки** и редактирование **конфигурационных файлов**. *Интерфейс командной строки — такой способ взаимодействия с системой, при котором каждая строка, передаваемая пользователем системе, интерпретируется системой как команда, которую необходимо выполнить*. Команды также можно группировать в пакетные исполняемые файлы ( **сценарии** ) для решения системных и прикладных задач. *Концепция конфигурационных файлов подразумевает использование для хранения настроек системы и прикладного ПО текстовых файлов, формат которых одновременно воспринимаем пользователем и программным обеспечением*. Необходимые знания *по* использованию интерфейса командной строки можно получить из соответствующих учебных пособий [[ 1 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.1)[[ 3 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.3)или справочника к дистрибутиву [[ 4 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.4).

Возможны различные варианты установки **Linux** на *компьютер*:

* *Монопольно одну ОС* (в данном случае можно использовать автоматическое разбиение диска программой установки Linux).
* *В качестве альтернативной ОС*. Выбор загружаемой ОС происходит в меню при включении компьютера. При таком способе установки обязательно необходимо сделать резервные копии ценных данных; высвободить место на жестком диске для разделов **Linux** (минимально требуются: раздел для *корневой файловой системы* и раздел подкачки). Далее в процессе установки потребуется установить в*загрузочном секторе* диска ( **MBR** – *Master Boot Record*) загрузчик-меню выбора ОС (стандартный Linux – *grub* или *lilo*, или альтернативный).
* *Использовать LiveCD* – ОС, загружаемая с CD или DVD.
* *Как виртуальную машину*.

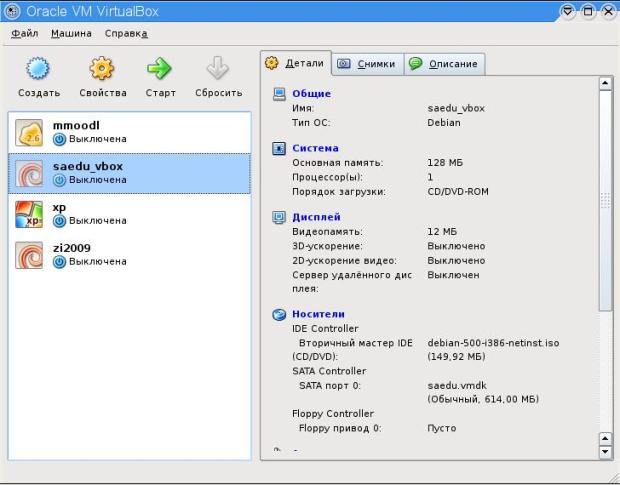
В рамках данного курса рекомендуется использовать последний вариант. Среда *виртуализации* не требует менять конфигурацию дисковых *разделов: диски* виртуальной машины будут представлены в виде файлов на диске компьютера; позволяет легко переключаться между основной системой и виртуальными машинами. *Виртуальная сеть дает возможность отлаживать различные сетевые конфигурации, запуская на одном физическом компьютере две и более систем*.

**Программное обеспечение виртуализации VirtualBox**

**Виртуализация** (в данном контексте) - *возможность запускать несколько виртуальных операционных систем (гостевых операционных систем) одновременно на одном физическом компьютере (хосте), то есть фактически создать несколько виртуальных компьютеров*. Данная возможность реализуется средствами специального *ПО* — **гипервизора или менеджера виртуальных машин**.

В данной работе в качестве гипервизора используется *программное обеспечение* Oracle/SUN **VirtualBox** [http://virtualbox.org](http://virtualbox.org/). *VirtualBox должен быть установлен как приложение операционной системы физического компьютера (хостовой системы* ). В качестве хостовых систем поддерживаются различные версии **Windows, Linux, Solaris** и **MAC OS X** (соответствующий пакет можно скачать с сайта).

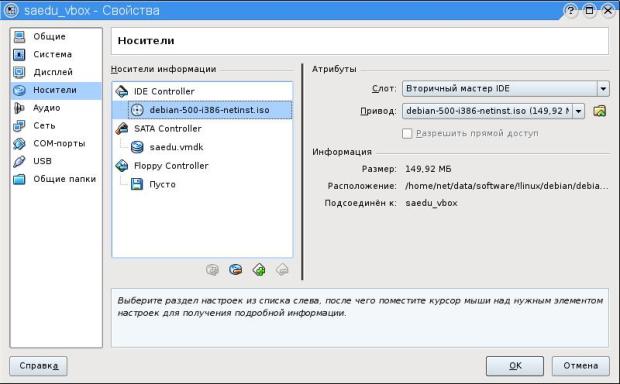
Управление виртуальными машинами осуществляется с помощью средства графического интерфейса пользователя, которое называется **менеджер VirtualBox** ( **VirtualBox manager** ). Рабочее окно программы ([рис. 1.1](http://www.intuit.ru/studies/courses/681/537/lecture/12105?page=1#image.1.1).) содержит текстовое *меню*, набор кнопок, слева - область со списком виртуальных машин, справа — вкладки "Детали", "Снимки", "Описания", относящиеся к выбранной в списке виртуальной машине.

[](http://www.intuit.ru/EDI/18_11_15_6/1447798956-19904/tutorial/998/objects/1/files/1_1.jpg)

**Рис. 1.1.**Менеджер VirtualBox.

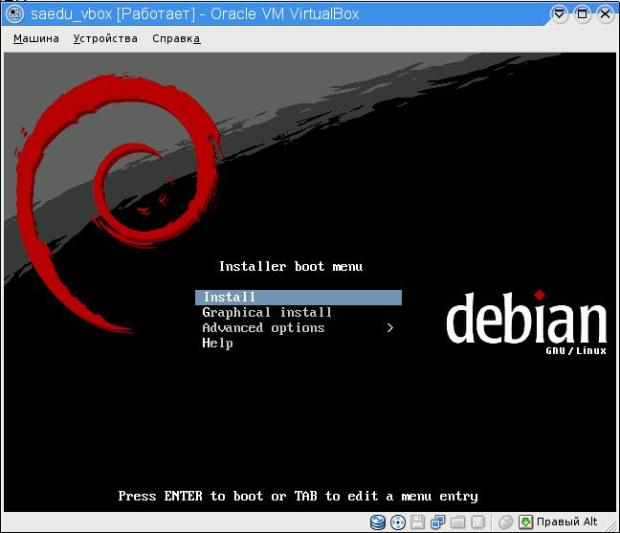
Кнопка " **Создать** " предназначена для вызова мастера создания виртуальной машины. В процессе работы мастера пользователю необходимо указать имя виртуальной машины, выбрать тип и версию операционной системы, указать размер памяти для виртуальной машины, создать новый виртуальный жесткий *диск* или использовать *файл* существующего жесткого диска.

Кнопка " **Свойства** " открывает окно настроек данной виртуальной машины. В частности, в списке свойств *пункт* "Носители" позволяет подключить к виртуальному приводу оптических дисков *диск*, вставленный в привод физического диска, а также iso-образ CD/*DVD*-диска ([рис. 1.2](http://www.intuit.ru/studies/courses/681/537/lecture/12105?page=1#image.1.2).)

[](http://www.intuit.ru/EDI/18_11_15_6/1447798956-19904/tutorial/998/objects/1/files/1_2.jpg)

**Рис. 1.2.**Подключение образа CD-диска к оптическому приводу виртуальной машины.

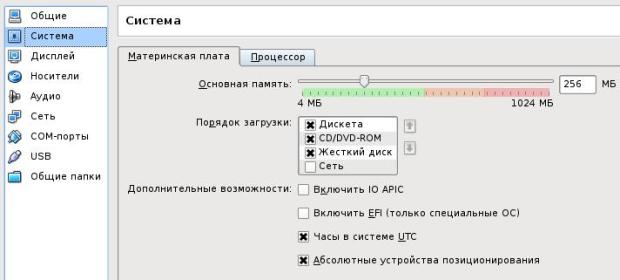
Кнопка " **Старт** " предназначена для запуска виртуальной машины. *Виртуальная машина* запускается в новом окне ([рис 1.3](http://www.intuit.ru/studies/courses/681/537/lecture/12105?page=1#image.1.3).). *Если кликнуть мышью в какую-либо область окна ВМ, то клавиатура и мышь перехватываются ВМ, для возврата управления в основную систему следует нажать хостовую клавишу (по умолчанию правый Ctrl)*.

[](http://www.intuit.ru/EDI/18_11_15_6/1447798956-19904/tutorial/998/objects/1/files/1_3.jpg)

**Рис. 1.3.**Работающая ВМ в среде VirtualBox.

*Меню* окна работающей **ВМ** позволяет: включить полноэкранный режим, послать сигнал **Ctrl+Alt+Del** виртуальной машине, сделать паузу в работе **ВМ**, выключить ВМ, а также управлять различными устройствами, подключаемыми к **ВМ**.

*Указать порядок выбора носителей для загрузки можно в свойствах виртуальной машины в пункте "Система" на вкладке "Материнская плата"* ([рис. 1.4](http://www.intuit.ru/studies/courses/681/537/lecture/12105?page=2#image.1.4)).

[](http://www.intuit.ru/EDI/18_11_15_6/1447798956-19904/tutorial/998/objects/1/files/1_4.jpg)

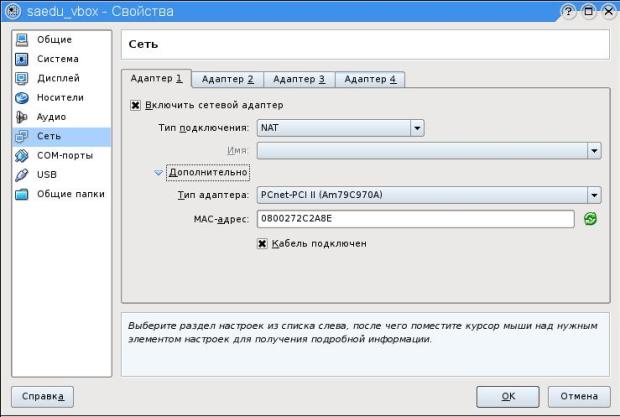
**Рис. 1.4.**Вкладка "Материнская плата" пункта "Система" свойств ВМ

*Перенос виртуальных машин между хостовыми системами осуществляется посредством импорта/экспорта: на источнике нужно сделать импорт ВМ (Файл — Импорт конфигурации), перенести полученные файлы на систему-назначение и на ней выполнить экспорт ВМ (Файл — Экспорт конфигурации)*.

Основные приемы и особенности работы с **VirtualBox** описаны в руководстве пользователя <http://www.virtualbox.org/manual/UserManual.html> .

**Сетевые возможности VirtualBox**

*Гипервизор VirtualBox позволяет виртуальным машинам использовать сетевые подключения хостовой системы, а также создавать виртуальные сети для виртуальных машин*. С помощью средств графического интерфейса пользователя для **ВМ** может быть настроено до 4-х сетевых адаптеров ( *пункт "Сеть" свойств виртуальной машины* -[рис. 1.5](http://www.intuit.ru/studies/courses/681/537/lecture/12105?page=2#image.1.5)).

[](http://www.intuit.ru/EDI/18_11_15_6/1447798956-19904/tutorial/998/objects/1/files/1_5.jpg)

**Рис. 1.5.**Настройка сети ВМ

Каждый сетевой *адаптер* может быть настроен на один из следующих режимов работы:

* **Режим "Не подключен"** (Not *attached*). В данном режиме адаптер присутствует в гостевой системе, но ведет себя так, как будто сетевой кабель в него не включен.
* **Режим "NAT"**. *В данном режиме адаптер использует сетевые настройки основной системы при взаимодействии с сетью физического узла и прочими внешними сетями*. Сетевая подсистема VirtualBox транслирует IP-трафик с исходным IP-адресом виртуальной машины в трафик с исходным адресом сетевого адаптера хостовой системы (трансляция сетевых адресов, *Network Address Translation*). Реализация**NAT** в **VirtualBox** имеет определенные ограничения, связанные с поддержкой протокола **ICMP**, широковещательного **UDP** -трафика и технологий *виртуальных частных сетей*.
* **Режим "Сетевой мост"** (*Bridged* networking). В данном режиме сетевой адаптер **ВМ** подключается к сетевому адаптеру хостовой системы и обрабатывает *сетевые пакеты* непосредственно в обход сетевого стека хостовой системы ( *адаптер хостовой системы работает с адаптером ВМ в режиме моста* ).
* **Режим "Внутренняя сеть"** (*Internal Networking*). *Данная внутренняя сеть представляет собой физический сегмент, объединяющий включенные в него виртуальные машины, причем внутренняя сеть не имеет доступа к основной системе и внешним сетям*.
* **Режим "Виртуальный адаптер хоста"** (Host-only networking). *Сеть объединяющая в данный сегмент хостовую систему и виртуальные машины, включенные в данный сегмент*. Для этого режима VirtualBox создает в хостовой системе программный сетевой интерфейс и устанавливает на нем IP-адрес.

*В режимах NAT и виртуального адаптера хоста VirtualBox предоставляет виртуальным машинам сервис динамической настройки протокола IP* ( **DHCP** — *Dynamic* *Host* *Configuration* *Protocol*). Соответствующий сетевой *адаптер*, настроенный в **ВМ** на динамическую конфигурацию в данных режимах будет получать *IP*-параметры автоматически от **DHCP** -сервера **VirtualBox**.

Примечание: В дальнейшем для установки через apt пакетов с DVD-диска, его необходимо предварительно примонтировать:

# mount -t iso9660 /dev/cdrom /media/cdrom

при этом, в /etc/apt/sources.*list* должно быть указано что-то вроде:

deb cdrom:[Debian GNU/Linux 6.0.0 \_Squeeze\_ - Official i386 DVD Binary-1

20110205-17:27]/ squeeze contrib main

**Задание 2. Сетевая подсистема Linux**

**Содержание:**

1. *Многоуровневый подход* к организации сетевых взаимодействий.
2. Средства настройки сетевой подсистемы **Linux**.

**Многоуровневый подход к организации сетевых взаимодействий**

*Процесс взаимодействия между узлами компьютерной сети является сложным* ["Транспортный и прикладной уровни модели сетевого взаимодействия."](http://www.intuit.ru/studies/courses/681/537/lecture/12113). Сложности связаны как с преобразованием "данные-сигналы-данные", так и с передачей сигнала по *гетерогенным* сетям. Для наглядной иллюстрации процесса передачи данных в сети *Интернет* можно, например, с помощью команды *traceroute* (tracert в *Windows*), изучить *список* промежуточных узлов, через которые проходят *сетевые пакеты* на пути от компьютера пользователя до какого-нибудь *Интернет*-ресурса [(пример 2.1)](http://www.intuit.ru/studies/courses/681/537/lecture/12106?page=1#example.2.1)

$ traceroute www.kernel.org

traceroute to www.kernel.org (130.239.17.4), 30 hops max, 60 byte packets

1 192.168.240.1 (192.168.240.1) 78.016 ms 77.871 ms 77.795 ms

2 \* \* \*

3 \* \* \*

4 core.10g.net.belpak.by (93.84.122.105) 109.257 ms 109.196 ms 109.132 ms

5 core.10g.net.belpak.by (93.84.122.18) 117.289 ms 117.227 ms 117.164 ms

6 193.232.249.102 (193.232.249.102) 108.876 ms 12.377 ms 12.257 ms

7 po31.l3-gw-1.mck.net.belpak.by (193.232.249.78) 22.113 ms 16.070 ms 15.949 ms

8 \* \* \*

9 83.229.226.101 (83.229.226.101) 55.643 ms msk-b4-hq-ae0.main.synterra.ru

(83.229.225.5) 55.571 ms 83.229.226.101 (83.229.226.101) 55.512 ms

10 ae7.RT.MR.MSK.RU.retn.net (87.245.253.169) 55.450 ms 55.388 ms 55.323 ms

11 ae2-6.RT.TC1.STO.SE.retn.net (87.245.233.134) 70.087 ms 70.025 ms 67.387 ms

12 \* \* \*

13 t1fre-ae0-v1.sunet.se (130.242.83.37) 98.493 ms umu-g.sunet.se (193.11.0.226)

99.859 ms t1fre-ae0-v1.sunet.se (130.242.83.37) 90.245 ms

14 umu-g.sunet.se (193.11.0.226) 112.213 ms u2k-te.gw.umu.se (130.239.0.201)

112.181 ms umu-g.sunet.se (193.11.0.226) 112.137 ms

15 pub4.kernel.org (130.239.17.4) 112.108 ms u2k-te.gw.umu.se (130.239.0.201)

117.516 ms 117.392 ms

Листинг 2.1. Трассировка маршрута (отображение списка промежуточных узлов) до узла www.kernel.org с помощью команды traceroute:

Задачу обеспечения сетевых коммуникаций решают с использованием **многоуровневого иерархического подхода**. *На узле программные компоненты данного уровня "общаются" только с соседними уровнями, при этом* **интерфейсы** *(предоставляемые функции) между соседними уровнями четко определены. Узлы взаимодействуют друг с другом посредством* **протоколов** *(правил, по которым обмениваются сообщениями программные компоненты одного уровня на разных узлах)* ["Использование виртуализации для изучения Linux"](http://www.intuit.ru/studies/courses/681/537/lecture/12105). Для передачи по сети *поток информации* на узле-отправителе разбивается на набор фрагментов (**единиц передачи данных** ), по мере продвижения по иерархии протоколов, *каждый протокол добавляет* (**инкапсулирует**) *свой заголовок к фрагменту передаваемой информации*. На узле-получателе происходит *обратный* процесс.

Существует несколько альтернативных **моделей сетевых взаимодействий** ["Использование виртуализации для изучения Linux"](http://www.intuit.ru/studies/courses/681/537/lecture/12105). В вопросах настройки сетевой подсистемы (в отличие от вопросов разработки протоколов и интерфейсов) *значение* данных моделей в большей степени методическое, направленное на унификацию терминологии и лучшее понимание процессов функционирования сети.

В данном пособии используется терминология (нестрогая) **модели стека протоколов TCP/IP**, которая включает четыре уровня:

* **Уровень сетевых интерфейсов** (доступа к сети). В рамках модели TCP/IP на данном уровне решаются две задачи: упаковки единиц передачи вышележащего уровня во **фреймы** (единицы передачи данного уровня) и получения физических адресов (**MAC-адресов**).
* **Сетевой уровень** (интернет-уровень, межсетевой уровень). Обеспечивает передачу **пакетов** (единиц передачи сетевого уровня) между гетерогенными сетями на основе IP-адресации (сетевой адресации) и группировки узлов в подсети. Основным протоколом данного уровня является **IP** (*Internet Protocol*).
* **Транспортный уровень**. Обеспечивает для фрагментов данных как гарантированную доставку (протокол **TCP**, единицы передачи - сегменты), так и доставку по возможности (протокол **UDP**, единицы передачи - **датаграммы** ). На данном уровне решается задача использования сетевого подключения несколькими приложениями (**порты**).
* **Прикладной уровень**. Включает средства преобразования и протоколы представления информационных потоков, непосредственно взаимодействующие с приложениями пользователя.

**Пример 2.2**. *Типичный процесс взаимодействия между сетевыми узлами по прикладному протоколу, использующему в качестве транспорта протокол TCP, можно описать следующим образом на узле-отправителе*:

* *На прикладном уровне осуществляются необходимые преобразования информационного потока (сжатие, шифрование и т. д.)*.
* *Информационный поток прикладного протокола на транспортном уровне разбивается на последовательность сегментов; в заголовке сегмента, в частности, указаны номера портов отправителя и получателя*.
* *На сетевом уровне сегмент вкладывается в пакет, заголовок, которого содержит IP-адреса отправителя и получателя*.
* *На уровне сетевых интерфейсов на основе сетевого пакета формируется фрейм, заголовок которого содержит MAC-адреса отправителя и получателя (в случае "нелокальной" доставки - маршрутизатора удаленных сетей)*.

На узле-получателе происходит *обратный* процесс.

**Примечание**: *В некоторых случаях на сетевом уровне и уровне сетевых интерфейсов используется дополнительная инкапсуляция* (технологии глобальных сетей, аутентификации, *виртуальные частные сети* и т.п.)

**Средства настройки сетевой подсистемы Linux**

Сетевая *поддержка* в **Linux** обеспечивается стандартными и дополнительными модулями ядра, программным обеспечением, входящим в состав конкретного дистрибутива, а также может быть расширена за счет использования прочих программных средств ["Доступ к локальной сети средствами Linux"](http://www.intuit.ru/studies/courses/681/537/lecture/12108)["Команды настройки протокола IP"](http://www.intuit.ru/studies/courses/681/537/lecture/12109).

Для управления сетевой подсистемой **Linux** существует множество средств: команды, скрипты, графические программы и т. д. Здесь приведены лишь наиболее известные из них.

*Для базовой настройки сетевой подсистемы можно использовать следующие группы команд*:

"Традиционные" **команды настройки сети** (пакет **net-tools** ) ["Постоянные сетевые конфигурации (на примере Debian/GNU Linux)"](http://www.intuit.ru/studies/courses/681/537/lecture/12110). В данном пакете присутствуют команды: управления сетевыми интерфейсами (ifconfig), таблицей маршрутизации (route), таблицей разрешения имен (arp), сетевой статистики (netstat), статусом сетевых устройств (mii-tool) и т.д. *Пакет net-tools использовался для управления сетью до ядра 2.4.; включается в современные дистрибутивы Linux для совместимости*.

Команды пакета **iproute2** ["Базовая диагностика сетевых подключений"](http://www.intuit.ru/studies/courses/681/537/lecture/12112). *Пакет iproute2 разработан для управления параметрами TCP/IP и контроля трафика в Linux (в частности маршрутизация и качество обслуживания). Для управления сетью в современных ядрах Linux (ядра версии 2.4-2.6) рекомендуется использовать средства данного пакета вместо команд* net-tools.

Пакет **iproute2** включает 3 основные утилиты:

-ip - *команда* для просмотра параметров и настройки сетевых интерфейсов, *IP*-адресов, таблиц маршрутизации, правил маршрутизации, таблиц преобразования, *IP*-туннелей и т.д.

-tc (*traffic* *control*) - *команда* для просмотра и настройки параметров управления трафиком (классификация трафика, дисциплины управления очереди для различных классов трафика)

-ss - *команда* для просмотра текущих соединений и открытых портов (аналог netstat ).

*Для настройки сети при загрузке системы (а также как средство настройки работающей системы) в Linux обычно используются наборы конфигурационных файлов и обрабатывающих их скриптов и программ*. В дистрибутивах применяются различные способы инициализации сетевой подсистемы, например **Debian** использует пакет **ifupdown** (см. ["Постоянные сетевые конфигурации (на примере Debian/GNU Linux)"](http://www.intuit.ru/studies/courses/681/537/lecture/12110)), а большинство **RPM** -основанных дистрибутивов — набор файлов вида **ifup-\*, ifdown-\*, ifcfg-\***, расположенных в каталоге /etc/sysconfig/network-scripts/ и т.п.

*В системах с установленным оконным менеджером для настройки сетевых параметров можно использовать средства графического интерфейса пользователя*. В качестве примера можно назвать пакет**NetworkManager** ["Транспортный и прикладной уровни модели сетевого взаимодействия."](http://www.intuit.ru/studies/courses/681/537/lecture/12113), совместимый с таким графическими окружениями, как **GNOME, KDE** и **XFCE** ; *интерфейс* **NetworkManager** позволяет настраивать проводные и беспроводные (**Wi-Fi, 3G, Bluetooth**) соединения, подключения к виртуальным частным сетям (**VPN**) и **ADSL** -провайдерам. Средства графического интерфейса пользователя есть также в пакете **Wicd** [[ 14 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.14).

*Для управления межсетевым экраном и средством манипулирования сетевыми пакетами (в том числе трансляция сетевых адресов - NAT) Netfilter стандартно используется утилита командной строки* **iptables** [[ 10 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.10)(см. ["Межсетевое экранирование в Linux"](http://www.intuit.ru/studies/courses/681/537/lecture/12118)и ["Обеспечение доступа в сеть Интернет"](http://www.intuit.ru/studies/courses/681/537/lecture/12119)). Над **Netfilter/iptables** существуют различные надстройки (в том числе и **GUI** -средства), например, **UFW/Gufw** — простые утилиты настройки межсетевого экрана уровня узла (**host-based**), используемые в дистрибутиве **Ubuntu** [[ 9 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.9). Примерами программ для настройки межсетевого экрана также могут служить такие проекты как:**Shorewall** [[ 9 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.9), **Firewall Builder** [[ 11 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.11), **Firestarter** [[ 12 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.12)и т. д.

В состав дистрибутивов **Linux** обычно также включаются разнообразные средства диагностики и мониторинга сети. Простейшими и общеупотребительными из них являются:

* Команда ping [[ 13 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.13). Простейшее средство проверки работоспособности удаленного узла.
* Команда *traceroute*. Позволяет отобразить все промежуточные узлы, которые проходят пакеты на пути к целевому узлу.
* Утилита tcpdump. Используется для перехвата и анализа сетевого трафика.

Более подробно эти и некоторые другие средства диагностики сети описаны в ["Базовая диагностика сетевых подключений"](http://www.intuit.ru/studies/courses/681/537/lecture/12112)и ["Анализ сетевого трафика как метод диагностики сети"](http://www.intuit.ru/studies/courses/681/537/lecture/12115).

**Ключевые термины**

**Гетерогенная сеть** - информационная *сеть*, в которой работают различные протоколы, используются технологии и оборудование различных фирм-производителей.

**Интерфейс** - формализованные правила, в соответствии с которыми взаимодействуют модули, реализующие протоколы соседних уровней модели сетевого взаимодействия (набор сервисов, предоставляемых данным уровнем соседнему).

**Протокол** - формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне модели сетевого взаимодействия в разных узлах.

**IP (Internet Protocol)** — маршрутизируемый протокол сетевого уровня без установления соединения.

**Сокет** — пара *IP*-*адрес*: *порт*, однозначно определяющая сетевой процесс; *точка доступа* прикладного процесса к сети.

**Краткие итоги**

1. К решению задачи обеспечения сетевых взаимодействий применяют многоуровневый иерархический подход, заключающийся в разбиении процесса коммуникации на набор уровней с четко определенными способами взаимодействия уровней на одном узле и на соседних узлах.
2. В данном пособии используется терминология модели *стека протоколов* TCP/IP, включающая следующие уровни: *уровень сетевых интерфейсов*, *сетевой уровень*, *транспортный уровень* и *прикладной уровень*.
3. Для управления сетевой подсистемой **Linux** существует множество средств, наиболее известными из которых являются средства пакетов **net-tools**, **iproute2**, **Netfilter/iptables**, утилиты диагностики сетевых подключений (ping, *traceroute*, tcpdump ), а также графические средства настройки сетевой подсистемы (например, **NetworkManager** ).

**Задание 3. Доступ к локальной сети средствами Linux**

**Содержание:**

1. Взаимодействие компьютерных систем в сети **Ethernet**.
2. Управление сетевыми интерфейсами в **Linux**.
3. Настройка физических параметров сетевого подключения.

**Взаимодействие компьютерных систем в сети Ethernet**

*Группа* технологий **Ethernet** [[ 20 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.20), [[ 7 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.7)*предназначена для организации взаимодействия компьютеров на уровне сетевых интерфейсов* (определяет характеристики среды передачи, типы соединений, способы коммуникации устройств и адресации, используемое оборудование и т. д.); *чаще всего используется для построения* **локальных сетей**.

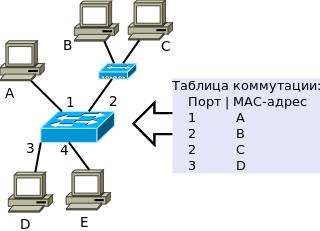
В качестве среды передачи в настоящее время в **Ethernet** используется *витая пара* (несколько пар изолированных проводников, скрученных между собой) или *оптоволоконный кабель*. *Полоса пропускания*различных технологий **Ethernet** варьирует от 10 Мбит/c до 10 Гбит/c. Все новые стандарты **Ethernet** *обратно совместимы* с базовым стандартом (условно говоря, *единица* передачи данных - *Ethernet-фрейм передается в любой среде без изменений). При этом, оборудование различных производителей, использующее различные технологии Ethernet, обычно, без проблем работает в одной сети*.

Для подключения компьютера к среде передачи (*витая пара*) используется **сетевой адаптер** с разъемом *RJ-45*. В данном сегменте сети каждый сетевой *адаптер* имеет уникальный **физический адрес (MAC-адрес)**, *размером 48 бит, который принято записывать в виде последовательности шестнадцатеричных цифр (например*, 00:1d:d9:43:0f:41). *Принцип действия сетевого адаптера — в том, что обрабатываются те Ethernet-фреймы, у которых указан его MAC-адрес в адресе получателя (а также специальные широковещательные фреймы, у которых MAC-адрес равен* FF:FF:FF:FF:FF:FF).

Сетевой *адаптер* компьютера подключается к **коммутатору** — многопортовому устройству, предназначенному для пересылки *Ethernet*-фреймов между портами. *Коммутатор* ведет **таблицу коммутации (MAC-адресов)**, *описывающую соответствие между MAC-адресами узлов и портами, к которым они подключены*. Записи в таблице **MAC-адресов** имеют определенное время жизни, по истечению которого они удаляются из таблицы.

Для передаваемых фреймов *коммутатор* анализирует **MAC-адреса**

1. Для передаваемых фреймов коммутатор анализирует **MAC-адрес** отправителя и получателя и содержимое *таблицы коммутации* ([рис. 3.1.](http://www.intuit.ru/studies/courses/681/537/lecture/12108?page=1#image.3.1)):
2. *Если MAC-адрес отправителя отсутствует в таблице, то он вносится туда с привязкой к порту*.
3. **Если MAC-адреса отправителя и получателя привязаны к одному и тому же порту, то фрейм не пересылается (фильтрация)**.
4. *Если MAC-адреса отправителя и получателя привязаны к различным портам, то фрейм отправляется в порт получателя*.
5. *Если MAC-адрес получателя неизвестен коммутатору, производится широковещательная рассылка фрейма по всем портам, за исключением порта источника*.
6. *Фреймы, у которых в адресе получателя указан широковещательный адрес* (FF:FF:FF:FF:FF:FF) *также рассылаются по всем портам, кроме исходного*.



**Рис. 3.1.**Коммутатор и таблица коммутации

**Управление сетевыми интерфейсами в Linux.**

*Физический сетевой интерфейс* — устройство, способное отправлять в физическую *сеть* и получать из сети атомарные единицы передачи — фреймы; предоставляется драйвером устройства и имеет определенное имя, которое используется для доступа к нему (например, драйвера **Ethernet** *ядро* именует как **eth0, eth1** и т.д.). Фактически, чтобы использовать *сетевой интерфейс*, нужно настроить на нем какой-либо протокол сетевого уровня (например, *протокол IP*).

Для управления сетевыми интерфейсами на уровне доступа к сети можно использовать команду ip из пакета **iproute2** с параметром link [[ 22 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.22).

# ip link show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 16436 qdisc noqueue state UNKNOWN

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP

qlen 1000 link/ether 00:1d:72:01:ab:93 brd ff:ff:ff:ff:ff:ff

3: irda0: <NOARP> mtu 2048 qdisc noop state DOWN qlen 8

link/irda 00:00:00:00 brd ff:ff:ff:ff

4: vboxnet0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN

qlen 1000 link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff

5: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER\_UP> mtu 1400 qdisc

pfifo\_fast state UNKNOWN qlen 3 link/ppp

Листинг 3.1. Вывод информации о сетевых интерфейсах (ip link show):

*Вывод* данной команды содержит пронумерованный *список* интерфейсов, присутствующих в системе. *Информация об интерфейсе содержит две строки: в первой указывается имя интерфейса, установленные флаги состояния, MTU (Maximum Transmission Unit, максимально допустимый размер фрейма в байтах), тип и размер очереди фреймов; во второй строке — тип соединения, MAC-адрес, широковещательный адрес и т. п*.

Некоторые *флаги состояния*:

* UP — устройство подключено и готово принимать и отправлять фреймы;
* *LOOPBACK* — интерфейс является локальным и не может взаимодействовать с другими узлами в сети;
* *BROADCAST* - устройство способно отправлять широковещательные фреймы;
* POINTTOPOINT — соединение типа "точка-точка"
* PROMISC — устройство находится в режиме "прослушивания" и принимает все фреймы.
* NOARP — отключена поддержка разрешения имен сетевого уровня.
* ALLMULTI — устройство принимает все групповые пакеты.
* NO-*CARRIER* — нет связи (не подключен кабель).
* DOWN — устройство отключено.

Можно также вывести информацию о данном интерфейсе, задав его имя в параметре dev:

# ip link show dev eth0

или просто:

# ip link show eth0

**Примечание**:У *параметров команд* пакета **iproute2** есть сокращенный *синтаксис*, например, ip link show eth0 можно записать как ip l sh eth0.

*Команда* ip link также позволяет менять некоторые свойства сетевого интерфейса. Для этого используется *параметр* set.

# ip link set mtu 1400 eth1

# ip link show eth1

3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1400 qdisc pfifo\_fast state

UP qlen 1000 link/ether 08:00:27:66:6a:4f brd ff:ff:ff:ff:ff:ff

Листинг 3.2. Уменьшение MTU сетевого интерфейса с помощью ip link set mtu:

**Примечание**: *Все произведенные изменения командами, описываемые в данном разделе, будут действительны до перезагрузки компьютера или перезапуска сетевой подсистемы скриптами инициализации*. Как редактировать постоянные сетевые конфигурации, описано в ["Постоянные сетевые конфигурации (на примере Debian/GNU Linux)"](http://www.intuit.ru/studies/courses/681/537/lecture/12110). "Постоянные сетевые конфигурации".

# ip link set address 00:11:11:12:FE:09 eth1

# ip link show eth1

3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1400 qdisc pfifo\_fast state

UP qlen 1000 link/ether 00:11:11:12:fe:09 brd ff:ff:ff:ff:ff:ff

Листинг 3.3. Смена MAC-адреса сетевого интерфейса с помощью ip link set address

Используя ip link, можно включить или отключить *сетевой интерфейс* [пример 3.4](http://www.intuit.ru/studies/courses/681/537/lecture/12108?page=1#example.3.4)

# ip link set down eth1

# ip link set up eth1

Листинг 3.4. Выключение (down) и включение (up) сетевого интерфейса с помощью ip link

Ниже приводится пример использования команды ifconfig из пакета **net-tools** для управления сетевым интерфейсом [пример 3.5](http://www.intuit.ru/studies/courses/681/537/lecture/12108?page=1#example.3.5).

# ifconfig eth1 down

# ifconfig eth1 up

# ifconfig eth1 mtu 1400

# ifconfig eth1 hw ether 00:11:12:13:14:15

# ifconfig eth1

eth1 Link encap:Ethernet HWaddr 00:11:12:13:14:15

inet6 addr: fe80::a00:27ff:fe66:6a4f/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1400 Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:9 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:0 (0.0 B) TX bytes:706 (706.0 B)

Листинг 3.5. Использование ifconfig для управления сетевым интерфейсом

**Настройка физических параметров сетевого подключения**

Физические параметры (скорость, технология **Ethernet**, тип дуплекса) сетевого подключения зависят от используемого оборудования и обычно настраиваются автоматически при подключении компьютера к сети.

В отдельных случаях несогласованной работы оборудования и драйверов адаптера параметры можно выставить вручную, используя утилиту ethtool. Данная *утилита* включена в стандартные *репозитории*Debian; для ее установки достаточно выполнить команду:

#apt-get install ethtool

# ethtool eth1

Settings for eth1:

Supported ports: [ TP ]

Supported link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100baseT/Full

1000baseT/Full

Supports auto-negotiation: Yes

Advertised link modes: 10baseT/Half 10baseT/Full

100baseT/Half 100baseT/Full

1000baseT/Full

Advertised auto-negotiation: Yes

Speed: 1000Mb/s

Duplex: Full

Port: Twisted Pair

PHYAD: 0

Transceiver: internal

Auto-negotiation: on

Supports Wake-on: umbg

Wake-on: d

Current message level: 0x00000007 (7)

Link detected: yes

Листинг 3.6. Просмотр параметров сетевого интерфейса eth1 с помощью ethtool

*Вывод* содержит информацию о поддерживаемых технологиях **Ethernet**, автоматическом определении сети, текущей скорости передачи, дуплексе и т. д.

# ethtool -i eth1

driver: e1000

version: 7.3.20-k2-NAPI

firmware-version: N/A

bus-info: 0000:00:08.0

Листинг 3.7. Вывод сведений об используемых драйверах с помощью ethtool (параметр -i)

# ethtool -s eth1 speed 10 duplex half

Листинг 3.8. Установка (параметр -s) скорости подключения 10 Мбит/с (параметр speed) и передачи в режиме полудуплекса (параметр duplex)

**Ключевые термины**

**Локальная сеть** ( **LAN** - *Local* *Area* *Network*) — высокоскоростная *сеть передачи данных*, занимающая небольшую территорию.

**Ethernet** — *группа* технологий уровня доступа к сети, использующих в качестве среды передачи витую пару и *оптоволоконный кабель*.

**Физический адрес** ( **MAC-адрес**, *Media* Access *Control*) — 48-битный номер, присваиваемый сетевому адаптеру и используемый для адресации на уровне доступа к сети.

**Коммутатор** — многопортовое устройство уровня доступа к сети, осуществляющее пересылку **Ethernet** - фреймов между портами на основе информации, извлекаемой из заголовков фреймов, а также поддерживаемых таблиц, определяющих направление пересылки (таблиц коммутации).

**Дуплекс** — режим работы приемопередающего устройства, при котором устройство может передавать и получать информацию одновременно.

**Краткие итоги**

1. Группа технологий **Ethernet** на основе витой пары и оптоволоконного кабеля широко используется для построения локальных (и не только) сетей.
2. Сетевой адаптер обычно обрабатывает те фреймы, в адресе получателя которых указан его адрес или *широковещательный адрес*, а все остальные игнорирует.
3. Коммутатор **Ethernet** в рамках широковещательного домена согласно собственной таблице коммутации пересылает фреймы между узлами, подключенными к его портам.
4. Для управления физическими интерфейсами применятся команда ip link из пакета **iproute2** (справку по синтаксису данной команды, а также других можно получить с помощью подсистемы помощи, например **man ip**). Также сетевые параметры можно изменять с помощью команды ifconfig пакета **net-tools**.
5. Для настройки характеристик физического подключения (скорость, технология, тип дуплекса используется команда ethtool.

**Задание 4. Команды настройки протокола IP**

**Содержание:**

1. Протокол IP как средство организации межсетевого взаимодействия.
2. Управление IP-адресами в **Linux**.
3. Несколько IP-адресов на одном сетевом интерфейсе.
4. Управление **ARP** -таблицей.
5. Управление маршрутами.

**Протокол IP как средство организации межсетевого взаимодействия**

*Сетевой уровень модели сетевого взаимодействия обеспечивает доставку пакетов для узлов составной сети* [1] (совокупности подсетей, построенных на основе различных технологий уровня интерфейсов и использующих различные типы магистральных каналов для взаимодействия).

Основой сетевого уровня является *интернет*-протокол **IP** (*Internet Protocol*), имеющий следующие ключевые особенности:

1. **IP** *внедряет данные вышележащих уровней в соответствующие единицы передачи информации - сетевые пакеты;*
2. **IP** *является протоколом без установления соединения*, т. е. не требует физического или логического канала и не гарантирует доставку пакетов;
3. **IP** представляет собой маршрутизируемый протокол. *Организация протокола IP обеспечивает схему доставки пакетов на основе IP-адресации и группировки сетевых узлов в подсети*.

*Каждый IP-пакет помимо данных вышележащего уровня содержит заголовок, в котором указаны IP-адреса отправителя и получателя*, а также описан ряд параметров: тип службы, идентификационную информацию, номер протокола вышележащего уровня, параметры, используемые при фрагментации, контрольную сумму и т.д. (более подробные сведения приведены в [[ 7 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.7)[[ 23 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.23)). *Длина* *IP*-заголовка составляет 20 *байт*, максимальная *длина* пакета ( **MTU** ) — 64Кбайт (реальная *длина* пакета ограничена возможностями передающей среды).

**IP-адрес** ( *сетевой адрес* ) *протокола IP версии 4 (IPv4) представляет собой 4-х байтовый номер, который принято записывать в десятичной форме, разделяя байты точкой* (например: 192.168.4.3). Схема адресации *составной сети* (в т.ч. сети *Интернет*) является независимой от способов адресации узлов в отдельных сетях [[ 7 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.7). *Сетевой адрес объединяет в себе информацию об адресе сети и адресе узла в данной сети*.

Для решения задачи определения сетевой части *IP*-адреса (количество *бит* в *IP*-адресе, отведенное для записи адреса сети, обычно располагаемое в "начале") наиболее распространен механизм **сетевых масок***(двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети)*.*Маска* может записываться в формате, подобном *IP*-адресу, а также путем указания количества *бит*, отведенных для *адреса сети* (например, 255.255.255.0 то же самое, что и /24 ). Результатом *операции* логического умножения *IP*-адреса и маски будет *адрес сети*, которой принадлежит данный узел.

Примеры интерпретации *IP*-адресов в зависимости от *маски сети*:

192.168.92.0/24 — *сеть*, включающая узлы от 1 до 254 ( 0 — *адрес сети*; 255 — *зарезервирован для широковещательных рассылок, каждый последний узел в сети)*.

192.168.92.5 255.255.255.0 — узел в сети 192.168.92.0.

192.168.92.130/25 — узел в сети 192.168.92.128, которая включает узлы от 129 до 254.

Манипулируя масками, можно объединять и разделять *диапазон* *IP*-адресов. Например, *сеть* 192.168.9.0/24 может быть разбита на две подсети путем увеличения сетевой части адреса на 1 *бит*:192.168.9.0/25 (узлы 1-126 ) и 192.168.9.128/25 (узлы 129-254 ).

**Примечание**: Существуют программные средства (в т.ч. online) для автоматического расчета подсетей на основе масок, например, сайт [[ 24 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.24)и соответствующая программа **ipcalc**, которая присутствует в*репозиториях* **Debian**.

Управление адресацией сети *Интернет* осуществляется по иерархической схеме (региональные центры выдачи *IP*-адресов - национальные центры — *Интернет*-провайдеры — клиенты -субклиенты и т. д.). В качестве *координатора* выступает некоммерческая организация **ICANN** (*Internet* Corporation for Assigned Names *and* Numbers). Для организации **локальных (автономных) сетей** (не имеющих непосредственного доступа к сети *Интернет*) используются зарезервированные диапазоны *IP*-адресов: 10.0.0.0/8, 172.16.0.0-172.31.0.0/12, 192.168.0.0/16.

**Управление IP-адресами в Linux**

Для того, чтобы *сетевой протокол* мог использовать физический *интерфейс*, необходимо добавить соответствующий *адрес* *(настроить логический интерфейс). В случае стека TCP/IP минимальными настройками протокола IP, позволяющими узлу взаимодействовать с другими узлами подсети в рамках локального сегмента (широковещательного домена) являются IP-адрес и маска подсети*.

Для конфигурирования протокола *IP* на сетевом интерфейсе с помощью командной строки можно использовать команду address (addr) *утилиты* ip пакета iproute2 или традиционную команду ifconfig.

**Примечание**: *Все произведенные изменения командами, описываемые в данном разделе, будут действительны до перезагрузки компьютера или перезапуска сетевой подсистемы скриптами инициализации*. Как редактировать постоянные сетевые конфигурации, описано в ["Постоянные сетевые конфигурации (на примере Debian/GNU Linux)"](http://www.intuit.ru/studies/courses/681/537/lecture/12110)

*Для просмотра списка IP-адресов используется команда* ip addr show [[ 24 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.24)[[ 25 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.25)[[ 26 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.26). *Вывод* команды содержит *список* сетевых интерфейсов, их параметров, а также *список* *IP*-адресов ([пример 4.1.](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=1#example.4.1)) на каждом интерфейсе. Можно вывести информацию только по данному интерфейсу, указав его имя (например, ip addr show eth0 ).

$ ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 16436 qdisc noqueue state UNKNOWN

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP qlen 1000

link/ether 00:1d:72:01:ab:93 brd ff:ff:ff:ff:ff:ff

inet 192.168.11.11/24 brd 192.168.11.255 scope global eth0

inet6 fe80::21d:72ff:fe01:ab93/64 scope link

valid\_lft forever preferred\_lft forever

3: irda0: <NOARP> mtu 2048 qdisc noop state DOWN qlen 8

link/irda 00:00:00:00 brd ff:ff:ff:ff

4: vboxnet0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000

link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff

5: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER\_UP> mtu 1400 qdisc

pfifo\_fast state UNKNOWN qlen 3 link/ppp inet 10.31.46.4 peer 172.20.20.1/32

scope global ppp0

Листинг 4.1. Отображение списка IP-адресов с использованием команды ip addr show

Каждая *запись* о сетевом адресе может содержать информацию о типе адреса ( inet — *адрес* *IPv4*, inet6 — *IPv6* и т. д.), непосредственно *адрес*, *маску подсети* (количество сетевых *бит*),*широковещательный адрес* данной сети, *область видимости* ( scope global — действителен везде, scope link — только для данного устройства, scope host — для данного узла) и имя логического интерфейса.

*Сведения подобного рода также предоставляет команда* ifconfig, *дополнительно отображая статистику работы сетевого интерфейса* ([пример 4.2](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=1#example.4.2)).

$ ifconfig eth0

eth0 Link encap:Ethernet HWaddr 00:1d:72:01:ab:93

inet addr:192.168.11.11 Bcast:192.168.11.255 Mask:255.255.255.0

inet6 addr: fe80::21d:72ff:fe01:ab93/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:69333 errors:0 dropped:0 overruns:0 frame:0

TX packets:69736 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:37148192 (37.1 MB) TX bytes:5619320 (5.6 MB)

Interrupt:16

Листинг 4.2. Вывод информации о сетевом интерфейсе eth0 с помощью ifconfig

*Для удаления IP-адреса используется команда* ip addr del [пример 4.3](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=1#example.4.3). При этом необходимо указать удаляемый *IP*-*адрес* ( 192.168.11.11 ), маску подсети (/24) и имя интерфейса ( dev eth0 ). Изменение параметров сети требует полномочий *суперпользователя*.

# ip address del 192.168.11.11/24 dev eth0

# ip addr show eth0

2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP qlen 1000

link/ether 00:1d:72:01:ab:93 brd ff:ff:ff:ff:ff:ff

inet6 fe80::21d:72ff:fe01:ab93/64 scope link

valid\_lft forever preferred\_lft forever

Листинг 4.3. Удаление IP-адреса с интерфейса eth0 с помощью ip addr del

*Для добавления IP-адреса используется команда* ip addr add ([пример 4.4](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=1#example.4.4).). При этом необходимо указать добавляемый *IP*-*адрес*, *маску подсети*, имя интерфейса, а также *широковещательный адрес* (в данном случае автоматически рассчитать *широковещательный адрес* на основе маски: *brd* +, можно также указать *адрес* вручную: *brd* 192.168.11.255 ).

# ip address add 192.168.11.11/24 brd + dev eth0

Листинг 4.4. Добавление IP-адреса на интерфейс eth0 c использованием ip addr add

Добавить *IP*-*адрес* можно также с помощью команды ifconfig ([пример 4.5](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=1#example.4.5)).

# ifconfig eth0 inet up 192.168.11.11 netmask 255.255.255.0

Листинг 4.5. Добавление IP-адреса на интерфейс eth0 с помощью ifconfig

**Несколько IP-адресов на одном сетевом интерфейсе**

В некоторых случаях необходимо использовать несколько *IP*-адресов на одном сетевом интерфейсе [[ 25 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.25).

*Пусть на сетевом интерфейсе eth1* установлен *IP*-*адрес* 192.168.11.10/24, необходимо, чтобы система принимала на этом интерфейсе входящие соединения на *адрес* 192.168.11.11. Добавим на *интерфейс* еще один *адрес*:

# ip addr add 192.168.11.11/24 brd + dev eth1

# ip addr show eth1

3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 08:00:27:66:6a:4f brd ff:ff:ff:ff:ff:ff

inet 192.168.11.10/24 brd 192.168.11.255 scope global eth1

inet 192.168.11.11/24 brd 192.168.11.255 scope global secondary eth1

inet6 fe80::a00:27ff:fe66:6a4f/64 scope link

valid\_lft forever preferred\_lft forever

Листинг 4.6. Добавление дополнительного IP-адреса

При добавлении адреса из той же сети, что и существующий, новый *адрес* становится дополнительным к существующему ( **secondary** ), и если удалить основной *адрес*, то будет удален и дополнительный.

**Примечание**: Команда ifconfig непосредственно не поддерживает несколько IP-адресов на одном сетевом интерфейсе (только в виде псевдонимов). Поэтому **secondary** - адрес в выводе ifconfigотображаться не будет:

# ifconfig eth1

eth1 Link encap:Ethernet HWaddr 08:00:27:66:6a:4f

inet addr:192.168.11.10 Bcast:192.168.11.255 Mask:255.255.255.0

inet6 addr: fe80::a00:27ff:fe66:6a4f/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:74 errors:0 dropped:0 overruns:0 frame:0

TX packets:74 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:8640 (8.4 KiB) TX bytes:7020 (6.8 KiB)

**Примечание**: *Для удаления всех сетевых адресов на интерфейсе можно использовать команду* ip addr *flush*:

# ip addr *flush* dev eth1

# ip addr add 192.168.11.11/24 brd + dev eth1 label eth1:add

# ip addr show eth1

3: eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 08:00:27:66:6a:4f brd ff:ff:ff:ff:ff:ff

inet 192.168.11.10/24 brd 192.168.11.255 scope global eth1

inet 192.168.11.11/24 brd 192.168.11.255 scope global secondary eth1:add

# ifconfig

eth1 Link encap:Ethernet HWaddr 08:00:27:66:6a:4f

inet addr:192.168.11.10 Bcast:192.168.11.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:756 errors:0 dropped:0 overruns:0 frame:0

TX packets:711 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:75800 (74.0 KiB) TX bytes:68550 (66.9 KiB)

eth1:add Link encap:Ethernet HWaddr 08:00:27:66:6a:4f

inet addr:192.168.11.11 Bcast:192.168.11.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

Листинг 4.7. Для реализации ifconfig - совместимого способа использования нескольких IP-адресов на одном сетевом интерфейсе следует в команде ip addr add использовать параметр label

**Управление ARP-таблицей**

В сегменте сети **Ethernet** для взаимодействия устройств друг с другом по протоколу *IP* необходимо, чтобы у передающего устройства были *логический* (*IP*-) и физический (MAC-) адреса получателя. *Для решения задачи определения MAC-адреса узла назначения по его IP-адресу в локальном сегменте используется протокол ARP (Address Resolution Protocol)* [[ 7 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.7)*.Принцип действия протокола ARP заключается в том, что отправитель сетевого пакета посылает широковещательный ARP-запрос (на широковещательный MAC-адрес FF:FF:FF:FF:FF:FF) с указанием искомого IP-адреса. ARP-запрос получают все устройства в сегменте, но только тот, чей IP-адрес является искомым, отправляет ARP-ответ (на MAC-адрес отправителя) с указанием своего MAC-адреса*.

Каждый узел локальной сети поддерживает **ARP-таблицу** (*кэш*), *которая устанавливает соответствие между IP и MAC-адресами узлов подсетей, в которые входит данный узел*. В **ARP-таблице** имеются два вида записей:

* **Динамические записи**, которые периодически обновляются с использованием *протокола ARP* (если запись "устаревает", то она удаляется).
* **Статические записи**, которые создаются пользователем с помощью соответствующих команд и существуют до тех пор, пока узел не будет выключен.

Используя команду ip neigh или традиционную команду arp можно манипулировать содержимым **ARP-таблицы**.

**Пример 4.8**. Просмотр *ARP*-таблицы.

Если узел 192.168.11.10 через *интерфейс* **eth1** осуществит сетевое взаимодействие с узлом 192.168.11.12 (например, с помощью утилиты **ping** ), то в его **ARP-таблице** появится новая *запись*, которую можно просмотреть командой ip neigh show (*параметр* dev указывает фильтровать записи относящиеся к интерфейсу **eth1** ):

# ping -c 1 192.168.11.12

PING 192.168.11.12 (192.168.11.12) 56(84) bytes of data.

64 bytes from 192.168.11.12: icmp\_seq=1 ttl=64 time=1.58 ms

--- 192.168.11.12 ping statistics ---

1 packets transmitted, 1 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 1.588/1.588/1.588/0.000 ms

# ip neigh show dev eth1

192.168.11.12 lladdr 08:00:27:23:22:97 REACHABLE

**Пример 4.9**. Добавление статической записи в *ARP*-таблицу.

Для добавления статической записи в **ARP-таблицу** можно использовать команду ip neigh add c указанием *IP*-адреса, **MAC-адреса** (*параметр* lladdr ) и сетевого интерфейса (*параметр* dev ):

# ip neigh add 192.168.11.100 lladdr 00:00:00:00:00:AA dev eth1

# ip neigh show dev eth1

192.168.11.100 lladdr 00:00:00:00:00:aa PERMANENT

*Для удаления записи из ARP-таблицы можно использовать команду* ip neigh del:

# ip neigh del 192.168.11.100 dev eth1

# arp -s 192.168.11.100 00:00:00:00:00:AA

# arp -i eth1

Address HWtype HWaddress Flags Mask Iface

192.168.11.100 ether 00:00:00:00:00:aa CM eth1

# arp -d 192.168.11.100

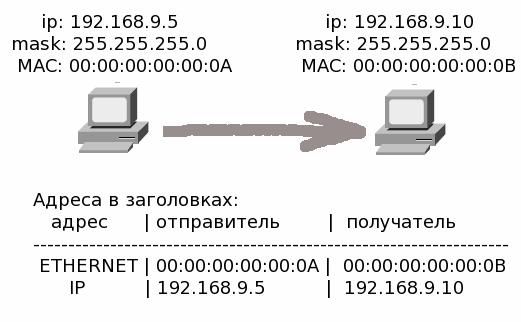
Листинг 4.10. Использование команды arp для манипулирования ARP-таблицей (ключ -s — добавить статическую запись, -i — фильтр интерфейса, -d— удалить запись)

**Управление маршрутами**

Доставку *сетевых пакетов* в рамках локального сегмента сети называют локальной.

**Пример 4.11**. *Локальная доставка пакетов*

*IP*-пакет с заголовком 192.168.9.5 > 192.168.9.10 ([рис. 4.1](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=3#image.4.1)) будет доставлен локально (*фрейм* отправляется непосредственно на **MAC-адрес** узла 10, т. к. согласно маске /24 узел 10 находится в одной подсети с узлом-отправителем).

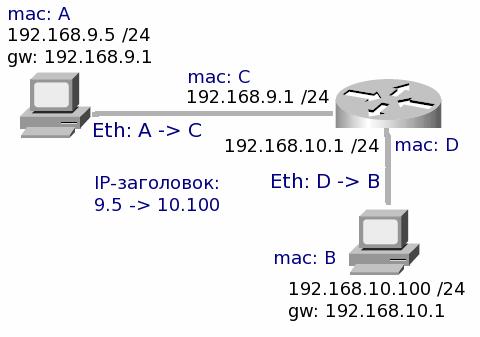


**Рис. 4.1.**Локальная доставка сетевого пакета

**Стандартный шлюз** (*шлюз* по умолчанию) - *IP-адрес интерфейса узла данной подсети, способный отправлять и принимать пакеты удаленных подсетей для данной подсети. Все IP-пакеты, предназначенные для узлов из других подсетей, данный узел отправляет на MAC-адрес стандартного шлюза* **("нелокальная" доставка** ).

**Пример 4.12**. *"Нелокальная" доставка пакетов*

*Сетевой пакет* 192.168.9.5 /24 > 192.168.10.100 ([рис. 4.2](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=3#image.4.2)) будет отправлен на **MAC-адрес** шлюза данной подсети (*шлюз* 192.168.9.1 — должен быть указан в настройках данного узла как *шлюз* по умолчанию).



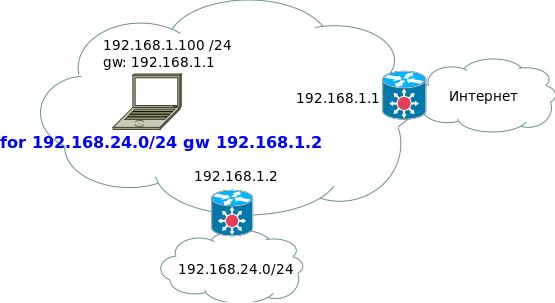
**Рис. 4.2.**Использование шлюза по умолчанию

Для отдельных сетей и узлов можно указывать маршруты через **альтернативные шлюзы**. При этом, *все шлюзы должны иметь IP-адреса из той же подсети, что и передающий узел, а также находится в одном широковещательном домене с передающим узлом* (*сеть* **Ethernet** ).

*Маршрутизация на узле (решение на какой шлюз отправлять пакет, предназначенный узлу "нелокальной" сети) осуществляется на основе* **таблицы маршрутизации**.

**Пример 4.13**. *Добавление альтернативного шлюза*.

Пусть в локальной сети 192.168.1.0/24 есть *шлюз* сети *Интернет* 192.168.1.1 (использующий, например трансляцию сетевых адресов - **NAT** ) и *шлюз* удаленной сети 192.168.24.0/24, имеющий *адрес* 192.168.1.2 ([рис.4.3](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=3#image.4.3)).



**Рис. 4.3.**Использование маршрута по умолчанию и альтернативного маршрута

Для того, чтобы *компьютер* с адресом 192.168.1.100/24 ([рис. 4.3](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=3#image.4.3)) имел *доступ* в *сеть* *Интернет* и к локальной сети 192.168.24.0/24, необходимо внести в таблицу маршрутизации соответствующие записи, например с помощью команды ip route add:

# ip route add default via 192.168.1.1

# ip route add 192.168.24.0/24 via 192.168.1.2

# ip route show

192.168.1.0/24 dev eth1 proto kernel scope link src 192.168.1.100

192.168.24.0/24 via 192.168.1.2 dev eth1

default via 192.168.1.1 dev eth

Первая *команда* добавляет *маршрут* по умолчанию ( default ) через узел 192.168.1.1 (*параметр* via ). Вторая *команда* устанавливает *маршрут* на *сеть* 192.168.24.0/24 через узел 192.168.1.2.

Для вывода на экран содержимого таблицы маршрутизации используется *команда* ip route show. В данном случае в таблице маршрутизации три записи: первая о том, что *сеть* 192.168.1.0/24 доступна непосредственно на интерфейсе **eth1** (*запись* добавляется автоматически), и две записи, добавленные пользователем: альтернативный *маршрут* и *маршрут* по умолчанию.

Те же действия (см. выше) можно выполнить, используя команду route:

# route add default gw 192.168.1.1

# route add -net 192.168.24.0 netmask 255.255.255.0 gw 192.168.1.2

# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1

192.168.24.0 192.168.1.2 255.255.255.0 UG 0 0 0 eth1

0.0.0.0 192.168.1.1 0.0.0.0 UG 0 0 0 eth

**Примечание**: При нескольких альтернативных маршрутах, возможных для данного узла выбирается тот, у которого *маска подсети* "более заполнена" [пример 4.14](http://www.intuit.ru/studies/courses/681/537/lecture/12109?page=3#example.4.14).

# ip route show

192.168.10.3 via 192.168.56.2 dev eth0

192.168.10.0/25 via 192.168.56.3 dev eth0

192.168.10.0/24 via 192.168.56.2 dev eth0

192.168.56.0/24 dev eth0 proto kernel scope link src 192.168.56.101

default via 192.168.56.1 dev eth0

Листинг 4.14. Выбор альтернативного маршрута

В данном примере есть несколько перекрывающихся маршрутов (1-3 записи). В соответствии с "заполненностью" маски, пакеты для узла 192.168.10.3 будут отправляться шлюзу 192.168.56.2, т. к. у этой записи самая "большая" *маска* (/32). Пакеты остальных узлов сети 192.168.10.0/25 будут отправляться шлюзу 192.168.56.3, так как у второй записи в таблице маршрутизации *маска* (/25) больше, чем у третьей (/24). Пакеты, предназначенные узлам 128-255 сети 192.168.10.0/24 будут отправляться шлюзу 192.168.56.2.

**Ключевые термины**

**IP (Internet Protocol)** — маршрутизируемый протокол сетевого уровня без установления соединения.

**IP-адрес (сетевой адрес)** — *адрес*, используемый протоколом *IP* для реализации схемы адресации в рамках *составной сети*.

**Маска подсети** — способ указания сетевой части *IP*-адреса, основанный на записи единиц в тех битах *IP*-адреса, которые указывают на *адрес сети*.

**ARP** (*Address Resolution* *Protocol*) — протокол, решающий задачу определения *MAC-адреса* по его *IP*-адресу в рамках сегмента сети (широковещательного домена).

**ARP-таблица** (*кэш*) — *таблица* соответствия *IP*-адресов узлов данной подсети их **MAC-адресам**, поддерживаемая узлом.

**Маршрутизация** — доставка *сетевого пакета* получателю по оптимальному маршруту.

**Стандартный шлюз** (*шлюз* по умолчанию) - *IP*-*адрес* интерфейса узла данной подсети, способный отправлять и принимать пакеты удаленных подсетей для данной подсети.

**Краткие итоги**

1. Протокол IP является маршрутизируемым протоколом без установления соединения. IP адрес представляет собой комбинацию номера сети и номера узла в данной сети. Количество бит, используемых для номера сети определяется по классу адреса или задается битовой маской.
2. Для того, чтобы узел мог взаимодействовать с другими узлами подсети в рамках локального сегмента (широковещательного домена) на нем должен быть установлен IP-адрес и определена *маска подсети*. Для управления IP-адресами в **Linux** можно использовать команду ip addr или традиционную команду ifconfig (подробный синтаксис и параметры большинства команд **Linux** можно посмотреть с помощью справочной подсистемы **man**, например: **man ifconfig** ).
3. В локальной сети данный узел для определения физических адресов по IP - адресам других узлов использует протокол **ARP** и локальный **ARP** -кэш (таблица). Управление **ARP** -таблицей осуществляется командой ip neigh или arp.
4. Для взаимодействия с удаленными подсетями на данном узле необходимо определить шлюз по умолчанию или/и альтернативные шлюзы. Все шлюзы должны находиться в той же подсети, что и исходный узел. Управление таблицей маршрутизации на узле может осуществляться командой ip route или route.

**Задание 5. Постоянные сетевые конфигурации (на примере Debian/GNU Linux)**

**Содержание:**

1. Автоматическая инициализация сетевой подсистемы
2. *Файл настроек* сетевых интерфейсов /etc/network/interfaces
3. Добавление постоянного статического маршрута
4. Несколько IP-адресов на одном сетевом интерфейсе
5. Несколько *виртуальных локальных сетей* ( **VLAN** ) на одном интерфейсе

**Автоматическая инициализация сетевой подсистемы**

Пакет **ifupdown** является комплексным средством настройки параметров сети, в частности используется для инициализации сети при загрузке операционной системы в **Debian GNU\Linux** (совместно с соответствующими скриптами ifupdown-clean, ifupdown, networking, расположенными в каталоге /etc/init.d) [[ 28 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.28).

**Примечание**: Начиная с версии 6.0 (*squeeze*), разработчики **Debian** в документации классифицируют пакет **ifupdown** как устаревший и рекомендуют пользоваться такими средствами как **NetworkManager** или**Wicd**. Такой подход себя оправдывает для рабочих станций с установленными средствами графического интерфейса пользователя. Для серверов продолжается использование **ifupdown** - стабильного, многофункционального, хорошо документированного средства управления сетевой подсистемой.

Пакет **ifupdown** содержит две команды ifup и ifdown для включения и отключения сетевого подключения ([пример 5.1](http://www.intuit.ru/studies/courses/681/537/lecture/12110?page=1#example.5.1)). Данные команды по умолчанию используют настройки, записанные в файле /etc/*network*/interfaces.

# ifdown eth1

# ifup eth1

Листинг 5.1. Использование ifdown и ifup для отключения и включения сетевого интерфейса eth1

*Для запуска, перезапуска и останова сетевой подсистемы следует использовать скрипт* /etc/init.d/networking *c параметрами* start, *restart* и stop *соответственно* ([пример 5.2](http://www.intuit.ru/studies/courses/681/537/lecture/12110?page=1#example.5.2)).

# /etc/init.d/networking restart

Листинг 5.2. Перезапуск сети с помощью /etc/init.d/networking

**Файл настроек сетевых интерфейсов /etc/network/interfaces**

*Файл /etc/network/interfaces имеет текстовый формат, пригодный для редактирования администратором системы c помощью текстового редактора*, в тоже время команды ifup и ifdown также способны его читать и распознавать указанные в нем настройки.

auto lo eth1 eth0

iface lo inet loopback

iface eth1 inet static

address 192.168.1.100

netmask 255.255.255.0

gateway 192.168.1.1

dns-nameservers 192.168.1.1

iface eth0 inet dhcp

Листинг 5.3. Файл /etc/network/interfaces

*Ключевое слово* auto ([пример 5.3](http://www.intuit.ru/studies/courses/681/537/lecture/12110?page=1#example.5.3)) c перечислением через *пробел* имен интерфейсов указывает включать данные интерфейсы при старте системы. *Ключевое слово* iface является описанием интерфейса (общий формат: iface <имя\_интерфейса> <тип адреса> <метод\_настройки> ). Так:

1. в строке iface lo inet *loopback* настраивается *локальный интерфейс lo* для взаимодействия приложений в рамках данного компьютера ( **loopback** ).
2. в строке iface eth1 inet static настраивается интерфейс eth1 на статический метод конфигурации (сетевые параметры указываются вручную, **static** ). Далее параметры статической настройки интерфейса и указаны (формат в общем виде: <опция> <значение> ): IP-адрес ( address ), *маска подсети* ( *netmask* ), шлюз по умолчанию ( gateway ), адреса *DNS-серверов* ( dns-*nameservers* ) и т.д.
3. в строке iface eth0 dhcp указывается настройка интерфейса **eth0** по *протоколу динамической конфигурации узла* ( **dhcp** ).

Протокол **DHCP** (*Dynamic* *Host* *Configuration* *Protocol*) *позволяет компьютеру автоматически получать по сети IP-адрес и другие параметры, необходимые для работы сетевого интерфейса*. Для использования протокола **DHCP** необходимо чтобы в данном широковещательном домене был настроен **DHCP-сервер**. При настройке сетевого устройства *компьютер* обращается к **DHCP-серверу** и получает требуемые сетевые параметры.

**Примечание**. Кроме статической и динамической настройки интерфейса, существует ручной способ конфигурирования ( **manual** ), который предполагает то, что интерфейс будет настроен с помощью средств, сторонних относительно ifupdown.

**Примечание**. Рекомендуется отключать сетевой интерфейс ( ifdown ) перед тем как вносить изменения в его конфигурацию, а затем включать ( ifup ).

**Примечание**. Синтаксис файла interfaces подробно описан на соответствующей странице помощи ( man interfaces ).

**Добавление постоянного статического маршрута**

Опции интерфейса в файле interfaces позволяют указать команды, которые необходимо выполнить при включении интерфейса (*опция* up ) и выключении (*опция* down ). В частности данные опции можно использовать для настройки дополнительных маршрутов ([пример 5.4](http://www.intuit.ru/studies/courses/681/537/lecture/12110?page=1#example.5.4).).

Настроим постоянную сетевую конфигурацию для решения задачи, описанной в примере 4.13 ["Команды настройки протокола IP"](http://www.intuit.ru/studies/courses/681/537/lecture/12109), с добавлением статического маршрута через альтернативный *шлюз* (приведен фрагмент файла /etc/*network*/interfaces):

iface eth1 inet static

address 192.168.1.100

netmask 255.255.255.0

up ip route add 192.168.24.0/24 via 192.168.1.2

gateway 192.168.1.1

Листинг 5.4. Добавление постоянного статического маршрута

Применим данную конфигурацию с помощью команд ifdown/ifup и просмотрим *таблицу маршрутизации*:

# ifdown eth1

# ifup eth1

# ip route show dev eth1

192.168.1.0/24 proto kernel scope link src 192.168.1.100

192.168.24.0/24 via 192.168.1.2

default via 192.168.1.1

**Несколько IP-адресов на одном сетевом интерфейсе**

Для решения задачи, описанной в примерах 4.6. и 4.7. ["Команды настройки протокола IP"](http://www.intuit.ru/studies/courses/681/537/lecture/12109)отредактируем *файл* /etc/*network*/interfaces следующим образом (приведен отрывок файла):

auto eth1 eth1:add

iface eth1 inet static

address 192.168.11.10

netmask 255.255.255.0

gateway 192.168.11.1

iface eth1:add inet static

address 192.168.11.11

netmask 255.255.255.0

Листинг 5.5. Добавление дополнительного IP-адреса

В данном случае *псевдоним* задается через двоеточие после имени интерфейса. Также необходимо указать автоматический *запуск* нового интерфейса в параметре auto. Для применения данной конфигурации необходимо выполнить последовательность команд:

# ifdown eth1; ifup eth1; ifup eth1:add

или

# /etc/init.d/networking restart

**Несколько виртуальных локальных сетей (VLAN) на одном интерфейсе**

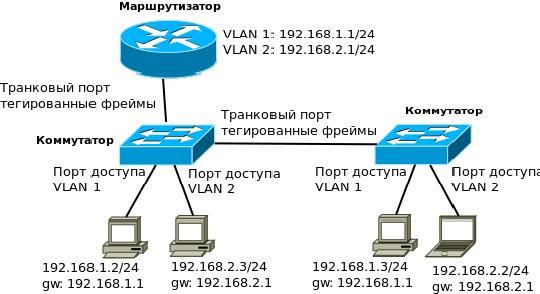
*Технология виртуальных локальных сетей (Virtual Local Area Network, VLAN) позволяет в рамках одной физической коммутируемой сети выделить несколько изолированных на канальном уровне виртуальных сетей*[[ 7 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.7)*. Для организации VLAN необходимо, чтобы телекоммуникационное оборудование (коммутаторы) поддерживало эту функцию; для взаимодействия виртуальных локальных сетей друг с другом используют средства сетевого уровня (маршрутизация)*.

Существуют два способа группировки компьютеров сети в виртуальные сети [[ 7 ]](http://www.intuit.ru/studies/courses/681/537/literature#literature.7):

* **Группировка портов коммутатора**. Каждый порт коммутатора сопоставляют соответствующей виртуальной сети.
* **Группировка MAC-адресов**. При данном подходе каждый *MAC-адрес* в сети должен быть прописан в той или иной виртуальной сети.

*Независимо от способа группировки коммутатор коммутирует фреймы (в том числе и широковещательные) в рамках VLAN и не коммутирует их между VLAN*.

*Для передачи данных между коммутаторами фреймы соответствующих виртуальных сетей маркируются (тегируются), т. е. в заголовке Ethernet-фрейма указывается номер виртуальной сети. Соответствующие порты, соединяющие коммутаторы друг с другом называются* **транковыми**. В современных сетях для пересылки *VLAN*-трафика между коммутаторами используется протокол **IEEE 802.1Q** (могут встречаться также фирменные протоколы, совместимые с оборудованием данного производителя).



**Рис. 5.1.**Пример использования группировки портов

На [рис. 5.1](http://www.intuit.ru/studies/courses/681/537/lecture/12110?page=2#image.5.1). изображен пример построения виртуальной локальной сети на основе группировки по портам. Порты, к которым подключены рабочие станции, настроены на коммутаторах как **порты доступа**, т. е. каждому порту доступа сопоставлен номер **VLAN** (1 или 2); при этом фреймы на этих портах "нетегированы". Коммутаторы и *маршрутизатор* соединены транковыми портами, на которых фреймы каждого **VLAN**нумеруются соответственно. На коммутаторах сети **VLAN 1** и **VLAN 2** изолированы друг от друга. Их взаимодействие происходит на маршрутизаторе на сетевом уровне (если фильтры маршрутизатора позволяют такое взаимодействие); для этого первой виртуальной сети сопоставлена IP-*подсеть* 192.168.1.0/24 и в ней *маршрутизатор* имеет *адрес* 192.168.1.1, вторая *сеть* — 192.168.2.0/24 и *адрес* интерфейса маршрутизатора - 192.168.2.1.

*Ядро Linux поддерживает тегированные фреймы стандарта 802.1Q* ([пример 5.6](http://www.intuit.ru/studies/courses/681/537/lecture/12110?page=2#example.5.6)). Для настройки **VLAN** используется *утилита* **vconfig** из пакета **vlan**. Ее необходимо установить из стандартных *репозиториев***Debian**:

# apt-get install vlan

Пусть требуется подключить *компьютер* с **Linux** в рамках схемы, изображенной на [рис. 5.1](http://www.intuit.ru/studies/courses/681/537/lecture/12110?page=2#image.5.1), таким образом чтобы он имел *доступ* как к первой виртуальной сети, так и второй. Для этого, предварительно необходимо настроить *порт* коммутатора, к которому подключается данный *компьютер*, как транковый. Затем в файле /etc/*network*/interfaces указать (приведен фрагмент файла):

auto vlan1

iface vlan1 inet static

address 192.168.1.4

netmask 255.255.255.0

vlan\_raw\_device eth1

auto vlan2

iface vlan2 inet static

address 192.168.2.4

netmask 255.255.255.0

vlan\_raw\_device eth1

Листинг 5.6. Настройка виртуальных частных сетей на сетевом интерфейсе

где, номер виртуальной сети определяется в строке описания интерфейса ( vlan1 — первый, vlan2 — второй), а физический *интерфейс* определяется опцией vlan\_raw\_device (в данном случае eth1 ).

Для применения изменений перезапустим сетевую подсистему:

# /etc/init.d/networking restart

Отрывок вывода команды ip addr show для данной конфигурации:

4: vlan1@eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP

link/ether 08:00:27:66:6a:4f brd ff:ff:ff:ff:ff:ff

inet 192.168.1.4/24 brd 192.168.1.255 scope global vlan1

inet6 fe80::a00:27ff:fe66:6a4f/64 scope link

valid\_lft forever preferred\_lft forever

5: vlan2@eth1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP

link/ether 08:00:27:66:6a:4f brd ff:ff:ff:ff:ff:ff

inet 192.168.2.4/24 brd 192.168.2.255 scope global vlan2

inet6 fe80::a00:27ff:fe66:6a4f/64 scope link

valid\_lft forever preferred\_lft foreve

**Ключевые термины**

**DHCP** (*Dynamic* *Host* *Configuration* *Protocol*) - протокол для автоматической настройки сетевых параметров с использованием специального сервера *DHCP*.

**DNS** (*Domain Name System*) — иерархическая система, позволяющая преобразовать символьные имена узлов и ресурсов *Интернет* в IP-адреса и обратно.

**Виртуальная локальная сеть** ( **VLAN**, *Virtual* *LAN*) — изолированный сегмент сети уровня сетевых интерфейсов, не привязанный к физическому расположению узлов.

**Краткие итоги**

1. Пакет **ifupdown** представляет собой комплексное средство настройки сетевых подключений в **Linux** (в частности **Debian GNU\Linux** ). Основными компонентами пакета являются команды ifup и ifdown, а также конфигурационный файл /etc/network/interfaces.
2. Файл /etc/network/interfaces можно редактировать с помощью любого текстового редактора (имея права суперпользователя). Секция описания интерфейса **iface** позволяет задать статический ( static ), динамический ( dhcp ) или ручной (с помощью команд, *manual* ) способ настройки интерфейса. Опции настройки интерфейсов дают возможность указать основные характеристики интерфейса, установить дополнительные IP-адреса, выполнить заданные действия при включении и выключении интерфейса и т. д.
3. **Linux** поддерживает работу с тегированными фреймами Ethernet стандарта **802.1Q** на уровне ядра. Для управления параметрами *виртуальных локальных сетей* используется утилита **vconfig** из пакета **vlan**. Соответствующую конфигурацию сетевого интерфейса можно описать в файле /etc/network/interfaces.