

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

А. В. Королькова, Д. С. Кулебов

Прикладные протоколы Интернет и www

Лабораторные работы

Москва

Российский университет дружбы народов
2012

УДК 004.72+004.057.4

ББК 73

К 68

Утверждено

РИС Учёного совета

Российского университета

дружбы народов

Рецензент —

начальник сектора телекоммуникаций УИТО РУДН
кандидат физико-математических наук, доцент *К. П. Ловецкий*

Королькова, А. В.

К 68 Прикладные протоколы Интернет и www [Текст] :
лабораторные работы / А. В. Королькова, Д. С. Кулябов. —
М. : РУДН, 2012. — 135 с. : ил.

Учебно-методическое пособие предназначено для студентов направлений «Прикладная математика и информатика», «Математика и компьютерные науки», «Фундаментальная информатика и информационные технологии», «Бизнес-информатика».

ISBN 978-5-209-04935-7

© Королькова А. В., Кулябов Д. С., 2012

© Российского университета дружбы народов, Издательство, 2012.

Содержание

Лабораторная работа № 1. Поисковые системы Интернет. Организация поисковых запросов.	5
1.1. Цель работы	5
1.2. Информация, необходимая для начала работы	5
1.3. Последовательность выполнения работы	6
1.4. Оформление результатов	9
Лабораторная работа № 2. Установка и конфигурация операционной системы на виртуальную машину	10
2.1. Цель работы	10
2.2. Информация, необходимая для начала работы	10
2.3. Последовательность выполнения работы	10
Лабораторная работа № 3. Настройка DNS-сервера	30
3.1. Цель работы	30
3.2. Информация, необходимая для начала работы	30
3.3. Основные понятия DNS	30
3.4. Сетевые утилиты	32
3.5. Последовательность выполнения работы	34
3.6. Контрольные вопросы.	49
Лабораторная работа № 4. Настройка DHCP-сервера и HTTP-сервера	50
4.1. Цель работы	50
4.2. Информация, необходимая для начала работы	50
4.3. Кратко о DHCP	50
4.4. Последовательность выполнения работы	51
4.5. Контрольные вопросы.	66
Лабораторная работа № 5. Установка и настройка почтового сервера	67
5.1. Цель работы	67
5.2. Информация, необходимая для начала работы	67
5.3. Последовательность выполнения работы	67
5.4. Контрольные вопросы.	86
Лабораторная работа № 6. Программирование клиентских и серверных сценариев WEB-приложений. HTML, технология PHP и СУБД MySQL	87
6.1. Цель работы	87
6.2. Информация, необходимая для начала работы	87
6.3. Теоретическая часть	89
6.4. Порядок выполнения работы	92

Лабораторная работа № 7. Форматирование и преобразование XML-документа с помощью XSL. XSLT-преобразование XML-документа.	106
7.1. Цель работы	106
7.2. Теоретический материал	106
7.3. Практическое задание: стилевые таблицы XSL и XML	107
7.4. Задание для самостоятельного выполнения.	114
Лабораторная работа № 8. Применение JavaScript для добавления интерактивных возможностей веб-страниц .	116
8.1. Цель работы	116
8.2. Информация, необходимая для начала работы	116
8.3. Примеры реализации интерактивных возможностей веб-страниц	122
8.4. Задания для самостоятельного выполнения	130
Предметный указатель	131
Список иллюстраций	131
Список таблиц	134

Лабораторная работа № 1. Поисковые системы Интернет. Организация поисковых запросов

1.1. Цель работы

Изучить возможности нескольких поисковых систем Интернет (не менее четырёх), например: <http://www.google.com>, <http://www.nigma.ru>, <http://arxiv.org>, <http://www.yandex.ru>.

При изучении функций и возможностей исследуемых поисковых систем следует использовать доступную документацию и интерактивные справочные системы.

1.2. Информация, необходимая для начала работы

Информационно-поисковая система (Information Retrieval System, IRS) — система, предназначенная для обеспечения поиска и отображения документов, представленных в базах данных.

Основные характеристики поисковых запросов:

- *релевантность* — степень соответствия найденной в поисковой системе информации запросу пользователя;
- *ранжирование* — построение поисковой машиной списка релевантных запросу сайтов, расположенных в соответствии с весом и ценностью запрашиваемой информации;
- *степень индексации сайта в поисковых системах* — показатель соотношения числа веб-страниц, проиндексированных конкретной поисковой системой, к действительному числу веб-страниц сайта;
- *пертинаентность* — соответствие полученных в результате поиска документов информационным потребностям пользователя, а не формальному запросу.

Для вычисления показателей качества поиска принято рассматривать табл. 1.1, которую заполняют по результатам поиска.

Таблица 1.1
Результаты поиска

Документы	Выданные	Невыданные
Релевантные	<i>a</i>	<i>c</i>
Нерелевантные	<i>b</i>	<i>d</i>

Некоторые показатели информационного поиска:

- коэффициент полноты (recall): $r = \frac{a}{a + c}$;
- коэффициент точности (precision): $P = \frac{a}{a + b}$;
- коэффициент аккуратности (accuracy): $acc = \frac{a + d}{a + b + c + d}$;
- ошибка (error): $err = \frac{b + c}{a + b + c + d}$.

1.3. Последовательность выполнения работы

1. Зафиксировать в отчете список критериев оценки эффективности поисковых систем. Предлагаются следующие критерии:
 - 1) полнота поиска;
 - 2) точность поиска.
 Перечень критериев оценки эффективности поисковых систем может быть изменен и/или расширен.
2. Назначить весовые коэффициенты (веса) принятым критериям. Весовой коэффициент — параметр, отражающий значимость, относительную важность, «вес» данного критерия в сравнении с другими критериями. Например, для точности поиска весовой коэффициент принимаем равным 0,8, для полноты поиска – 0,2. Сумма всех весов должна быть равной 1.
3. Сформулировать и зафиксировать в отчёте несколько запросов (как на русском, так и на английском языке) по темам «Сетевые технологии» и «Протоколы Интернет», например:
 - виртуальная машина VirtualBox;
 - Berkeley Internet Name Daemon;
 - mathematical model of the protocol ...;
 - Linux.
 Запрос должен включать в себя несколько (не менее двух) ключевых слов.
4. Выполнить каждый запрос в каждой исследуемой поисковой системе. Зафиксировать общее количество выданных каждым поисковиком ответов.
5. Оптимизировать запрос так, чтобы количество выдаваемой поисковиком информации не превышало 100 пунктов. Зафиксировать итоговую форму запроса в отчете.
6. Из полученных списков результатов выбрать следующую информацию:
 - общее количество найденных документов (Δ);
 - количество релевантных документов различной ценности (РД).
 При определении количества релевантных документов должно быть использовано понятие «когнитивная релевантность», или «перти-

- нентность». Количество релевантных документов оценивается при просмотре текста первых 10 найденных документов. При этом определяется ценность найденной информации (степень удовлетворения информационных потребностей пользователя найденным документом). Ценность информации определяется по 3-балльной шкале: 2 — имеет ценность, 1 — имеет частичную ценность, 0 — не имеет ценности. Результаты выполнения запросов свести в табл. 1.2.
7. Выполнить первичную обработку результатов, сведя их в табл. 1.3:
- Вычислить средние арифметические значения показателей \bar{D} , $\bar{P}D(2)$, $\bar{P}D(1)$ и $\bar{P}D(0)$ для каждой поисковой системы.
 - Определить место каждой поисковой системы по критерию «Полнота поиска». Для определения места использовать среднее количество найденных документов \bar{D} . Система, которая нашла больше документов, считается лучшей. Самой лучшей системе присваивается место 1, самой худшей — место N (N — количество исследуемых систем).
 - Определить коэффициент точности поиска P для каждой системы по следующей формуле:
- $$P = \frac{a}{a + b},$$
- где a — число релевантных документов, выданных поисковой системой в ответ на запрос, $a = \bar{P}D(2) + 0.5 * \bar{P}D(1)$; b — число нерелевантных документов, выданных поисковой системой в ответ на запрос, $b = \bar{P}D(0)$.
- Определить место каждой поисковой системы по критерию «Точность поиска». Система, имеющая большее значение коэффициента точности поиска P , считается лучшей. Самой лучшей системе присваивается место 1, самой худшей — место N (N — количество исследуемых систем).
 - Вычислить коэффициент поискового шума S по следующей формуле: $S = 1 - P$.
8. Вычислить сумму мест каждой исследуемой системы R по следующей формуле:
- $$R = \sum_{i=1}^m w_i(N - q_i + 1),$$
- где m — количество критериев оценки, i — номер критерия оценки поисковой системы, w_i — весовой коэффициент для критерия оценки i , q_i — место поисковой системы по критерию оценки i . N — количество исследуемых систем.
9. Определить лучшую поисковую систему из исследуемых.

Таблица 1.2

Результаты выполнения запросов

Тема	google		nigma		arxiv.org		yandex	
	Д	РД	Д	РД	Д	РД	Д	РД
Запрос 1: «...»	2	1	0	2	1	0	2	1
Запрос 2: «...»								
Запрос 3: «...»								

Таблица 1.3

Результаты сравнительного анализа поисковых систем Интернет

Критерий	google	nigma	arxiv.org	yandex
Полнота поиска ($\text{вес} = \dots$) — среднее количество найденных документов (Д)				
Место (полнота поиска)				
Среднее количество пертинентных документов (РД2)				
Среднее количество частично пертинентных док. (РД1)				
Среднее количество непертинентных документов (РД0)				
Коэффициент точности поиска Р (вес - ...)				
Место (точность поиска)				
Коэффициент поискового шума (S)				
Сумма мест (R)				

1.4. Оформление результатов

В качестве предварительной информации привести заполненную табл. 1.2.

Основные результаты работы оформить в виде табл. 1.3.

Сформулировать выводы, проанализировав полученные таблицы.

Лабораторная работа № 2. Установка и конфигурация операционной системы на виртуальную машину

2.1. Цель работы

Приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

2.2. Информация, необходимая для начала работы

Лабораторная работа подразумевает установку на виртуальную машину VirtualBox (<https://www.virtualbox.org/>) операционной системы Linux, дистрибутив Fedora (<http://fedoraproject.org/ru/>).

Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук, так и дома. Описание выполнения работы приведено для дисплейного класса:

- Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 8GB свободного места на жёстком диске;
- ОС Linux Gentoo (<http://www.gentoo.ru/>);
- VirtualBox верс. 4.1.6 или старше.

2.3. Последовательность выполнения работы

Загрузить в дисплейном классе операционную систему Linux. Осуществить вход в систему.

Запустить терминал. Перейти в каталог /var/tmp:

```
cd /var/tmp
```

Создать каталог с именем пользователя (желательно совпадающим с логином студента в дисплейном классе), например, avkorolkova:

```
mkdir avkorolkova
```

Перейти в общий каталог, где размещён образ виртуальной машины:

```
cd /afs/dk.sci.pfu.edu.ru/common/files
```

Скопировать образ Fedora-15-i686-Live-Desktop.iso виртуальной машины в созданный на предыдущем шаге каталог:

```
cp Fedora-15-i686-Live-Desktop.iso /var/tmp/avkorolkova
```

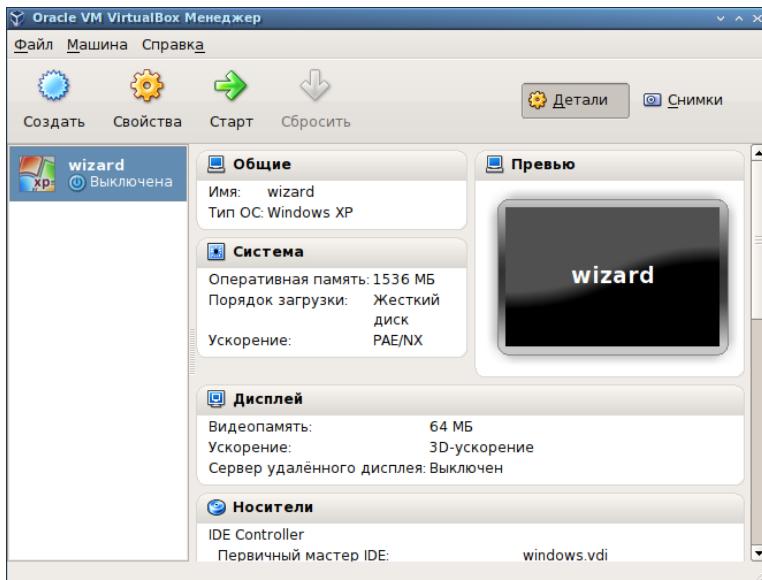


Рис. 2.1. Менеджер VirtualBox

Запустить виртуальную машину (рис. 2.1), введя `VirtualBox` в командной строке.

Проверить в свойствах VirtualBox месторасположение каталога для виртуальных машин. Для этого в VirtualBox выбрать «Файл», «Свойства», вкладка «Общие». В поле «Папка для машин» (рис. 2.2) должно стоять

`/var/tmp/имя_пользователя`

где `имя_пользователя` — логин (учётная запись) студента в дисплейном классе. Если указан другой каталог, то **изменить его**, как указано выше.

Создать новую виртуальную машину. Для этого в VirtualBox выбрать «Машина», «Создать» (рис. 2.3).

Указать имя виртуальной машины — `Base`, тип операционной системы — `Linux, Fedora` (рис. 2.4). Указать размер основной памяти виртуальной машины — `768 МБ` (рис. 2.5).

Задать конфигурацию жёсткого диска — загрузочный, `VDI (VirtualBox Disk Image)`, динамический виртуальный диск (рис. 2.6–2.8).

Задать размер диска — `40 ГБ`, его расположение — в данном случае `/var/tmp/имя_пользователя/Base/Base.vdi` (рис. 2.9–2.11).

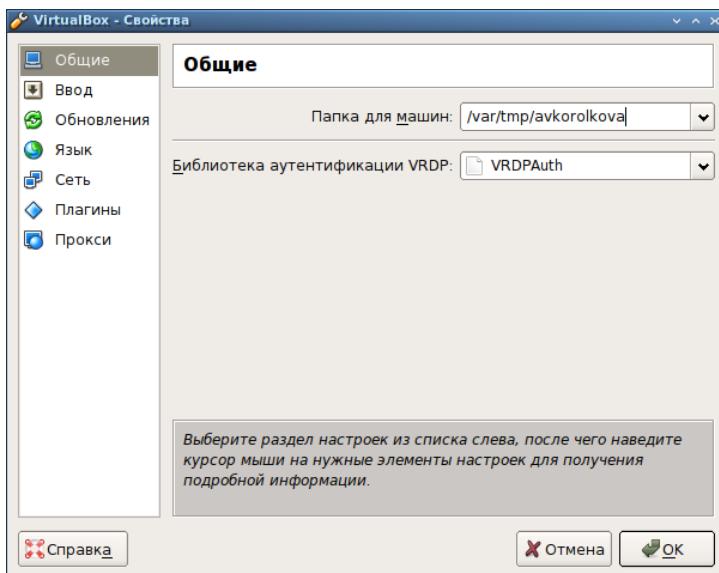


Рис. 2.2. Окно «Свойства» VirtualBox

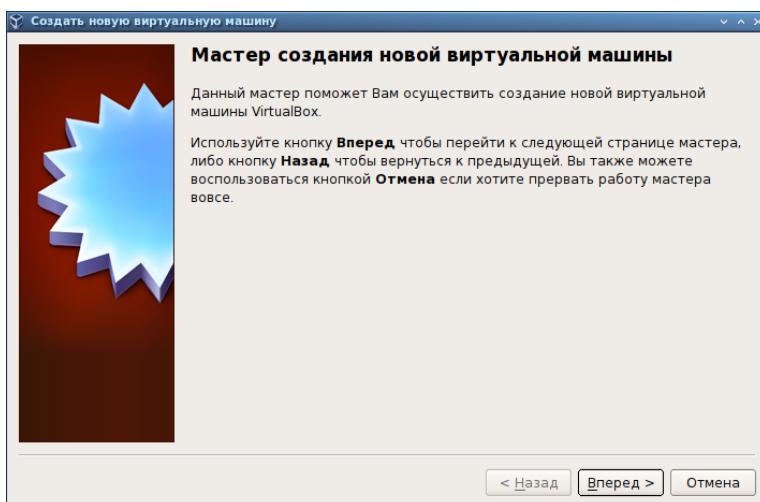


Рис. 2.3. Окно «Создать виртуальную машину»

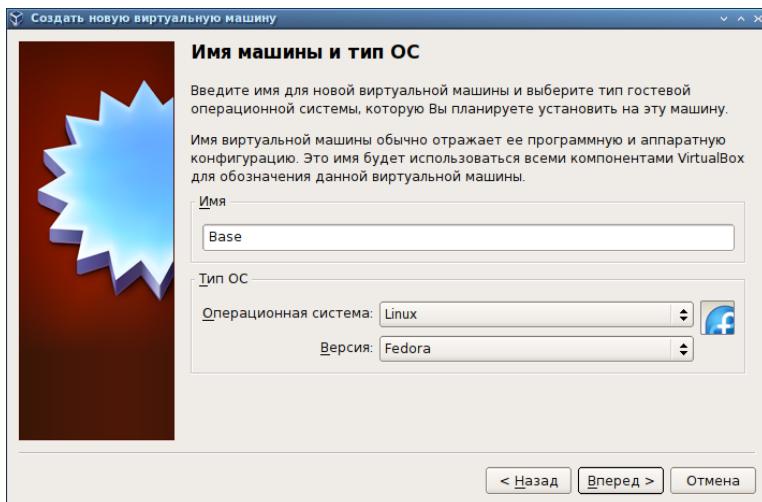


Рис. 2.4. Окно «Имя машины и тип ОС»

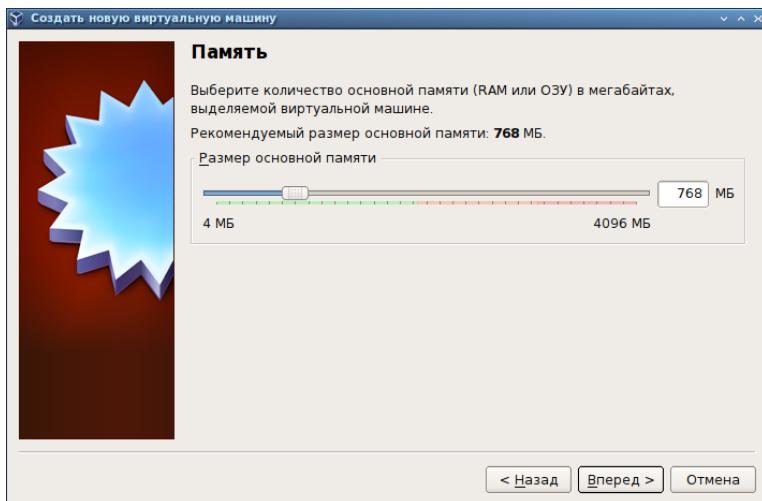


Рис. 2.5. Окно «Размер основной памяти»

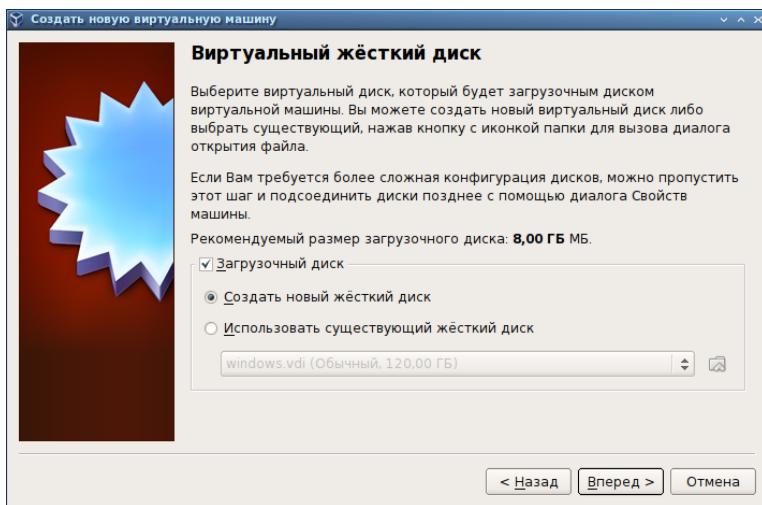


Рис. 2.6. Окно «Виртуальный жёсткий диск»

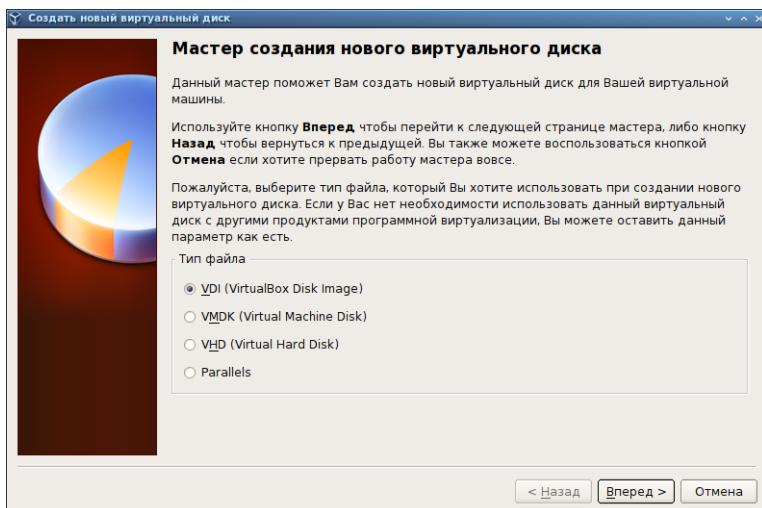


Рис. 2.7. Окно «Мастер создания нового виртуального диска»

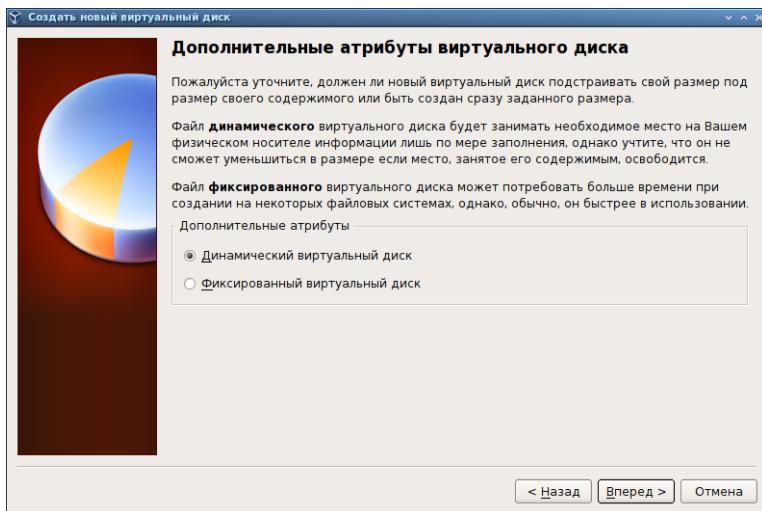


Рис. 2.8. Окно «Дополнительные атрибуты виртуального диска»

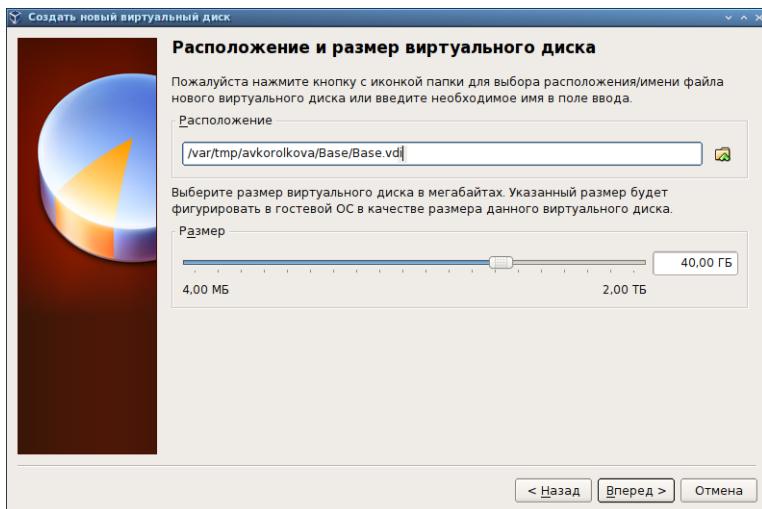


Рис. 2.9. Окно «Расположение и размер виртуального диска»

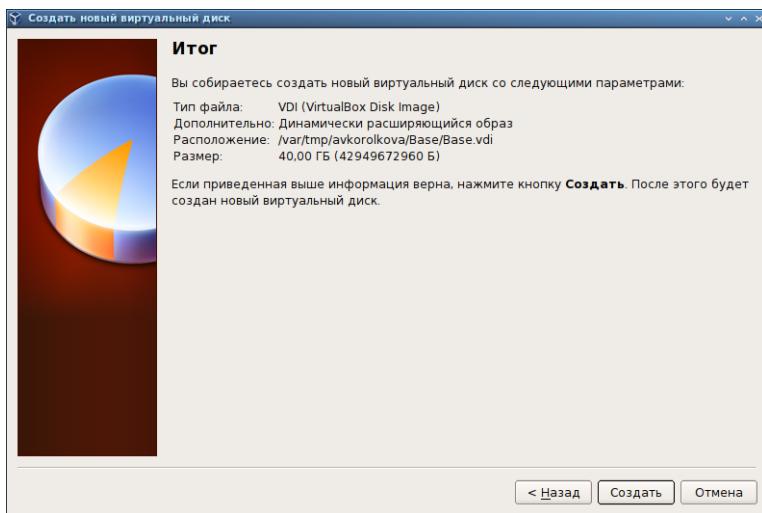


Рис. 2.10. Первое итоговое окно конфигурации виртуального диска

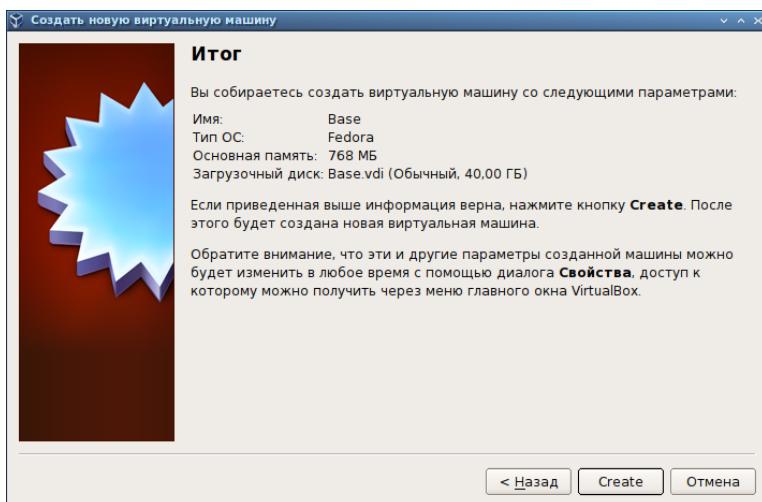


Рис. 2.11. Второе итоговое окно конфигурации виртуального диска

Выделить в окне менеджера VirtualBox виртуальную машину Base, и открыть окно «Свойства».

Проверить, что папка для снимков виртуальной машины Base имеет путь /var/tmp/имя_пользователя/Base/Snapshots. Для этого надо выбрать в VirtualBox «Свойства» виртуальной машины Base, «Общие», вкладка «Дополнительно» (рис. 2.12).

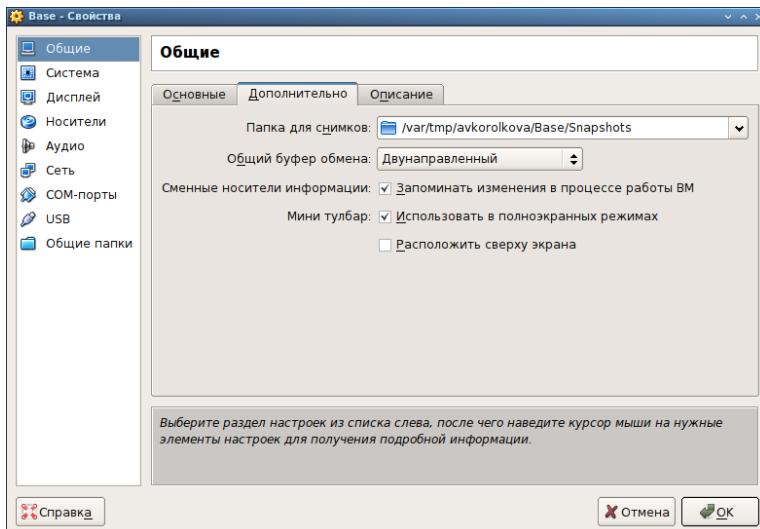


Рис. 2.12. Окно «Свойства» виртуальной машины Base

Выбрать в VirtualBox «Свойства», «Носители» виртуальной машины Base (рис. 2.13).

Добавить новый привод оптических дисков (рис. 2.14) и выбрать образ Fedora-15-i686-Live-Desktop.iso (рис. 2.15–2.16).

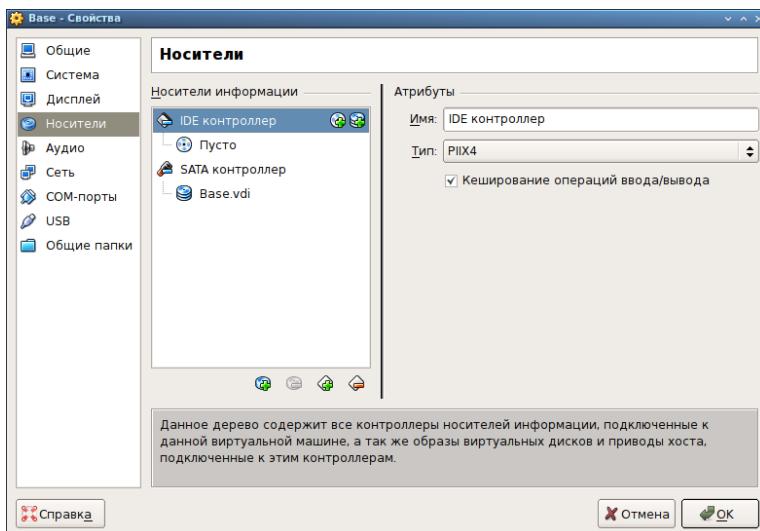


Рис. 2.13. Окно «Носители» виртуальной машины Base

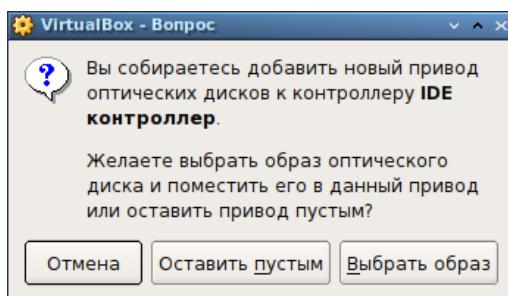


Рис. 2.14. Окно «Добавить новый привод оптических дисков» виртуальной машины Base

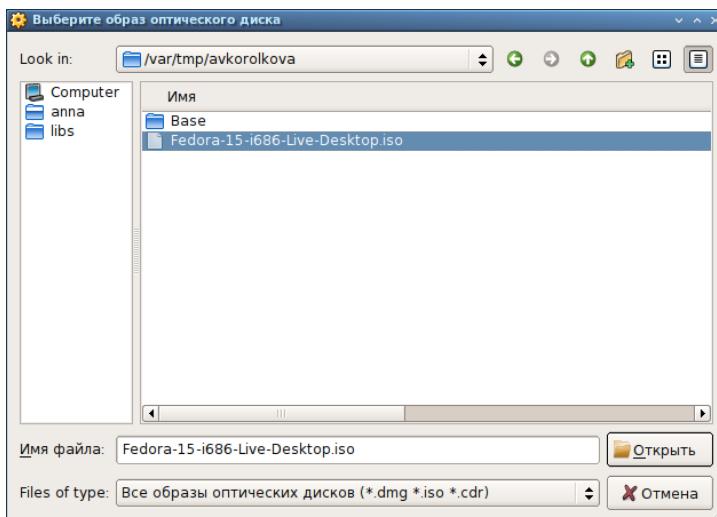


Рис. 2.15. Окно «Выбрать образ оптического диска» виртуальной машины Base

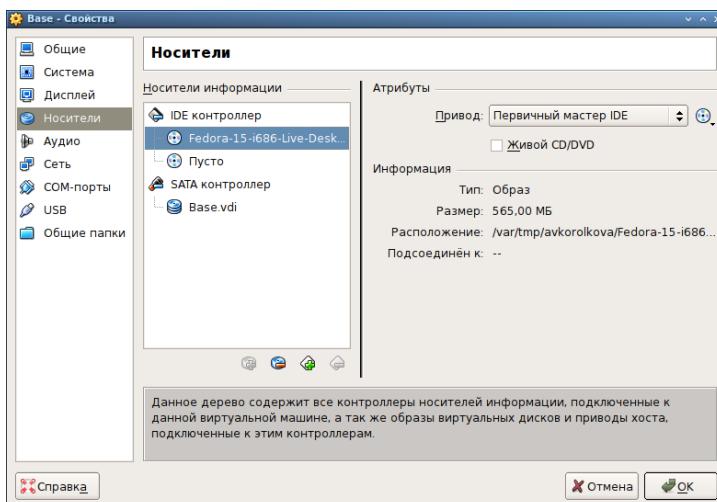


Рис. 2.16. Окно «Носители» виртуальной машины Base. Выбор образа оптического диска

Запустить виртуальную машину Base, выбрать «Applications», «System Tools», «Install to Hard Drive» (рис. 2.17).

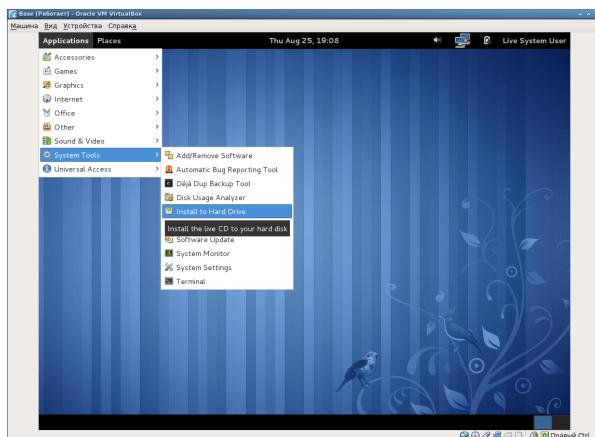


Рис. 2.17. Установка образа на жёсткий диск

Установить русский язык (рис. 2.18).

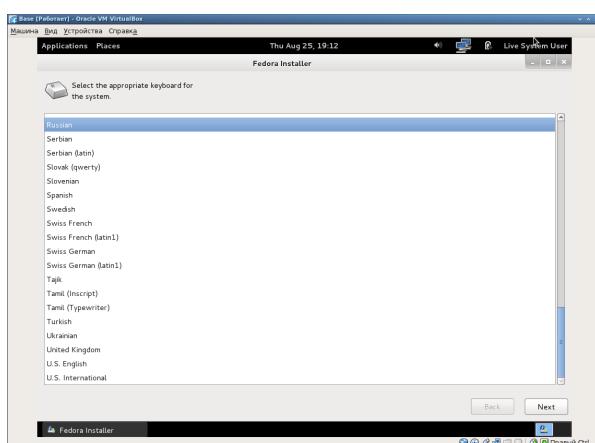


Рис. 2.18. Виртуальная машина Base. Установка русского языка

Указать «Basic Storage Devices» (рис. 2.19). В окне конфигурации жёсткого диска выбрать «Yes, discard any data» (рис. 2.20).

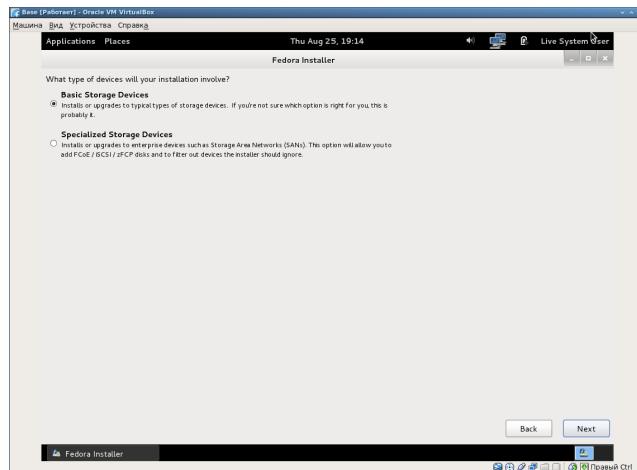


Рис. 2.19. Виртуальная машина Base

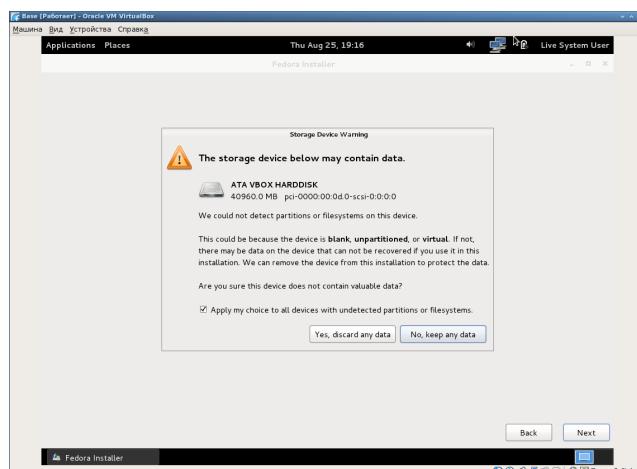


Рис. 2.20. Конфигурация жёсткого диска

В качестве имени машины указать «имя_пользователя.localdomain» (рис. 2.21). Указать часовой пояс «Москва» (рис. 2.22).

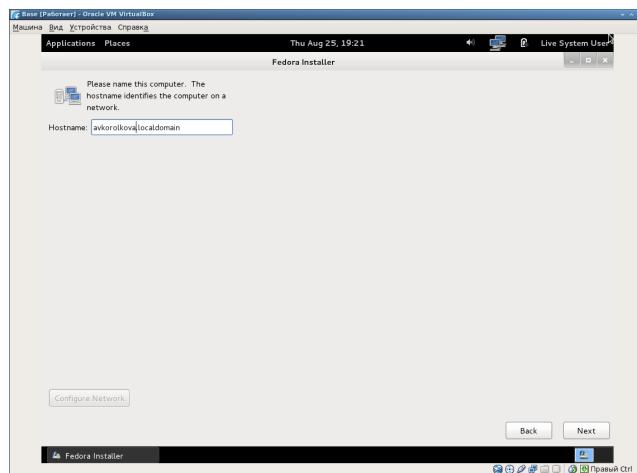


Рис. 2.21. Задать сетевое имя виртуальной машины

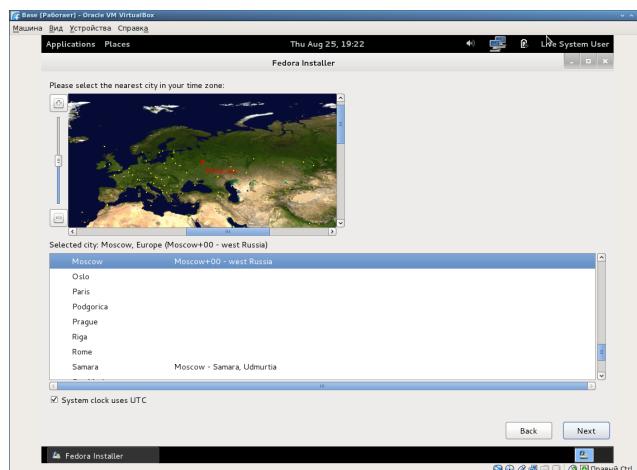


Рис. 2.22. Указать часовой пояс «Москва»

Установить пароль для root (рис. 2.23).

При конфигурировании размера жёсткого диска указать «Использовать всё свободное место» и выбрать «Review and modify partitions layout» (рис. 2.24).

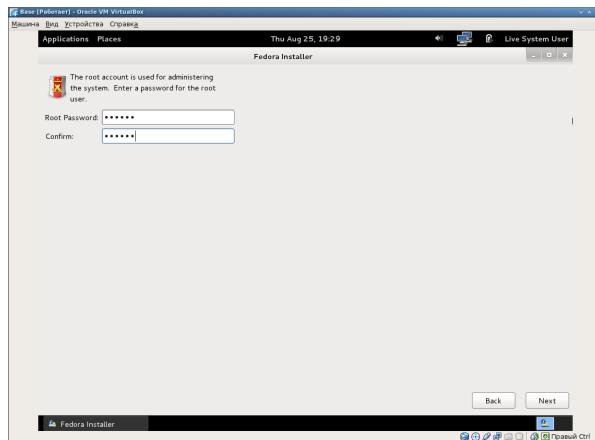


Рис. 2.23. Установка пароля для root

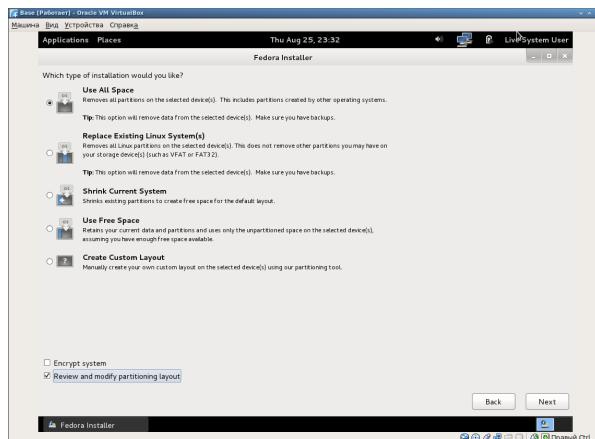


Рис. 2.24. Конфигурирование размера жёсткого диска

Проверить конфигурацию виртуальной машины (рис. 2.25).

Скопировать образ на жёсткий диск (рис. 2.26).

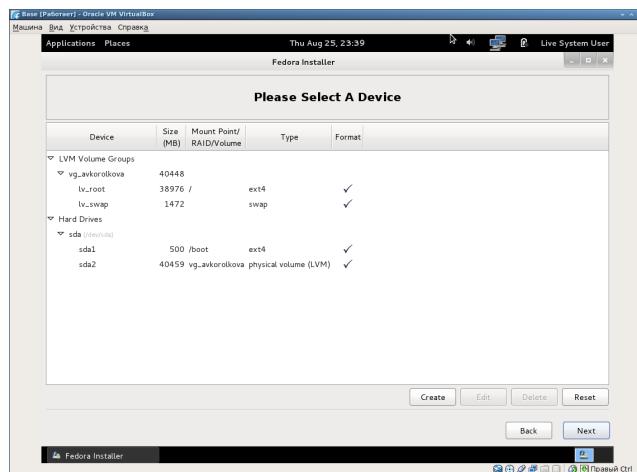


Рис. 2.25. Виртуальная машина Base. Проверка конфигурации

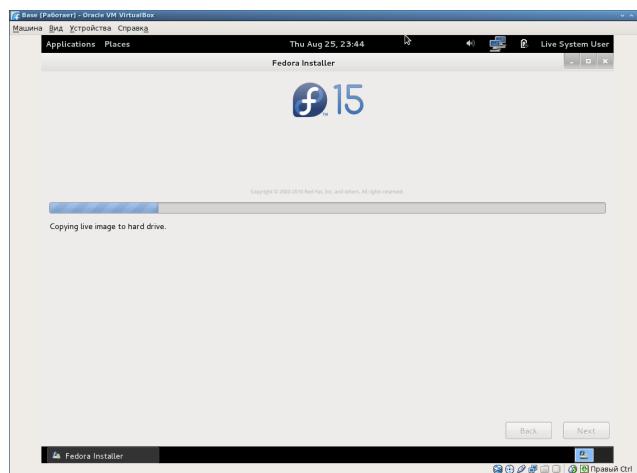


Рис. 2.26. Копирование образа на жёсткий диск

Завершить установку виртуальной машины (рис. 2.27).

Остановить виртуальную машину (рис. 2.28).

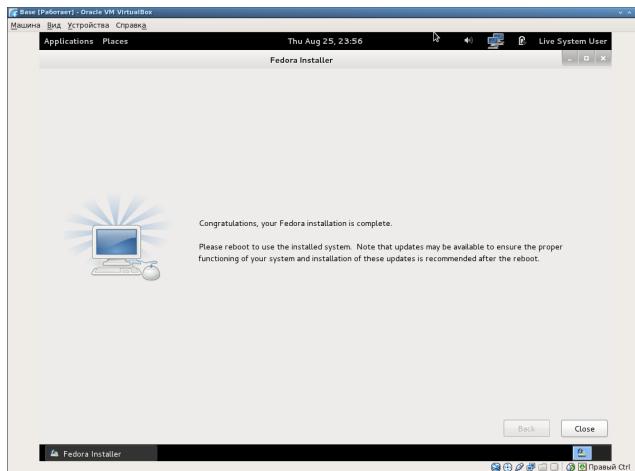


Рис. 2.27. Виртуальная машина Base. Завершение установки

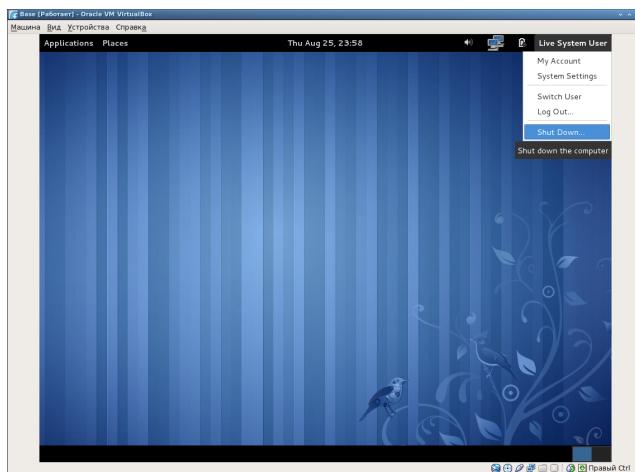


Рис. 2.28. Остановка виртуальной машины Base

В VirtualBox для машины Base отключить носитель информации с образом, выбрав «Свойства», «Носители», «Fedora-15-...iso», «Удалить устройство» (рис. 2.29).

Запустить виртуальную машину Base и настроить её (рис. 2.30–

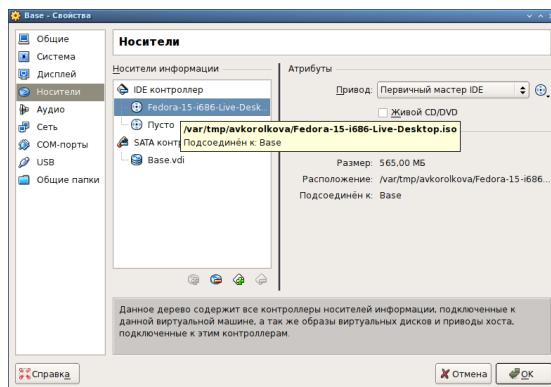


Рис. 2.29. Отключение носителя информации в VirtualBox

2.32).

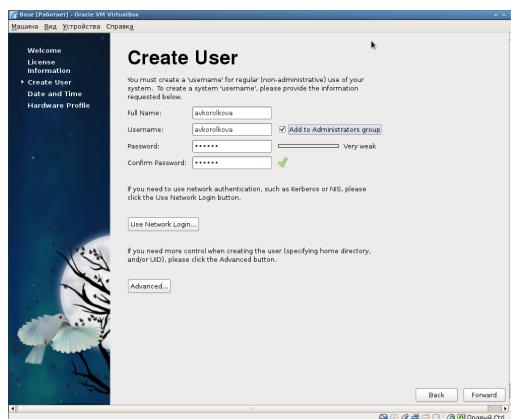


Рис. 2.30. Настройка виртуальной машины: учётная запись, далее «Add to Administrators group»

Подключиться к виртуальной машине с помощью созданной учётной записи (рис. 2.33).

На виртуальной машине Base запустить терминал, перейти под учетную запись `root` с помощью команды `su`.

Перейти в каталог `/etc`:
`cd /etc`

С помощью редактора `vi` в файле `/etc/yum.conf` прописать адрес и порт прокси-сервера. Для этого выполнить следующие действия:

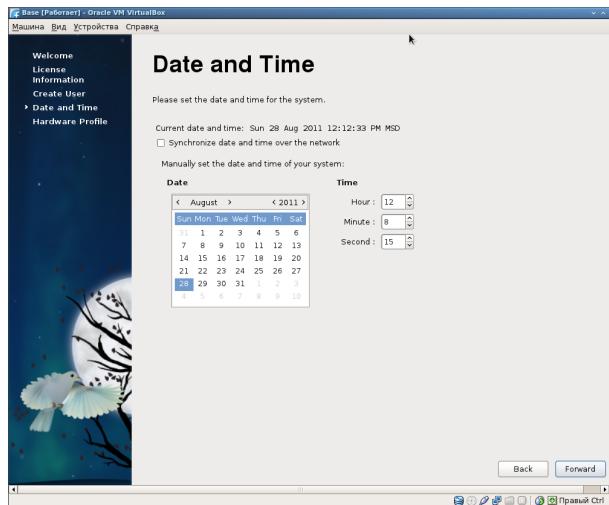


Рис. 2.31. Настройка виртуальной машины: дата и время



Рис. 2.32. Настройка виртуальной машины: профиль устройств

```
vi yum.conf
```

в открывшемся файле, используя команды редактора vi (см. man vi

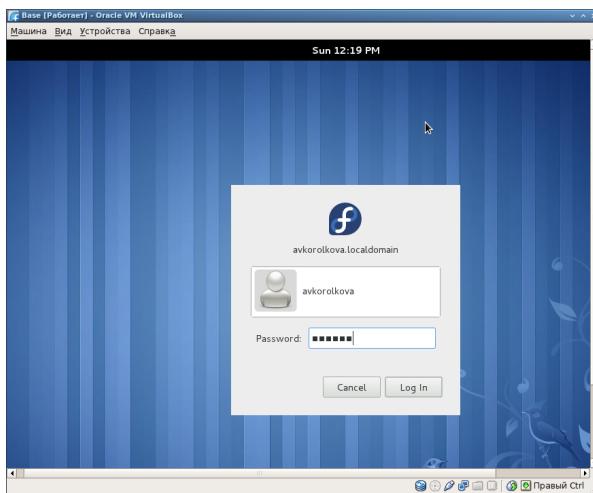


Рис. 2.33. Подключение к виртуальной машине

или, например, <http://ru.wikipedia.org/wiki/Vi>, <http://lib.ru/unixhelp/vi.txt>), прописать строку:

```
proxy=http://proxy.dk.sci.pfu.edu.ru:3128
```

Для тех, кто работает не в дисплейном классе, или ничего не надо прописывать, или нужно прописать адрес прокси-сервера своей сети.

С помощью команды `yum update` обновить системные файлы и установить необходимые программы, например,

```
yum update
yum install mc bind bind-utils httpd php
```

После установки необходимых программ можно завершить работу виртуальной машины. Её конфигурация сохранится на жёстком диске в директории `/var/tmp/avkorolkova/Base`.

Для того чтобы другие виртуальные машины могли использовать машину Base и её конфигурацию как базовую, необходимо произвести следующие действия. В VirtualBox в меню выбрать «Файл», «Менеджер виртуальных носителей», «Жёсткие диски» и, выделив «Base.vdi», указать «Освободить» (рис. 2.34).

Затем в консоли перейти в директорию с конфигурацией виртуальной машины (`/var/tmp/avkorolkova/Base`) и набрать следующую команду:

```
VBoxManage modifyhd --type multiattach Base.vdi
```

Теперь на основе виртуальной машины Base можно создать машину Host2, выбрав в VirtualBox «Машина», «Создать» и в «Мастере

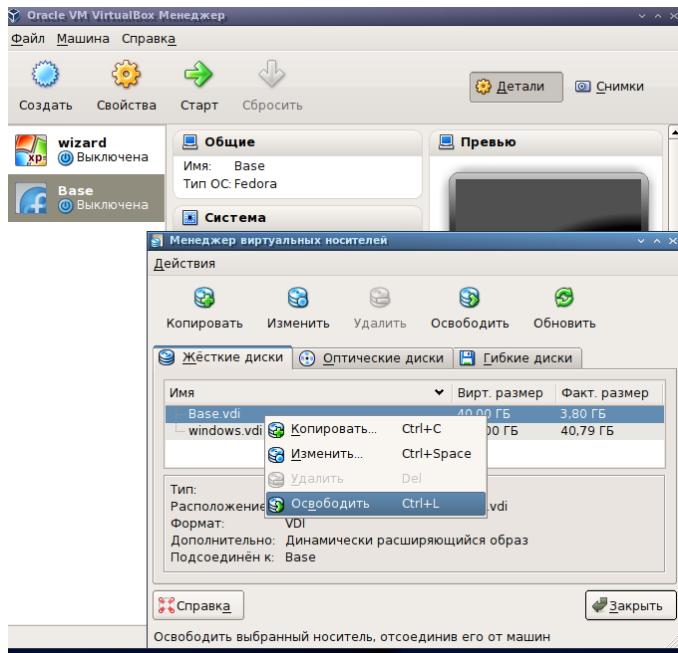


Рис. 2.34. Множественный доступ к виртуальной машине

создания новой виртуальной машины» указав в качестве имени машины Host2, в качестве типа операционной системы — Linux, версия «Fedora», а при конфигурации виртуального жёсткого диска выбрав «Использовать существующий жёсткий диск» Base.vdi.

Лабораторная работа № 3. Настройка DNS-сервера

3.1. Цель работы

Приобретение практических навыков по установке и конфигурированию DNS-сервера, усвоение студентами принципов работы системы доменных имён.

3.2. Информация, необходимая для начала работы

Перед началом выполнения работы:

- должна быть выполнена лабораторная работа по установке и конфигурированию операционной системы на виртуальной машине Base;
- на виртуальной машине Base должны быть установлены `bind` и `bind-utils`;
- должен быть установлен множественный доступ к виртуальной машине Base;
- необходимо ознакомиться с лекционным материалом по DNS, про читать документацию к пакету `bind` и утилитам `ifconfig`, `ping`, `dig`, `host`.

3.3. Основные понятия DNS

Система доменных имён (Domain Name System, DNS) — распределённая система (распределённая база данных), ставящая в соответствие доменному имени хоста (компьютера или другого сетевого устройства) IP-адрес и наоборот.

DNS-сервер — специализированное программное обеспечение для обслуживания DNS.

DNS-клиент — специализированная библиотека (или программа) для работы с DNS.

В качестве серверов доменных имён чаще всего используются различные версии BIND (Berkeley Internet Name Domain), <http://www.isc.org/software/bind>.

Зона — логический узел в дереве имён.

Домен — название зоны в системе доменных имён Интернета. Структура доменного имени отражает порядок следования зон в иерархическом виде.

Поддомен (subdomain) — имя подчинённой зоны.

Спецификация DNS определяет **два типа DNS-серверов**:

- *первичный мастер-сервер (primary master)* — производит загрузку данных для зоны из файла на машине-сервере;
- *дополнительный, или вторичный, мастер-сервер (secondary master)* — получает данные зоны от другого DNS-сервера, называемого его *мастером (master server)*.

Файлы данных зоны — файлы, из которых первичные DNS-серверы производят чтение зональных данных. Вторичные DNS-серверы также могут загружать зональные данные из файлов.

Директивы управления:

- директива **\$ORIGIN** определяет текущее имя домена;
- директива **\$INCLUDE** используется для того, чтобы в файл описания зоны можно было включить содержание другого файла.

Формат записи:

```
[<comment>]
$ORIGIN [<comment>]
$INCLUDE [] [<comment>]
```

В квадратные скобки [] заключены необязательные параметры, а в угловые скобки < > — сущности.

RR-записи описывают все узлы сети в зоне и помечают делегирование поддоменов. **Типы записи описания ресурсов:**

- SOA-запись — указывает на авторитетность для зоны
- NS-запись — перечисляет DNS-серверы зоны
- A — отображение имён узлов в адреса
- PTR — отображение адресов в имена узлов
- CNAME — каноническое имя (для псевдонимов)
- MX — отображение имён почтовых серверов

Формат записи SOA:

```
[zone] [ttl] IN SOA origin contact (
    serial refresh retry expire minimum)
```

- *zone* — имя зоны;
- *ttl* — время кэширования (в SOA всегда пустое, определяется директивой управления **\$TTL**);
- *IN* — класс данных Internet;
- *origin* — доменное имя primary master сервера зоны;
- *contact* — почтовый адрес лица, осуществляющего администрирование зоны (так как символ @ имеет особый смысл при описании зоны, то вместо него в почтовом адресе используется символ «.»);
- *serial* — серийный номер файла зоны в нотации ГГГГММДДВВ (учёт изменений файла описания зоны);
- *refresh* — интервал времени, после которого slave-сервер обязан обратиться к master-серверу с запросом на верификацию своего описания зоны;

- *retry* — интервал времени, после которого slave-сервер должен повторить попытку синхронизировать описание зоны с master сервером;
- *expire* — интервал времени, после которого slave-сервер должен прекратить обслуживание запросов к зоне, если он не смог в течение этого времени верифицировать описание зоны, используя информацию с master сервера;
- *minTTL* — время негативного кэширования (negative caching), т.е. время кэширования ответов, которые утверждают, что установить соответствие между доменным именем и IP-адресом нельзя.

Формат записи NS:

[*domain*] [*ttl*] IN NS [*server*]

где *domain* — имя домена, для которого сервер, указанный последним аргументом записи NS, поддерживает описание зоны; *server* — доменное имя сервера.

Формат адресной записи:

[*host*] [*ttl*] IN A [*address*]

где *host* — доменное имя хоста; *address* — IP-адрес машины.

Формат PTR-записи имеет следующий вид:

[*name*] [*ttl*] IN PTR [*host*]

где *name* — номер (не реальный IP-адрес машины, а имя в специальном домене in-addr.arpa или в одной из его зон); *host* — доменное имя хоста.

Формат MX-записи:

[*name*] [*ttl*] IN MX [*preference*] [*host*]

где *name* — имя машины или домена, на который может отправляться почта; *preference* — приоритет почтового сервера, имя которого (поле *host*) указано последним аргументом в поле данных MX-записи.

Формат записи CNAME:

[*nickname*] [*ttl*] IN CNAME [*host*]

где поле *nickname* определяет синоним для канонического имени, которое задается в поле *host*.

3.4. Сетевые утилиты

3.4.1. Утилита ping

Утилита ping предназначена для проверки соединений в сетях на основе TCP/IP.

Утилита отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT, Round Trip Time) позволяет определять двусторонние задержки (RTT) по маршруту и частоту потери пакетов, т.е. косвенно определять загруженность на каналах передачи данных и промежуточных устройствах.

Полное отсутствие ICMP-ответов может также означать, что удалённый узел (или какой-либо из промежуточных маршрутизаторов) блокирует ICMP Echo-Reply или игнорирует ICMP Echo-Request.

3.4.2. Команда ifconfig

Команда `ifconfig` используется для конфигурирования и диагностики сетевых интерфейсов операционной системы.

Формат команды ifconfig:

```
ifconfig interface [address [parameters]]
```

где `interface` — имя интерфейса, `address` — IP-адрес, который требуется назначить интерфейсу. Это может быть IP-адрес или имя, которое `ifconfig` будет искать в файле `/etc/hosts`.

Если `ifconfig` используется только с именем интерфейса, он показывает конфигурацию этого интерфейса. Когда `ifconfig` вызывается без параметров, он показывает все интерфейсы, которые конфигурированы в системе; опция `-a` вынуждает показать бездействующие интерфейсы.

3.4.3. Утилита dig

Утилита `dig` (domain information groper) предоставляет пользователю интерфейс командной строки для обращения к системе DNS, позволяет формировать запросы о доменах DNS-серверам. Утилита `dig` входит в стандартный комплект DNS сервера BIND.

3.4.4. Утилита host

Утилита `host` производит простейшее преобразование имени с помощью DNS.

Формат команды host:

```
host [-l] [-v] [-w] [-r] [-d] [-t querytype]
      [-a] host [server]
```

Здесь `-l` — выводит полную информацию о домене, `-v` — использует подробный формат при выводе результатов, `-w` — заставляет команду `host` ожидать ответа, `-r` — выключает режим рекурсии, `-d` — включает режим отладки, `-t querytype` — определяет тип запроса, `-a` — восстанавливает все записи в DNS.

Формат команды dig:

```
dig [@server] domain [query-type] [query-class]
      [+query-option] [-dig-option] [%comment]
```

Здесь `server` — имя DNS-сервера. В качестве имени можно указать как имя хоста, так и его IP-адрес.

Параметр `query-type` — тип исходной записи, который можно указать в запросе (A, SOA, NS и MX). Для получения всей информации о домене можно указать `query-type anu`.

Параметр `query-class` — класс сетевой информации, который также можно указывать в запросе. По умолчанию этот параметр всегда будет IN для сети Internet.

Параметр `+query-option` используется для изменения значения параметра в пакете DNS или для изменения формата вывода результатов работы `dig`.

Более подробную информацию по работе с утилитой `dig` можно найти в руководстве `man`.

3.5. Последовательность выполнения работы

Загрузить в дисплейном классе операционную систему Linux. Осуществить вход в систему. Запустить терминал. Запустить VirtualBox (рис. 3.1), введя в командной строке `VirtualBox`.

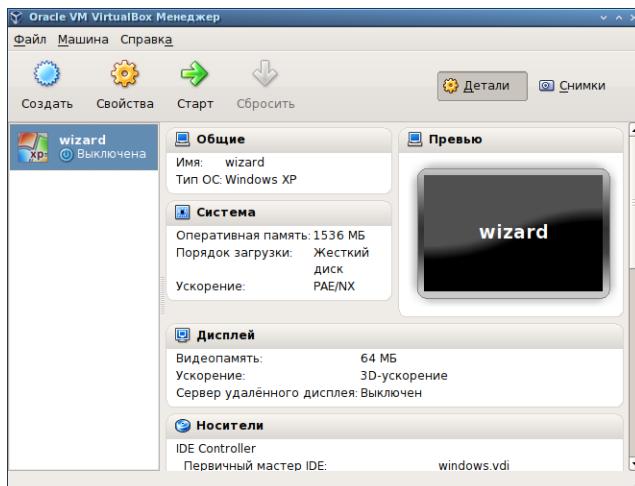


Рис. 3.1. Менеджер VirtualBox

Проверить в свойствах VirtualBox месторасположение каталога для виртуальных машин. Для этого в VirtualBox выбрать «Файл», «Свойства», вкладка «Общие». В поле «Папка для машин» (рис. 3.2) должно стоять `/var/tmp/имя_пользователя`, где имя_пользователя — логин студента в дисплейном классе. Если указан другой каталог, то изменить его, как указано выше.

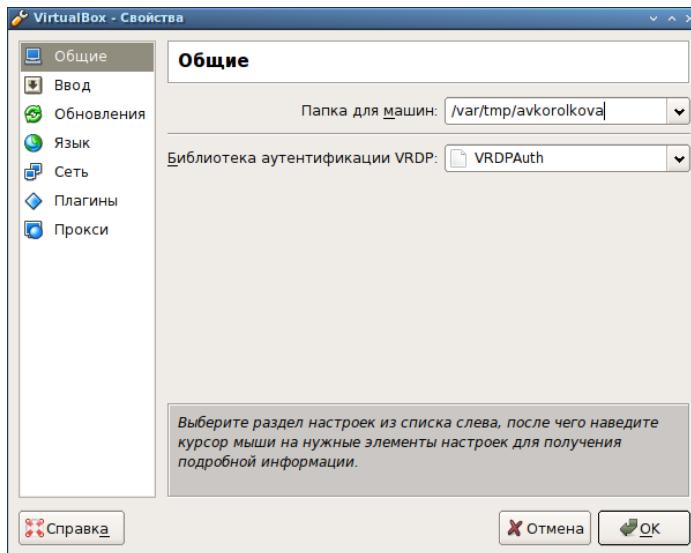


Рис. 3.2. Окно «Свойства» VirtualBox

Создать виртуальную машину Server с операционной системой Linux, дистрибутив Fedora (рис. 3.3).

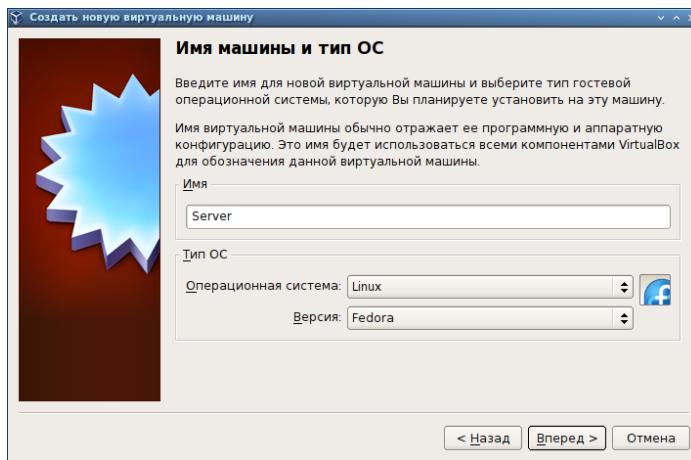


Рис. 3.3. Создание в VirtualBox виртуальной машины Server

Указать размер основной памяти виртуальной машины (рис. 3.4).

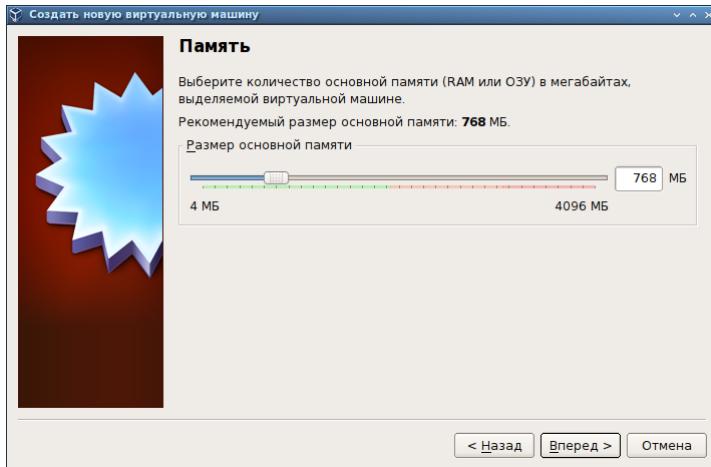


Рис. 3.4. Окно «Размер основной памяти» виртуальной машины Server

Задать размер и конфигурацию жёсткого диска (рис. 3.5, 3.6), выбрав «Использовать существующий жесткий диск» Base.vdi.

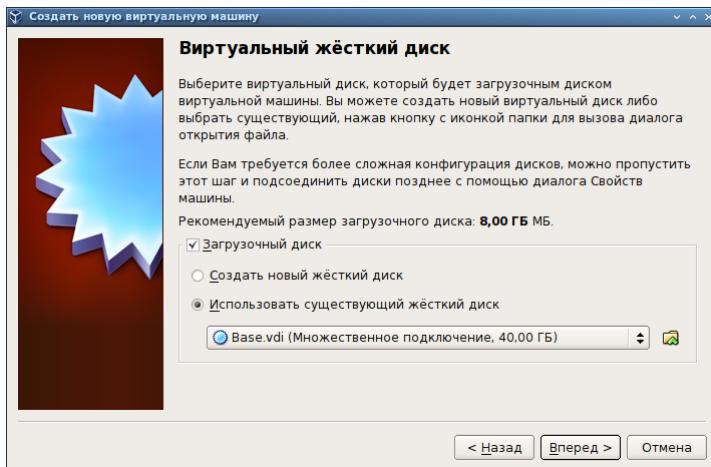


Рис. 3.5. Окно «Виртуальный жёсткий диск» виртуальной машины Server

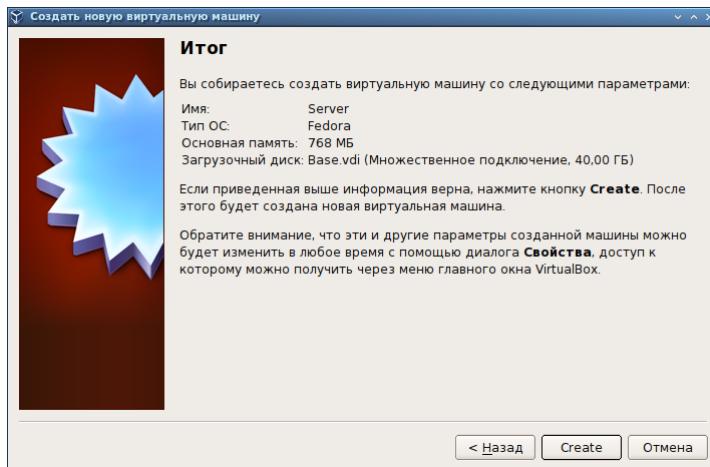


Рис. 3.6. Итоговое окно конфигурации виртуальной машины Server

В свойствах виртуальной машины Server в качестве типа сетевого подключения («Свойства», «Сеть») задать тип подключения «Внутренняя сеть» (рис. 3.7).

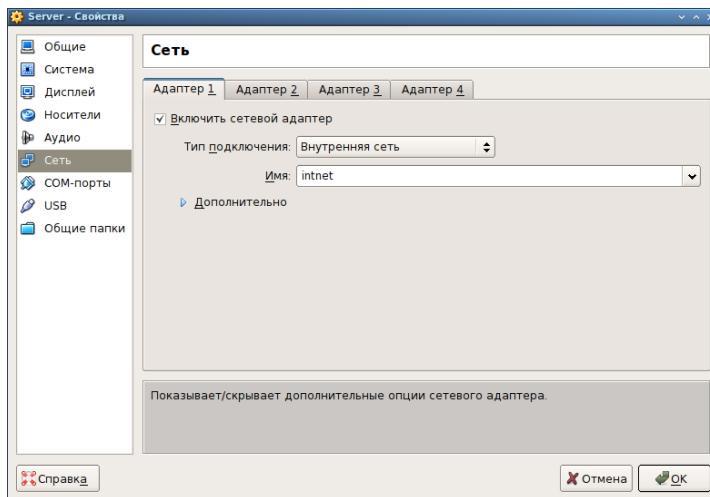


Рис. 3.7. Тип сетевого подключения виртуальной машины Server

Запустить виртуальную машину Server. Открыть «Сетевые соединения», выбрав при нажатии правой кнопки мыши «Изменить соединение...» (рис. 3.8).

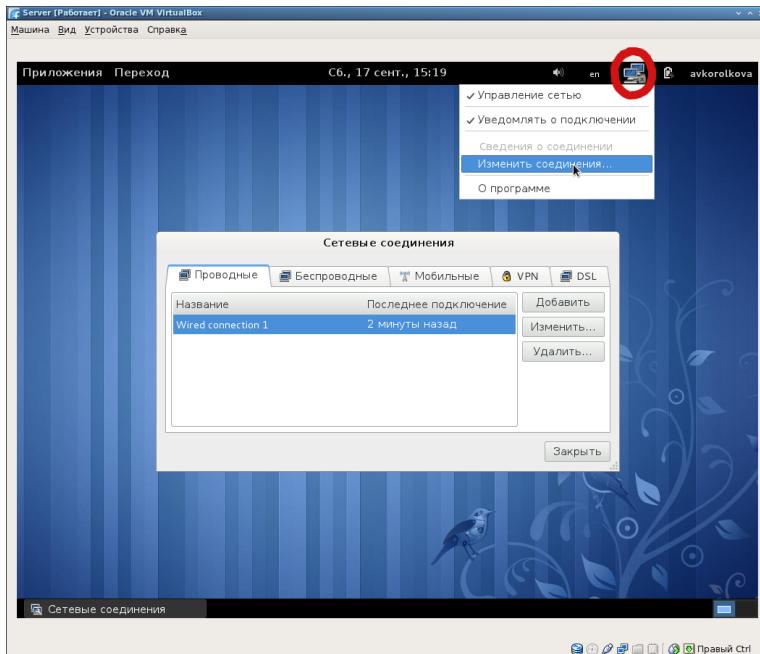


Рис. 3.8. Изменение сетевого подключения виртуальной машины Server

Выбрав вкладку «Параметры IPv4», изменить соединение «Wired connection 1», следующим образом (рис. 3.9):

- способ настройки: указать «Вручную»;
 - в поле «Адреса» добавить:
 - адрес: 192.168.0.100
 - маска: 255.255.255.0
 - шлюз: 192.168.0.100
 - серверы DNS: 192.168.0.100
 - домены поиска: telesys-stud.net
- Поскольку сама виртуальная машина Server является DNS-сервером, то адреса в полях «Адрес», «Шлюз», «Серверы DNS» совпадают.
- Проверить текущее сетевое соединение, выбрав при нажатии правой кнопки мыши на менеджере сетевых соединений «Сведения о соединении» (рис. 3.10).

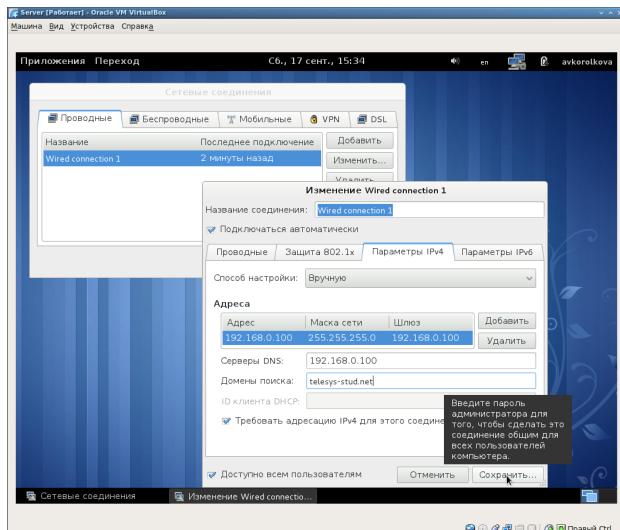


Рис. 3.9. Изменение «Wired connection 1» виртуальной машины Server

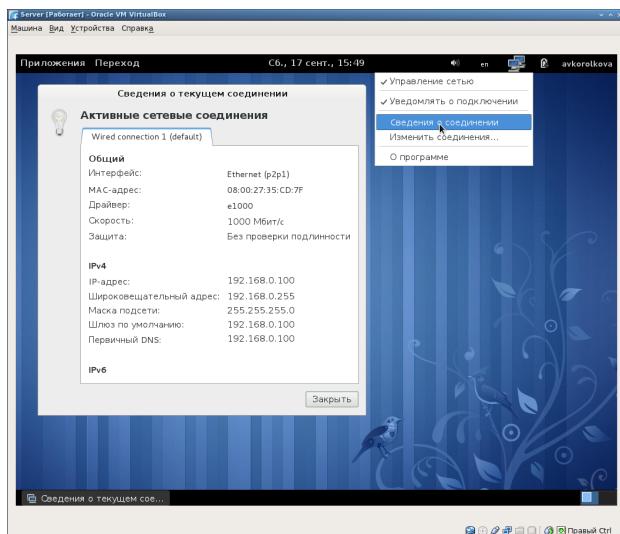


Рис. 3.10. Сведения о сетевом соединении виртуальной машины Server

Запустить терминал. Перейти с помощью команды `su` в привилегированный режим. Для удобства дальнейшей работы запустить файловый менеджер Midnight Commander с помощью команды `mc`. Справку по работе с `mc` можно найти в руководстве `man` или на сайте <https://www.midnight-commander.org/wiki/ru/doc>.

Перейти в каталог `/etc/sysconfig`:

```
cd /etc/sysconfig
```

В файле `network` задать имя хоста в домене `telesys-stud.net` (рис. 3.11):

```
HOSTNAME=server.telesys-stud.net
```

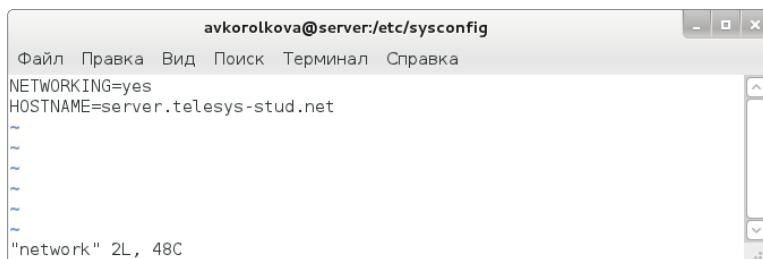


Рис. 3.11. Задать имя хоста в домене telesys-stud.net

Скопировать из `/etc` в `/etc/named` файл `named.rfc1912.zones`.

```
cp /etc/named.rfc1912.zones /etc/named/.
```

Переименовать его в `telesys-stud.net`:

```
cd /etc/named
mv named.rfc1912.zones telesys-stud.net
```

Отредактировать файл `/etc/named.conf` (рис. 3.12):

- в поле `options` указать сеть, хосты которой могут иметь доступ к DNS:

```
allow-query { localhost; 192.168.0.0/16; }
...
```

- в поле `options` закомментировать строки:

```
// dnssec-enable yes;
// dnssec-validation yes;
// dnssec-lookaside auto;
```

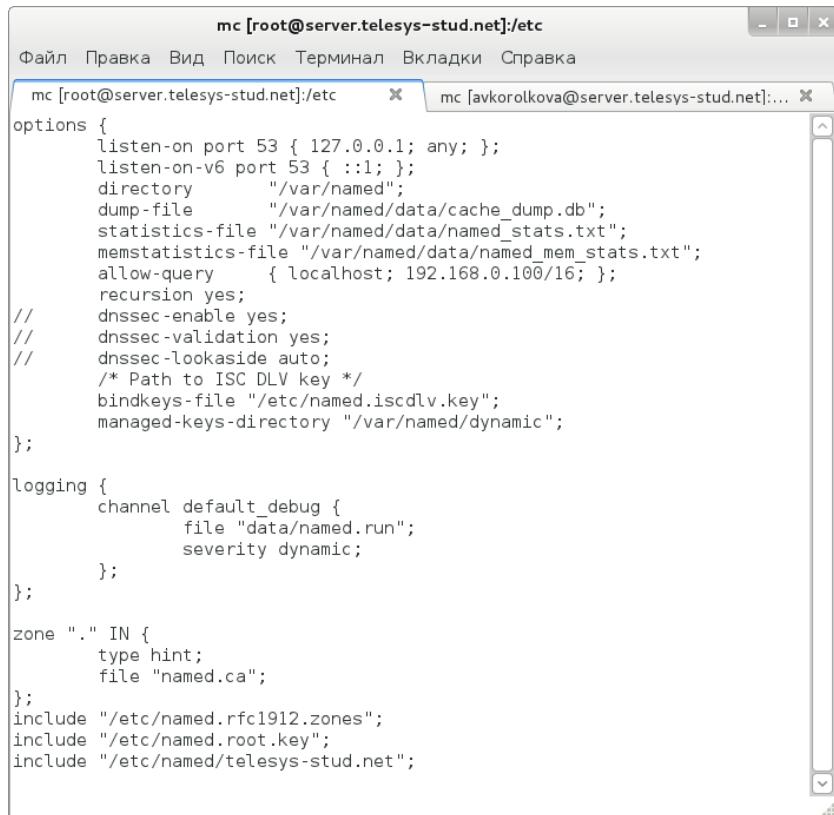
так как стандарт DNSSEC является экспериментальным и не будет использоваться;

- разрешить прослушивать порт 53 с любого интерфейса (любого хоста):

```
listen-on port 53 { 127.0.0.1; any; }
...
```

- прописать:

```
include "/etc/named/telesys-stud.net";
```



```

mc [root@server.telesys-stud.net]:/etc
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc × mc [avkorolkova@server.telesys-stud.net]:... ×
options {
    listen-on port 53 { 127.0.0.1; any; };
    listen-on-v6 port 53 { ::1; };
    directory      "/var/named";
    dump-file      "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query    { localhost; 192.168.0.100/16; };
    recursion yes;
// dnssec-enable yes;
// dnssec-validation yes;
// dnssec-lookaside auto;
/* Path to ISC DLV key */
bindkeys-file "/etc/named.iscdlv.key";
managed-keys-directory "/var/named/dynamic";
};

logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};

zone "." IN {
    type hint;
    file "named.ca";
};
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
include "/etc/named/telesys-stud.net";

```

Рис. 3.12. Разрешение доступа к DNS

Прописать в `/etc/named/telesys-stud.net` прямую и обратную мастер-зоны домена `telesys-stud.net` (рис. 3.13):

```

zone "telesys-stud.net" IN {
    type master;
    file "master/fz/telesys-stud.net";
    allow-update { none; };
};

zone "0.168.192.in-addr.arpa" IN {
    type master;
    file "master/rz/192.168.0";
    allow-update { none; };
};

```

```

mc [root@server.telesys-stud.net]:/etc/named
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc/named... mc [avkorolkova@server.telesys-stud.net]...
// named.rfc1912.zones:
//
// Provided by Red Hat caching-nameserver package
//
// ISC BIND named zone configuration for zones recommended by
// RFC 1912 section 4.1 : localhost TLDs and address zones
// and http://www.ietf.org/internet-drafts/draft-ietf-dnsop-default-local-
zones-02.txt
// (c)2007 R W Franks
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

zone "telesys-stud.net" IN {
    type master;
    file "master/fz/telesys-stud.net";
    allow-update { none; };
};

zone "0.168.192.in-addr.arpa" IN {
    type master;
    file "master/rz/192.168.0";
    allow-update { none; };
};

```

Рис. 3.13. Прямая и обратная мастер-зоны домена telesys-stud.net

Перейти в директорию `/var/named` и создать каталог `master`, а в нём — каталоги `fz` и `rz`:

```

cd /var/named
mkdir master
cd master
mkdir fz rz

```

Скопировать из `/var/named` в каталог `/var/named/master/fz` файл `named.localhost`, переименовать его в `telesys-stud.net`:

```

cp /var/named/named.localhost /var/named/master/fz/.
cd /var/named/master/fz
mv named.localhost telesys-stud.net

```

Изменить файл `/var/named/master/fz/telesys-stud.net`, указав необходимые DNS-записи для прямой зоны (рис. 3.14):

```
$TTL 1D
@ IN SOA @ server.telesys-stud.net. (
    2011101500 ; serial
    1D; refresh
    1H; retry
    1W; expire
    3H); minimum
NS      @
        A      192.168.0.100
$ORIGIN telesys-stud.net.
ns      A      192.168.0.100
```



Рис. 3.14. Конфигурация прямой мастер-зоны домена `telesys-stud.net`

В файле `/var/named/master/fz/telesys-stud.net`:

- запись SOA указывает, на каком сервере хранится эталонная информация о данном домене `telesys-stud.net`;
- символ `@` обозначает имя текущей зоны: `server.telesys-stud.net.` ; класс — `IN` (`Internet`), тип — `SOA`;
- сервер `server.telesys-stud.net` — основной сервер имен зоны;
- запись `NS` (`name server`) указывает на DNS-сервер для данного домена;

- запись A (address record) — запись адреса связывает хост с IP-адресом;
- TTL (время жизни) — определяет время в секундах, в течение которого запись ресурса этой зоны остается действительной в кэше других серверов;
- serial — серийный номер файла зоны, он должен увеличиваться каждый раз при внесении изменений в данные домена;
- refresh (обновление) — время в секундах, определяющее частоту проверок вторичным сервером серийного номера на первичном и запускающее новый обмен, если на первичном имеются новые данные;
- retry (повтор) — этим значением определяется время задержки в секундах между повторными попытками обновления сведений о зоне;
- expire (срок) — если повторные попытки обновления не привели к успеху в течение этого времени, вторичный сервер уничтожает свою копию данных файла(ов) зоны и прекращает отвечать на запросы для этого домена, что позволяет остановить повторение и циркуляцию очень старых и потенциально неточных данных.

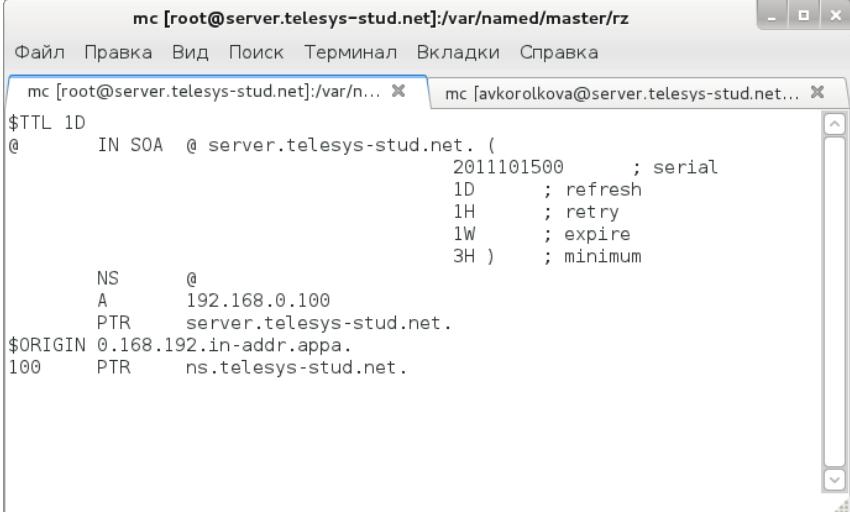
Скопировать из `/var/named` в каталог `/var/named/master/rz` файл `named.loopback`, переименовать его в `192.168.0`:

```
cp /var/named/named.loopback /var/named/master/rz/
cd /var/named/master/rz
mv named.loopback 192.168.0
```

Изменить файл `/var/named/master/rz/192.168.0`, указав необходимые DNS-записи для обратной зоны (рис. 3.15):

```
$TTL 1D
@ IN SOA @ server.telesys-stud.net. (
    2011101500 ; serial
    1D; refresh
    1H; retry
    1W; expire
    3H); minimum
NS      @
A       192.168.0.100
PTR    server.telesys-stud.net.
$ORIGIN 0.168.192.in-addr.arpa.
100    PTR ns.telesys-stud.net.
```

Здесь PTR (Pointer) — запись-указатель обратной зоны.



```

mc [root@server.telesys-stud.net]:/var/named/master/rz
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/var/named/master/rz... mc [avkorolkova@server.telesys-stud.net...]
$TTL 1D
@ IN SOA @ server.telesys-stud.net. (
                                2011101500      ; serial
                                1D              ; refresh
                                1H              ; retry
                                1W              ; expire
                                3H )            ; minimum
NS      @
A       192.168.0.100
PTR    server.telesys-stud.net.
$ORIGIN 0.168.192.in-addr.apa.
100    PTR    ns.telesys-stud.net.

```

Рис. 3.15. Конфигурация обратной мастер-зоны домена telesys-stud.net

Проверка работоспособности DNS.

- Запустить два терминала и перейти в них в привилегированный режим.
- В одном из них запустить мониторинг системных сообщений:
`tail -f /var/log/messages`
- В другом терминале попробовать запустить DNS-службу:
`/etc/init.d/named start`
- При неудачном запуске (рис. 3.16) проверить и исправить права доступа:
`chown -R named:named /etc/named`
`chown -R named:named /var/named`
- Еще раз запустить DNS-службу и убедиться, что все работает (рис. 3.17).

С помощью команды `dig` (рис. 3.18) проверяем работоспособность запросов к зоне:

`dig @localhost telesys-stud.net`

или с помощью команды `host` (рис. 3.19–3.21):

`host -l telesys-stud.net`

`host -a telesys-stud.net`

`host -t A telesys-stud.net`

`host -t PTR 192.168.0.100`

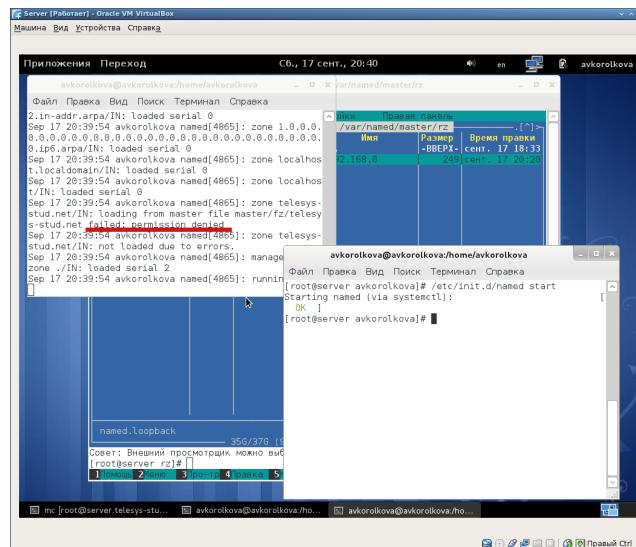


Рис. 3.16. Запуск DNS-службы. Неверные права доступа named

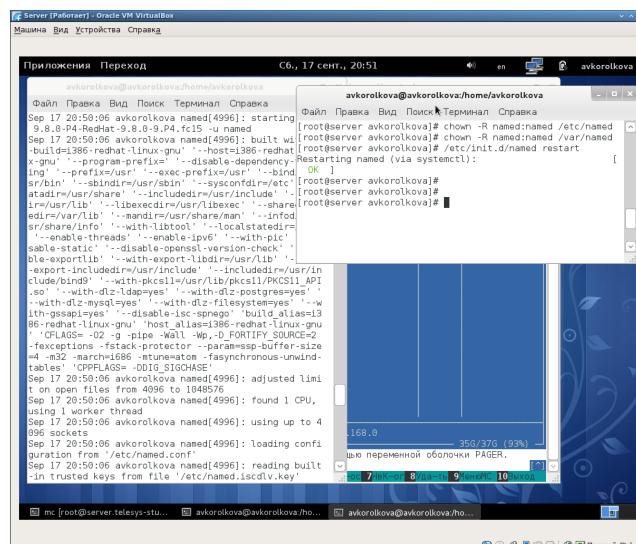
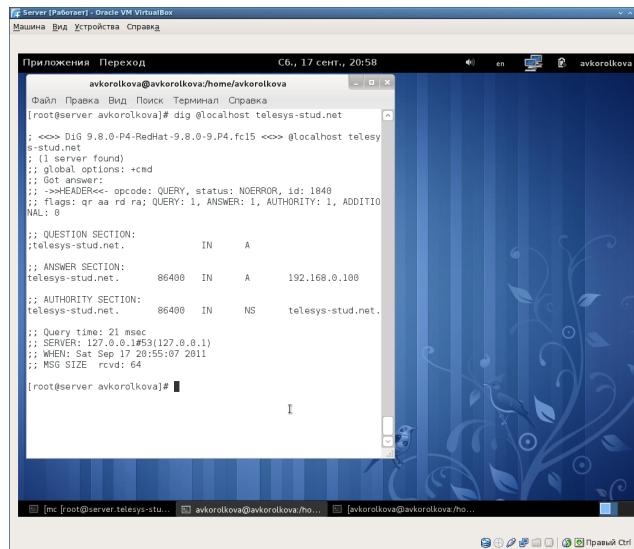


Рис. 3.17. Корректный запуск DNS-службы



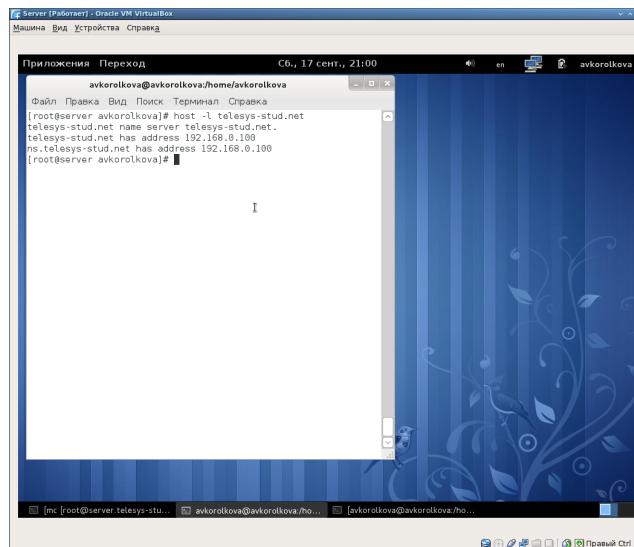
```

Server [Faboraet] - Oracle VM VirtualBox
Машине Вид Устройства Справка

Приложения Переход       Сб, 17 сентя., 20:58
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
[root@server avkorolkova]# dig @localhost telesys-stud.net
; <>> DIG 9.8.0-P4-RedHat-9.8.0-9.P4.fc15 <>> @localhost telesys-stud.net
; <-->| s-stud.net
; [1 server found]
; global options: +cmd
; Got answer:
; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 1848
; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
; 
; QUESTION SECTION:
;telesys-stud.net.           IN  A
; 
;ANSWER SECTION:
telesys-stud.net.      86400  IN  A   192.168.0.100
; 
;AUTHORITY SECTION:
telesys-stud.net.      86400  IN  NS   telesys-stud.net.
; 
; Query time: 21 msec
; SERVER: 127.0.0.1#53[127.0.0.1]
; WHEN: Sat Sep 17 20:55:07 2011
; MSS SIZE: rcvd: 64
[root@server avkorolkova]#

```

Рис. 3.18. Проверка с помощью команды dig



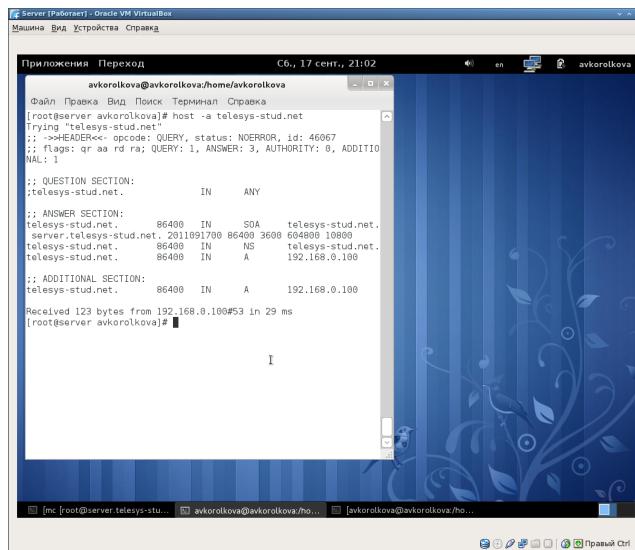
```

Server [Faboraet] - Oracle VM VirtualBox
Машине Вид Устройства Справка

Приложения Переход       Сб, 17 сентя., 21:00
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
[root@server avkorolkova]# host -l telesys-stud.net
Name: telesys-stud.net
telesys-stud.net has address 192.168.0.100
ns.telesys-stud.net has address 192.168.0.100
[root@server avkorolkova]#

```

Рис. 3.19. Проверка с помощью команды host -l



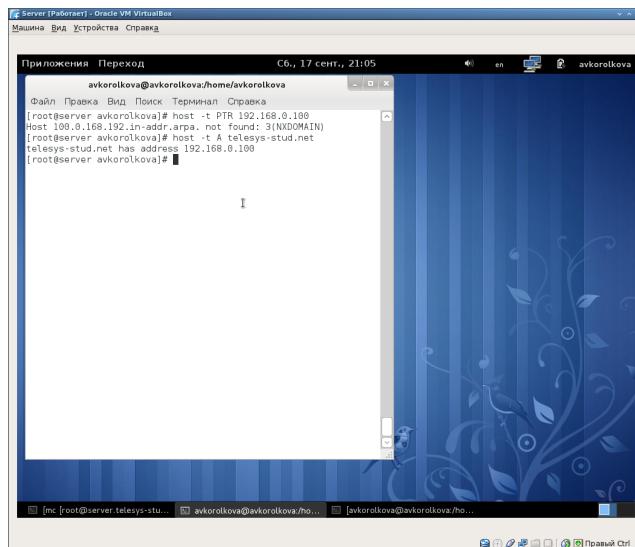
```

Server [Pobortser] - Oracle VM VirtualBox
Машине Вид Устройства Справка

Приложения Переход       Сб., 17 сентя., 21:02
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
[root@server avkorolkova]# host -a telesys-stud.net
Trying "telesys-stud.net"
;; >>HEADER<< opcode: QUERY, status: NOERROR, id: 46067
;; flags: qr a rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
;QUESTION SECTION:
;telesys-stud.net.           IN      ANY
;; ANSWER SECTION:
telesys-stud.net.          86400   IN      SOA     telesys-stud.net.
server.telesys-stud.net. 2011091700 86400 3600 604800 10800
telesys-stud.net.          86400   IN      NS      telesys-stud.net.
telesys-stud.net.          86400   IN      A       192.168.0.108
;; ADDITIONAL SECTION:
telesys-stud.net.          86400   IN      A       192.168.0.108
Received 123 bytes from 192.168.0.108#53 in 29 ms
[root@server avkorolkova]#

```

Рис. 3.20. Проверка с помощью команды host -a



```

Server [Pobortser] - Oracle VM VirtualBox
Машине Вид Устройства Справка

Приложения Переход       Сб., 17 сентя., 21:05
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
[root@server avkorolkova]# host -t PTR 192.168.0.108
Host not found: no address for 192.168.0.108
[root@server avkorolkova]# host -t A telesys-stud.net
telesys-stud.net has address 192.168.0.108
[root@server avkorolkova]#

```

Рис. 3.21. Проверка с помощью команды host -t A и host -t PTR

3.6. Контрольные вопросы

1. Что такое DNS?
2. Чем отличается прямая зона от обратной?
3. Что указывается в файле resolv.conf?
4. Какие типы записи описания ресурсов есть в DNS и для чего они используются?
5. Для чего используется домен in-addr.arpa?
6. Для чего нужен демон named?
7. В чём заключаются основные функции slave-сервера и master-сервера?
8. Какие параметры отвечают за время обновления зоны?
9. Как обеспечить защиту зоны от скачивания и просмотра?
10. Какая запись RR применяется при создании почтовых серверов?
11. Как протестировать работу сервера доменных имён?

Лабораторная работа № 4. Настройка DHCP-сервера и HTTP-сервера

4.1. Цель работы

Приобретение практических навыков по установке и конфигурированию DHCP-сервера и HTTP-сервера.

4.2. Информация, необходимая для начала работы

Перед началом выполнения работы:

- должна быть выполнена лабораторная работа по установке и конфигурированию операционной системы на виртуальной машине Base;
- должна быть выполнена лабораторная работа по установке и конфигурированию DNS-сервера.

4.3. Кратко о DHCP

Протокол динамической конфигурации узла (Dynamic Host Configuration Protocol, DHCP) – сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

DHCP:

- работает по модели «клиент-сервер»;
- позволяет избежать ручной настройки компьютеров сети;
- выделяет каждому компьютеру произвольный свободный IP-адрес из определённого администратором диапазона;
- передача данных осуществляется через протокол UDP, при этом сервер принимает сообщения от клиентов на порт 67 и отправляет сообщения клиентам на порт 68.

Некоторые из наиболее часто используемых опций DHCP:

- IP-адрес маршрутизатора по умолчанию;
- маска подсети;
- адреса серверов DNS;
- имя домена DNS.

4.4. Последовательность выполнения работы

4.4.1. Установка DHCP-сервера

Загрузить в дисплейном классе операционную систему Linux. Осуществить вход в систему.

Запустить терминал. Запустить VirtualBox, введя `VirtualBox` в командной строке.

В свойствах виртуальной машины Server в качестве типа сетевого подключения задать «NAT» (рис. 4.1).

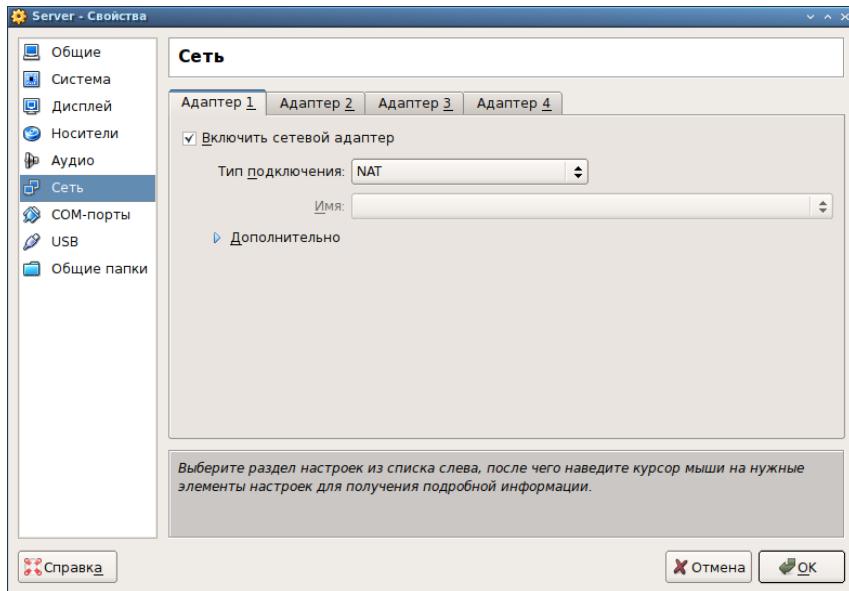


Рис. 4.1. Тип сетевого подключения виртуальной машины Server

Запустить виртуальную машину Server. Открыть «Сетевые соединения», выбрав при нажатии правой кнопки мыши «Изменить соединение...» (рис. 4.2).

Добавить соединение «Wired connection2» и изменить его, выбрав вкладку «Параметры IPv4» и указав «Способ настройки» «Автоматически (DHCP)» (рис. 4.3). Отключить сетевое соединение «Wired connection1» и подключить «Wired connection2».

На виртуальной машине Server запустить терминал и перейти с помощью команды `su` в привилегированный режим (рис. 4.4).

Запустить команду `ifconfig` для проверки адреса (рис. 4.5).

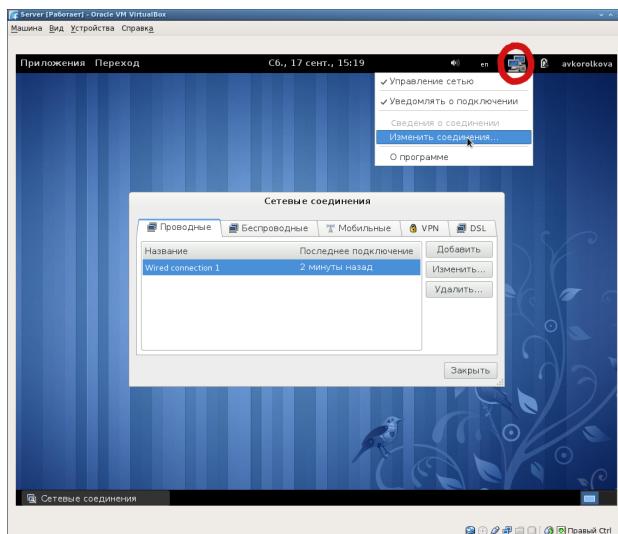


Рис. 4.2. Изменение сетевого подключения виртуальной машины Server

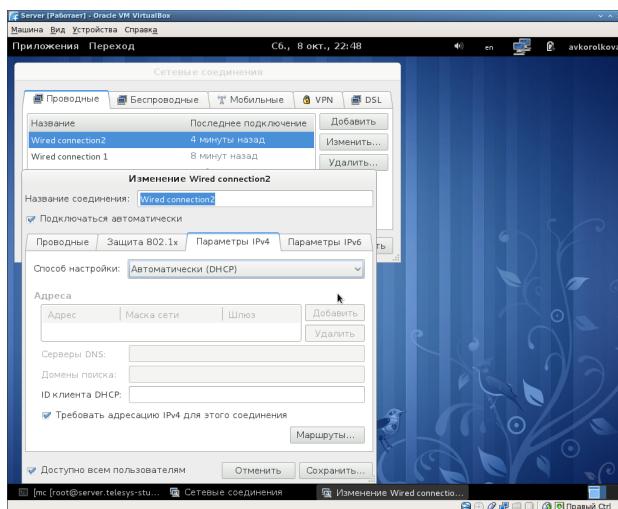


Рис. 4.3. Тип сетевого подключения виртуальной машины Server

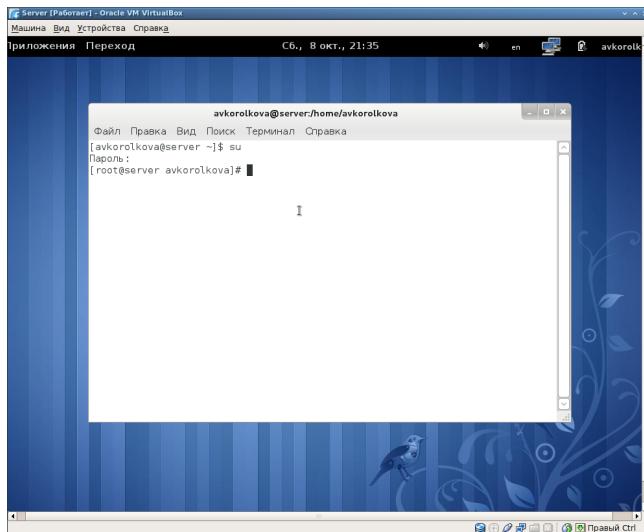


Рис. 4.4. Переход в привилегированный режим

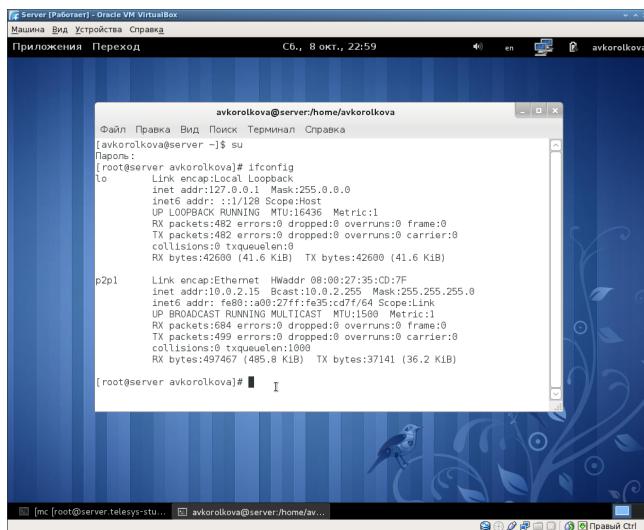


Рис. 4.5. Результат команды ifconfig

Адрес для машин в дисплейном классе при подключении сетевого соединения виртуальной машины через NAT должен быть типа

```
inet addr:10.X.X.X Bcast:10.X.X.255 Mask:255.255.255.0
```

Если же команда `ifconfig`, запущенная на виртуальной машине Server, выдаёт адрес типа

```
inet addr:198.168.X.X Bcast:192.168.X.255 Mask:255.255.255.0
```

то подключение сетевого соединения виртуальной машины осуществляется через внутреннюю сеть и внешняя сеть для этой виртуальной машины не будет доступна. В этом случае нужно в *свойствах виртуальной машины Server* в качестве типа сетевого подключения задать «NAT» (см. рис. 4.1).

Установить `dhclient` и `httpd` с помощью команды (рис. 4.6):
`yum install dhclient httpd`

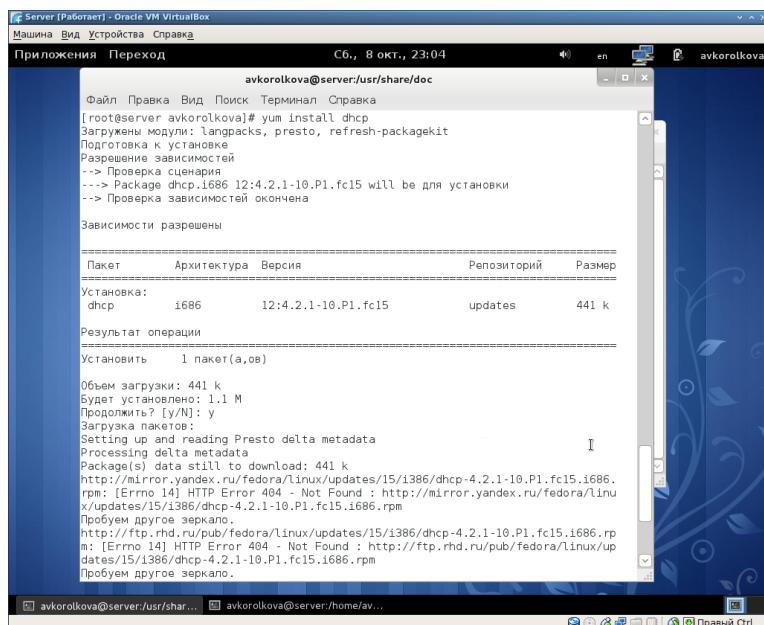


Рис. 4.6. Установка dhclient и httpd

Выйти из операционной системы виртуальной машины Server.

4.4.2. Настройка DHCP-сервера

В свойствах виртуальной машины Server в качестве типа сетевого подключения задать «Внешняя сеть» (рис. 4.7).

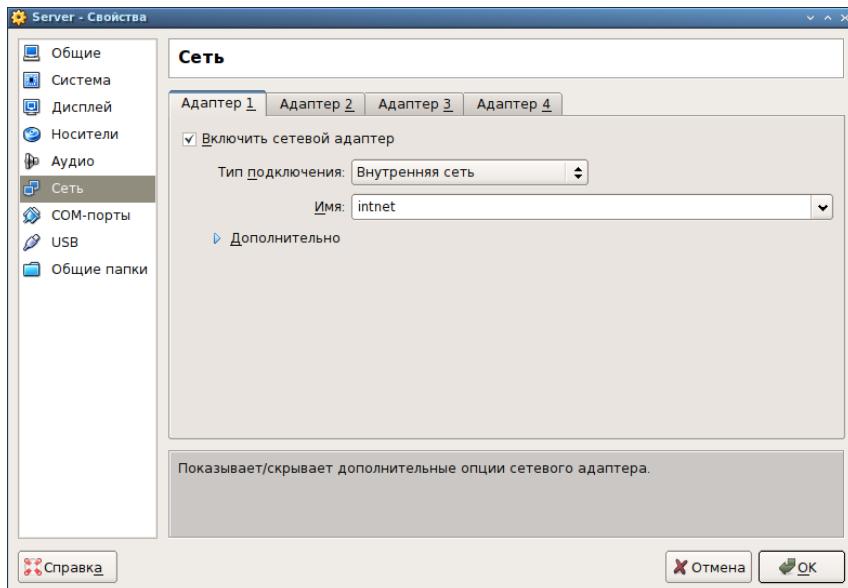


Рис. 4.7. Тип сетевого подключения виртуальной машины Server

Запустить виртуальную машину Server и подключить сетевое соединение «Wired connection1».

На виртуальной машине Server запустить терминал и перейти с помощью команды `su` в привилегированный режим.

Перейти в каталог `/etc/sysconfig/network-scripts/`, сделать копию файла `ifcfg-lo` с названием `ifcfg-p2p1` (рис. 4.8), где `p2p1` — название интерфейса, выдаваемое командой `ifconfig` (рис. 4.9):

```
cd /etc/sysconfig/network-scripts/
cp ifcfg-lo ifcfg-p2p1
```

Для удобства дальнейшей работы можно запустить файловый менеджер Midnight Commander с помощью команды `mc`. Справку по работе с `mc` можно найти, воспользовавшись руководством `man`, или на сайте <https://www.midnight-commander.org/wiki/ru/doc>.

mc [root@server.telesys-stud.net]:/etc/sysconfig/network-scripts

Файл Правка Вид Поиск Терминал Вкладки Справка

```
[root@server sysconfig]# cd /etc/sysconfig/network-scripts/
[root@server network-scripts]# ls
ifcfg-lo          ifdown-isdn   ifup-eth    ifup-ppp
ifcfg-Wired_connection_1  ifdown-post  ifup-ipp   ifup-routes
ifcfg-Проводное_соединение_1 ifdown-ppp  ifup-ipsec  ifup-sit
ifdown             ifdown-routes ifup-ipv6  ifup-tunnel
ifdown-bnep        ifdown-sit   ifup-ixp   ifup-wireless
ifdown-eth         ifdown-tunnel ifup-isdn  init.ipv6-global
ifdown-ipp         ifup        ifup-pip    net.hotplug
ifdown-ipsec       ifup-aliases ifup-plusb network-functions
ifdown-ipv6        ifup-bnep   ifup-post  network-functions-ipv6
[root@server network-scripts]# cp ifcfg-lo ifcfg-p2p1
```

Рис. 4.8. Создание конфигурационного файла интерфейса p2p1

mc [root@server.telesys-stud.net]:/var/named/master

Файл Правка Вид Поиск Терминал Вкладки Справка

```
[root@server ~]# ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:1242 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1242 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:112636 (109.9 KiB)  TX bytes:112636 (109.9 KiB)

p2p1    Link encap:Ethernet HWaddr 08:00:27:35:CD:7F
        inet addr:192.168.0.100 Bcast:192.168.0.255 Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe35:cd7f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:992 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1401 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:146016 (142.5 KiB)  TX bytes:1248792 (1.1 MiB)

[root@server ~]#
```

Рис. 4.9. Результат команды ifconfig

Отредактировать файл ifcfg-p2p1 (рис. 4.10):

```
DEVICE=p2p1
IPADDR=192.168.0.100
NETMASK=255.255.255.0
NETWORK=192.168.0.0
BROADCAST=192.168.0.255
ONBOOT=yes
NAME=p2p1
GATEWAY=192.168.0.100
```

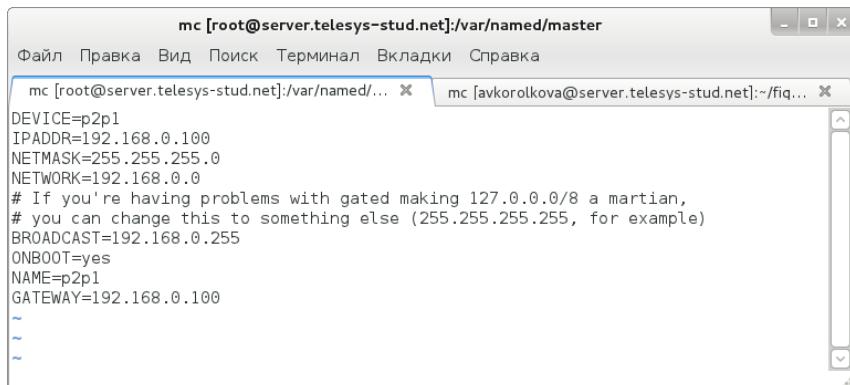


Рис. 4.10. Файл ifcfg-p2p1

В файле ifcfg-p2p1:

- DEVICE=p2p1 — название интерфейса, выдаваемое ifconfig;
- IPADDR=192.168.0.100 — ip-адрес Server;
- NETWORK=192.168.0.0 — адрес сети;
- BROADCAST=192.168.0.255 — broadcast-адрес;
- ONBOOT=yes — указание, что при загрузке Server соединение работает;
- NAME=p2p1 — обращение к интерфейсу при использовании ifup;
- GATEWAY=192.168.0.100 — ip-адрес шлюза.

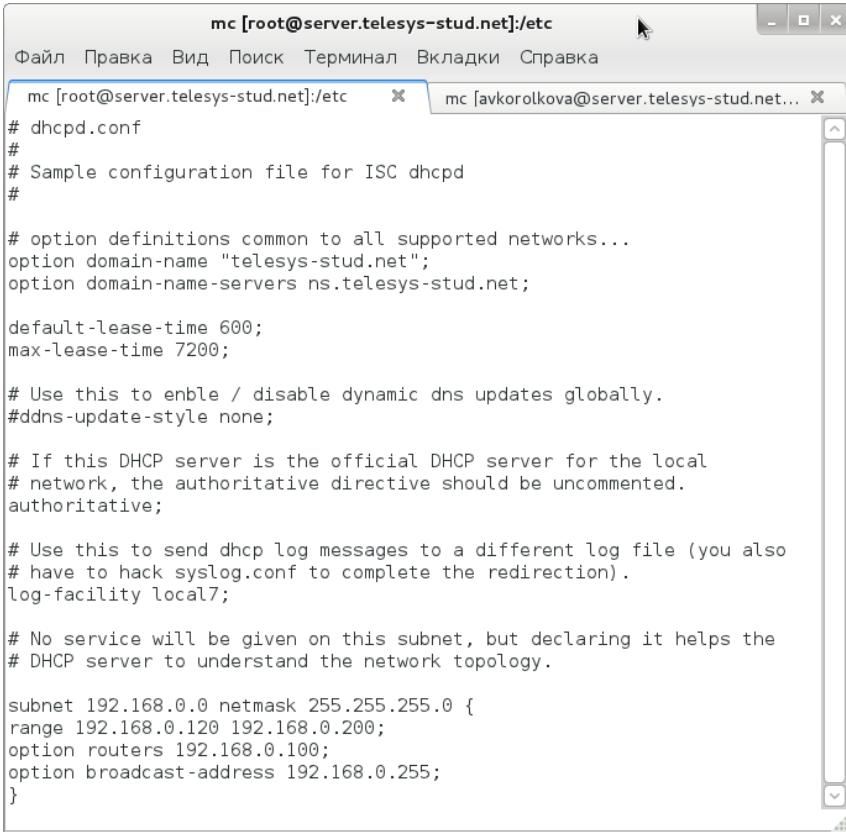
Из каталога /usr/share/doc/dhcp-4.2.1 скопировать пример конфигурационного файла dhcpcd.conf.sample в каталог /etc и переименовать его в dhcpcd.conf:

```
cp /usr/share/doc/dhcp-4.2.1/dhcpcd.conf.sample /etc/
cd /etc
mv dhcpcd.conf.sample dhcpcd.conf
```

Отредактировать файл `dhcpd.conf` (рис. 4.11):

- в качестве `option domain-name` указать "`telesys-stud.net;`"
- в качестве `option domain-name-servers` указать `ns.telesys-stud.net;`
- раскомментировать `authoritative;`
- задать конфигурацию dhcp-сети:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.120 192.168.0.200;
    option routers 192.168.0.100;
    option broadcast-address 192.168.0.255;
}
```



```
mc [root@server.telesys-stud.net]:/etc
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc × mc [avkorolkova@server.telesys-stud.net... ×
# dhcpd.conf
#
# Sample configuration file for ISC dhcpcd
#
# option definitions common to all supported networks...
option domain-name "telesys-stud.net";
option domain-name-servers ns.telesys-stud.net;

default-lease-time 600;
max-lease-time 7200;

# Use this to enable / disable dynamic dns updates globally.
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.120 192.168.0.200;
    option routers 192.168.0.100;
    option broadcast-address 192.168.0.255;
}
```

Рис. 4.11. Файл `/etc/dhcpd.conf`

Перейти в каталог `/etc/sysconfig/` и в файле `dhcpd` прописать (рис. 4.12):

```
DHCPDARGS="-cf /etc/dhcpd.conf p2p1"
```

указывая тем самым, что для соединения `p2p1` применяется конфигурационный файл для DHCP-сервера.

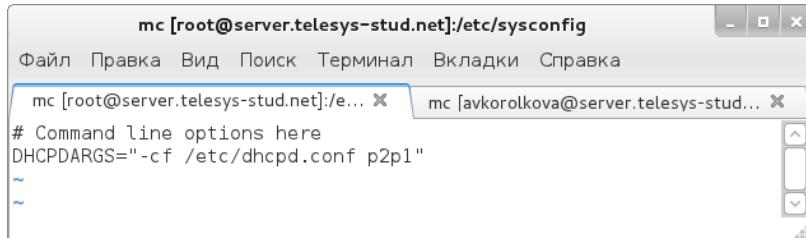


Рис. 4.12. Файл `/etc/sysconfig/dhcpd`

В файле `/var/named/master/fz/telesys-stud.net` добавить запись для прямой зоны DNS-сервера (рис. 4.13):

dhcp	A 192.168.0.100
------	-----------------



Рис. 4.13. Файл `/var/named/master/fz/telesys-stud.net`

В файле `/var/named/master/rz/192.168.0` добавить запись для обратной зоны DNS-сервера (рис. 4.14):

```
100 PTR dhcp.telesys-stud.net.
```

```

mc [root@server.telesys-stud.net]:/var/named/master/rz
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/var/named/master/rz... × mc [avkorolkova@server.telesys-stud.net... ×
$TTL 1D
@ IN SOA @ server.telesys-stud.net. (
    2011101500 ; serial
    1D ; refresh
    1H ; retry
    1W ; expire
    3H ) ; minimum
NS      @
A       192.168.0.100
PTR     server.telesys-stud.net.
$ORIGIN 0.168.192.in-addr.appa.
100    PTR    ns.telesys-stud.net.
100    PTR    dhcp.telesys-stud.net.

```

Рис. 4.14. Файл `/var/named/master/rz/192.168.0`

Запустить DNS-сервер:

```
/etc/init.d/named start
```

Запустить DHCP-сервер:

```
systemctl start dhcpcd.service
```

4.4.3. Проверка работы DHCP

В VirtualBox создать виртуальную машину Client. В свойствах виртуальной машины Client в качестве типа сетевого подключения задать «Внутренняя сеть». Запустить виртуальную машину Client.

На виртуальной машине Client перейти в привилегированный режим и проверить, что выдаётся адрес из диапазона 192.168.0.120 – 192.168.0.200 (рис. 4.15). Проверить доступность DHCP-сервера с помощью команды `ping` (см. рис. 4.15). Проверить доступность по ip-адресу виртуальной машины Client с DHCP-сервера с помощью команды `ping`.

```

Client [Работает] - Oracle VM VirtualBox
Машина Вид Устройства Справка
Приложения Переход
Вс., 9 окт., 00:21 en avkorolkova
avkorolkova@avkorolkova:~$ su
[avkorolkova@avkorolkova ~]$ su
[avkorolkova@avkorolkova ~]$ ifconfig
lo Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:440 (440.0 b) TX bytes:440 (440.0 b)

p2p1 Link encap:Ethernet HWaddr 00:0c:29:ff:fe:05
      inet addr:192.168.0.129 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::0c29:fffe%p2p1/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:53 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:6769 (6.6 KiB) TX bytes:10155 (9.9 KiB)

[avkorolkova@avkorolkova ~]$ ping 192.168.0.100
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.
64 bytes from 192.168.0.100: icmp_req=1 ttl=64 time=1.04 ms
64 bytes from 192.168.0.100: icmp_req=2 ttl=64 time=2.03 ms
^C
--- 192.168.0.100 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 1.047/1.538/2.030/0.493 ms
[avkorolkova@avkorolkova ~]$

```

Рис. 4.15. Результат ifconfig и ping на виртуальной машине Client

4.4.4. Настройка HTTP-сервера

На виртуальной машине Server должен быть установлен httpd. Запустить httpd:

```
systemctl start httpd.service
```

Перейти в /var/www/html и создать файл index.html:

```
cd /var/www/html/
touch index.html
```

Отредактировать index.html:

```
<body>
  Test!!!
</body>
```

Запустить Firefox и задать адрес сайта `http://localhost` (рис. 4.16).

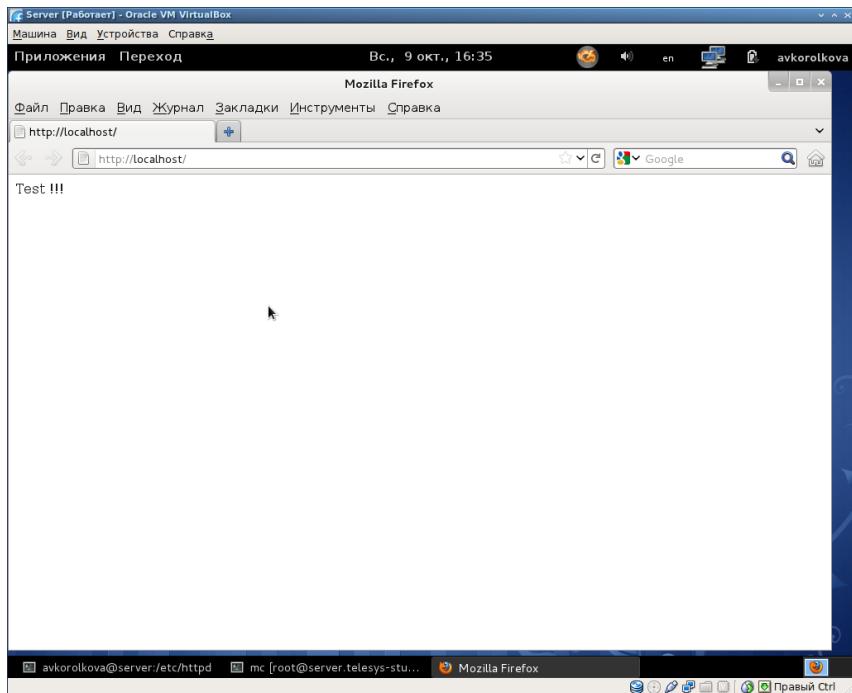


Рис. 4.16. Запуск `http://localhost` на виртуальной машине Server

В файле `/var/named/master/fz/telesys-stud.net` добавить запись для прямой зоны DNS-сервера (рис. 4.17):

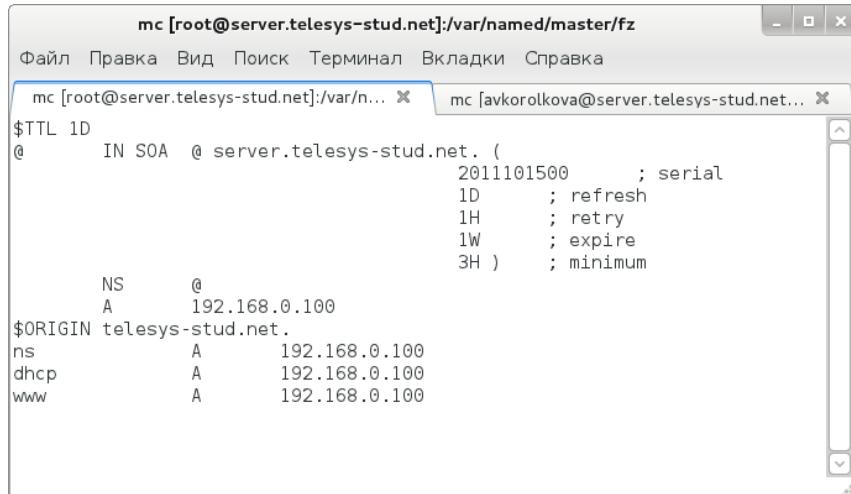
www A 192.168.0.100

В файле `/var/named/master/rz/192.168.0` добавить запись для обратной зоны DNS-сервера (рис. 4.18):

100 PTR www.telesys-stud.net.

Перезапустить DNS-сервер:

`/etc/init.d/named restart`

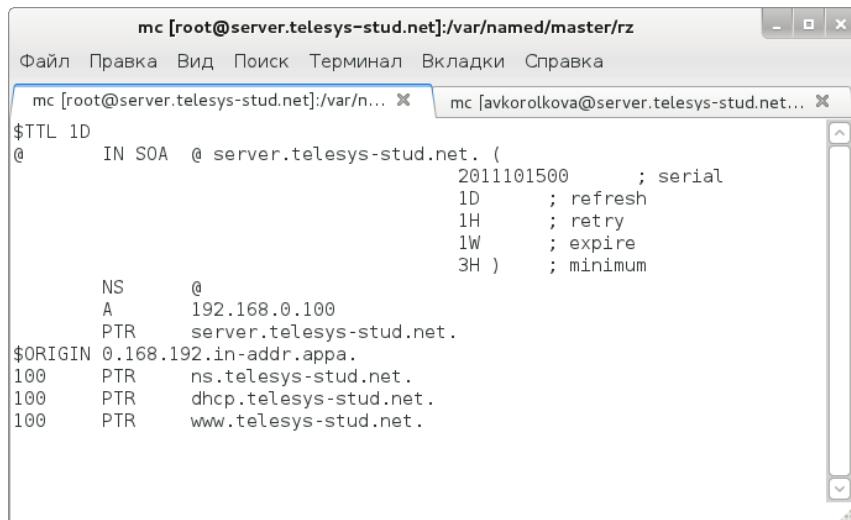


```

mc [root@server.telesys-stud.net]:/var/named/master/fz
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/var/named/master/fz... mc [avkorolkova@server.telesys-stud.net...
$TTL 1D
@ IN SOA @ server.telesys-stud.net. (
                                2011101500      ; serial
                                1D              ; refresh
                                1H              ; retry
                                1W              ; expire
                                3H )            ; minimum
NS      @
A       192.168.0.100
$ORIGIN telesys-stud.net.
ns      A       192.168.0.100
dhcp   A       192.168.0.100
www    A       192.168.0.100

```

Рис. 4.17. Файл /var/named/master/fz/telesys-stud.net



```

mc [root@server.telesys-stud.net]:/var/named/master/rz
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/var/named/master/rz... mc [avkorolkova@server.telesys-stud.net...
$TTL 1D
@ IN SOA @ server.telesys-stud.net. (
                                2011101500      ; serial
                                1D              ; refresh
                                1H              ; retry
                                1W              ; expire
                                3H )            ; minimum
NS      @
A       192.168.0.100
PTR    server.telesys-stud.net.
$ORIGIN 0.168.192.in-addr.apa.
100    PTR    ns.telesys-stud.net.
100    PTR    dhcp.telesys-stud.net.
100    PTR    www.telesys-stud.net.

```

Рис. 4.18. Файл /var/named/master/rz/192.168.0

Настроить межсетевой экран на Server, задав в терминале:

```
iptables --flush
system-config-firewall
```

и в открывшемся окне (рис. 4.19) указав в качестве доверенных служб DNS, WWW.

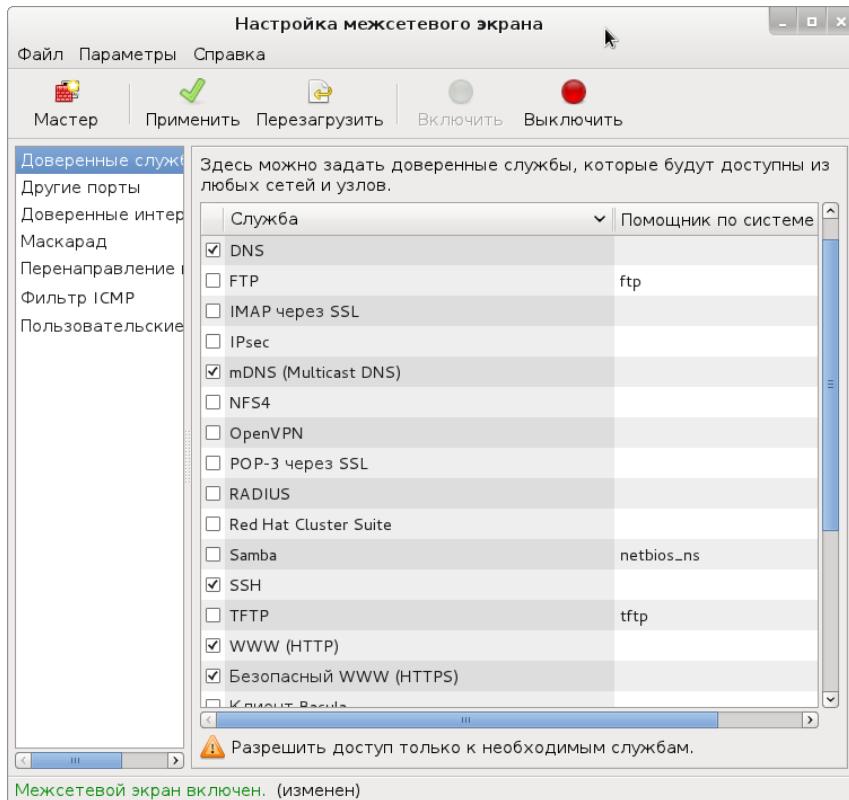


Рис. 4.19. Доверенные службы в межсетевом экране

На виртуальной машине Client: проверить доступность DNS-сервера с помощью команды `ping` (рис. 4.20), запустить Firefox, отключить прокси, в качестве адреса сайта задать `http://www.telesys-stud.net` (рис. 4.21). В случае недоступности `www.telesys-stud.net` отключить на Server и Client межсетевой экран.

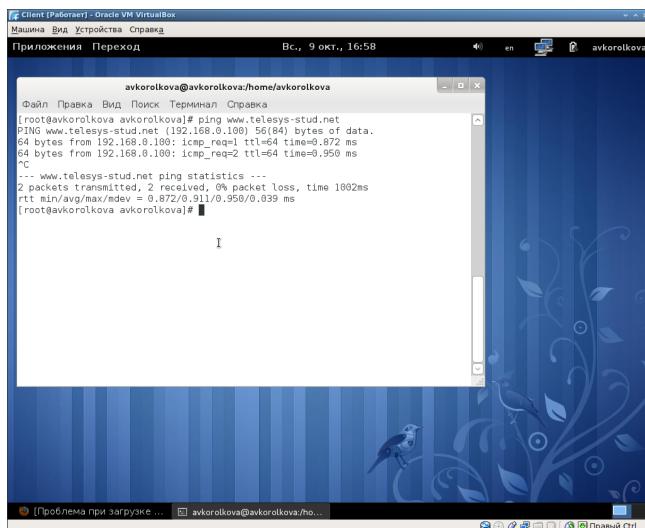


Рис. 4.20. Результат команды ping на виртуальной машине Client

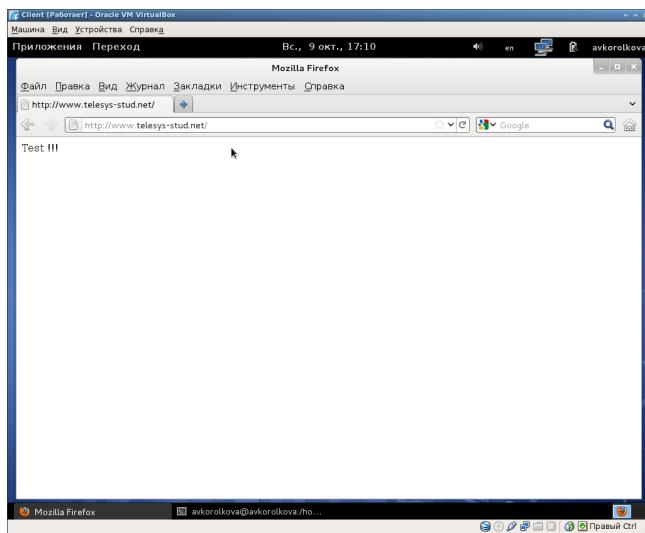


Рис. 4.21. <http://www.telesys-stud.net> на виртуальной машине Client

4.5. Контрольные вопросы

1. За что отвечает протокол DHCP? В каких файлах обычно находятся настройки его работы? За что отвечает каждый из файлов?
2. За что отвечает протокол HTTP?
3. Дайте определение понятию «сетевая маска». Приведите примеры.
4. Дайте определение понятию «broadcast-адрес». Приведите примеры.
5. Что такое gateway?
6. Приведите примеры с пояснениями использования сетевых утилит для диагностики сетевых соединений.

Лабораторная работа № 5. Установка и настройка почтового сервера

5.1. Цель работы

Приобретение практических навыков по установке и конфигурированию почтового сервера.

5.2. Информация, необходимая для начала работы

Перед началом выполнения работы:

- должна быть выполнена лабораторная работа по установке и конфигурированию операционной системы на виртуальной машине Base;
- должна быть выполнена лабораторная работа по установке и конфигурированию DNS-сервера;
- должна быть выполнена лабораторная работа по установке и конфигурированию DHCP-сервера.

5.3. Последовательность выполнения работы

5.3.1. Установка почтового сервера

Загрузить в дисплейном классе операционную систему Linux. Осуществить вход в систему.

Запустить терминал. Запустить VirtualBox, введя *VirtualBox* в командной строке.

В свойствах виртуальной машины Server в качестве типа сетевого подключения задать «NAT» (рис. 5.1).

Запустить виртуальную машину Server. Отключить сетевое соединение «Wired connection1» и подключить «Wired connection2».

На виртуальной машине Server запустить терминал и перейти с помощью команды *su* в привилегированный режим. Установить IMAP-сервер *dovecot*, SMTP-сервер *postfix*, *telnet* (рис. 5.2):

```
yum install dovecot postfix telnet
```

Выключить виртуальную машину Server.

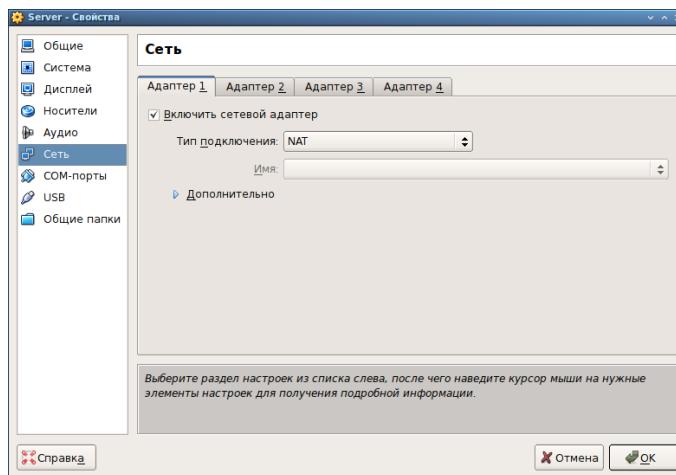


Рис. 5.1. Тип сетевого подключения виртуальной машины Server

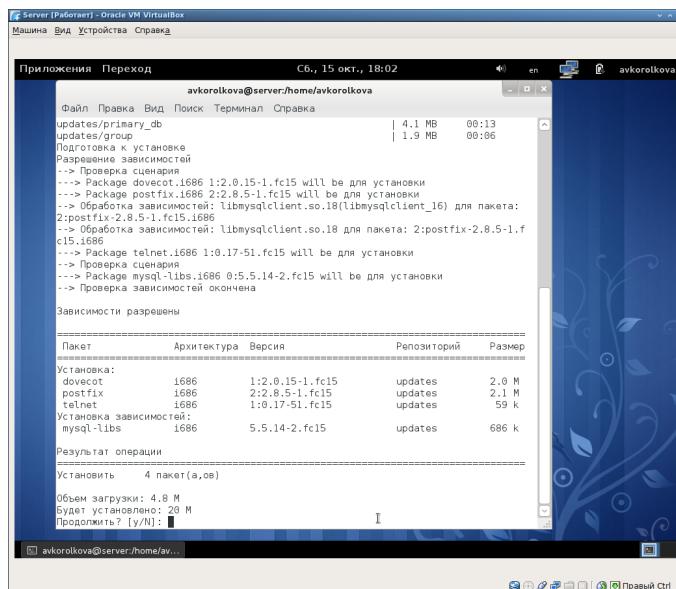


Рис. 5.2. Установка dovecot, postfix, telnet

5.3.2. Настройка почтового сервера

В свойствах виртуальной машины Server в качестве типа сетевого подключения задать «Внутренняя сеть» (рис. 5.3).

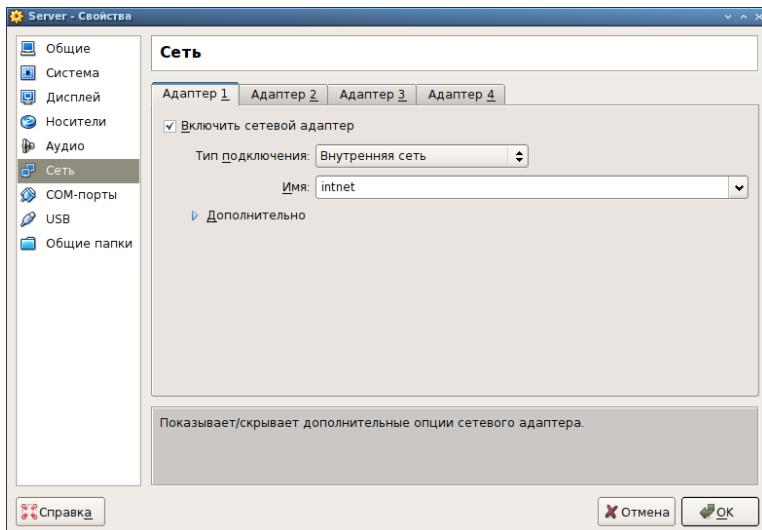


Рис. 5.3. Тип сетевого подключения виртуальной машины Server

Запустить виртуальную машину Server и подключить сетевое соединение «р2р1».

На виртуальной машине Server запустить терминал и перейти с помощью команды `su` в привилегированный режим. Перейти в каталог `/etc/dovecot` и открыть для редактирования файл `dovecot.conf`.

В файле `dovecot.conf` (рис. 5.4):

- раскомментировать строку:

```
protocols = imap pop3 lmtp
```

указав используемые почтовые протоколы;

- раскомментировать строку, указывающую на директорию для данных dovecot:

```
base_dir = /var/run/dovecot/
```

- раскомментировать строку, задающую выдаваемое клиенту сообщение:

```
login_greeting = Dovecote ready.
```

- указать, с какой сети разрешён доступ на почтовый сервер:

```
login_trusted_networks = 192.168.0.0/16
```

- для отображения отладочной информации указать:

```
verbose_proctitle = yes
```

```

mc [root@server.telesys-stud.net]:/etc/dovecot
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc/dovecot × mc [avkorolkova@server.telesys-stud.net]:~/fig... ×

# Protocols we want to be serving.
protocols = imap pop3 lmtp

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
#listen = *, ::

# Base directory where to store runtime data.
base_dir = /var/run/dovecot/

# Greeting message for clients.
login_greeting = Dovecot ready.

# Space separated list of trusted network ranges. Connections from these
# IPs are allowed to override their IP addresses and ports (for logging and
# for authentication checks). disable_plaintext_auth is also ignored for
# these networks. Typically you'd specify your IMAP proxy servers here.
login_trusted_networks = 192.168.0.0/16

# Separe separated list of login access check sockets (e.g. tcpwrap)
#login_access_sockets =

# Show more verbose process titles (in ps). Currently shows user name and
# IP address. Useful for seeing who are actually using the IMAP processes
# (e.g. shared mailboxes or if same uid is used for multiple accounts).
verbose_proctitle = yes

```

Рис. 5.4. Файл /etc/dovecot/dovecot.conf

Перейти в каталог `/etc/dovecot/conf.d/` и открыть для редактирования файл `10-mail.conf`. Указать, что каталог для почтовых сообщений располагается в домашней директории пользователя в каталоге `Maildir` (рис. 5.5):

```
mail-location = maildir:~/Maildir
```

```
mc [root@server.telesys-stud.net]:/etc/dovecot/conf.d
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc/dovec... mc [avkorolkova@server.telesys-stud.net]:~/fiq...
# <doc/wiki/MailLocation.txt>
#
#mail_location =
mail_location = maildir:~/Maildir

# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.
#
```

Рис. 5.5. Файл /etc/dovecot/conf.d/10-mail.conf

Перейти в каталог `/etc/postfix/` и открыть для редактирования файл `main.cf`:

- указать имя почтового сервера и имя домена (рис. 5.6):

```
myhostname = mail.telesys-stud.net
mydomain = telesys-stud.net
```

```
mc [root@server.telesys-stud.net]:/etc/postfix
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc/postfix mc [avkorolkova@server.telesys-stud.net]:~/fiq...
#
# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
#myhostname = host.domain.tld
myhostname = mail.telesys-stud.net

# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
# $mydomain is used as a default value for many other configuration
# parameters.
#
mydomain = telesys-stud.net
```

Рис. 5.6. Файл /etc/postfix/main.cf: имя почтового сервера и имя домена

- в качестве интерфейса для получения почты указать `all` (рис. 5.7):

`inet_interface = all`

```

mc [root@server.telesys-stud.net]:/etc/postfix
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc/postfix  X  mc [avkorolkova@server.telesys-stud.net]:~/fiqs...
# RECEIVING MAIL

# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on. By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
inet_interfaces = all
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost
#inet_interfaces = localhost

```

Рис. 5.7. Файл /etc/postfix/main.cf: интерфейс для получения почты

- в качестве места назначения почты указать (рис. 5.8):

```
mydestination = $myhostname, localhost.$mydomain,
                localhost, $mydomain
```

```

mc [root@server.telesys-stud.net]:/etc/postfix
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc/postfix  X  mc [avkorolkova@server.telesys-stud.net]:~/fiqs-m...
# The local machine is always the final destination for mail addressed
# to user@[the.net.work.address] of an interface that the mail system
# receives mail on (see the inet_interfaces parameter).
#
# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
#mydestination = $myhostname, localhost.$mydomain, localhost
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain, mail.$mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#               mail.$mydomain, www.$mydomain, ftp.$mydomain

```

Рис. 5.8. Файл /etc/postfix/main.cf: место назначения почты

- для доступа пользователей к почте указать (рис. 5.9):

```
local_recipient_maps =
```

это означает, что данные по учетной записи пользователя берутся из соответствующих файлов сервера.

```

mc [root@server.telesys-stud.net]:/etc/postfix
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/etc/postfix x mc [avkorolkova@server.telesys-stud.net]:~/figs-m...
#
# Details are described in the LOCAL_RECIPIENT_README file.
#
# Beware: if the Postfix SMTP server runs chrooted, you probably have
# to access the passwd file via the proxymap service, in order to
# overcome chroot restrictions. The alternative, having a copy of
# the system passwd file in the chroot jail is just not practical.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify a bare username, an @domain.tld
# wild-card, or specify a user@domain.tld address.
#
#local_recipient_maps = unix:passwd.byname $alias_maps
#local_recipient_maps = proxy:unix:passwd.byname $alias_maps
local_recipient_maps =

```

Рис. 5.9. Файл /etc/postfix/main.cf: доступ пользователей к почте

- указать доверенную подсеть (рис. 5.10):

```

mynetworks_style = subnet
mynetworks = 192.168.0.0/16, 127.0.0.0/8

```

- указать каталог для доставки почты (рис. 5.11):

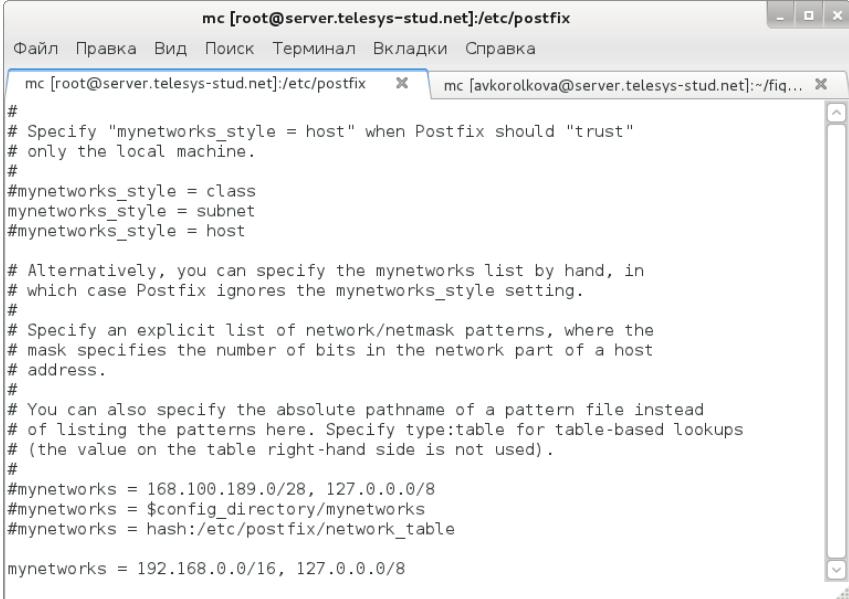
```
home_mailbox = Maildir/
```

В файле /var/named/master/fz/telesys-stud.net добавить запись для прямой зоны DNS-сервера (рис. 5.12):

```
mail          A 192.168.0.100
```

В файле /var/named/master/rz/192.168.0 добавить запись для обратной DNS-сервера (рис. 5.13):

```
100          PTR mail.telesys-stud.net.
```



mc [root@server.telesys-stud.net]:/etc/postfix

Файл Правка Вид Поиск Терминал Вкладки Справка

mc [root@server.telesys-stud.net]:/etc/postfix x mc [avkorolkova@server.telesys-stud.net]:~/fiq... x

```

#
# Specify "mynetworks_style = host" when Postfix should "trust"
# only the local machine.
#
#mynetworks_style = class
mynetworks_style = subnet
#mynetworks_style = host

# Alternatively, you can specify the mynetworks list by hand, in
# which case Postfix ignores the mynetworks_style setting.
#
# Specify an explicit list of network/netmask patterns, where the
# mask specifies the number of bits in the network part of a host
# address.
#
# You can also specify the absolute pathname of a pattern file instead
# of listing the patterns here. Specify type:table for table-based lookups
# (the value on the table right-hand side is not used).
#
#mynetworks = 168.100.189.0/28, 127.0.0.0/8
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table

mynetworks = 192.168.0.0/16, 127.0.0.0/8

```

Рис. 5.10. Файл /etc/postfix/main.cf: доверенная подсеть



mc [root@server.telesys-stud.net]:/etc/postfix

Файл Правка Вид Поиск Терминал Вкладки Справка

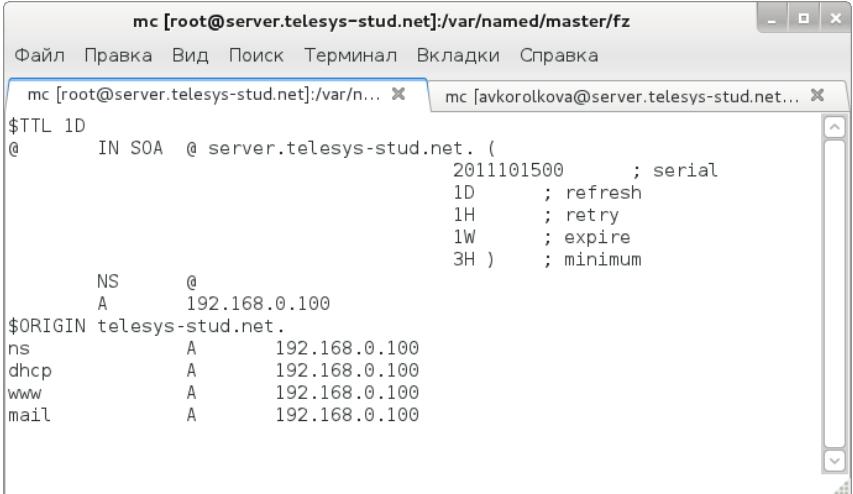
mc [root@server.telesys-stud.net]:/etc/postfix x mc [avkorolkova@server.telesys-stud.net]:~/fiq... x

```

# DELIVERY TO MAILBOX
#
# The home_mailbox parameter specifies the optional pathname of a
# mailbox file relative to a user's home directory. The default
# mailbox file is /var/spool/mail/user or /var/mail/user. Specify
# "Maildir/" for qmail-style delivery (the / is required).
#
#home_mailbox = Mailbox
home_mailbox = Maildir/

```

Рис. 5.11. Файл /etc/postfix/main.cf: каталог для доставки почты

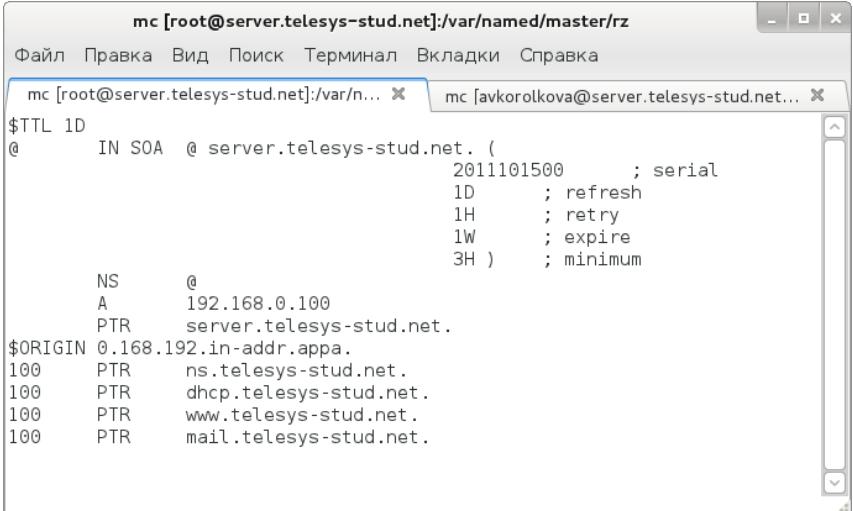


```

mc [root@server.telesys-stud.net]:/var/named/master/fz
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/var/named/master/fz... mc [avkorolkova@server.telesys-stud.net... ×
$TTL 1D
@      IN SOA  @ server.telesys-stud.net. (
                      2011101500      ; serial
                      1D            ; refresh
                      1H            ; retry
                      1W            ; expire
                      3H )          ; minimum
NS      @
A       192.168.0.100
$ORIGIN telesys-stud.net.
ns      A       192.168.0.100
dhcp    A       192.168.0.100
www     A       192.168.0.100
mail    A       192.168.0.100

```

Рис. 5.12. Файл /var/named/master/fz/telesys-stud.net



```

mc [root@server.telesys-stud.net]:/var/named/master/rz
Файл Правка Вид Поиск Терминал Вкладки Справка
mc [root@server.telesys-stud.net]:/var/named/master/rz... mc [avkorolkova@server.telesys-stud.net... ×
$TTL 1D
@      IN SOA  @ server.telesys-stud.net. (
                      2011101500      ; serial
                      1D            ; refresh
                      1H            ; retry
                      1W            ; expire
                      3H )          ; minimum
NS      @
A       192.168.0.100
PTR    server.telesys-stud.net.
$ORIGIN 0.168.192.in-addr.apa.
100    PTR    ns.telesys-stud.net.
100    PTR    dhcp.telesys-stud.net.
100    PTR    www.telesys-stud.net.
100    PTR    mail.telesys-stud.net.

```

Рис. 5.13. Файл /var/named/master/rz/192.168.0

Запустить DNS-сервер, DHCP-сервер, SMTP-сервер, IMAP-сервер (рис. 5.14):

```
/etc/init.d/named start
systemctl start dhcpcd.service
killall sendmail
postfix start
/etc/init.d/dovecot start
```

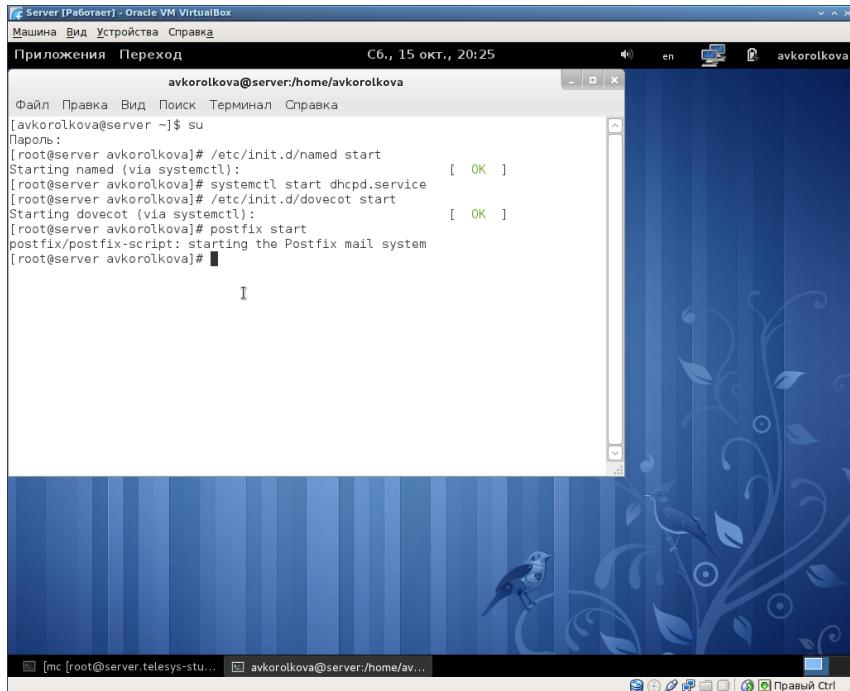


Рис. 5.14. Запуск named, dhcpcd, dovecot, postfix

Для просмотра отладочных сообщений используйте команду:

```
tail -f /var/log/messages
```

и команду

```
tail -f /var/log/maillog
```

5.3.3. Проверка работы почтового сервера

На виртуальной машине Server запустить и настроить почтовый клиент Evolution, выбрав «Приложения», «Офис», «Электронная почта и календарь Evolution» (рис. 5.15, 5.16).

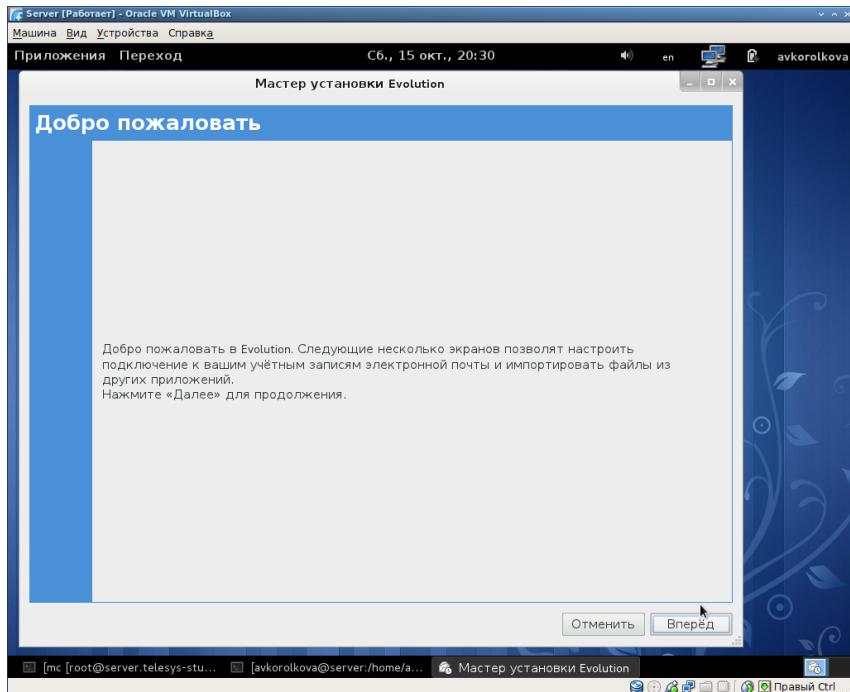


Рис. 5.15. Запуск Evolution

Задать e-mail в виде `имя_пользователя@telesys-stud.net` (рис. 5.17).

Задать IMAP-сервер `mail.telesys-stud.net` (рис. 5.18) и параметры получения почты (рис. 5.19).

Задать SMTP-сервер `mail.telesys-stud.net` (рис. 5.20) и информацию об учётной записи (рис. 5.21), завершить настройку и запустить Evolution (рис. 5.22, 5.23).

После запуска Evolution введите пароль для заданной учетной записи (согласно настройкам пароль совпадает с паролем для пользователя системы).

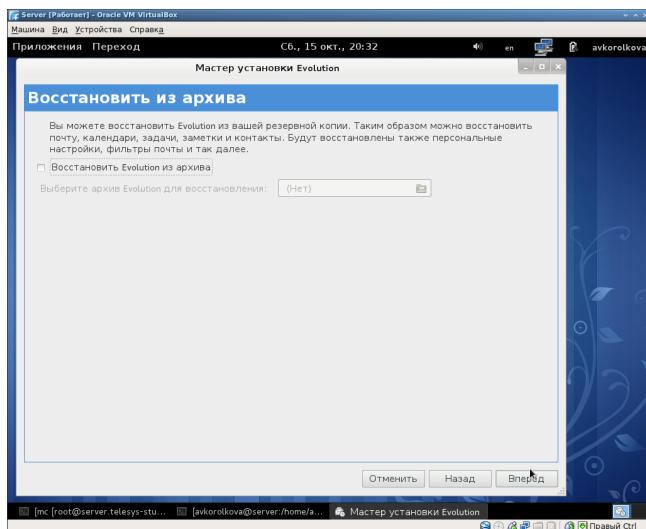


Рис. 5.16. Настройка Evolution

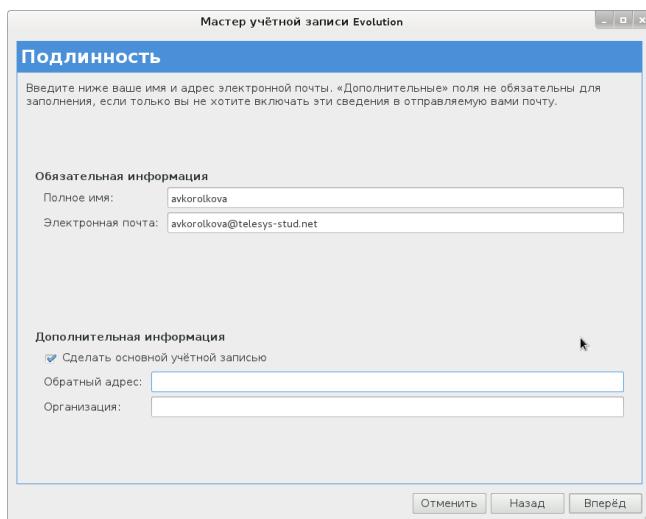


Рис. 5.17. Настройка Evolution. Указание e-mail

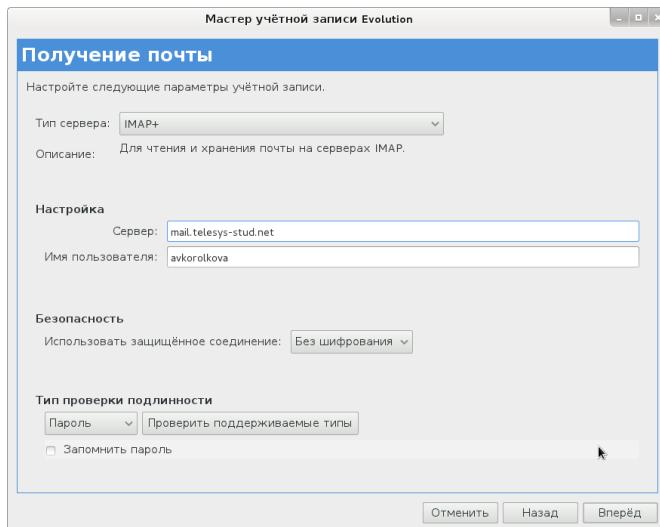


Рис. 5.18. Настройка Evolution. IMAP-сервер

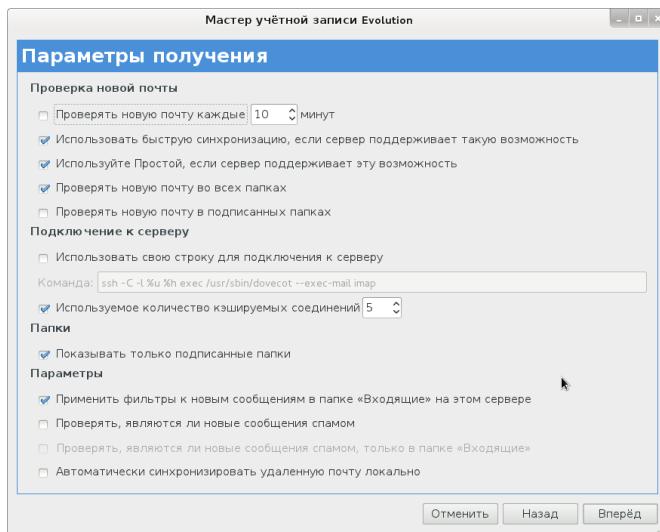


Рис. 5.19. Настройка Evolution. Параметры получения почты

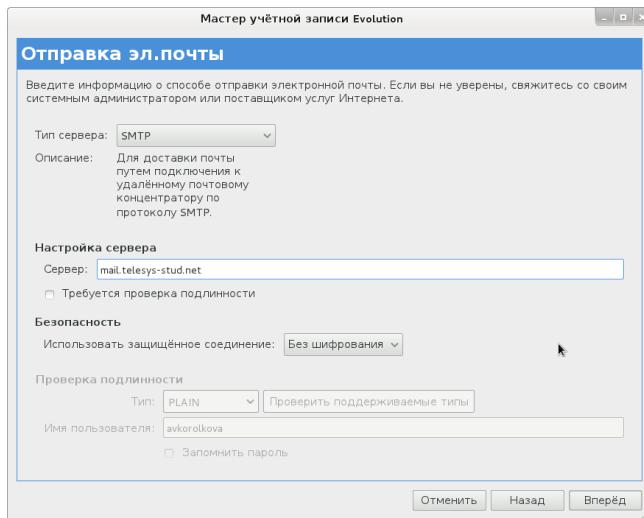


Рис. 5.20. Настройка Evolution. SMTP-сервер

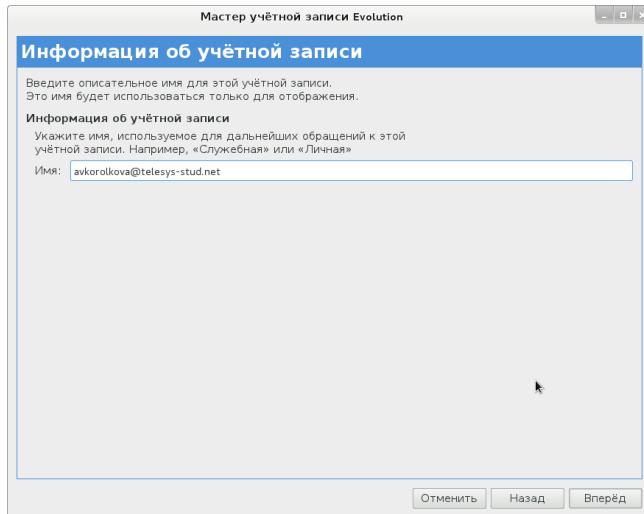


Рис. 5.21. Настройка Evolution. Информация об учётной записи

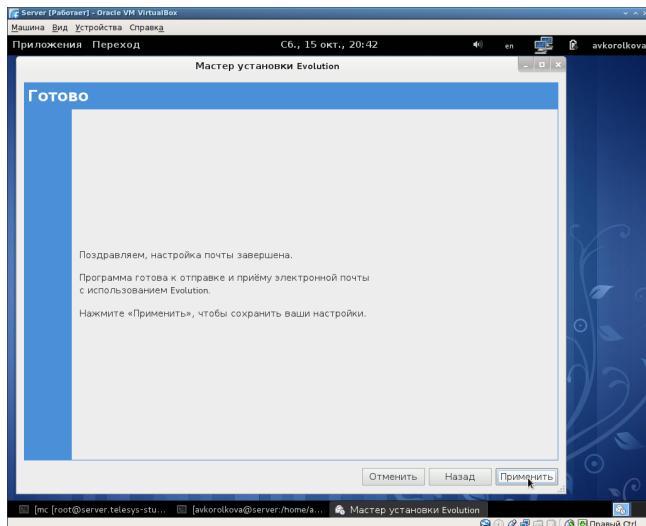


Рис. 5.22. Настройка Evolution. Завершение настройки

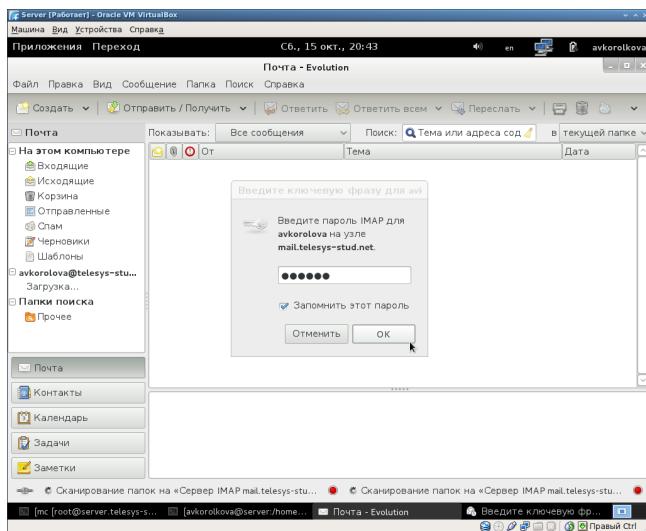


Рис. 5.23. Запуск Evolution

Создать тестовое письмо, отправить его на свой адрес и проверить, что письмо придёт (рис. 5.24).

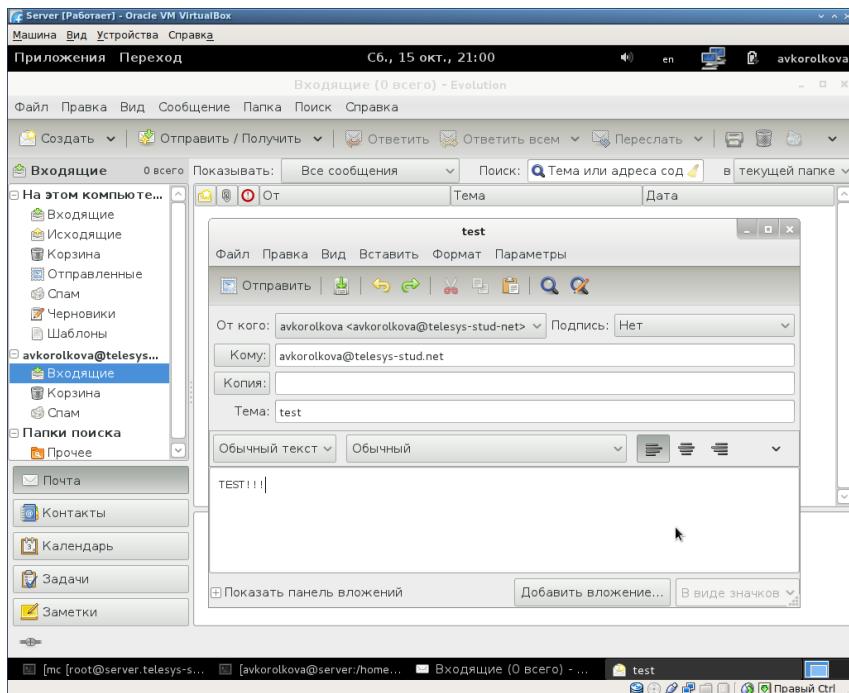


Рис. 5.24. Тестовое письмо

5.3.4. Проверка работы почтового клиента на виртуальной машине Client

В VirtualBox в свойствах виртуальной машины Client в качестве типа сетевого подключения задать «NAT».

Не отключая виртуальную машину Server, запустить виртуальную машину Client. Перейти в привилегированный режим. Установить telnet:

```
yum install telnet
```

Выключить виртуальную машину Client.

В VirtualBox в свойствах виртуальной машины Client в качестве типа сетевого подключения задать «Внутренняя сеть».

Не отключая виртуальную машину Server, запустить виртуальную машину Client. В терминале перейти в привилегированный режим и проверить, что выдается адрес из диапазона 192.168.0.120 – 192.168.0.200. Проверить доступность почтового сервера с помощью команды ping:

```
ping mail.telesys-stud.net
```

Проверить доступность по ip-адресу виртуальной машины Client почтового сервера с помощью команды ping.

Запустить и настроить Evolution (по аналогии с произведенными действиями на Server). Создать тестовое письмо, отправить его на свой адрес и проверить, что письмо придёт.

5.3.5. Работа с почтой с помощью telnet

На виртуальной машине Client в терминале перейти в привилегированный режим и подключиться с помощью протокола telnet к почтовому серверу по протоколу POP3 (через порт 110), ввести логин для подключения и пароль (рис. 5.25):

```
telnet mail.telesys-stud.net 110
user имя_пользователя
pass ваш_пароль
```

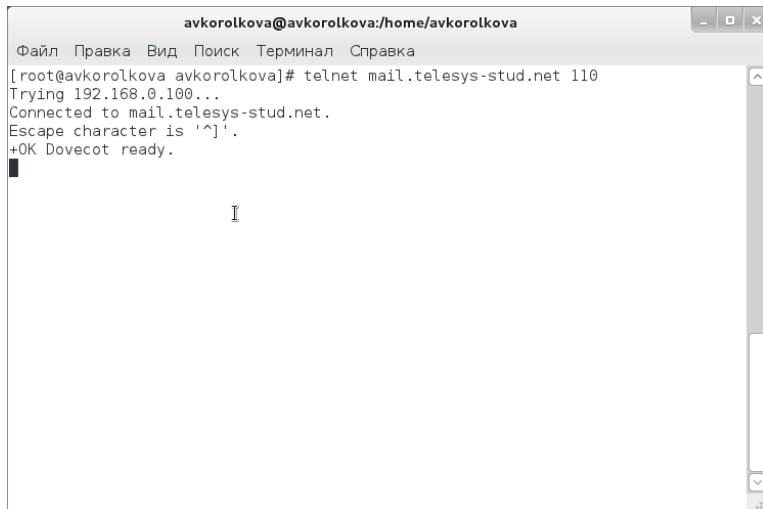
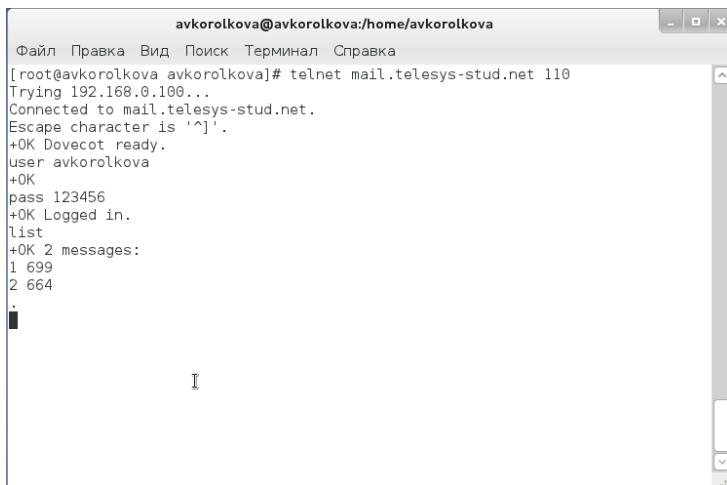


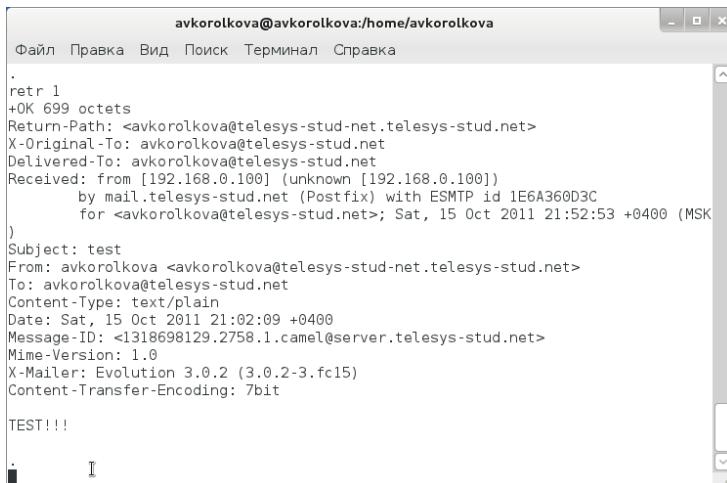
Рис. 5.25. Подключение по telnet к почтовому серверу

С помощью команды **list** получить список писем (рис. 5.26). С помощью команды **retr 1** получить первое письмо из списка (рис. 5.27).



```
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
[root@avkorolkova avkorolkova]# telnet mail.telesys-stud.net 110
Trying 192.168.0.100...
Connected to mail.telesys-stud.net.
Escape character is '^'.
+OK Dovecot ready.
user avkorolkova
+OK
pass 123456
+OK Logged in.
list
+OK 2 messages:
1 699
2 664
```

Рис. 5.26. Просмотр списка писем на сервере

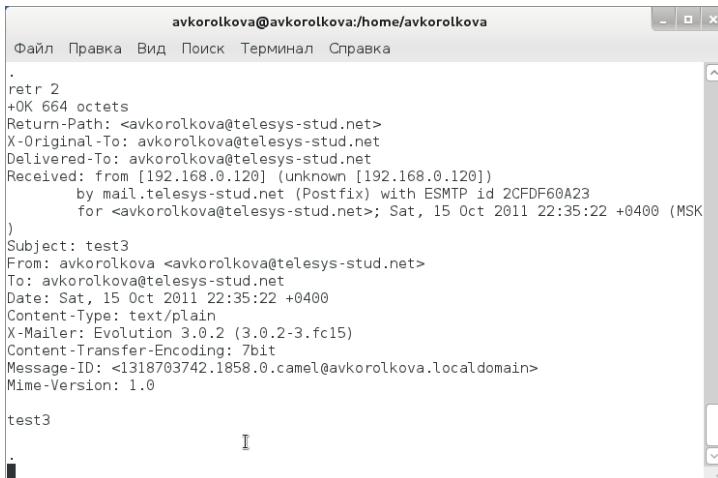


```
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
.
retr 1
+OK 699 octets
Return-Path: <avkorolkova@telesys-stud.net.telesys-stud.net>
X-Original-To: avkorolkova@telesys-stud.net
Delivered-To: avkorolkova@telesys-stud.net
Received: from [192.168.0.100] (unknown [192.168.0.100])
    by mail.telesys-stud.net (Postfix) with ESMTP id 1E6A360D3C
        for <avkorolkova@telesys-stud.net>; Sat, 15 Oct 2011 21:52:53 +0400 (MSK)
)
Subject: test
From: avkorolkova <avkorolkova@telesys-stud.net.telesys-stud.net>
To: avkorolkova@telesys-stud.net
Content-Type: text/plain
Date: Sat, 15 Oct 2011 21:02:09 +0400
Message-ID: <1318698129.2758.1.camel@server.telesys-stud.net>
Mime-Version: 1.0
X-Mailer: Evolution 3.0.2 (3.0.2-3.fc15)
Content-Transfer-Encoding: 7bit

TEST!!!
```

Рис. 5.27. Содержание первого письма

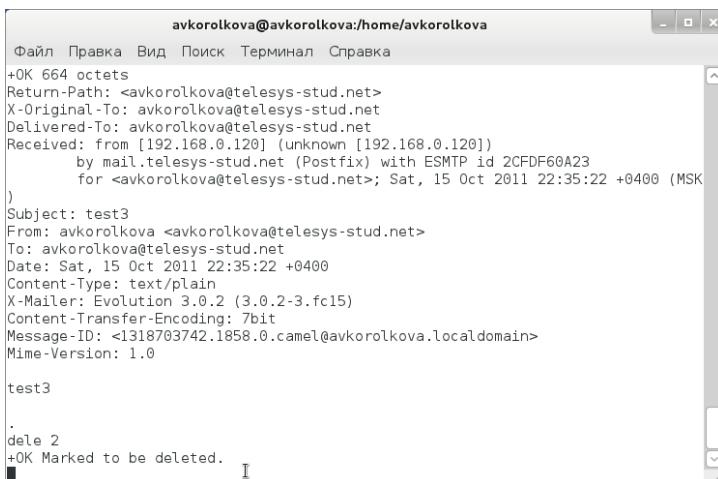
С помощью команды `retr 2` получить второе письмо из списка (рис. 5.28). С помощью команды `delete 2` удалить второе письмо (рис. 5.29). С помощью команды `quit` завершить сеанс работы.



```
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
.
retr 2
+OK 664 octets
Return-Path: <avkorolkova@telesys-stud.net>
X-Original-To: avkorolkova@telesys-stud.net
Delivered-To: avkorolkova@telesys-stud.net
Received: from [192.168.0.120] (unknown [192.168.0.120])
    by mail.telesys-stud.net (Postfix) with ESMTP id 2CFDF60A23
    for <avkorolkova@telesys-stud.net>; Sat, 15 Oct 2011 22:35:22 +0400 (MSK)
)
Subject: test3
From: avkorolkova <avkorolkova@telesys-stud.net>
To: avkorolkova@telesys-stud.net
Date: Sat, 15 Oct 2011 22:35:22 +0400
Content-Type: text/plain
X-Mailer: Evolution 3.0.2 (3.0.2-3.fc15)
Content-Transfer-Encoding: 7bit
Message-ID: <1318703742.1858.0.camel@avkorolkova.localdomain>
Mime-Version: 1.0

test3
```

Рис. 5.28. Содержание второго письма



```
avkorolkova@avkorolkova:/home/avkorolkova
Файл Правка Вид Поиск Терминал Справка
.
+OK 664 octets
Return-Path: <avkorolkova@telesys-stud.net>
X-Original-To: avkorolkova@telesys-stud.net
Delivered-To: avkorolkova@telesys-stud.net
Received: from [192.168.0.120] (unknown [192.168.0.120])
    by mail.telesys-stud.net (Postfix) with ESMTP id 2CFDF60A23
    for <avkorolkova@telesys-stud.net>; Sat, 15 Oct 2011 22:35:22 +0400 (MSK)
)
Subject: test3
From: avkorolkova <avkorolkova@telesys-stud.net>
To: avkorolkova@telesys-stud.net
Date: Sat, 15 Oct 2011 22:35:22 +0400
Content-Type: text/plain
X-Mailer: Evolution 3.0.2 (3.0.2-3.fc15)
Content-Transfer-Encoding: 7bit
Message-ID: <1318703742.1858.0.camel@avkorolkova.localdomain>
Mime-Version: 1.0

test3

.
+OK Marked to be deleted.
```

Рис. 5.29. Удаление второго письма

5.4. Контрольные вопросы

1. За что отвечает протокол SMTP?
2. За что отвечает протокол IMAP?
3. За что отвечает протокол POP3?
4. В чём назначение dovecot? В каких файлах обычно находятся настройки его работы? За что отвечает каждый из файлов?
5. В чём назначение postfix? В каких файлах обычно находятся настройки его работы? За что отвечает каждый из файлов?
6. Приведите пример заголовка письма с пояснениями его полей.
7. Что такое telnet? В чём его назначение?
8. Приведите синтаксис команды telnet.
9. Приведите примеры использования telnet для соединения с почтовыми серверами (по протоколам SMTP, IMAP, POP3).
10. Приведите примеры основных команд telnet для работы с почтовыми протоколами.

Лабораторная работа № 6.

Программирование клиентских и серверных сценариев WEB-приложений. HTML, технология PHP и СУБД MySQL

6.1. Цель работы

Приобретение практических навыков по программированию сценариев WEB-приложений.

6.2. Информация, необходимая для начала работы

Перед началом выполнения работы:

- должна быть выполнена лабораторная работа по установке и конфигурированию операционной системы на виртуальной машине Base;
- должна быть выполнена лабораторная работа по установке и конфигурированию DNS-сервера;
- должна быть выполнена лабораторная работа по установке и конфигурированию DHCP-сервера;
- должна быть выполнена лабораторная работа по установке и конфигурированию HTTP-сервера.

В рамках лабораторной работы должно быть разработано Web-приложение «Успеваемость студентов», состоящее из двух Web-страниц:

- на первой странице «Default» пользователь должен выбирать факультет, специальность и студенческую группу из соответствующих раскрывающихся списков (рис. 6.1);
- передача выбранных пользователем параметров на сервер выполняется нажатием на ссылку «Просмотр»;
- в результате выполняется переход на страницу «Browse», где отображается информация об успеваемости выбранной студенческой группы (рис. 6.2);
- для реализации Web-приложения используются язык гипертекстовой разметки HTML, таблицы стилей CSS;
- программирование серверных сценариев выполняется с помощью технологий PHP, MySQL;
- база данных реализуется с помощью СУБД MySQL;

- программный код Transact-SQL создания базы данных и внесения в нее тестовой информации приведен в приложении к работе и должен быть запущен на выполнение из соответствующей визуальной среды.

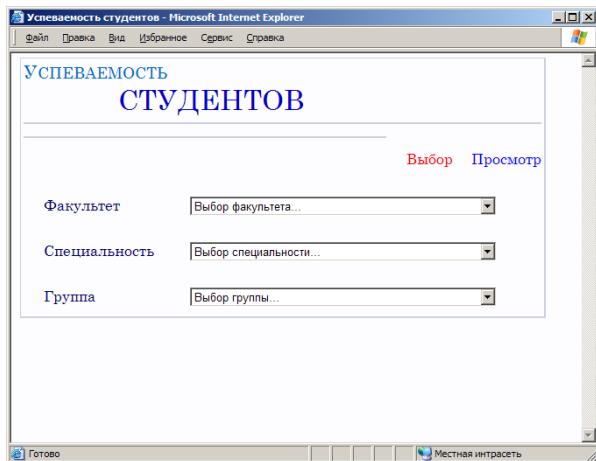


Рис. 6.1. Страница «Default» Web-приложения «Успеваемость студентов»

Студент	Дисциплина	Дата	Оценка
Иванов И.И.	Математика	06.12.2009	5
Петров П.П.	Математика	06.12.2009	4
Сидоров С.С.	Математика	06.12.2009	3
Иванов И.И.	Физика	08.12.2009	3
Петров П.П.	Физика	08.12.2009	4
Сидоров С.С.	Физика	08.12.2009	5
Иванов И.И.	Химия	10.12.2009	5
Петров П.П.	Химия	10.12.2009	4
Сидоров С.С.	Химия	10.12.2009	3

Рис. 6.2. Страница «Browse» Web-приложения «Успеваемость студентов»

6.3. Теоретическая часть

6.3.1. Web-приложения. Основные понятия

Web-приложения представляют собой особый тип программ, построенных по архитектуре «клиент-сервер» (Web-приложение находится и выполняется на сервере, клиент при этом получает только результаты работы).

Код Web-приложения исполняется на сервере, в качестве интерфейса системы выступает браузер, а в качестве среды, посредством которой происходит обмен данными, — Интернет.

Запрос — обращение браузера к серверу.

Отклик — ответ сервера браузеру.

В общем виде функционирование Web-приложений определяется работой трех основных компонентов:

- Web-сервера;
- Web-браузера;
- HTTP-протокола.

Web-сервер — любое управляющее компьютером программное обеспечение, которое дает ему возможность выполнять запросы на документы или другие данные.

Web-серверы:

- способны обрабатывать как статические HTML-файлы, так и запросы страниц, содержащих код, который выполняется на сервере;
- поддерживают сохранение данных запросов, что позволяет связывать Web-страницы в Web-приложении;
- поддерживают постоянные соединения с Web-браузерами, непрерывно ожидая дополнительные запросы

Web-браузер — независимое от платформы средство отображения Web-страниц, написанных на языке HTML.

Каждый запрос Web-страницы, отправленный Web-серверу, приводит к очистке окна Web-браузера и отображению новой Web-страницы.

Рендеринг — визуализация HTML-кода в окне клиентского браузера.

Для запроса Web-страниц у Web-браузера и возврата откликов браузеру применяется протокол HTTP. HTTP-сообщения обычно передаются между Web-сервером и Web-браузером через порт 80, а при использовании защищенного HTTP (HTTPS) — через порт 443.

Работа по протоколу HTTP:

- программа-клиент (браузер) устанавливает TCP-соединение с сервером (стандартный номер порта — 80) и выдает ему HTTP-запрос;
- сервер обрабатывает этот запрос и выдает HTTP-ответ клиенту.

Web-сайт считается **статическим**, если он состоит просто из набора HTML-файлов, ссылающихся друг на друга.

Динамический Web-сайт в ответ на запросы Web-браузера пользователя может не просто выдавать заранее подготовленные страницы HTML, а создавать страницы «динамически», на основе записей в базе данных. Такой подход к созданию сайта позволяет также отделить его содержимое от внешнего вида, появляется возможность быстро менять оформление сайта или даже предлагать пользователям несколько вариантов оформления на выбор.

Сценарий — специальная программа, написанная на особом языке программирования (скриптовом языке, языке сценариев), расширяющая функциональные возможности Web-страницы по сравнению с обычным HTML-документом.

Серверные сценарии хранятся и работают на сервере и передают в браузер пользователя только результат своей работы.

Клиентские сценарии загружаются в браузер пользователя вместе с текстом Web-страницы и исполняются в самом браузере.

Web-страница, которая передается в браузер пользователя, в самом общем виде может быть представлена как множество элементов управления, размещенных на пустой Web-странице.

Серверные элементы привязаны не к разметке, а к функциональности, которую нужно обеспечить; программируются с помощью серверного кода, обрабатывающего события элементов. В промежутках между обращениями к серверу серверные элементы управления автоматически поддерживают (сохраняют) свое состояние.

Применение клиентских элементов управления целесообразно, если для них требуется собственный клиентский сценарий, связанный с событиями элемента управления.

6.3.2. Язык PHP

Язык PHP — язык сценариев, которые внедряются в страницы HTML для исполнения на стороне сервера.

Как правило, для отделения PHP-кода от кода HTML используются символы `<?php код ?>`, реже — `<? код ?>`, а также

```
<script language="php">код</script>
```

Web-страница PHP имеет расширение `.php` и состоит из трех разделов:

- *директивы страницы* — используются для настройки и определяют, как должна обрабатываться страница. Например, так можно задать подключение внешних файлов;
- *код* — программный код, реализующий выполняемые на сервере операции;
- *разметка страницы* — это HTML-код страницы, включающий тег `body` и его содержимое.

Для разработки PHP-страниц можно использовать как текстовые редакторы, так и более «продвинутые» визуальные HTML-редакторы.

6.3.3. СУБД MySQL

Приложения PHP, использующие для хранения информации базу данных, как правило, работают с системой **MySQL**. Для работы с MySQL PHP располагает достаточно широким набором функций — от установки соединения с базой данных до извлечения отдельных значений, полученных в результате выполнения запроса (рис. 6.1).

Таблица 6.1
Функции взаимодействия с базой данных MySQL в PHP-приложениях

Название класса	Описание класса
<code>mysql_connect</code>	Соединение с источником данных
<code>mysql_select_db</code>	Выбор базы данных для дальнейшей работы
<code>mysql_query</code>	Отправка SQL-запросов серверу
<code>mysql_fetch_array</code>	Представление результата запроса в виде ассоциативного массива
<code>mysql_result</code>	Доступ к отдельному полю записи результата запроса

6.3.4. Дополнительные справочные материалы

- Справочники по HTML:
<http://html.manual.ru/>, <http://htmlbook.ru/html>
- Справочники и самоучитель по CSS:
<http://css.manual.ru/>, <http://htmlbook.ru/css>,
<http://htmlbook.ru/samcss>
- Руководство по PHP: <http://www.php.ru/manual/>
- Справочное руководство по MySQL:
<http://www.mysql.ru/docs/man/>
- Вступление в PHP и MySQL:
<http://www.mysql.ru/docs/tkachenko/>
- Илья Кантор. Подключение и выполнение javascript:
<http://javascript.ru/tutorial/foundation/start>
- Учебник JavaScript: <http://learn.javascript.ru/>
- On-line редактор JavaScript и HTML:
<http://learn.javascript.ru/play>

6.4. Порядок выполнения работы

6.4.1. Разработка Web-приложения

1. В своей домашней директории создать каталог, в котором будут размещены файлы Web-приложения, и назвать его `lab06-php`.
2. Создать файл и сохранить его под именем `index.php`.
3. Открыть файл `index.php` в редакторе (например, `geany` или `bluefish`).
4. Ввести открывающий и закрывающий теги корневого элемента HTML и элемента `head`:

```
<html>
  <head>
    </head>
</html>
```

5. Задать заголовок страницы `index.php`. Для этого между открывающим и закрывающим тегами элемента `head` ввести следующий код:

```
<title>Успеваемость студентов</title>
```

6. Создать CSS-таблицы стилей:

- (a) В каталоге с вашим веб-приложением создать каталог `images`, в нём создать новый файл `style.css`. Открыть его на редактирование.
- (b) Определить стиль представления на HTML-странице горизонтальной линии:

```
.line {
  height: 2px;
  border: #cbd1e0 2px solid
}
```

- (c) Создать стиль представления на HTML-странице основной таблицы:

```
.pageborder {
  border-right: #cbd1e0 2px solid;
  border-top: #cbd1e0 1px solid;
  background: none transparent scroll repeat 0% 0%;
  border-left: #cbd1e0 1px solid;
  color: #000000;
  border-bottom: #cbd1e0 2px solid
}
```

- (d) Построить стиль представления на HTML-странице текста «Успеваемость» в заголовке страницы:

```
.text1 {
    font-size: 16pt;
    color: #0066cc;
    font-variant: small-caps;
    font-family: 'Century Schoolbook',
}
```

- (e) Задать стиль представления на HTML-странице текста «студентов» в заголовке страницы:

```
.text2 {
    font-size: 24pt;
    text-transform: uppercase;
    color: #0000cc;
    font-family: 'Century Schoolbook',
}
```

- (f) Задать стиль представления текста `.text3` на основе стиля `.text1` с размером шрифта `13pt`.

- (g) Задать стиль представления таблицы `.headborder` на основе стиля `.pageborder` с шириной правой и нижней границ, равной `2px` и `1px` соответственно:

```
.headborder {
    border-right: #cbd1e0 2px solid;
    border-top: #cbd1e0 1px solid;
    background: none transparent scroll repeat 0% 0%;
    border-left: #cbd1e0 1px solid;
    color: #000000;
    border-bottom: #cbd1e0 2px solid
}
```

- (h) Аналогичным образом задать стиль представления таблицы `.textborder` на основе стиля `.headborder` без верхней границы.

- (i) Задать стиль представления текста `.headtext` на основе стиля `.text1` без малых прописных букв и с размером шрифта `13pt`.

- (j) Задать стиль представления текста `.tabletext` с размером шрифта `10pt`.

- (k) Сохранить файл `style.css`

7. Задать ссылку на таблицу стилей в файле `index.php`. Для этого между открывающим и закрывающим тегами элемента `head` ввести следующий код:

```
<link href="images/style.css"  
      rel="stylesheet" type="text/css"/>
```

8. Задать тип содержимого для корректного отображения и передачи текстовых данных, содержащих кириллические символы. Для этого между открывающим и закрывающим тегами элемента **head** ввести следующий код:

```
<meta http-equiv="Content-Type"  
      content="text/html; charset=utf8">
```

9. Создать «тело» документа. Для этого после закрывающего тега элемента **head** ввести следующий код:

```
<body>  
</body>
```

10. Создать форму для отправки данных серверу.

- (а) Создать серверный элемент **form**. Для этого между открывающим и закрывающим тегами элемента **body** ввести следующий код:

```
<form>  
</form>
```

- (б) Задать имя формы. Для этого в качестве атрибута элемента **form** указать:

```
name="form1"
```

- (с) Указать метод **POST** формы и страницу, на которую будут направлены данные формы. Для этого в элемент **form** ввести следующие атрибуты:

```
action="browse.php" method="post"
```

11. Сформировать структуру Web-страницы **index.php** в форме одноколончной таблицы из двух строк.

- (а) Внутри элемента **form** создать «каркас» таблицы шириной **600px**, к которой применён стиль **pageborder**:

```
<table class="pageborder" border="0" width="600">  
</table >
```

(b) Создать «заголовочную» строку «Успеваемость студентов».

i. Создать новую строку таблицы. Внутри строки задать единственный столбец, ширина которого совпадает с шириной самой таблицы (ширина столбца – 100%). Для этого между открывающим и закрывающим тегами элемента **table** ввести следующий код:

```
<tr>
  <td width="100%">
    </td>
</tr>
```

ii. В ячейке таблицы (внутри элемента **td**) ввести текстовую строку «Успеваемость» и применить к ней стиль **text1**:

```
<span class="text1">Успеваемость</span>
```

iii. Там же ввести новую текстовую строку «студентов» с использованием стиля **text2**:

```
<br/>
<span class="text2">
  &ampnbsp &ampnbsp &ampnbsp &ampnbsp &ampnbsp &ampnbsp
  студентов
</span>
```

iv. Создать две горизонтальные линии и применить к ним стиль **line**. Ширина второй линии – 70%, выравнивание – по левому краю:

```
<br/>
<hr class="line" />
<hr class="line" width="70%" align="left" />
```

12. Создать управляющие элементы отправки запросов формы серверу.

(a) Создать новую строку таблицы. Внутри строки задать единственный столбец, ширина которого совпадает с шириной самой таблицы (ширина столбца – 100%), а текст внутри столбца имеет размер 17 px и выровнен по правому краю. Для этого между открывающим и закрывающим тегами элемента **table** из п. 11(a) после строки из п. 11(b)i ввести следующий код:

```
<tr>
<td width="100%" style="height: 17px; text-align: right;" >
</td>
</tr>
```

- (б) Внутри тега **td** создать текстовый фрагмент «Выбор», к которому применен следующий стиль: цвет текста — RGB-код #ff0000, размер шрифта — 13pt, шрифт — Century Schoolbook:

```
<span style="color: #ff0000; font-size:13pt;
font-family: 'Century Schoolbook'">
    Выбор &nbsp;&nbsp;
</span>
```

- (с) Затем создать текстовый фрагмент «Просмотр» внутри элемента управления А (гиперссылка), к которому применен следующий стиль: цвет текста — RGB-код #669999, размер шрифта — 13pt, шрифт — Century Schoolbook. В качестве реакции на выбор гиперссылки определить отправку формы на сервер:

```
<span style="color: #669999; font-size:13pt;
font-family: 'Century Schoolbook'">
    <a href="javascript:document.form1.submit();"
        style="text-decoration: none">Просмотр
    </a>
</span>
```

13. Создать столбцы — параметры запроса.

- (а) Создать новую двухколоночную таблицу в новой (третьей по счёту) строке таблицы из п. 11(а). Таблица должна быть выровнена по центру и иметь ширину 550 px.

```
<tr>
<td><br/>
    <table align="center" width="550">
        </table>
    </td>
</tr>
```

- (б) В созданной внутренней таблице создать строку для запроса параметра «Факультет».

- i. Создание новой строки таблицы из п. 11(a) шириной 50 px с верхним вертикальным выравниванием:

```
<tr height="50px" valign="top">
</tr>
```

- ii. Создать первый столбец строки из п. 13(b)i. К тексту «Факультет» столбца применить стиль **text3**:

```
<td style="width: 163px">
  <span class="text3">
    Факультет
  </span>
</td>
```

- iii. Создать второй столбец строки из п. 13(b)i с элементом управления **select** (раскрывающийся список) шириной 350 px с именем **faculty**:

```
<td>
  <select name="faculty" style="width: 350px">
    <option selected="selected">
      Выбор факультета...
    </option>
  </select>
</td>
```

- (c) Создать строки для запроса параметров «Специальность» и «Группа» (аналогично строке из п. 13(b)). Имена соответствующих элементов управления — **spec** и **gr**.

14. Заполнить раскрывающиеся списки **select** информацией о факультетах, специальностях и группах, полученной из базы данных.

- (a) Установить соединение с базой данных MySQL. Для этого нужно создать сценарий PHP и разместить его, например, до всего HTML-кода страницы **index.php**:

```
<?php
$link = @mysql_connect("localhost", "root") or
  die("Невозможно соединиться с сервером");
$db=@mysql_select_db("university") or
  die("Нет такой базы данных");
?>
```

- (b) Затем установить кодировку utf8 для корректной передачи и получения данных из базы данных. Для этого в сценарий PHP из предыдущего пункта добавить следующий код:

```
@mysql_query("SET SESSION
    Character_set_results = utf8;");
@mysql_query("SET SESSION
    Character_set_client = utf8;");
@mysql_query("SET SESSION
    Character_set_results = utf8;");
@mysql_query("SET SESSION
    Collation_connection = utf8_general_ci;");
@mysql_query("SET SESSION
    Character_set_connection = utf8;");
```

- (c) Затем выполнить запрос к таблице **faculty** (Факультет) для получения данных о факультетах из базы данных. Для этого добавить в сценарий PHP следующий код:

```
$f_query="select * from 'faculty'";
$f=mysql_query($f_query);
$f_count = mysql_num_rows($f);
```

- (d) Аналогичным образом выполнить запросы информации о специальностях и группах (т.е. добавить код в сценарий PHP, указав имена соответствующих элементов управления — **spec** и **gr** и выполнив запрос к соответствующим таблицам базы данных: **speciality** и **groups**).

- (e) Заполнить раскрывающийся список **faculty** информацией о факультетах. Для этого в соответствующем элементе **select** построить цикл по результату выборки из базы данных. После закрывающего тега элемента **option** родительского элемента **select** из п. 13(b)iii ввести следующий код:

```
<?php
    for($i=0;$i<$f_count;$i++)
    {
?
<option value="
    <?php
        echo mysql_result($f,$i,'id')
    ?>
">
    <?php
        echo mysql_result($f,$i,'name')
    ?>
</option>
<?php
    }
?>
```

- (f) Аналогичным образом заполнить списки специальностей и групп.
15. Разработать клиентский сценарий проверки введенных пользователем данных `check()`.

- (a) Определить обработчик события отправки формы на сервер (`submit`). Для этого в открывающем теге элемента `form` из п. 10(a)) задать атрибут:

```
onsubmit="javascript:check()"
```

- (b) Создать клиентский сценарий `check()`, введя следующий код между открывающим и закрывающим тегами элемента `head`:

```
<script language="javascript">
    function check()
    {
        f = document.getElementById('faculty').value;
        s = document.getElementById('spec').value;
        g = document.getElementById('gr').value;
        if(f=='0' || s=='0' || g=='0')
        {
            alert('Один из параметров не был выбран!');
        }
    }
</script>
```

16. В папку `lab06-php`, содержащую файлы Web-приложения, добавить новый файл с именем `browse.php`.
17. Открыть файл `browse.php` в редакторе, ввести корневой HTML-элемент, элементы `head` и `title` (см. п. 4, 5), присоединить стилевой файл CSS (см. п. 7), элемент `body` (см. п. 9).
18. Сформировать заголовок и основную структуру таблицы `browse.php`, аналогичную странице `index.php` (см.пп. 11–12).
19. Создать новую таблицу во второй строке таблицы из п. 18 для вывода информации об успеваемости студентов выбранной группы.
- (a) Создать «каркас» таблицы. Для этого между открывающим и закрывающим тегами элемента `td` ввести теги элемента `table`. Задать ширину таблицы в 500 px, выравнивание по центру и использование стиля `headborder`:
- ```
<table width="500" align="center" class="headborder">
</table>
```
- (b) Создать строку таблицы с четырьмя столбцами–заголовками информации об успеваемости. Первые два столбца должны занимать по 30% от общей ширины таблицы, а последние два — по 20%. К ячейкам таблицы применить стиль представления текста `headtext`:

```

<tr>
 <td width="30%">
 Студент
 </td>
 <td width="30%" align="center">
 Дисциплина
 </td>
 <td width="20%" align="center">
 Дата
 </td>
 <td width="20%" align="center">
 Оценка
 </td>
</tr>

```

- (c) Аналогичным образом создать таблицу того же формата, что и в п. 19, к которой применён стиль `textborder`. Задать одну строку, аналогичную строке из п. 19(b), ячейки которой не содержат никаких данных.
20. Выполнить запрос к базе данных на получение информации об успеваемости студенческой группы.
- Установить соединение с базой данных и кодировку запросов и ответов (см. п. 14) в сценарии PHP.
  - Создать переменную `gr` и записать в нее значения параметра `gr`, переданного со страницы `index.php`:  
`$gr = $_POST['gr'];`
  - Создать переменную `usp_query` для хранения текста запроса к базе данных, для извлечения информации об успеваемости студентов заданной переменной `gr` группы:  
`$usp_query="SELECT 'fio', 'subject', 'ocenka', 'data'`  
`FROM 'uspev', 'student'`  
`WHERE 'uspev'. 'student' = 'student'. 'zk' AND 'gr' = '$gr';`
  - Выполнить запрос к базе данных:  
`$usp=mysql_query($usp_query);`
  - Вычислить количество строк, полученных в результате выполнения запроса:  
`$usp_count = mysql_num_rows($usp);`
21. Создать таблицу для вывода на страницу информации об успеваемости заданной студенческой группы.
- Создать новую таблицу, аналогичную той, что была построена в предыдущем пункте. Применить к таблице стиль `textborder`:  
`<table width="500" align="center" class="textborder">`  
`</table>`
  - Организовать цикл по результатам выборки информации об успеваемости из базы данных. На каждой итерации цикла создавать новую строку:

```

<?php
 for($i=0;$i<$usp_count;$i++)
 {
?
 <tr>
 </tr>
<?php
 }
?

```

- (c) В строке из п. 21(b) задать столбец для отображения информации о фамилии, имени и отчестве (ФИО) студента (первое поле из результата запроса):

```

<td width="30%">

 <?php
 echo mysql_result($usp,$i,0)
 ?>

</td>

```

- (d) Аналогичным образом определить столбцы для представления информации о предметах (второе поле в запросе), датах (четвертое поле) и оценках (третье поле).

22. Проверить работоспособность созданного Web-приложения (раздел 6.4.3).

#### 6.4.2. Содержимое файла с базой данных university.sql

```

DROP DATABASE ‘university’;
CREATE DATABASE ‘university’
DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

CREATE TABLE ‘faculty’ (
 ‘ID’ VARCHAR(7) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
 ‘name’ VARCHAR(27) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
 PRIMARY KEY (‘ID’)
) ENGINE = MYISAM ;

CREATE TABLE ‘speciality’ (
 ‘ID’ VARCHAR(7) NOT NULL ,
 ‘name’ VARCHAR(27) NOT NULL ,
 ‘faculty’ VARCHAR(7) NOT NULL ,
 PRIMARY KEY (‘ID’) ,
 FOREIGN KEY (‘faculty’) REFERENCES Faculty(‘ID’)
) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_general_ci;

CREATE TABLE ‘groups’ (
 ‘ID’ INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
 ‘number’ INT NOT NULL ,

```

```

'spec' VARCHAR(7) NOT NULL ,
FOREIGN KEY ('spec') REFERENCES speciality('ID')
) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_general_ci;

CREATE TABLE 'student' (
 'zk' INT NOT NULL PRIMARY KEY ,
 'fio' VARCHAR(27) NOT NULL ,
 'gr' INT NOT NULL ,
 FOREIGN KEY ('gr') REFERENCES groups('ID')
) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_general_ci;

CREATE TABLE 'uspev' (
 'ID' INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
 'student' INT NOT NULL ,
 'subject' VARCHAR(25) NOT NULL ,
 'ocenka' INT NOT NULL ,
 'data' DATE NOT NULL ,
 FOREIGN KEY ('student') REFERENCES student('zk')
) ENGINE = MYISAM CHARACTER SET utf8 COLLATE utf8_general_ci;

INSERT INTO 'faculty' ('ID' , 'name')
VALUES (
 'ФМиЕН', 'физ.мат.' ,
), (
 'ИИЖ', 'инженерный' ,
);

INSERT INTO 'speciality' ('ID' , 'name' , 'faculty')
VALUES (
 'АД', 'авиац. двиг. и энерг. уст.' , 'ИИЖ' ,
), (
 'ЭАТ', 'экспл. авиац. техн.' , 'ИИЖ' ,
);

INSERT INTO 'speciality' ('ID' , 'name' , 'faculty')
VALUES (
 'HK', 'Математика и компьютерные науки' , 'ФМиЕН' ,
), (
 'НИ', 'Информационные технологии' , 'ФМиЕН' ,
);

INSERT INTO 'groups' ('ID' , 'number' , 'spec')
VALUES (
 '01', '01', 'АД' ,
), (
 '02', '02', 'АД' ,
);

INSERT INTO 'groups' ('ID' , 'number' , 'spec')
VALUES (
 '03', '03', 'ЭАТ' ,
), (
 '04', '04', 'ЭАТ' ,
);

INSERT INTO 'groups' ('ID' , 'number' , 'spec')
VALUES (

```

```
'05', '05', 'HK'
), (
'06', '06', 'HK'
);

INSERT INTO `groups` (`ID` , `number` , `spec`)
VALUES (
'07', '07', 'НИ'
), (
'08', '08', 'НИ'
);

INSERT INTO `student` (`zk` , `fio` , `gr`)
VALUES (
'111111', 'Иванов И.И.', '07'
), (
'222222', 'Петров П.П.', '07'
);
INSERT INTO `student` (`zk` , `fio` , `gr`)
VALUES (
'333333', 'Сидоров С.С.', '07'
);

INSERT INTO `uspev` (`ID` , `student` , `subject` , `ocenka` , `data`)
VALUES (
NULL , '111111', 'Математика', '5', '2012-06-12'
), (
NULL , '222222', 'Математика', '4', '2012-06-12'
);

INSERT INTO `uspev` (`ID` , `student` , `subject` , `ocenka` , `data`)
VALUES (
NULL , '333333', 'Математика', '3', '2012-06-12'
), (
NULL , '111111', 'Физика', '3', '2012-08-12'
);

INSERT INTO `uspev` (`ID` , `student` , `subject` , `ocenka` , `data`)
VALUES (
NULL , '222222', 'Физика', '4', '2012-08-12'
), (
NULL , '333333', 'Физика', '5', '2012-08-12'
);

INSERT INTO `uspev` (`ID` , `student` , `subject` , `ocenka` , `data`)
VALUES (
NULL , '111111', 'Информатика', '5', '2012-10-12'
), (
NULL , '222222', 'Информатика', '4', '2012-10-12'
);

INSERT INTO `uspev` (`ID` , `student` , `subject` , `ocenka` , `data`)
VALUES (
NULL , '333333', 'Информатика', '3', '2012-10-12'
);
```

### 6.4.3. Проверка работоспособности Web-приложения

- В свойствах виртуальной машины Server в качестве типа сетевого подключения задать «NAT».
- Запустить виртуальную машину Server.
- Подключить сетевое соединение «Wired connection2».
- На виртуальной машине Server запустить терминал и перейти с помощью команды `su` в привилегированный режим.
- Установить `php`, `mysql`, `mysql-server`, `php-mysql`.
- По протоколу `ssh` из-под виртуальной машины Server подсоединиться к своему домашнему каталогу на обычном компьютере. Для этого на виртуальной машине Server в `mc` выбрать «Правая панель», Shell-соединение. В строке набрать

```
имя_пользователя@ip-адрес
```

где `имя_пользователя` — ваш логин в ДК, `ip-адрес` — ip-адрес компьютера в ДК (посмотреть командой `/sbin/ifconfig`).

- на виртуальной машине Server на «Левой панели» `mc` перейти в каталог `/var/www/html/`.
- Скопировать разработанное вами Web-приложение на виртуальной машине Server.
- На «Правой панели» `mc` перейти в каталог `/afs/dk.sci.pfu.edu.ru/common/files/MySQL`.
- Скопировать оттуда на виртуальную машину Server каталог с базой данных MySQL.
- Выйти из `mc`. Выключить виртуальную машину Server.
- В свойствах виртуальной машины Server в качестве типа сетевого подключения задать «Внутренняя сеть».
- Запустить виртуальную машину Server и подключить сетевое соединение «p2p1».
- На виртуальной машине Server запустить терминал и перейти с помощью команды `su` в привилегированный режим.
- Запустить DNS-сервер:

```
systemctl start named.service
```

- Запустить DHCP-сервер:

```
systemctl start dhcpcd.service
```

- Стартовать `mysql`, `httpd`:

```
systemctl start mysqld.service
```

```
systemctl start httpd.service
```

- Создать базу university, перейдя предварительно в директорию с файлом university.sql:

```
mysqladmin -u root -p create university
mysql -u root -p university < university.sql
```

- Запустить Firefox и в качестве адреса сайта задать <http://localhost>, проверить разработанное приложение.
- Если на виртуальной машине Server всё работает, то проверить на виртуальной машине Client, запустив на ней Firefox и в качестве адреса сайта указав <http://www.telesys-stud.net>.

# Лабораторная работа № 7. Форматирование и преобразование XML- документа с помощью XSL. XSLT- преобразование XML-документа

## 7.1. Цель работы

Знакомство с методами форматирования и преобразования XML-документов на основе XSLT-преобразований.

## 7.2. Теоретический материал

*XML (eXtensible Markup Language)* предназначен не для форматирования и отображения данных, а для описания данных. В отличие от HTML, в XML в общем случае теги не определены. При создании XML-файлов автор создает свои собственные теги.

Тело документа XML состоит из элементов разметки (*markup*) и непосредственно содержимого документа — данных (*content*). XML-теги предназначены для определения элементов документа, их атрибутов и других конструкций языка.

Любой XML-документ должен всегда начинаться с инструкции `<?xml?>`, внутри которой можно задать номер версии языка и другие параметры, необходимые программе-анализатору в процессе разбора документа.

В общем случае XML-документы должны удовлетворять следующим требованиям:

- в заголовке документа помещается объявление XML, в котором указывается язык разметки документа, номер его версии и дополнительная информация;
- каждый открывающий тег, определяющий некоторую область данных в документе, обязательно должен иметь закрывающий тег;
- в XML учитывается регистр символов;
- все значения атрибутов, используемых в определении тегов, должны быть заключены в кавычки;
- вложенность тэгов в XML строго контролируется, поэтому необходимо следить за порядком следования открывающих и закрывающих тегов;
- вся информация, располагающаяся между начальным и конечным тегами, рассматривается в XML как данные, поэтому учитываются все символы форматирования (пробелы, переводы строк, табуляции не игнорируются, как в HTML).

Для того чтобы использовать данные, определяемые элементами XML, например, отображать их на экране пользователя, необходимо использовать программу-анализатор (*XML-парсер браузера*), *стилевые таблицы* или *сценарий*, написанный на скриптовом языке. Например, для преобразования XML-документов можно использовать методы, реализованные Java Script, XSL (XSLT/XPath).

XSLT используется для преобразования XML-документа в другой XML-документ либо в документ другого распознаваемого браузерами типа, например HTML или XHTML.

С помощью XSLT можно добавлять или удалять элементы и атрибуты в результирующем документе. Также возможна перегруппировка и сортировка элементов, фильтрация элементов при отображении и многое другое. Фактически XSLT преобразует исходное XML-дерево в результирующее XML-дерево. XSLT использует XPath для поиска информации в XML-документе, т.е. XPath является инструментом навигации по элементам и атрибутам XML-документов.

В процессе преобразования XSLT использует XPath для поиска частей исходного документа, соответствующих одному или более заданным шаблонам. Когда соответствие найдено XSLT преобразует найденную часть исходного документа в соответствующую часть результирующего документа. XSLT является W3C рекомендацией с ноября 1999 года. Подавляющее большинство браузеров имеют поддержку XML и XSLT.

### 7.3. Практическое задание: стилевые таблицы XSL и XML

Создайте простой XML-файл ex01.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<tutorial>
 <title>"Путешествие дилетантов"</title>
 <author>Булат Окуджава</author>
</tutorial>
```

Откройте этот файл в браузере. Вы увидите только код файла ex01.xml или сообщение об ошибке. Необходимо преобразовать XML-файл, добавив шаблон преобразования — XSL-файл.

Создайте XSL-файл ex01.xsl со следующим содержанием:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:template match="/">
```

```

<html>
 <body>
 <p><xsl:value-of select="//title"/></p>
 <p><xsl:value-of select="//author"/></p>
 </body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Первая строка файла содержит тег элемента `xsl:stylesheet`. Атрибуты элемента — номер версии и ссылка на пространство имен. Эти атрибуты элемента `xsl:stylesheet` являются обязательными. Пространство имен — это все имена элементов и их атрибутов, которые могут использоваться в XSL-файле. Для XSL-файлов ссылка на пространство имен является стандартной. Заметим, что XSL-файл является одной из разновидностей XML-файлов. Он не содержит пользовательских данных, но формат его тот же самый. Файл содержит элемент верхнего уровня `xsl:stylesheet`, а далее идет дерево правил преобразования.

Элемент `stylesheet` содержит три шаблона, каждый из которых вложен в элемент `template`.

У элемента `template` есть атрибут `match`, значением которого является образец (`pattern`) в форме выражения XPath. С ним сравнивается узел дерева, к которому применяется шаблон. В образце можно записать только путь, причём находясь в каком-то узле, можно «увидеть» кроме него самого только его атрибуты и узлы-потомки.

Элемент `xsl:value-of` вычисляет выражение, записанное в его обязательном атрибуте `select`, и преобразует его в строку.

XML-файл `ex01.xml` модифицируйте следующим образом:

```

<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type='text/xsl'
 href="ex01.xsl" tppabs="ex01.xsl"?>
<NewDataSet>
 <tutorial>
 <title>Путешествие дилетантов</title>
 <author>Булат Окуджава</author>
 </tutorial>
</NewDataSet>

```

Первая строка информирует браузер о том, что файл имеет формат XML. Атрибут `version` является обязательным. Атрибут `encoding` является не обязательным, но если в тексте есть русские буквы, то необходимо вставить этот атрибут, в противном случае XML-файл просто не будет обрабатываться.

Вторая строка указывает ссылку на XSL-файл.

Следующие строки — тело XML-файла. Оно состоит из элементов, которые в совокупности образуют древовидную структуру. Элементы идентифицируются тегами и могут быть вложены друг в друга. Элементы могут иметь атрибуты, значения которых тоже могут обращаться в соответствии с шаблоном. На верхнем уровне XML-файла всегда находится только один элемент.

Откройте файл `ex01.xml` в браузере и убедитесь, что отображается только информативная его часть без служебной информации.

Порядок вывода строк определяется только содержанием шаблона преобразования — XSL-файла. При необходимости шаблон можно изменить, не затрагивая основной XML-файл, например:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <html>
 <body>
 <p><xsl:value-of select="//author"/></p>
 <p>
 <xsl:value-of select="//title"/>
 </p>
 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

Создайте XML-файл `ex02.xml`:

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type='text/xsl'
 href="ex02.xsl" tppabs="ex02.xsl"?>
<tutorial>
 <car caption="Автомобиль: " marka="Opel">
 <carInfo number="34c666" color="белый" />
 </car>
</tutorial>
```

В этом файле информация будет храниться в виде значений атрибутов.

Создайте файл `ex02.xsl`:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
```

```
<html>
 <body>
 <P><xsl:value-of select="//car/@caption"/>
 <xsl:value-of select="//car/@marka"/>.
 <xsl:value-of select="//car/carInfo/@number"/>,
 <xsl:value-of select="//carInfo/@color"/>.</P>
 </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Здесь в элементе //car/@Marka имя элемента и имя атрибута разделены парой символов "/@".

Создайте XML-файл ex03.xml:

```
<?xml version="1.0" encoding="utf-8" ?>

<?xml-stylesheet type='text/xsl'
 href="ex03.xsl" tppabs="ex03.xsl"?>
<tutorial>
<avto>
 <cars>
 <carsCaption>Автомобили</carsCaption>
 <carsCaptionMarka>Марка</carsCaptionMarka>
 <carsCaptionNumber>Номер</carsCaptionNumber>
 <carsCaptionColor>Цвет</carsCaptionColor>
 <car>
 <carMarka>Opel</carMarka>
 <carNumber>34c666</carNumber>
 <carColor>Белый</carColor>
 </car>
 <car>
 <carMarka>BMW</carMarka>
 <carNumber>34rt66</carNumber>
 <carColor>Индиго</carColor>
 </car>
 <car>
 <carMarka>Mersedes</carMarka>
 <carNumber>341111</carNumber>
 <carColor>черный</carColor>
 </car>
 <car>
 <carMarka>Mazda</carMarka>
 <carNumber>456678</carNumber>
 <carColor>синий</carColor>
 </car>
```

```

 </cars>
</avto>
</tutorial>
```

Создайте файл ex03.xsl:

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<table border="1" >
 <tr bgcolor="#C1FFC1">
 <td align="center">
 <xsl:value-of select="//carsCaptionMarka"/>
 </td>
 <td align="center">
 <xsl:value-of select="//carsCaptionNumber"/>
 </td>
 <td align="center">
 <xsl:value-of select="//carsCaptionColor"/>
 </td>
 </tr>
 <xsl:for-each select="tutorial/avto/cars/car">
 <tr bgcolor="#11dd33">
 <td><xsl:value-of select="carMarka"/></td>
 <td align="right"><xsl:value-of select="carNumber"/>
 <xsl:value-of select="carNumber/@caption"/>
 </td>
 <td><xsl:value-of select="carColor"/></td>
 </tr>
 </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Элемент `<xsl:for-each select="tutorial/avto/cars/car">` позволяет выбрать и просмотреть все группы информации, полный путь к которым задаётся списком тегов `"tutorial/avto/cars/car"`.

Для того чтобы отсортировать информацию, можно воспользоваться тегом `xsl:sort`, который имеет следующий синтаксис:

```

<xsl:sort
 select="выражение"
```

```
datatype="тип данных"
order={"ascending" | "descending"}
case-order={"upper-first" | "lower-first"} />
```

Атрибут `select` содержит выражение, оцениваемое для каждого узла множества и являющееся критерием сортировки. Атрибут `order` определяет порядок сортировки: значение `"ascending"` («восходящий») означает сортировку по возрастанию и является значением по умолчанию, значение `"descending"` («нисходящий») означает сортировку по убыванию. Если необходимо провести сортировку по нескольким критериям (например, по фамилии и по имени), то в этом случае первым записывается элемент `xsl:sort`, выполняющий сортировку по фамилии, затем элемент `xsl:sort`, выполняющий сортировку по имени.

В рассмотренном выше примере, чтобы отсортировать таблицу по марке автомобиля, нужно после строки

```
<xsl:for-each select="tutorial/avto/cars/car">
```

добавить

```
<xsl:sort order="ascending" select="carMarka"/>
```

Здесь используются два атрибута: атрибут `order` — способ сортировки (по возрастанию (`ascending`) или по убыванию (`descending`)) и атрибут `select` — имя поля, по которому производится сортировка.

Для того чтобы применить фильтр (например, вывести только строки таблицы, в которых номер автомобиля начинается с цифры 3 или больше), нужно в рассмотренный ранее пример внести изменения:

```
...
<xsl:for-each select="tutorial/avto/cars/car">
<xsl:sort order="ascending" select="carMarka"/>
<xsl:if test="carNumber>3">
<tr bgcolor="#11dd33">
 <td><xsl:value-of select="carMarka"/></td>
 <td align="right"><xsl:value-of select="carNumber"/>
 <xsl:value-of select="carNumber/@caption"/>
 </td>
 <td><xsl:value-of select="carColor"/></td>
</tr>
</xsl:if>
</xsl:for-each>
...
```

Следующий пример (ex04.xml, ex04.xsl) демонстрирует «перекрашивание» фона элемента в разные цвета в зависимости от автора и года издания.

```
<?xml version="1.0" encoding="utf-8" ?>
<?xmlstylesheet type='text/xsl' href="ex04.xsl" tppabs="ex04.xsl"?>

<catalog title="Библиотека фантастики">
 <book title="Неукротимая Планета" author="Г.Гаррисон" year="1985" />
 <book title="Крыса из нержавеющей стали" author="Г.Гаррисон" year="1981" />
 <book title="Конные Варвары" author="Г.Гаррисон" year="1990" />
 <book title="Ведмак" author="А.Сапковски" year="2000" />
 <book title="Властелин Колец" author="Р.Толкиен" year="1940" />
 <book title="Хоббит или туда и обратно" author="Р.Толкиен" year="1937" />
 <book title="Бездна голодных глаз" author="Г.Л.Олди" year="1997" />
 <book title="Герой должен быть один" author="Г.Л.Олди" year="2001" />
 <book title="Корпорация 'Бессмертие'" author="Р.Шекли" year="1980" />
</catalog>

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

 <xsl:template match="/">
 <html>
 <body>
 <h3><xsl:value-of select="catalog/@title" /></h3>
 <table width="600" border="0">
 <tr style="color: white; background: black;">
 <th>Title</th>
 <th>Author</th>
 <th>Year</th>
 </tr>
 <xsl:apply-templates />
 </table>
 </body>
 </html>
 </xsl:template>

 <xsl:template match="book">
 <tr>
 <xsl:choose>
 <xsl:when test="@year < 1950">
 <xsl:attribute name="bgcolor">#FFDDDD</xsl:attribute>
 </xsl:when>
 <xsl:when test="@year >= 1950 and @year < 2000">
 <xsl:attribute name="bgcolor">#DDFFDD</xsl:attribute>
 </xsl:when>
 <xsl:when test="@year >= 2000">
 <xsl:attribute name="bgcolor">#DDDDFF</xsl:attribute>
 </xsl:when>
 </xsl:choose>
 <td><xsl:value-of select="@title" /></td>
 <td><xsl:value-of select="@author" /></td>
 <td><xsl:value-of select="@year" /></td>
 </tr>
 </xsl:template>
</xsl:stylesheet>
```

```
<td><xsl:value-of select="@year" /></td>
</tr>
</xsl:template>
</xsl:stylesheet>
```

Элемент `xsl:choose` содержит один или несколько элементов `xsl:when` и необязательный элемент `xsl:otherwise`. При обработке `xsl:choose` процессор поочередно вычисляет выражения, содержащиеся в атрибутах `test` элементов `xsl:when`, приводит их к булевому типу и выполняет содержимое первого (и только первого) элемента `xsl:when`, тестовое выражение которого равно `true`. В случае если ни одно из тестовых выражений не дало «истину» в результате и в элементе `xsl:choose` присутствует `xsl:otherwise`, процессор выполнит его содержимое.

## 7.4. Задание для самостоятельного выполнения

1. Экзаменационная ведомость
  - (а) Создайте XML-документ с описанием экзаменационной ведомости.
  - (б) Создайте для него XSL-файл для XSLT-преобразования исходного XML-документа в HTML-страницу, содержащую заголовок ведомости и таблицу оценок со столбцами: номер, студент, студенческий билет, оценка в баллах от 0 до 100, оценка от «2» до «5».
  - (с) значение в последней колонке таблицы должно получаться динамически в зависимости от значения баллов в предпоследнем столбце по следующему принципу: если число баллов от 0 до 50, то оценка «2»; если число баллов от 51 до 68, то оценка «3»; если число баллов от 69 до 85, то оценка «4»; если число баллов от 86 до 100, то оценка «5».
  - (д) Реализуйте сортировку по возрастанию фамилий студентов.
2. База студентов
  - (а) В базе студентов хранятся: ФИО студента, фотография, факультет, группа, дата поступления, оценки за экзамены с датой их сдачи, ФИО, должностью и кафедрой преподавателя, принявшего экзамен.
  - (б) Студент может пересдавать экзамен несколько раз разным преподавателям или одному и тому же, но при этом должность преподавателя может к этому времени измениться.
3. Расписание занятий
  - (а) Расписание занятий у каждой группы свое. Для каждого дня недели расписание разное. При этом некоторые дни недели могут быть свободны. Кроме того, расписание на четной и нечетной неделе может отличаться.

- (b) В расписании указывается название предмета, тип (лекция, практика, лабораторные работы), аудитория (корпус и номер), в которой будет проходить занятие, ссылка на информацию о преподавателе, которая хранится в отдельном файле.
- (c) В информацию о преподавателе входит ФИО, должность, кафедра, университет, контактная информация (телефон рабочий, домашний, e-mail). Необходимо учесть, что рабочий телефон может быть не один, домашнего телефона может не быть или быть несколько, e-mail может не быть или быть несколько.

## Лабораторная работа № 8. Применение JavaScript для добавления интерактивных возможностей веб-страниц

### 8.1. Цель работы

Приобретение навыков программирования простых сценариев на JavaScript.

### 8.2. Информация, необходимая для начала работы

Язык JavaScript предназначен для создания интерактивных HTML-документов. С его помощью можно создавать приложения, выполняемые как на стороне клиента, так и на стороне сервера.

*Программа (сценарий)* на языке JavaScript представляет собой последовательность операторов, разделённых точкой с запятой (;), если они расположены в одной строке.

*Литерал* — простейшие данные, которыми может оперировать программа. Они не могут изменяться.

Литералы целого типа могут быть заданы в десятичном (со знаком или без знака), шестнадцатеричном или восьмеричном представлении. В литералах вещественного типа разделителем разрядов служит точка. Строковый литерал представляется последовательностью символов, заключенной в одинарные или двойные кавычки.

Для хранения данных используются переменные, которые в сценарии представляются с помощью идентификаторов. Идентификатор должен начинаться с буквы латинского алфавита либо с символа подчёркивания. Переменная определяется и/или инициализируется с помощью оператора `var`. Значение переменной изменяется в результате выполнения оператора присваивания.

Переменные, описанные в сценарии как в части `<head>`, так и в части `<body>`, имеют одну и ту же область действия, доступны любому сценарию текущего документа. Такие переменные называются *глобальными*. Локальные переменные определены в теле функции.

*Выражения* строятся из литералов, переменных, знаков операций, скобок. В результате вычисления выражения получается единственное значение, которое может быть либо числом (целым или вещественным), либо строкой, либо логическим значением. Выражения формируются из операндов и обозначений *операций*. Некоторые операции JS приведены в табл. 8.1–8.3

**Арифметические операции в JS**

**Таблица 8.1**

Операция	Название
+	Сложение
-	Вычитание
*	Умножение
/	Деление
%	Остаток от деления целых чисел
++	Увеличение значения операнда на единицу
--	Уменьшение значения операнда на единицу

**Операции отношения в JS**

**Таблица 8.2**

Операция	Название
<	Меньше
<=	Меньше или равно
==	Равно
!=	Не равно
>=	Больше или равно
>	Больше

**Операции отношения в JS**

**Таблица 8.3**

Операция	Название
!	Логическое НЕ
&&	Логическое И
	Логическое ИЛИ

Литерал `null` используется для обозначения неопределённого значения. Если переменной присвоено значение `null`, то она считается инициализированной.

Сценарии, написанные на языке JavaScript, могут располагаться непосредственно в HTML-документе между тегами `<script>` и `</script>`:

```
<html>
 <head>
 <meta charset="utf-8"/>
 <title>Первый сценарий в документе</title>
 </head>

 <body>
 <p>Страница, содержащая сценарий.</p>

 <script>
 <!--

 var a=8; h=10
 document.write ("Площадь прямоугольного \
 треугольника равна ", a*h/2,".")

 //-->
 </script>

 <p>Конец формирования страницы, содержащей сценарий</p>
 </body>
</html>
```

В приведённом выше сценарии описываются и инициализируются две переменные: `a` и `h`. Затем значение выражения `a*h/2` записывается в документ. Для формирования вывода в HTML-страницу используется метод `write` объекта `document`.

Основным элементом языка JavaScript является функция:

```
function F (V) {S}
```

где `F` — идентификатор функции, задающий имя, по которому можно обращаться к функции; `V` — список параметров функции, разделяемых запятыми; `S` — тело функции, в нём задаются действия, которые нужно выполнить, чтобы получить результат. Необязательный оператор `return` определяет возвращаемое функцией значение.

Описание функции не может быть вложено в описание другой функции.

## ПРИМЕР 1. Использование сценария с функцией

```
<html>
 <head>
 <meta charset="utf-8"/>
 <title>Использование сценария с функцией</title>
 <script language="JavaScript">
 <!-- //
 function care (a, h) { return a*h/2 }
 //-->
 </script>
 </head>
 <body>
 <p>Начало отображения страницы
 со сценарием и функцией.</p>
 <script>
 <!--
 var al=4; hl=16
 var s=care (al,hl)
 document.write("При вызове функции \
 получено значение ", s,".");
 //-->
 </script>
 <p>Конец формирования страницы.</p>
 </body>
</html>
```

## ПРИМЕР 2. Обработка значений из формы

```
<html>
 <head>
 <meta charset="utf-8"/>
 <title>Обработка значений из формы</title>
 <script language="JavaScript">
 <!--//
 function care(a, h)
 { var s= (a*h) / 2;
 document.write ("Площадь прямоугольного \
 треугольника равна ", s);
 return s;
 }
 //-->
 </script>
 </head>
 <body>
 <p>Пример сценария со значениями из формы</p>
 <form name="form1">
 Основание: <input type="text" size=5 name="st1"><hr>
```

```

Высота: <input type="text" size=5 name="st2"><hr>
<input type="button" value=Вычислить
 onClick="care(document.form1.st1.value,
 document.form1.st2.value)">
 <input type="reset" value=Обновить>
</form>
</body>
</html>

```

В рассмотренном примере 2 параметр обработки события (`onClick`) задаёт действия, выполняемые при обработке события «щелчок мышью по кнопке Вычислить» (`button`). Кнопка «Обновить» очищает поля формы.

Таблица 8.4  
События и элементы документов HTML

Событие	Объекты	Когда происходит событие
Abort	image	Отказ от загрузки изображения
Blur	windows, элементы формы	Потеря объектом фокуса
Change	text, textarea, select	Изменение значения элемента
Click	button, radio, checkbox, submit, reset, link	Щелчок на элементе или связи
DragDrop	windows	Перетаскивается мышью объект в окно браузера
Error	image, windows	Ошибка при загрузке документа или изображения
Focus	windows, элементы формы	Окно или элемент формы получает фокус
KeyDown	document, image, link, textarea	Нажатие клавиши клавиатуры
KeyPress	document, image, link, textarea	Удержание нажатой клавиши клавиатуры
KeyUp	document, image, link, textarea	Отпускаются клавиши клавиатуры
Load	Тело документа	Загружается документ в браузер

**Таблица 8.4**  
**События и элементы документов HTML (продолжение)**

<b>Событие</b>	<b>Объекты</b>	<b>Когда происходит событие</b>
MouseDown	document, button, link	Нажатие кнопки мыши
MouseOut	area, link	Перемещение курсора из области изображения или связи
MouseOver	link	Перемещение курсора над связью
MouseUp	document, button, link	Отпускается кнопка мыши
Move	windows	Пользователь или сценарий перемещает окно
Reset	form	Нажатие кнопки Reset формы
Resize	windows	Пользователь или сценарий изменяет размеры окна
Select	text, textarea	Выбирается поле ввода элемента формы
Submit	form	Нажатие кнопки Submit формы
Unload	Тело документа	Пользователь закрывает документ

**Таблица 8.5**  
**Методы объекта Math**

<b>Метод объекта</b>	<b>Описание метода</b>
abs	Абсолютное значение
sin, cos, tan	Тригонометрические функции
log	Натуральный логарифм
exp	Экспонента
min	Наименьшее значение двух аргументов
max	Наибольшее значение двух аргументов
pow	Показательная функция
sqrt	Квадратный корень

Для организации ветвлений используют условный оператор:  
`if B {s1} else {s2}`  
где B — выражение логического типа; s1 и s2 — операторы.

### 8.3. Примеры реализации интерактивных возможностей веб-страниц

ПРИМЕР 3. Простой калькулятор

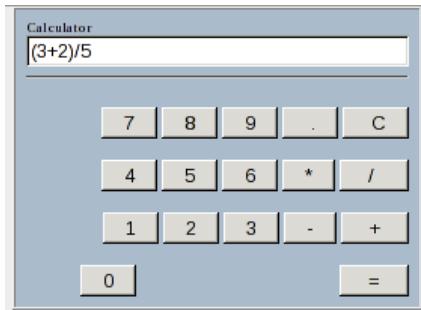


Рис. 8.1. Простой калькулятор на JS

```
<html>
<head>
<meta charset="utf-8"/>
<title>JavaScript - Calculator</title>
<style>
A:Link{ Color: #000000; Text-decoration: underline}
A:Visited{ Color: #000000; Text-decoration: underline}
A:Hover{ Color: #000000; Text-decoration: none}
td, body {font-family: verdana, arial, helvetica; font-size:11px;}
</style>

<script language="JavaScript">
<!-- Begin
// DECLARE AND INITIALIZE VARIABLES
var zero = '0';
var one = '1';
var two = '2';
var three = '3';
var four = '4';
var five = '5';
var six = '6';
var seven = '7';
var eight = '8';
var nine = '9';
var add = '+';
var subtract = '-';
var multiply = '*';
var divide = '/';
var point = '.';
var clear = 'C';
var equals = '=';</pre>

```

```

var multiply = '*';
var divide = '/';
var decimal = '.';

// FUNCTION: compute
function compute(f)
{
 f.view.value = eval(f.view.value);
}

// FUNCTION: getNum
function getNum(f, val)
{
 f.view.value += val;
}
// End -->
</script>
</head>

<body bgcolor="#EDEDED" text="#000000" link="#000000"
 topmargin="0" leftmargin="0">
<center>

<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=1 BORDERCOLOR=CCCCCC>
<TR><TD>
<TABLE CELLPADDING=7 CELLSPACING=0 BGCOLOR=AABBCC>

<!-------BEGIN CALCULATOR----->
<FORM NAME="calc">
<TR>
 <TD COLSPAN=2>
 <DIV ALIGN=LEFT><SMALL>
 Calculator
 </SMALL></DIV>
 <INPUT TYPE=text NAME="view" SIZE=30 action="compute(this)">

 <HR>
 </TD>
</TR>

<TR>
 <TD COLSPAN=2><DIV ALIGN=RIGHT>
 <INPUT TYPE=button VALUE=" 7 " onClick="getNum(this.form, seven)">
 <INPUT TYPE=button VALUE=" 8 " onClick="getNum(this.form, eight)">
 <INPUT TYPE=button VALUE=" 9 " onClick="getNum(this.form, nine)">
 <INPUT TYPE=button VALUE=" ." onClick="getNum(this.form, decimal)">
 <INPUT TYPE=reset VALUE=" C ">
 </DIV></TD>
</TR>
<TR>
 <TD COLSPAN=2><DIV ALIGN=RIGHT>
 <INPUT TYPE=button VALUE=" 4 " onClick="getNum(this.form, four)">
 <INPUT TYPE=button VALUE=" 5 " onClick="getNum(this.form, five)">
 <INPUT TYPE=button VALUE=" 6 " onClick="getNum(this.form, six)">
 <INPUT TYPE=button VALUE=" * " onClick="getNum(this.form, multiply)">
 <INPUT TYPE=button VALUE=" / " onClick="getNum(this.form, divide)">
 </DIV></TD>
</TR>

```

```

<TR>
<TD COLSPAN=2><DIV ALIGN=RIGHT>
 <INPUT TYPE=button VALUE=" 1 " onClick="getNum(this.form, one)">
 <INPUT TYPE=button VALUE=" 2 " onClick="getNum(this.form, two)">
 <INPUT TYPE=button VALUE=" 3 " onClick="getNum(this.form, three)">
 <INPUT TYPE=button VALUE=" - " onClick="getNum(this.form, subtract)">
 <INPUT TYPE=button VALUE=" + " onClick="getNum(this.form, add)">
</DIV></TD>
</TR>
<TR>
<TD><DIV ALIGN=CENTER>
 <INPUT TYPE=button VALUE=" 0 " onClick="getNum(this.form, zero)">
</DIV>
</TD>
<TD>
<DIV ALIGN=RIGHT>
 <INPUT TYPE=button VALUE=" = " onClick="compute(this.form)">
</DIV>
</TD>
</TR>
</FORM>
<!-------END CALCULATOR----->

</TABLE>
</TD></TR></TABLE>

<!-- BODY END HERE -->

</body>
</html>

```

#### ПРИМЕР 4. Вертикальное меню

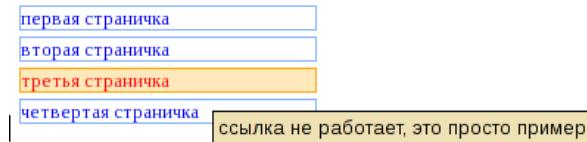


Рис. 8.2. Вертикальное меню на JS

```

<!DOCTYPE HTML>
<html>
 <head>
 <meta charset="utf-8"/>
 <style type="text/css">
 <!--
 A.menu {
 display: block;
 BORDER: 1px solid #7EAEFF;
 background: White;
 WIDTH: 200px;

```

```

 TEXT-DECORATION: none;
 }
 A.menu:hover {
 BORDER: 1px solid Orange;
 BACKGROUND-COLOR: #ffe8ba;
 TEXT-DECORATION: none;
 color: Red;
 }
 .menu {
 FONT-SIZE: 10pt;
 FONT-FAMILY: arial, helvetica, sans-serif;
 }
-->
</style>
</head>

<body>
 <table border=0>
 <tr>
 <td><A class="menu"
 title="ссылка не работает, это просто пример" href="#">
 первая страничка
 </td>
 </tr>
 <tr>
 <td><A class="menu"
 title="ссылка не работает, это просто пример" href="#">
 вторая страничка
 </td>
 </tr>
 <tr>
 <td><A class="menu"
 title="ссылка не работает, это просто пример"
 href="#" target=_blank> третья страничка
 </td>
 </tr>
 <tr>
 <td><A class="menu"
 title="ссылка не работает, это просто пример"
 href="#" target=_blank> четвертая страничка
 </td>
 </tr>
 </table>
</body>
</html>
```

## ПРИМЕР 5. Представление данных с помощью списка на JS

Требуется написать сценарий обработки анкеты переводчика. Переводчик может владеть несколькими языками. Сведения о тех языках, которыми он владеет, требуется задать с помощью списка. За знание каждого языка назначается определенная сумма. После анализа всех выбранных значений определяется вознаграждение.

```
<html>
 <head>
 <meta charset="utf-8"/>
```

## Анкета переводчика

Выберите язык, которым Вы владеете в совершенстве

русский  
английский  
французский  
немецкий

Вознаграждение:

Рис. 8.3. Представление данных с помощью списка на JS

```
<title>Данные об языках, представленные с помощью списка</title>
<script language="JavaScript">

</script>
</head>

<body>
<form name="form1">
 <h4>Анкета переводчика</h4>
 <p>Выберите язык, которым Вы владеете в совершенстве
 </p>
 <select name="forma" size=4 multiple>
 <option value="русский">русский
 <option value="английский">английский
 <option value="французский">французский
 <option value="немецкий">немецкий
 <option value="китайский">китайский
 <option value="японский">японский
 </select>
 <p>Вознаграждение:

 <input type="text" name="res" size=10>
 </p>
 <input type="button" value="Определить"
 onClick="form1.res.value=testsel()">
</form>
</body>
```

```

<input type="reset" value="Отменить">
</form>
</body>
</html>

```

### ПРИМЕР 6. Пятница 13

Требуется реализовать сценарий, определяющий, в какие месяцы года 13-е число попадает на пятницу.

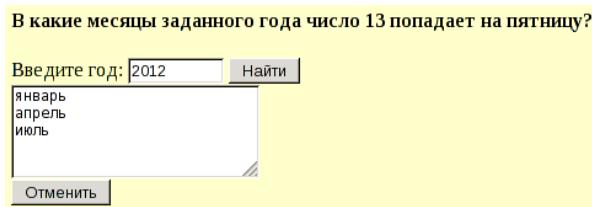


Рис. 8.4. Месяцы в указанном году, в которых есть пятница 13 (на JS)

```

<html>
 <head>
 <meta charset="utf-8"/>
 <title>В какие месяцы года 13 число попадает на пятницу?</title>
 <script language="JavaScript">
 <!-- // -->
 function def13(obj)
 {
 var t= new Date()
 var c=""
 var y=Number(obj.fye.value)
 for (var i=0; i<=11; i++)
 {
 t.setYear(y)
 t.setMonth(i)
 t.setDate(13)
 if ((t.getDay())==5)
 c = c+ fmon(i)+ "\r\n"
 }
 obj.res.value = c
 }

 function fmon(mont)
 {
 var s;
 switch (mont)
 {
 case 0: s="январь"; break;
 case 1: s="февраль"; break;
 case 2: s="март"; break;
 }
 }
 </script>
 </head>
 <body>
 В какие месяцы заданного года число 13 попадает на пятницу?
 <input type="text" value="2012" name="fye">
 <input type="button" value="Найти" onclick="def13(this.form)">
 <input type="button" value="Отменить" onclick="history.back()">
 </body>
</html>

```

```

 case 3: s="апрель"; break;
 case 4: s="май"; break;
 case 5: s="июнь"; break;
 case 6: s="июль"; break;
 case 7: s="август"; break;
 case 8: s="сентябрь"; break;
 case 9: s="октябрь"; break;
 case 10: s="ноябрь"; break;
 case 11: s="декабрь"; break;
 }
 return s
}

// -->
</script>
</head>

<body bgcolor="#FFFFCC">
<h4>В какие месяцы заданного года число 13 попадает на пятницу?</h4>
<form name="form1">
 Введите год: <input type="text" size="8" name="fy">
 <input type="button" value="Найти" onClick="def13(form1)">

 <textarea Cols="30" rows="4" name="res"></textarea>

 <input type="reset" value="Отменить">
</form>
</body>
</html>

```

#### ПРИМЕР 7. Дата и время изменения страницы

```

<html>
 <head><meta charset="utf-8"/>
 <title>Дата и время изменения страницы</title>
 </head>

 <body>
 <script language="JavaScript">
 document.write("Last update:" + document.lastModified);
 </script>
 </body>
</html>

```

#### ПРИМЕР 8. Дата посещения Web-страницы

Требуется реализовать сценарий, определяющий дату и время посещения Web-страницы.

```

<html>
 <head>
 <meta charset="utf-8"/>
 <script language="JavaScript">
 function NDays(r)
 { var s="";
 switch (r)
 {
 case 0: s="воскресенье"; break;
 case 1: s="понедельник"; break;
 case 2: s="вторник"; break;

```

```

 case 3: s="среда"; break;
 case 4: s="четверг"; break;
 case 5: s="пятница"; break;
 case 6: s="суббота"; break;
 }
 return s
}
function NMonths (r)
{ var s="";
 switch (r)
 {
 case 0: s="января"; break;
 case 1: s="февраля"; break;
 case 2: s="марта"; break;
 case 3: s="апреля"; break;
 case 4: s="мая"; break;
 case 5: s="июня"; break;
 case 6: s="июля"; break;
 case 7: s="августа"; break;
 case 8: s="сентября"; break;
 case 9: s="октября"; break;
 case 10: s="ноября"; break;
 case 11: s="декабря"; break;
 }
 return s
}
function DateTime()
{ var now = new Date()
 var str= "Вы посетили эту страницу
" +
 year = (1900 + now.getYear()) + ""
 str += now.getDate() + " " + NMonths(now.getMonth()) + " "
 + year + " года
" +
 str += "в " + now.getHours() + ":" + now.getMinutes() + ":" +
 now.getSeconds () + "
" +
 str += "Сегодня - " + NDays(now.getDay()) + "" +
 document.write(str)
}
</script>
</head>
<body>
<h4 align=center>
 Пример использования функций определения даты и времени
</h4>
<center>
<script language = "JavaScript">
 DateTime()
</script>

</center>
</body>
</html>
```

**Пример использования функций определения даты и времени**

Рис. 8.5. Дата и время посещения Web-страницы (на JS)

#### 8.4. Задания для самостоятельного выполнения

1. Реализовать в калькуляторе функции возведения в степень, экспоненты, логарифма, взятия корня, вычисления процентов.
2. Напишите сценарий, который по заданному времени определяет количество минут, прошедших от начала текущего занятия.
3. Напишите сценарий, который для заданной даты определяет, сколько дней прошло после некоторого события.
4. В анкете приводятся данные о семи сотрудниках: фамилия и дата приема на работу. Напишите сценарий вычисления стажа работы (в годах) и определения максимального числа сотрудников с одинаковым стажем.
5. Напишите сценарий, который позволяет посчитать стоимость предполагаемой покупки. Задается список продуктов, цена за единицу товара и количество экземпляров.
6. В анкете заполняется информация о шести студентах: фамилия и четыре оценки за сессию. Напишите сценарий определения категорий студента и количество человек в каждой категории. Категории определяются следующим образом: сдавшие все экзамены на «5» относятся к категории «отличники», имеющие хотя бы одну «2» — к категории «неуспевающие», все остальные студенты относятся к категории «успевающие».

## Предметный указатель

<b>C</b>	<b>U</b>
CSS 87, 91, 92, 99	UDP 50
<b>D</b>	<b>V</b>
DHCP 50, 51, 54, 66	VirtualBox 6, 10, 11, 17, 25, 28, 34,
DNS 30, 31, 33, 34, 38, 40, 45, 49,	51, 60, 67, 82, 83
50, 64	
DNS-клиент 30	
DNS-сервер 30, 43, 60, 62, 76, 104	
<b>H</b>	<b>W</b>
HTML 87, 89–93, 97, 99, 106, 107,	web-браузер 89
114, 116, 118	web-приложение 87, 89, 104
HTTP 66, 89	web-сервер 89
<b>I</b>	<b>X</b>
ICMP 32, 33	XML 106–109, 114
ifconfig 33	XSL 107–109, 114
IMAP 67, 76, 77, 86	XSLT 106, 107, 114
IP-адрес 30, 32, 33, 50, 57, 104	
<b>J</b>	<b>B</b>
java script 91, 107, 116, 118	Весовой коэффициент 6, 7
<b>M</b>	<b>K</b>
Midnight Commander 40, 55	Клиент-сервер 50, 89
MySQL 87, 91, 97, 104	<b>L</b>
<b>P</b>	Литерал 116, 118
PHP 87, 90, 91, 97, 98, 100, 104	<b>П</b>
POP3 83, 86	Пертингентность 5, 6
<b>R</b>	<b>P</b>
RTT 32	Ранжирование 5
<b>S</b>	Релевантность 5, 6
SMTP 67, 76, 77, 86	<b>C</b>
SOA 31, 34, 43	Сценарий 90, 97–99, 107, 116, 125,
<b>T</b>	127, 128, 130
TCP/IP 32, 50	
telnet 82, 83, 86	

## Список иллюстраций

2.1	Менеджер VirtualBox . . . . .	11
2.2	Окно «Свойства» VirtualBox . . . . .	12
2.3	Окно «Создать виртуальную машину» . . . . .	12
2.4	Окно «Имя машины и тип ОС» . . . . .	13
2.5	Окно «Размер основной памяти» . . . . .	13
2.6	Окно «Виртуальный жёсткий диск» . . . . .	14
2.7	Окно «Мастер создания нового виртуального диска» . . .	14
2.8	Окно «Дополнительные атрибуты виртуального диска» .	15
2.9	Окно «Расположение и размер виртуального диска» . .	15
2.10	Первое итоговое окно конфигурации виртуального диска	16
2.11	Второе итоговое окно конфигурации виртуального диска	16
2.12	Окно «Свойства» виртуальной машины Base . . . . .	17
2.13	Окно «Носители» виртуальной машины Base . . . . .	18
2.14	Окно «Добавить новый привод оптических дисков» виртуальной машины Base . . . . .	18
2.15	Окно «Выбрать образ оптического диска» виртуальной машины Base . . . . .	19
2.16	Окно «Носители» виртуальной машины Base. Выбор образа оптического диска . . . . .	19
2.17	Установка образа на жёсткий диск . . . . .	20
2.18	Виртуальная машина Base. Установка русского языка .	20
2.19	Виртуальная машина Base . . . . .	21
2.20	Конфигурация жёсткого диска . . . . .	21
2.21	Задать сетевое имя виртуальной машины . . . . .	22
2.22	Указать часовой пояс «Москва» . . . . .	22
2.23	Установка пароля для root . . . . .	23
2.24	Конфигурирование размера жёсткого диска . . . . .	23
2.25	Виртуальная машина Base. Проверка конфигурации .	24
2.26	Копирование образа на жёсткий диск . . . . .	24
2.27	Виртуальная машина Base. Завершение установки . . .	25
2.28	Остановка виртуальной машины Base . . . . .	25
2.29	Отключение носителя информации в VirtualBox . . . . .	26
2.30	Настройка виртуальной машины: учётная запись, далее «Add to Administrators group» . . . . .	26
2.31	Настройка виртуальной машины: дата и время . . . . .	27
2.32	Настройка виртуальной машины: профиль устройств .	27
2.33	Подключение к виртуальной машине . . . . .	28
2.34	Множественный доступ к виртуальной машине . . . . .	29
3.1	Менеджер VirtualBox . . . . .	34
3.2	Окно «Свойства» VirtualBox . . . . .	35

3.3	Создание в VirtualBox виртуальной машины Server . . . . .	35
3.4	Окно «Размер основной памяти» виртуальной машины Server . . . . .	36
3.5	Окно «Виртуальный жёсткий диск» виртуальной машины Server . . . . .	36
3.6	Итоговое окно конфигурации виртуальной машины Server . . . . .	37
3.7	Тип сетевого подключения виртуальной машины Server . . . . .	37
3.8	Изменение сетевого подключения виртуальной машины Server . . . . .	38
3.9	Изменение «Wired connection 1» виртуальной машины Server . . . . .	39
3.10	Сведения о сетевом соединении виртуальной машины Server . . . . .	39
3.11	Задать имя хоста в домене telesys-stud.net . . . . .	40
3.12	Разрешение доступа к DNS . . . . .	41
3.13	Прямая и обратная мастер-зоны домена telesys-stud.net . . . . .	42
3.14	Конфигурация прямой мастер-зоны домена telesys-stud.net . . . . .	43
3.15	Конфигурация обратной мастер-зоны домена telesys-stud.net . . . . .	45
3.16	Запуск DNS-службы. Неверные права доступа named . . . . .	46
3.17	Корректный запуск DNS-службы . . . . .	46
3.18	Проверка с помощью команды dig . . . . .	47
3.19	Проверка с помощью команды host -l . . . . .	47
3.20	Проверка с помощью команды host -a . . . . .	48
3.21	Проверка с помощью команды host -t A и host -t PTR . . . . .	48
4.1	Тип сетевого подключения виртуальной машины Server . . . . .	51
4.2	Изменение сетевого подключения виртуальной машины Server . . . . .	52
4.3	Тип сетевого подключения виртуальной машины Server . . . . .	52
4.4	Переход в привилегированный режим . . . . .	53
4.5	Результат команды ifconfig . . . . .	53
4.6	Установка dhcp и httpd . . . . .	54
4.7	Тип сетевого подключения виртуальной машины Server . . . . .	55
4.8	Создание конфигурационного файла интерфейса p2p1 . . . . .	56
4.9	Результат команды ifconfig . . . . .	56
4.10	Файл ifcfg-p2p1 . . . . .	57
4.11	Файл /etc/dhcpd.conf . . . . .	58
4.12	Файл /etc/sysconfig/dhcpd . . . . .	59
4.13	Файл /var/named/master/fz/telesys-stud.net . . . . .	59
4.14	Файл /var/named/master/rz/192.168.0 . . . . .	60
4.15	Результат ifconfig и ping на виртуальной машине Client . . . . .	61
4.16	Запуск http://localhost на виртуальной машине Server . . . . .	62
4.17	Файл /var/named/master/fz/telesys-stud.net . . . . .	63
4.18	Файл /var/named/master/rz/192.168.0 . . . . .	63

4.19	Доверенные службы в межсетевом экране . . . . .	64
4.20	Результат команды ping на виртуальной машине Client .	65
4.21	http://www.telesys-stud.net на виртуальной машине Client	65
5.1	Тип сетевого подключения виртуальной машины Server .	68
5.2	Установка dovecot, postfix, telnet . . . . .	68
5.3	Тип сетевого подключения виртуальной машины Server .	69
5.4	Файл /etc/dovecot/dovecot.conf . . . . .	70
5.5	Файл /etc/dovecot/conf.d/10-mail.conf . . . . .	71
5.6	Файл /etc/postfix/main.cf: имя почтового сервера и имя домена . . . . .	71
5.7	Файл /etc/postfix/main.cf: интерфейс для получения почты	72
5.8	Файл /etc/postfix/main.cf: место назначения почты . .	72
5.9	Файл /etc/postfix/main.cf: доступ пользователей к почте	73
5.10	Файл /etc/postfix/main.cf: доверенная подсеть . . . . .	74
5.11	Файл /etc/postfix/main.cf: каталог для доставки почты .	74
5.12	Файл /var/named/master/fz/telesys-stud.net . . . . .	75
5.13	Файл /var/named/master/rz/192.168.0 . . . . .	75
5.14	Запуск named, dhcpcd, dovecot, postfix . . . . .	76
5.15	Запуск Evolution . . . . .	77
5.16	Настройка Evolution . . . . .	78
5.17	Настройка Evolution. Указание e-mail . . . . .	78
5.18	Настройка Evolution. IMAP-сервер . . . . .	79
5.19	Настройка Evolution. Параметры получения почты . .	79
5.20	Настройка Evolution. SMTP-сервер . . . . .	80
5.21	Настройка Evolution. Информация об учётной записи .	80
5.22	Настройка Evolution. Завершение настройки . . . . .	81
5.23	Запуск Evolution . . . . .	81
5.24	Тестовое письмо . . . . .	82
5.25	Подключение по telnet к почтовому серверу . . . . .	83
5.26	Просмотр списка писем на сервере . . . . .	84
5.27	Содержание первого письма . . . . .	84
5.28	Содержание второго письма . . . . .	85
5.29	Удаление второго письма . . . . .	85
6.1	Страница «Default» Web-приложения «Успеваемость сту- дентов» . . . . .	88
6.2	Страница «Browse» Web-приложения «Успеваемость сту- дентов» . . . . .	88
8.1	Простой калькулятор на JS . . . . .	122
8.2	Вертикальное меню на JS . . . . .	124
8.3	Представление данных с помощью списка на JS . . . .	126
8.4	Месяцы в указанном году, в которых есть пятница 13 (на JS) . . . . .	127
8.5	Дата и время посещения Web-страницы (на JS)	130

## Список таблиц

1.1 Результаты поиска . . . . .	5
1.2 Результаты выполнения запросов . . . . .	8
1.3 Результаты сравнительного анализа поисковых систем Интернет . . . . .	8
6.1 Функции взаимодействия с базой данных MySQL в PHP- приложениях . . . . .	91
8.1 Арифметические операции в JS . . . . .	117
8.2 Операции отношения в JS . . . . .	117
8.3 Операции отношения в JS . . . . .	117
8.4 События и элементы документов HTML . . . . .	120
8.4 События и элементы документов HTML (продолжение) .	121
8.5 Методы объекта Math . . . . .	121

Учебное издание

**Анна Владиславовна Королькова  
Дмитрий Сергеевич Кулябов**

**Прикладные протоколы Интернет и www**

**Лабораторные работы**

Учебно-методическое пособие

Тематический план 2012 г., № 21

Технический редактор *H. A. Ясько*

Издание подготовлено в авторской редакции  
Компьютерная вёрстка *A. B. Королькова, Д. С. Кулябов*

Подписано в печать 7.08.2012 г. Формат 60×84/16. Печать офсетная.  
Усл. печ. л. 8,13. Тираж 100 экз. Заказ № 369.

---

Российский университет дружбы народов  
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3

---

Типография РУДН  
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3, тел. 952-04-41