ХАБАРОВ С. П., ЖУК Ю. А.

СЕТЕВЫЕ ТЕХНОЛОГИИ ВЗАИМОДЕЙСТВИЯ UBUNTU И WINDOWS ПЛАТФОРМ

Санкт-Петербург
«Наука и техника»
2013

Хабаров С. П., Жук Ю. А. Сетевые технологии взаимодействия Ubuntu и Windows платформ // С. П. Хабаров. — СПб.: Наука и техника, 2013. — 369 с.

В книге в популярной форме с использованием виртуальных машин рассматриваются вопросы сетевого взаимодействия в гетерогенных сетях на базе Windows и Linux платформ. Даны краткие сведения по ОС Ubuntu, её командному языку и основным утилитам сетевого администрирования. Рассмотрены вопросы разделения ресурсов и удалённого доступа к ним.

Большое число примеров иллюстрируют взаимодействия Windows и Ubuntu платформ: путем монтирования по SMB или gvfs, на основе удаленных рабочих RDP и VNC столов, на базе файлового сервера Samba, путем организации FTP и WEB серверов. Дается представление о возможностях удаленного доступа к узлам ЛВС с мобильных устройств на базе iOS и Android.

Книга предназначена для специалистов в области информационных систем и технологий, для студентов и аспирантов, а также для специалистов в других областях.

Рецензенты:

УТКИН Л. В., д. т. н., профессор, проректор по научной работе, заведующий кафедрой управления, автоматизации и системного анализа Санкт-Петербургского государственного лесотехнического университета

КАШЕВНИК А. М., к. т. н., старший научный сотрудник лаборатории интегрированных систем автоматизации Санкт-Петербургского института информатики и автоматизации Российской академии наук

ISBN 978-5-94387-941-8

ОГЛАВЛЕНИЕ

Приложение

ПРЕДИСЛОВИЕ

Основная цель данной книги — дать общее представление об организации простейших офисных или домашних ЛВС, включающих в свой состав ПК с различными операционными системами. Одной из таких систем наряду с Windows является Linux, а Ubuntu — наиболее популярный в настоящее время клон этой системы. Но если сетевое взаимодействие Windows-компьютеров на пользовательском уровне большинству знакомо даже по школьной программе, то этого нельзя сказать при использовании Linux-компьютеров, а тем более при их совместном использовании.

Linux — отличная операционная система, но от Windows не уйти. Windows будет окружать нас всегда — будь то домашняя, корпоративная сеть или интернет-кафе. Нам предстоит постоянно обмениваться документами с Windows-компьютерами, так как далеко не все пользователи предпочитают работать в Linux, а некоторые даже не знают о ее существовании и разнообразии ее клонов.

Какая из систем лучше? Об этом пусть спорят профессионалы, хотя и у них различные точки зрения в зависимости от той сферы деятельности, которой они занимаются. Мы же с вами постараемся составить лишь общее представление о том:

- что такое Ubuntu, как операционная система класса Linux;
- как подключить Linux-компьютеры к сети и настроить их сетевое взаимодействие;
- как совместно использовать сетевые ресурсы Windows- и Linux-компьютеров в ЛВС;
- как организовать удаленный доступ к ПК для целей сетевого администрирования.

Все эти вопросы будут рассмотрены лишь поверхностно, чтобы дать вам начальные сведения и привить элементарные навыки, которые вы можете совершенствовать в дальнейшем, используя другие источники информации.

В частности, что касается Ubuntu, то она будет рассмотрена в очень ограниченном объеме, который необходим для организации сетевого взаимодействия компьютеров. Познакомимся только с файловым менеджером и терминалом, а также их использованием для целей администрирования как отдельного компьютера, так и сети в целом.

Трудно предположить, что без начальных навыков по инсталляции Ubuntu и организации сети кто-либо допустит вас в локальную сеть работающих компьютеров, хотя бы и домашнюю. В этих условиях, изложение базируется на том факте, что вы будете использовать виртуализацию как Windows, так и Ubuntu на своем рабочем Windows-компьютере. В зависимости от его мощности, у вас появится возможность организовывать на нем сеть от 2-х до 5-ти узлов с разными операционными системами. Этого вполне достаточно для изучения предлагаемого материала.

При таком подходе просто необходимо иметь хотя бы минимальные представления о том, что такое виртуализация, какое программное обеспечение для этого применяется, как оно используется и настраивается. Если вы не знакомы с этой технологией, то следует познакомиться с ней до начала прочтения данной книги по литературным или Интернет источникам. В Приложении приведены некоторые выдержки из таких источников, позволяющих лучше понять и изучить содержимое данной книги.

В книге в популярной форме с использованием виртуальных машин рассматриваются вопросы сетевого взаимодействия в гетерогенных сетях на базе Windows и Linux платформ. Даны краткие сведения по ОС Ubuntu, её командному языку и основным утилитам сетевого администрирования. Рассмотрены вопросы разделения ресурсов и удалённого доступа к ним. Большое число примеров иллюстрируют взаимодействия Windows и Ubuntu платформ. Дается представление о возможностях удаленного доступа к узлам ЛВС с мобильных устройств на базе iOS и Android.

Авторы не ставили перед собой цель подробного описания тех или иных программных сред, превращая книгу в техническое руководство. Скорее это методическое пособие, позволяющее с единых позиций рассмотреть все аспекты современных сетевых технологий.

1. ВИРТУАЛИЗАЦИЯ КАК ИНСТРУМЕНТ ИЗУЧЕНИЯ ГЕТЕРОГЕННЫХ СЕТЕЙ

1.1. Понятия сетевой технологии и межсетевого взаимодействия

В базовом варианте термин сетевая технология — это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств, достаточный для построения вычислительной сети. Слово «достаточный» подчеркивает то, что это минимальный набор средств, с помощью которых можно построить работоспособную сеть. Но ее можно улучшить, например, за счет выделения в ней подсетей, что потребует кроме протоколов стандарта Ethernet, еще и протокола IP, а также дополнительных устройств — маршрутизаторов. Такая сеть будет более надежной и быстродействующей, но за счет надстроек над средствами технологии Ethernet, которая составила базис сети.

В настоящее время термин сетевая технология все чаще применяется в его расширенном толковании, как любого набора средств и правил для построения сети и доступа к ним. Например, «технология баз данных». «технология сквозной маршрутизации», «технология IP-телефония».

Иногда сетевые технологии называют базовыми технологиями, имея в виду то, что на их основе строится базис любой сети. Примерами базовых сетевых технологий могут служить наряду с Ethernet такие известные технологии локальных сетей, как Token Ring и FDDI, или же технологии территориальных сетей X.25 и frame relay. Для получения работоспособной сети в этом случае достаточно приобрести программные и аппаратные средства, относящиеся к одной базовой технологии.

До недавнего времени проблемы межсетевого взаимодействия не очень волновали пользователей и системных администраторов. Они уютно себя чувствовали в замкнутом мире PC-совместимых компьютеров. Однако пора монокультурного развития сетей закончилась. Организации приобретают, например, бизнес-серверы Hewlett-Packard, графические станции Sun или Silicon Graphics и другую не менее достойную аппаратуру с разнообразными операционными системами.

Прежде чем говорить о межсетевом взаимодействия, уточним, что понимается под термином «сеть». В узком смысле − это совокупность компьютеров, соединенных между собой в соответствии с одной из

базовых топологий, использующих для передачи пакетов один из протоколов канального уровня, определенный для этой топологии. В широком смысле — это совокупность компьютеров и телекоммуникационного оборудования, обеспечивающая информационный обмен компьютеров в сети и доступ к распределенным ресурсам.

Когда две или более сетей организуют совместную транспортную службу, то такой режим взаимодействия обычно называют межсетевым взаимодействием (internetworking), а для обозначения составной сети часто используют термин интерсеть (internetwork или internet). Интерсеть обеспечивает только передачу пакетов, не занимаясь их содержанием.

Если термин internetworking обозначает взаимодействие сетей на нижних уровнях, непосредственно связанных с транспортировкой пакетов, то в понятие interoperability входит обеспечение согласования верхних уровней стека коммуникационных протоколов, реализуемых серверами и редиректорами операционных систем, а также некоторыми сетевыми приложениями.

В данное время наиболее часто под межсетевым взаимодействием понимается взаимодействие вычислительных машин в неоднородной (гетерогенной) сети. Использование разных аппаратных и программных компонентов в гетерогенной сети ведет к проблеме обеспечения межсетевого взаимодействия. Источник проблемы — несовпадение используемых наборов коммуникационных протоколов.

1.2. Понятие гетерогенной сети

Только небольшое количество сетей обладает однородностью (гомогенностью) программного и аппаратного обеспечения. Однородными чаще всего являются сети, которые состоят из небольшого количества компонентов от одного производителя. Но дни сетей от одного производителя миновали. Нормой сегодняшнего дня являются сети неоднородные (гетерогенные),

Гетерогенные сети — это сети, которые состоят из различных рабочих станций, операционных систем и приложений, а для реализации взаимодействия между компьютерами используются различные протоколы.

Кроме этого, термин гетерогенная сеть также используется в беспроводных сетях с использованием различных технологий доступа. Например, сеть, которая предоставляет услуги через беспроводную локальную сеть, но способна поддерживать службу при переходе на сотовую связь называется беспроводной гетерогенной сетью.

Следует отметить, что даже на небольших предприятиях сети разных отделов строят на основе задач конкретных групп сотрудников. Например, в инженерном отделе используют рабочие станции SPARC фирмы Sun Microsystems, потому что им нужны приложения, работающие только в

среде UNIX. В рекламном отделе — компьютеры Macintosh, поскольку они наилучшим образом подходят для создания презентационных материалов. В отделе продаж уже были стационарные Acer с Windows XP и появились новые ноутбуки Sony Vaio с Windows 7. И всем им нужна общая база данных, доступ в Интернет и почта. Задачей информационного отдела является интегрирование всего этого в единый прозрачный организм. Другими словами, создать работоспособную гетерогенную сеть.

Добавление в вычислительную сеть новых, чужеродных элементов может происходить при всякой значительной реорганизации предприятия. Например, при смене его владельца. В этом случае вновь приобретенное предприятие и его вычислительное оборудование также должны быть интегрированы в общую структуру предприятия нового владельца.

Разнообразие всех компонентов, из которых строится сеть, порождает еще большее разнообразие структур сетей, получающихся из этих компонентов. А если продолжить далее и рассмотреть более сложные образования, получающиеся в результате объединения отдельных сетей в единую большую сеть, то становится понятным то множество проблем, связанных с проектированием, администрированием и управлением такой гетерогенной интерсетью. В идеале это объединение неоднородных сетей должно быть прозрачным для пользователя.

Все изложенное относится, может быть не в таком объеме и к небольшим офисным или даже домашним сетям. Особенностью этих сетей является то, что в большинстве случаев функции пользователя и сетевого администратора совмещаются в одном лице. На вашем рабочем месте или дома — ноутбук с Windows, смартфон Samsung с Android, интернет-планшет Apple iPad с операционной системой iOS, да еще в офисе все ваши документы лежат на сервере на основе Windows 2008 Server и т. д.

В этих условиях ясно, что даже рядовой пользователь, а тем более тот, кто хочет стать специалистом в области информационных технологий, должен иметь представление о современных гетерогенных сетях и возможном межсетевом взаимодействии между узлами этой сети. А еще лучше получить хотя бы начальные навыки в этих сетевых технологиях и познакомиться с программными продуктами, их поддерживающими.

Трудно предположить, что не имея начальных навыков вы будете экспериментировать со своей офисной или домашней сетью. В этих условиях вам будет полезна виртуализация как Windows, так и Ubuntu Linux. Одним из самых важных достоинств виртуальных машин является возможность их объединения в виртуальные сети, что позволяет на одном компьютере моделировать поведение распределенных систем, состоящих как из приложений для конечного пользователя, так и различного рода серверов в гетерогенной среде. Гибкость виртуальных машин в отношении выделяемых им ресурсов и широкие возможности по обслуживанию и оптимизации производительности позволяет легко управлять множеством

различных конфигураций виртуальных машин и создавать независимые от оборудования, приложения, «упакованные» в виртуальные машины. Затем эти компоненты, состоящие из виртуальных машин, могут быть в различных вариантах объединены в сеть для моделирования различных систем.

1.3. Общие сведения о виртуализации

Когда на реальный компьютер устанавливают операционную систему, она подстраивается под его аппаратное обеспечение (процессор, материнскую плату, жесткие диски и т. д.) и начинает функционировать, используя в своей работе ресурсы данного аппаратного обеспечения. Но ее можно обмануть и подсунуть ей вместо реального «виртуальное железо», используя специальные программные продукты для виртуализации. Данные программы позволяют создать внутри одной операционной системы специальные изолированные программные среды (виртуальные машины), которые заменяют реальное аппаратное обеспечение для установленных внутри них операционных систем.

Виртуальные машины так хорошо эмулируют реальное аппаратное обеспечение, что установленные внутри них операционные системы даже не подозревают, что они используют не реальные «железки», а всего лишь виртуальные устройства. На рис. 1.1. приведена упрощенная структура взаимодействия аппаратуры, хостовой и гостевых ОС.

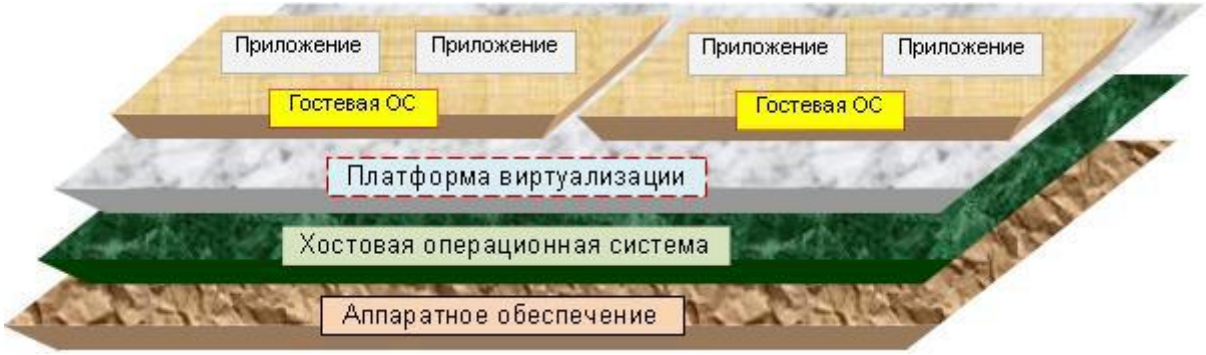

Рис. 1.1. Структура взаимодействия хостовой и гостевой ОС.

На рис. 1.1 видно, что прослойкой между хостовой и гостевыми операционными ОС является платформа виртуализации. Именно она позволяет внутри хостовой ОС организовать несколько виртуальных машин. Виртуальная машина исполняет некоторый машино-независимый код (например, байт-код, р-код) или машинный код реального процессора. Помимо процессора виртуальная машина может эмулировать работу как отдельных компонентов аппаратного обеспечения, так и целого реального компьютера (включая BIOS, оперативную память, жесткий диск и другие периферийные устройства). В последнем случае на виртуальной машине,

как и на реальном компьютере, можно устанавливать операционные системы. Например, Windows можно запускать в виртуальной машине под Linux или наоборот. На одном компьютере может функционировать несколько виртуальных машин, и на каждой из них может быть оригинальная операционная система. Это можно использовать, например, для имитации нескольких серверов на одном реальном сервере с целью оптимизации ресурсов последнего. Круг использования этой технологии достаточно широкий, но в основном виртуальные машины используются для:

- установки программ, несовместимых с ОС компьютера;
- защиты информации;
- тестирования программного и/или аппаратного обеспечения;
- создания переносных пользовательских сред, отвязанных от конкретного оборудования;
- запуска вредоносных программ с целью их исследования;
- эмуляции локальной компьютерной сети.

Концепция виртуальной машины как совокупности ресурсов, которые эмулируют поведение реальной машины, появилась в Кембридже в конце 1960-х годов как расширение концепции виртуальной памяти. В данный момент на рынке платформ виртуализации представлены продукты различных производителей как платные, так и бесплатные. Наиболее популярны среди простых и бесплатных программ:

- Virtual Box от компании Oracle, работает в Windows, Linux и т. д.;
- VMware Player, работает в Windows, Linux.;
- Virtual PC от компании Microsoft, работает только в Windows.

При использовании Virtual PC стоит обратить внимание на то, что в Windows XP поддерживаются только версии Virtual PC 2004 и 2007, а в Windows 7 только Windows Virtual PC, выпускаемая в качестве обновления к операционной системе.

Что особенно для нас важно — это то, что наши читатели при изучении этого пособия смогут на своих компьютерах использовать сразу несколько машин, а настроив их сетевые интерфейсы и протокольные стеки, организовать на своем компьютере гетерогенную небольшую ЛВС с сетевым обменом и распределением ресурсов между отдельными виртуальными узлами этой сети.

Кроме того, при таком подходе в любой виртуальной машине можно свободно менять сетевые настройки, а также конфигурации операционных систем, без особых опасений за последующее состояние операционной системы и, как следствие, состояние и самого компьютера.

Работа с виртуальными мини-ЛВС наложила некоторый отпечаток на выбор исследуемых операционных систем. Основное требование — это минимально необходимые вычислительные ресурсы, чтобы без проблем запустить несколько машин на одном компьютере. Так как особых

различий в сетевых настройках Windows разных версий не существует, особенно в объеме нашего изучения, то при дальнейшем изложении материала будем ссылаться на использование в наших гетерогенных мини-ЛВС виртуальных машин на базе Windows 98 и Windows XP.

1.4. Основы работы с Microsoft Virtual PC

Microsoft Virtual PC представляет собой платформу виртуализации, позволяющую создавать внутри хостовой операционной системы виртуальные машины, у которых есть BIOS, оперативная память, жесткий диск (выделенное место на жестком диске реального ПК), и могут эмулироваться различные периферийные устройства. Microsoft Virtual PC допускает на одном реальном компьютере функционирование нескольких виртуальных с разными гостевыми ОС.

Установка Microsoft Virtual PC

- Зайдите на официальный сайт продукта Microsoft Virtual PC 2007. (http://www.microsoft.com/en-us/download/details.aspx?id=4580)
- Скачайте и запустите установочный файл Microsoft Virtual PC.
- В окне Microsoft Virtual PC Wizard нажмите Next, а в следующем окне выберите I accept the terms in the license agreement –> Next.
- Затем введите Product Key –> Next –> Install –> Finish.

Создание новой виртуальной машины в Microsoft Virtual PC

- Нажмите Пуск –> Программы –> Microsoft Virtual PC.
- В окне New Virtual Machine Wizard нажмите Next.
- Далее выберите режим «Create a virtual machine (Use default settings to create a virtual machine; …)», нажмите Next.
- В следующем окне введите имя для новой виртуальной машины.
- Нажатием кнопки Browse выберите место для машины на диске, а затем Сохранить (<диск>:\New Virtual Machine.vmc) –> Next.
- В следующем окне в выпадающем списке выберите систему, которую будете устанавливать (по умолчанию – Other) –> Next.
- В новом окне надо задать объем RAM («Adjusting the RAM») или принять предложенный («Using the recommended») –> Next.
- Затем - «A new virtual hard disk» –> Next –> Next –> Finish.

Установка операционной системы на виртуальный ПК

- В окне «New Virtual Machine – Microsoft Virtual PC» появится надпись «Reboot and Select proper Boot device or Insert Boot Media in selected Boot device».

- Вставьте в лоток привода установочный диск и в меню выберите CD –> «Use Physical Drive <буква_диска>:». Если программа установки не запустится, выберите меню Action –> Reset.
- Начнется установка ОС, которая ничем не отличается от обычной инсталляции (включая выбор раздела, форматирование этого раздела, копирование файлов установки и т. д.).
- По мере установки ОС файл New-Virtual-Machine-Hard-Disk.vhd будет увеличиваться в размере.
- В определенный момент установки появится сообщение, что перемещения указателя мыши теперь возможны только в окне виртуальной машины. Для выхода за пределы этого окна, то есть для возврата в основную ОС, надо нажать правый Alt. Для входа в окно виртуальной – щелкнуть мышью внутри этого окна.
- После окончания установки ОС можно настроить как обычно.

Установка Virtual Machine Additions

Эта надстройка к Virtual PC обеспечивает обмен данными между виртуальной и основной машиной. Для того, чтобы установить эту надстройку необходимо выполнить следующую последовательность действий:

- Выберите Action –> Install or Update Virtual Machine Additions и в окне Virtual Machine нажмите Continue.
- Затем в окне Virtual Machine Additions – Install Shield Wizard нажмите Next –> Готово –> Да.
- Перезагрузите виртуальную машину. Создайте на диске основной машины новую папку. Ее будем использовать для обмена данными между основной и виртуальной машинами.
- Выберите меню Edit –> Settings в окне Virtual Machine или нажмите кнопку Settings в окне Virtual PC Console.
- Появится окно «Некоторые настройки этой виртуальной машины были временно отключены и не могут быть изменены во время работы виртуальной машины или в состоянии сохранения», нажмите кнопку OK.
- В окне Settings for Virtual Machine выберите Shared Folders, нажмите кнопку Share Folder… и в окне Обзор папок выберите папку, созданную для обмена.
- В выпадающем списке Drive letter задайте букву виртуального диска, который будет соответствовать на виртуальной машине общей папке основного компьютера.
- Наконец, установите флажок Share every time, нажмите OK –> OK и теперь общая папка будет доступна с обеих машин.

Подключение CD/DVD-привода к виртуальной машине

Если для установки какой-либо новой программы на виртуальной машине надо использовать CD/DVD-привод основной, то следует в меню Microsoft Virtual PC выбрать CD –> «Use Physical Drive <буква_диска>:».

После этого, используя, например, «Мой компьютер» на виртуальной машине, можно получить доступ к CD/DVD-приводу. Для отключения привода следует в меню выбрать CD –> «Release Physical Drive <буква_диска>:», а чтобы извлечь диск надо использовать CD –> Eject CD.

Консоль Virtual PC

Доступ ко всем виртуальным машинам осуществляется из консоли Virtual PC. В ней размещен список всех установленных виртуальных машин, кнопки для добавления новых (New…), удаления существующих (Remove) и просмотра/изменения настроек (Settings) существующих.

Рис 1.2. Консоль MS Virtual PC в среде Windows XP.

Работа с виртуальными машинами в среде MS Virtual PC

После установки операционной системы можно начать работу. Для того чтобы включить или выключить любую виртуальную машину, надо выделить ее в списке консоли и использовать кнопки «Start» или «Close... ».

Рис.1.3 Виртуальная машина с Windows 98 в среде MS Virtual PC
на основном компьютере с Windows XP.

Управление мышкой и клавиатурой передается автоматически после запуска виртуальной машины. В процессе работы с виртуальной машиной для пользователя доступны следующие основные режимы.

- Полноэкранный режим.

Вход и выход из полноэкранного режима виртуальной машины осуществляется клавишами [Alt - Enter].

- Установка новых приложений внутри виртуальной машины.

Инсталляция новых приложений внутри виртуальной машины Virtual PC осуществляется точно так же, как и на обычном компьютере.

- Копирование и вставка.

Операции копирования и вставки осуществляются через меню Edit выбором пунктов копирование (Copy) и вставка (Paste) либо комбинацией Alt+C и Alt+V соответственно.

- Приостановка и мгновенное восстановление состояния ВМ.

Для приостановки работы и сохранения состояния ВМ следует в основном меню MS Virtual PC выбрать Action -> Close... и в появившемся окне выбрать Save state.

Для временной приостановки работы ВМ в любой момент времени надо в основном меню MS Virtual PC выбрать Action -> Pause либо использовать комбинацию клавиш Alt+P. Возврат из режима паузы производится аналогично.

- Выключение виртуальной машины.

Для выключения достаточно, находясь внутри виртуальной ОС выбрать Пуск —> Завершение работы или в меню MS Virtual PC выбрать Action —> Close... и в появившемся окне выбрать Turn off PC.

Настройка сетевого взаимодействия виртуальных машин Virtual PC

Один из самых для нас важных и интересных вариантов использования виртуальных машин — это организация на одном физическом компьютере виртуальной сети, где одновременно работают несколько виртуальных машин.

Давайте разберемся, как в Virtual PC 2007 выполняется добавление сетевых устройств в состав виртуальных машин и настраиваются режимы их работы. Для этого в списке консоли Virtual PC выбираем нужную нам виртуальную машину. Используя кнопку Setting консоли, открываем окно настройки (Settings) виртуальной машины.

В левой части окна настроек виртуальной машины выводится список всех доступных для этой машины устройств, которые может настраивать пользователь этой машины. Сейчас нас интересуют только сетевые настройки, поэтому выберем строку Networking, и в правой части окна

появляется вкладка «Networking». Скорее всего, вы на экране своего дисплея увидите нечто подобное рис. 1.4.

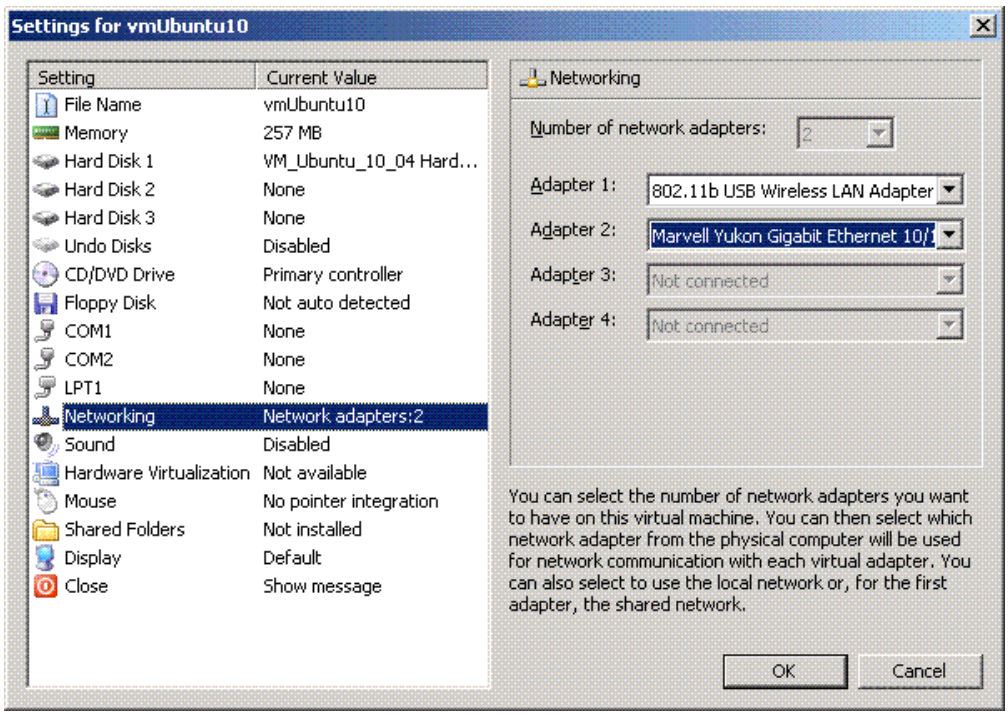

Рис.1.4. Настройка сетевых интерфейсов виртуальной машины.

В строчке «Adapter 1» выбран сетевой адаптер вашего основного компьютера. Что это значит? Платформа Virtual PC позволяет гостевым и хостовой системам совместно использовать ресурсы физического сетевого адаптера с помощью трех различных моделей сетевого взаимодействия или оградить виртуальную машину от внешнего сетевого взаимодействия.

➢ Если в строчке «Adapter» выбран ваш физический сетевой адаптер, это значит, что сетевой адаптер виртуальной машины напрямую подключен к сетевому адаптеру хоста. При таком типе сетевого взаимодействия виртуальная машина будет видеться из внешней сети и вести себя так, будто бы это отдельный компьютер в сети. Если в сети используется DHCP-сервер, виртуальная машина получит самостоятельный IP адрес в этой сети.

Такой тип сетевого взаимодействия применяется, когда из внешней сети необходимо обращаться к ресурсам виртуальной машины и работать с ней, как с полноценным клиентом сети. Например, гостевая система является файл-сервером. Иногда такой тип сетевого взаимодействия также называют Bridged Networking.

➢ При выборе варианта Local only виртуальные машины на одном хосте смогут взаимодействовать между собой, но им будут недоступны внешние сетевые соединения хоста. Такой тип сетевого взаимодействия применяется, когда требуется построить какую-либо модель сетевого

взаимодействия между несколькими машинами, однако внешние сетевые соединения не требуются. Например, такой тип взаимодействия идеален для проверки работы связки «сервер приложений» — «сервер баз данных».

➢ Вариант Not connected означает, что виртуальная машина не будет принимать участие в сетевом взаимодействии и виртуальный сетевой адаптер как устройство не будет включен.

➢ Последний вариант сетевого взаимодействия, который может быть выбран для виртуальной машины, это «Shared Networking». При использовании такого варианта, программа Virtual PC, действуя как DHCP-сервер, выдает виртуальной машине IP-адрес из диапазона 192.168.131.1 — 192.168.131.253. Также Virtual PC при этом является ещё и NAT-сервером (NAT — Network Address Translation).

Виртуальные машины, использующие этот тип сетевого соединения, спрятаны за NAT-сервером по отношению к внешней сети хоста. Они могут инициировать соединения с ее клиентами, но члены внешней сети не могут инициировать соединения с виртуальными машинами хоста. Такой тип сетевого взаимодействия идеален, когда требуется, например, выходить в Интернет из виртуальной машины, максимально при этом спрятав виртуальную машину от атак извне.

Как видно, Virtual PC 2007 предлагает различные модели сетевого взаимодействия, каждый из которых наиболее подходит в какой-либо конкретной ситуации. К тому же платформа Virtual PC позволяет создавать до 4-х сетевых адаптеров для одной виртуальной машины, что создает обширные возможности для экспериментов.

Несколько общих замечаний

Выбирая место расположения виртуальной машины, следует учесть, что после установки на виртуальную машину операционной системы и прикладных программ файл виртуальной машины может достигать нескольких гигабайт! Поэтому необходимо выбирать такой диск для VM, на котором есть достаточно свободного места.

На виртуальной машине все, или почти все, происходит как на настоящем компьютере, — даже иногда появляется знаменитый «синий экран смерти».

Иногда невозможно запустить установку новой операционной системы на виртуальной машине. В этом случае рекомендуется создать новую виртуальную машину, загрузить ее и заново запустить установку операционной системы.

Были зафиксированы случаи, когда виртуальная система создавалась вредоносным кодом для управления инфицированным компьютером. Например, вирус PMBS (1993 г.) и руткит SubVirt (2006 г.) создавали

виртуальную систему, которой ограничивались пользователи и все защитные программы, в том числе антивирусы и брандмауэры.

Автоматический запуск гостевой ОС при входе в основную

Для автоматического запуска гостевой ОС при входе в основную ОС прежде всего надо создать нового пользователя в основной ОС. Для начинающих пользователей желательно, чтобы они имели ограниченную учетную запись, а загрузка виртуальной ОС выполнялась в полноэкранном режиме без панелей меню и статуса Virtual PC.

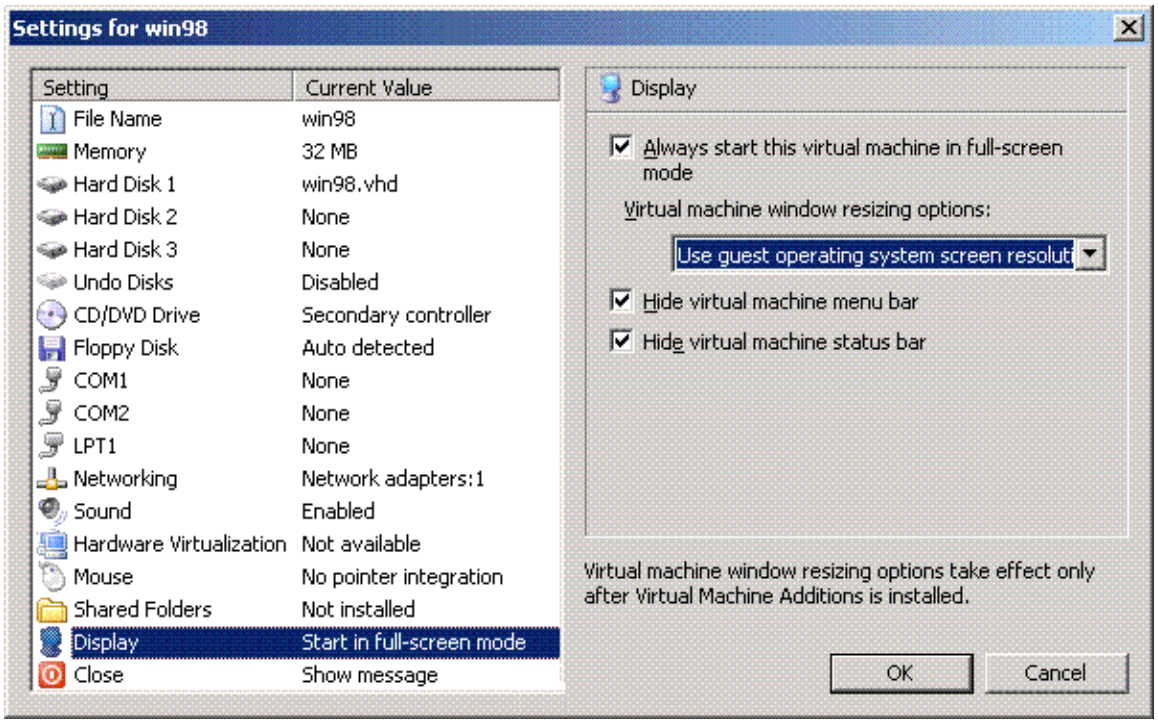

Рис.1.5. Настройка вида экрана виртуальной машины.

Пусть для примера гостевой операционной системой будет MS Windows 98, а хостовой — MS Windows XP. Создадим в хостовой Windows XP нового пользователя с именем, например, win98 и таким же паролем. Для автоматизации процесса загрузки виртуальной Windows 98 при входе пользователя win98 в Windows XP можно использовать три подхода:

- Ярлык в автозагрузку.
  - Находим файл, описывающий нашу виртуальную машину, например D:\win98_one\Win98_one.vmc.
  - Правой кнопкой мышки вызываем всплывающее меню и выбираем опцию «Создать ярлык».
  - Вновь созданный файл «Ярлык для win98.vmc» копируем в папку автозагрузки пользователя с именем win98, то есть в папку C:\Documents_and_Settings\win98\Главное_меню\Программа\Автозагрузка\.

BAT-файл в автозагрузку.

- Создаем текстовый командный файл, например StartWin98.bat, состоящий всего из одной строки:

```
"C:\Program Files\Microsoft Virtual PC\Virtual PC.exe" -
startvm "D:\win98_one\Win98_one.vmc" -singlepc
```

- Этот файл копируем в папку автозагрузки пользователя win98.

➢ Файл VBScript в автозагрузку.

- Создаем текстовый автоматически выполняемый файл на языке VBScript, например StartWin98.vbs, состоящий из нескольких операторов скриптового Visual Basic:

```
Dim WshShell, oExec, fileVM
Set WshShell = CreateObject("WScript.Shell")
fileVM="D:\win98_one\Win98_one.vmc"

Set oExec = WshShell.Exec("C:\Program Files\Microsoft
Virtual PC\Virtual PC.exe "+" -singlepc -startvm "+fileVM)
Do While oExec.Status = 0
     WScript.Sleep 1500
Loop

WshShell.Run "C:\WINDOWS\system32\shutdown.exe -l -f -t 0"
```

- Вновь созданный файл StartWin98.vbs копируем в папку автозагрузки для пользователя win98.

Следует отметить, что третий вариант не только запускает виртуальную MS Windows 98 при входе пользователя с именем win98 в основную ОС, но и отслеживает окончание работы виртуальной ОС, при котором производится автоматическое завершение сеанса активного пользователя, то есть пользователя с именем win98.

Чтобы пользователь не мог входить в хостовой компьютер до старта виртуального, следует установить в регистре:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrenVersion\Winlogon
DefaulDomainName (REG_SZ) = [Domain] (Vista Only)
DefaultUserName  (REG_SZ) = [Username]
DefaultPassword  (REG_SZ) = [Password]
AutoAdminLogon (REG_DWORD) = 1
```

В тех случаях, когда при завершение работы гостевой ОС смены пользователя оказывается недостаточно, можно организовать полную автоматическую перезагрузку хостовой ОС. Для этого достаточно в команде shutdown вместо ключа -l использовать ключ -r при том же

интервале времени ожидания до перезагрузки. Использование ключа принудительного завершение всех процессов -f зависит от конкретной ситуации. Особенно при подключении внешних сетевых процессов и ресурсов.

1.5. Общие сведения об Ubuntu

Ubuntu — это современная полнофункциональная операционная система, основанная на ядре Linux Debian. Каждые шесть месяцев все поправки, которые были внесены в Debian за последние полгода, вносятся в Ubuntu. Дистрибутив поддерживается и спонсируется компанией Canonical Ltd. Операционная система Ubuntu распространяется и всегда будет распространяться абсолютно бесплатно.

Устанавливая Ubuntu на свой компьютер, вы получаете полный набор всех необходимых для работы приложений, а все недостающее в стандартной поставке можно легко скачать из Интернета безо всяких ограничений и на совершенно законных основаниях (GPL, GNU и OSS). Официальная документация по системе доступна на сайте разработчика (рис.1.6).

Рис. 1.6. Официальная документация на сайте разработчика.

Ubuntu работает на большинстве современных компьютеров, поэтому существуют сборки Ubuntu для различных архитектур ПК. Самыми распространенными являются i386 и amd64. Версия amd64 предназначена для компьютеров, поддерживающих 64-битные вычисления. Все современные компьютеры с многоядерными процессорами их поддерживают (как AMD, так и Intel). Архитектура i386 является гораздо более старой, однако 64-битные процессоры полностью с ней совместимы. Поэтому версия Ubuntu для нее будет работать практически на всех компьютерах, включая современные многоядерные, но не будет поддерживать все возможности новых процессоров. В общем, все просто: если у вас новый компьютер, то рекомендуется использовать версию amd64, если же старый, то вам ничего кроме i386 не остается.

У каждой версии Ubuntu кроме номера есть также кодовое имя. Все кодовые имена состоят из двух начинающихся на одну и ту же букву английских слов: прилагательного и названия животного. Например, версия Ubuntu 10.04 LTS носит имя «Lucid Lynx» (Ясная Рысь). Начиная с Ubuntu 6.06 «Dapper Drake» первые буквы слов меняются в соответствии с алфавитом. Приставка LTS означает Long Term Support, то есть релиз с долгосрочной поддержкой. Ubuntu 10.04 Desktop будет получать все обновления безопасности до апреля 2013 года.

Минимальными системными требованиями для Ubuntu являются 384Mb оперативной памяти и хоть какой-нибудь процессор, однако не рекомендуется запускать Ubuntu на 384Mb. Для более-менее комфортной работы надо минимум 512Mb. Если у вас мало оперативной памяти, посмотрите в сторону более «легких» дистрибутивов, например в сторону Xubuntu.

Загрузить дистрибутив Ubuntu можно, посетив русскоязычный сервер http://ubuntu.ru/get или официальное русское зеркало Ubuntu на сервере Yandex (http://mirror.yandex.ru/ubuntu-releases/). Дистрибутивный диск Ubuntu выполнен в виде Live CD. Это означает, что с него можно запустить Ubuntu, не устанавливая ее на жесткий диск. Это очень удобно, так как можно попробовать дистрибутив перед его установкой.

И еще несколько общих замечаний об Ubuntu**.** Слово «ubuntu» в переводе с одного из африканских языков означает «гуманность по отношению к другим». Нужно отметить, что название полностью оправдывает себя. Прежде всего это проявляется в простой программе установки, удобном интерфейсе пользователя, а также в тщательной локализации. Ubuntu поставляется всего на одном компакт-диске. Это связано с тем, что:

- Во-первых, до версии 11.04 Ubuntu использовала только графическую среду GNOME, а среда KDE не входила в состав дистрибутива, что позволяло сэкономить много места.
- Во-вторых, дистрибутив комплектовался по правилу «одна задача — одна программа». Часто в состав дистрибутивов входит несколько проигрывателей, несколько браузеров и т. д. Здесь все иначе. Да, с одной стороны нет выбора. Но с другой стороны лучшее решение проблемы выбора — это отсутствие самого выбора. Ведь когда много всего, пользователь не знает, что ему лучше использовать. Кроме этого, в дистрибутив далеко не всегда включаются проверенные программы.
- В-третьих, дистрибутив Ubuntu ориентирован в основном на работу в графическом режиме, поэтому отсутствуют некоторые консольные утилиты, что также позволило немного сократить размер дистрибутива. Но основное сокращение производится за счет «одна задача — одна программа».

Ubuntu — один из самых популярных дистрибутивов в мире. На наших просторах пока популярны Fedora Core и Mandriva, а некоторые даже и не слышали такое название — «Ubuntu», но есть основания думать, что ситуация изменится. Что же касается популярности Fedora Core и Mandriva, то это объясняется тем, что дистрибутив Red Hat Linux (предшественник Fedora Core) был одним из самых первых и самых удачных дистрибутивов Linux, которые появились на наших просторах.

Затем последовал Mandrake (предшественник Mandriva), который оказался более удобным и простым, чем Red Hat, и этим заслужил популярность. С новыми версиями этих дистрибутивов не все просто: Fedora Core в некоторых моментах оставляет желать лучшего, а разработчики Mandriva, похоже, решили сделать коммерческий дистрибутив. Поэтому сейчас есть все основания, чтобы у нас начали развиваться альтернативные дистрибутивы, которые ничем не уступают той же Fedora Core.

Теперь о совместимости. Ubuntu полностью совместим с Debian, поскольку он основан именно на этом дистрибутиве. Но нужно помнить, что Debian несовместим с Fedora Core/Mandriva и другими SysV-дистрибутивами. Это означает, что:

- Во-первых, у Ubuntu другая система инициализации. Так, например, в Red Hat-совместимых дистрибутивах используется система инициализации стиля System V, а в Debian (Ubuntu) используется система инициализации стиля BSD. Но как пользователя система инициализации будет волновать вас меньше всего.
- Во-вторых, RPM-пакеты, которые используются в Red Hat-совместимых дистрибутивах (Fedora Core, Mandriva), установить в Ubuntu у вас не получится — там используется другой формат пакетов. Но для Debian, а значит и для Ubuntu, разработано не меньше программ, поэтому ущемленными вы себя чувствовать не будете. Тем более что Ubuntu позволяет устанавливать и обновлять программное обеспечение непосредственно из репозиториев (хранилищ пакетов) Debian. Фактически, с технической точки зрения, Ubuntu — это тот же Debian, но с более новыми версиями пакетов, которые входят в состав дистрибутива.

У Ubuntu есть одна интересная особенность: вы не можете по умолчанию войти в систему как пользователь root. В Linux-системах это самый «главный» пользователь, обладающий максимальными привилегиями.

Для выполнения команд, требующих прав root, нужно использовать утилиту sudo. С одной стороны, это неудобно, с другой — безопасно, ведь дистрибутив рассчитан на начинающих пользователей, которые с правами

root могут натворить много чего ненужного. Вместе с тем опытные пользователи в Ubuntu могут сделать полноценную учетную запись root.

Следует отметить, что кроме Ubuntu существует еще несколько модификаций дистрибутива таких, как Kubuntu, Edubuntu и Xubuntu.

Kubuntu — то же самое, что и Ubuntu, только основан на базе графической среды KDE, а не GNOME. Системные требования такие же. В состав дистрибутива входят программы, основанные на библиотеке Qt (лежит в основе KDE), а не Gtk+ (это основа GNOME);

Edubuntu — версия Ubuntu, «заточенная» для школ и других образовательных заведений. Содержит весь необходимый набор программного обеспечения для организации образовательного процесса;

Xubuntu — облегченная версия Ubuntu, основанная на базе графического менеджера Xfce, что позволяет запускать операционную систему на компьютерах с объемом оперативной памяти 64 Мбайт (но для полноценной работы рекомендуется не меньше 128 Мбайт ОЗУ).

Релизы LTS поддерживаются Canonical дольше, чем другие релизы Ubuntu. Исторически первой такой версией была Ubuntu 6.06. Обновления выполнялись в течение трех лет для пользовательских версий и пяти лет для серверных. К версии 6.06 Dapper Drake вышло два таких обновления. Обновления версии 8.04 Hardy Heron выходили более регулярно, вышло четыре обновления, но больше не ожидается. Для версии 10.04 Lucid Lynx вышло 4 обновления, последний доступный образ имеет версию 10.04.4.

Основываясь именно на этом релизе, изложен материал данного пособия. Но жизнь продолжается и, как говорил Гераклит из Эфеса еще в 554 году до нашей эры: «Все течет, все меняется (древнегреческое — Paula rhei)». И особенно стремительно в информационных технологиях.

К моменту подготовки данной рукописи к изданию стало известно, что текущим LTS релизом является Ubuntu 12.04 LTS Precise Pangolin, начиная с которого поддержка LTS-релизов Ubuntu составляет 5 лет как для серверов, так и для ПК и планшетов. Таким образом, обновления с исправлением проблем безопасности для Ubuntu 12.04 LTS Desktop, как и для Ubuntu 12.04 LTS Server, будут выпускаться до апреля 2017 года.

1.6. Обзор новых и перспективных версий Ubuntu

Пользовательский интерфейс ранних версий характеризовался оттенками коричневого и оранжевого цветов. Начиная с версии 10.04 цветовая гамма изменена в сторону черного и фиолетового цветов. В версиях до 11.04 основой Ubuntu была среда рабочего стола GNOME, которая разрабатывалась для того, чтобы обеспечить свободный, простой и

интуитивный интерфейс, предлагая полный диапазон современных настольных приложений. Помимо тех приложений, которые включены в GNOME, релиз Ubuntu содержал дополнительное программное обеспечение, такое как OpenOffice.org (LibreOffice начиная с версии 11.04), web-браузер Mozilla Firefox.

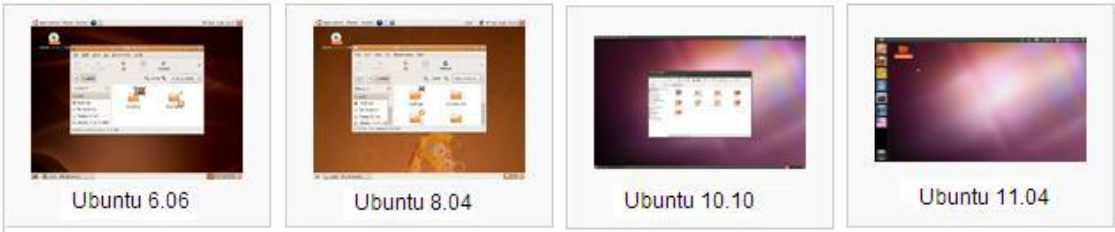

Рис. 1.7. Пользовательский интерфейс ранних версий Ubuntu.

Начиная с версии 11.04 «Natty Narwhal», стандартная для Ubuntu среда рабочего стола GNOME заменена на Unity. Она использовалась в Ubuntu Netbook Edition 10.10, а потом перекочевала на Desktop версию и стала в ней графической оболочкой по умолчанию. Теперь стандартный образ Ubuntu стал мультиплатформенным и может устанавливаться как на нетбуках, так и на ноутбуках и десктопных компьютерах.

Рис. 1.8. Пользовательский интерфейс Ubuntu 12.04 LTS.

В версии Ubuntu 12.04 LTS «Precise Pangolin» окружение рабочего стола по-прежнему Unity, но теперь уже с усовершенствованной строкой для поиска пунктов меню запущенных приложений. В состав данной версии включены следующие компоненты: GNOME3.4.1, Firefox 11, LibreOffice 3.5.2, Thunderbird 11 и др. Доступны версии для различных официально поддерживаемых архитектур, таких как i386, amd64, ARM.

В новой версии Ubuntu разработчики постарались внедрить самые популярные онлайн ресурсы непосредственно в графическую оболочку

системы. Онлайн сайты в браузере теперь становятся органичными и родными приложениями Ubuntu с помощью технологии WebApp. Любой пользователь получил возможность всегда быть в курсе всех событий за счет просмотра новостей и получать уведомления о поступивших новых сообщениях в социальных сетях с помощью системы уведомлений самой системы.

Реализован механизм удаленного входа в систему. Достаточно настроить удаленную учетную запись (Ubuntu Remote Login Account) на нужной удаленной машине и спокойно пользоваться этой функцией. Появилась опция Online Accounts, которая позволяют централизованно давать доступ и контролировать «кто и к чему» имеет в данный момент возможность подсоединения. Теперь легко отказать всем в доступе к какому-нибудь онлайн ресурсу.

Если ранее была возможность только открыть найденный файл программой по умолчанию, то теперь пользователь имеет возможность предварительного просмотра самого файла или информации о нем перед тем, как открыть данный файл. Эта функция доступна при нажатии правой клавиши мыши по интересующему файлу.

Достаточно сильно изменился в новой Ubuntu 12.10 Менеджер обновлений. Он заменен новым Software Updater'ом, который выделен в отдельную программу. В новом релизе реализовано и поддерживается полное шифрование раздела. В ходе установки системы предлагаются дополнительные опции – полное шифрование диска и применение LVM (Logical Volume Manager). Полное шифрование гарантирует практически 100 % защиту ваших персональных данных в случае кражи вашего компьютера, а использование LVM позволит расширить раздел с данными без нанесения вреда работающей системе за счет использования дополнительного винчестера.

Ubuntu 12.10 – это попытка внести новое и яркое. В ней внедрены разные онлайн дополнения, стирающие грань между web и компьютером. Она вышла с множеством обновленных популярных пакетов программ и новым более удобным и дружелюбным в использовании интерфейсом для Ubuntu One.

Вообще у Ubuntu довольно интересная перспектива развития — с каждым релизом она начинает все меньше нравиться специалистам (но не всем), а в то же время больше нравится людям, которые ничего не понимают в компьютерах.

Canonical Ltd — частная компания, основанная южноафриканским предпринимателем Марком Шаттлвортом для популяризации проектов свободного программного обеспечения. Она зарегистрирована на острове Мэн и имеет сотрудников по всему миру, включая главный офис в Лондоне и вспомогательные офисы в Бостоне, Монреале, Тайбэе, Сан-Паулу и Шанхае.

В день представления Ubuntu 12.10 Марк Шаттлворт анонсировал имя новой Ubuntu 13.04 – «Raring Ringtail» (Нетерпеливый Лемур). Это первый из двух коротких релизов перед Ubuntu 14.04 LTS.

Кроме этого заявлено о работе над версией Ubuntu для смартфонов (Ubuntu Phone OS), которая позволит Ubuntu сделать еще один шаг к созданию мощной, универсальной операционной системы для десктопов, облаков и множества других различных устройств.

Рис. 1.9. Перспективы Ubuntu (с сайта http://www.ubuntu.com).

По этому поводу Марк Шаттлворт сказал; «Мы определили новую эру конвергенции технологий, с одной единой операционной системой, лежащей в основе облачных вычислений, дата-центров, персональных компьютеров и потребителей электроники».

Другая значимая характеристика Ubuntu Phone заключается в заимствовании от другого проекта, а именно запуска Ubuntu под Android, которая позволяет развернуть полноценное рабочее окружение с помощью подключение через док-станцию (рис. 1.9). Для этого смартфон должен соответствовать минимальным системным требованиям: четырехъядерный процессор A9 ARM, 1 Гб оперативной памяти, 32 Гб Flash + SD.

Особенность такого переключения довольно интересна, так как при выводе на экран будет появляться новый адаптированный ПК-интерфейс, а не тот, который используется в телефоне. Обратной стороной такого решения станет необходимость создания двух различных интерфейсов для смартфона с поддержкой касаний и обычной десктоп версии.

Марк Шаттлворт в своем блоге написал, что главный акцент в следующих циклах будет сделан на планшеты, смартфоны и телевизоры. «Пора взглянуть на ядро Ubuntu через призму мобильных устройств: ключевыми метриками должны стать время работы от батареи, число запущенных процессов, потребление памяти», — пишет он.

Но это пока перспективы, а наша задача познакомиться с Ubuntu, как с представителем Linux-узлов в гетерогенных сетях и их взаимодействием с Windows системами. Поэтому переходим к установке операционных систем Ubuntu в качестве виртуальных машин под Windows.

2. ВИРТУАЛИЗАЦИЯ UBUNTU НА WINDOWS

Поставим перед собой задачу создать на своем компьютере виртуальную машину и установить на нее Ubuntu как гостевую операционную систему. Для решения этой задачи нам надо определиться:

- с типом операционной системы рабочего компьютера, на котором проводим виртуализацию;
- со средой виртуализации;
- с версией Ubuntu для гостевой операционной системы.

Оставляя в стороне разговор о достоинствах и недостатках тех или иных продуктов, можно констатировать, что в процессе обучения в настоящее время в основном используются компьютеры с Windows XP, Vista или Windows 7. Как показало тестирование, с этими ОС практически без сбоев работает MS Virtual PC 2007, выбор которой во многом определяется ее бесплатным распространением.

Что касается Ubuntu, то здесь будем использовать следующий подход. В качестве начальной выберем Ubuntu 6.10, которая показала свою устойчивую работу, хорошо описана в учебной литературе и в Интернете, что не вызовет у вас трудностей в поиске дополнительной информации.

Кроме того, эта ОС займет небольшой объем памяти вашего базового компьютера, на котором в дальнейшем будем виртуализировать еще несколько ОС. Исходя из этого, именно Ubuntu 6.10 будем использовать на начальном этапе лабораторного практикума.

Для обеспечения решения поставленной задачи понадобятся:

- Компьютер с Windows XP SP2, Windows Vista Ultimate или Windows 7.
- Ubuntu 6.10 (http://ubuntu-releases.optus.net/edgy/ubuntu-6.10-alternate-i386.iso)
- Ubuntu версии 10.04 или выше (http://ubuntu.ru/get).
- Среда виртуализации: Virtual PC 2007 + SP1 (для Windows XP и Vista) или Windows Virtual PC (для Windows 7).

Будем считать, что вы уже скачали и установили Virtual PC 2007, хорошо разобрались в ее работе, настройках и готовы к инсталляции гостевых операционных систем. Если так, то вперед, и начинаем с виртуализации Ubuntu 6.10

2.1. Создание виртуальной машины в MS Virtual PC

Первый шаг – скачать установочный образ Ubuntu. В нашем случае мы не можем использовать нормальный образ (ubuntu-6.10-desktop-i386.iso) по одной простой, но важной причине: виртуальный графический адаптер Virtual PC 2007 поддерживает только 16-битную глубину цвета, для своих виртуальных машин, а Ubuntu Live CD использует по умолчанию 24 бита. Не будем отрицать, что Ubuntu выглядит замечательно, но Virtual PC 2007, к сожалению, плохо это поддерживает.

Итак, чтобы заставить Ubuntu работать в Virtual PC 2007, придется использовать текстовый режим установки, при котором не задействован графический xserver. Нам понадобится альтернативный установочный образ — ubuntu-6.10-alternate-i386.iso, размер которого 713MB.

Второй шаг – в консоли MS Virtual PC выбираем режим создания новой виртуальной машины (рис. 2.1).

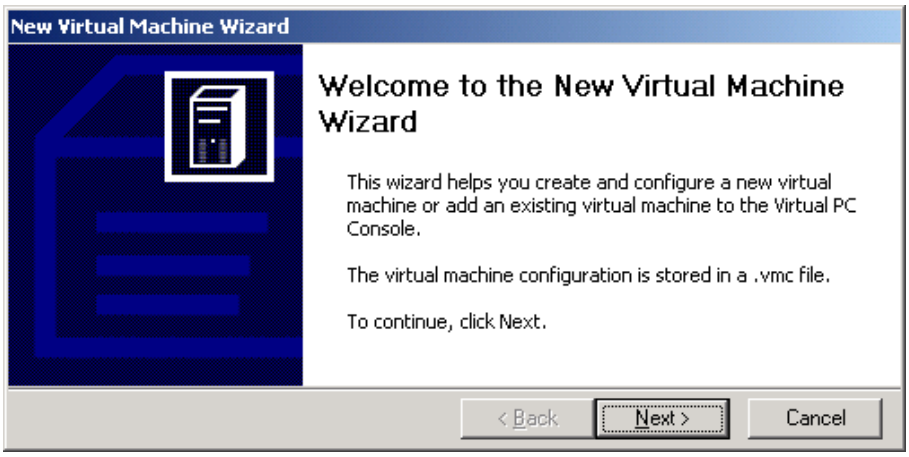

Рис. 2.1. Virtual PC 2007 – Новая виртуальная машина.

В появившемся окне (рис. 2.2) выбираем опцию Create a virtual machine и нажимаем Next.

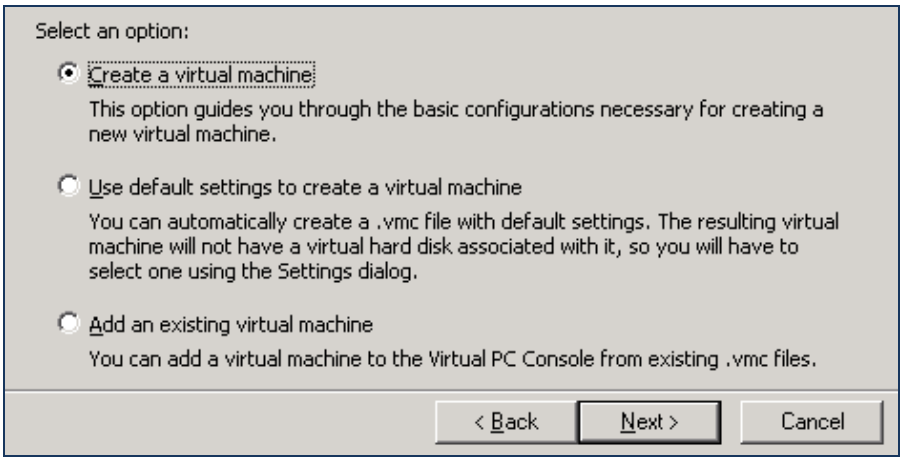

Рис. 2.2. Создаем новую виртуальную машину.

Далее определяем место вновь созданной машины на диске и даем ей какое-либо имя, например «VM_Ubuntu» (рис. 2.3).

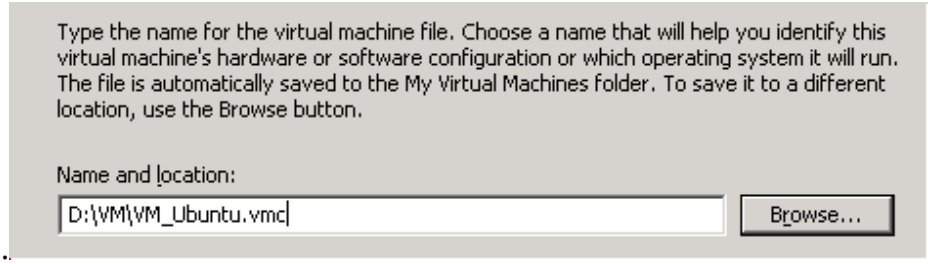

Рис. 2.3. Местоположение новой виртуальной машины.

Из списка возможных для виртуализации операционных систем, выбираем пункт «Other» (рис. 2.4).

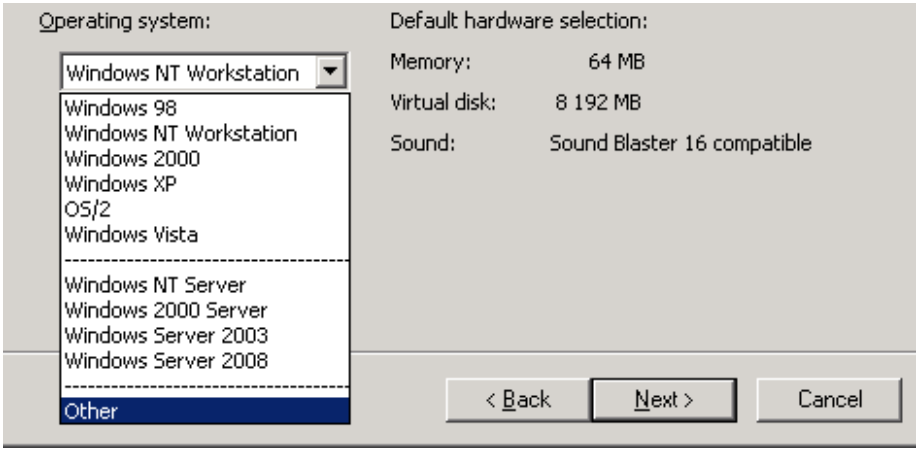

Рис. 2.4. Выбор операционной системы.

Для этого случая Virtual PC Console по умолчанию задает объем оперативной памяти для виртуальной машины — 128MB. Нам нужно как минимум 256 MB, а еще лучше 512 MB (рис. 2.5).

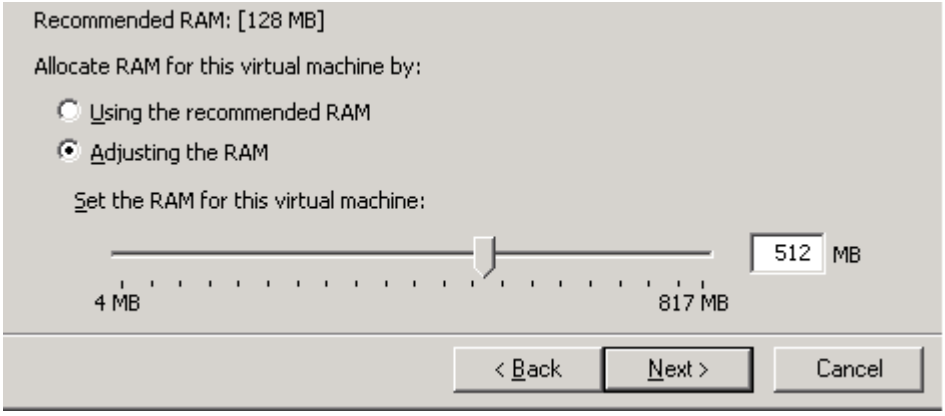

Рис. 2.5. Указание требуемого объема ОЗУ для виртуальной машины.

На следующем этапе надо создать виртуальный жесткий диск для новой виртуальной машины. Именно на нем будет установлена операционная система, а в дальнейшем храниться необходимое для работы программное

обеспечение. На рис. 2.6 определен размер в 4GB, хотя можно и значительно меньше, так как собственно ОС Ubuntu не требует так много места, но сделан резерв для дополнительного программного обеспечения.

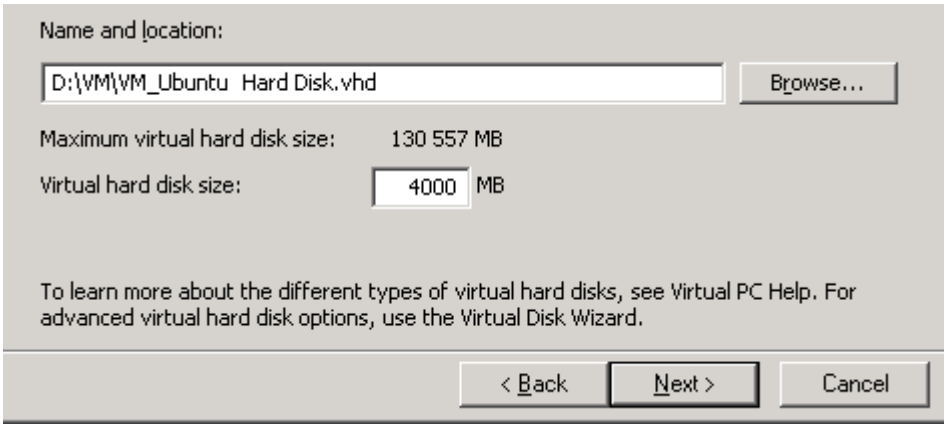

Рис. 2.6. Задание требуемого объема виртуального жесткого диска.

Если не планируется устанавливать обновления и программы, то можно сделать намного меньше. Все это зависит от мощности вашего рабочего компьютера, а так как придется на этом же компьютере виртуализировать и другие ОС, то следует беречь его память.

Третий шаг – мы успешно завершили создание на рабочем компьютере виртуальной машины. Теперь требуется провести ее настройку. Для этого в Virtual PC Console и выбираем режим «Settings».

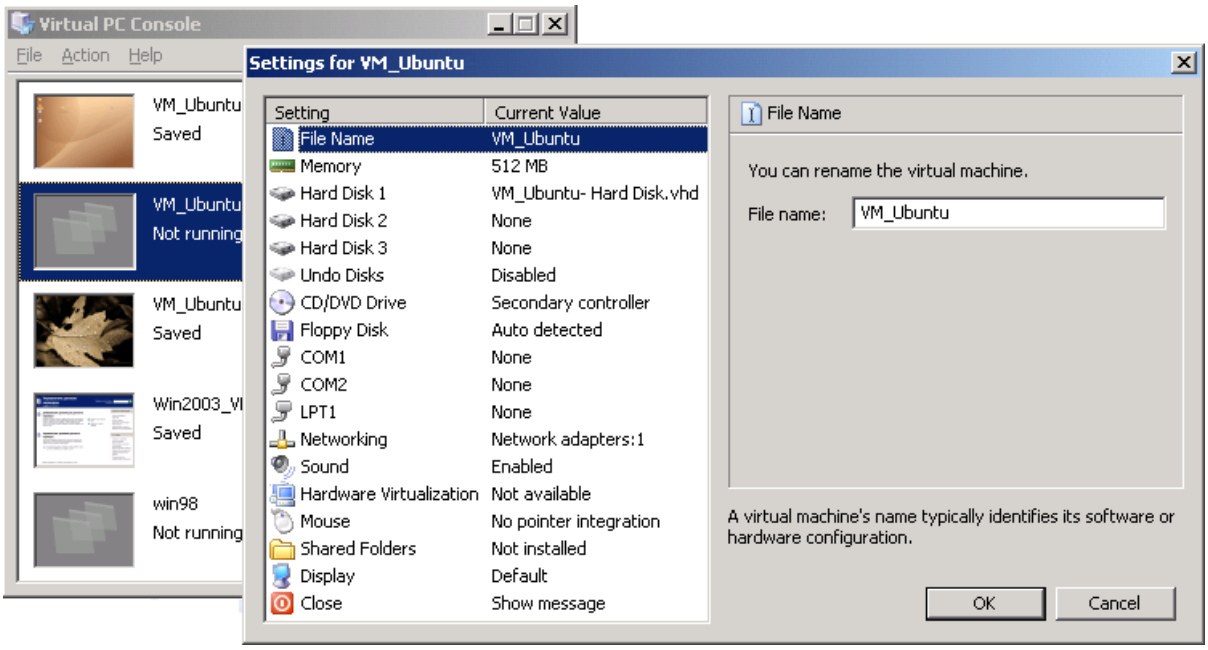

Рис. 2.7. Окно «Setting» для новой виртуальной машины VM_Ubuntu.

В окне «Setting» (рис. 2.7) можно изменить любые характеристики виртуальной машины. Например, отключить дисковод и звуковой адаптер, а также установить нужный режим отображения виртуальной машины на

экране (опция Display). Особо важна для сетевого взаимодействия виртуальных машин настройка их сетевых адаптеров (рис. 2.8).

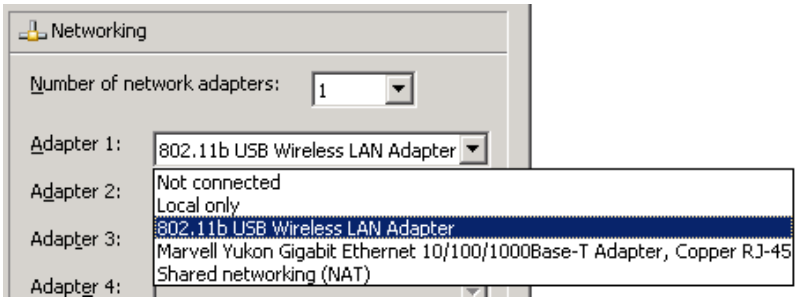

Рис. 2.8. Настройки сетевых адаптеров виртуальной машины.

В Virtual PC 2007 можно подключить до четырех адаптеров, настроив их на разный тип. На начальном этапе изучения Ubuntu можно установить режим «Not connected».

В дальнейшем, при изучении процесса взаимодействия настраиваемой виртуальной машины с другими Windows- или Linux-компьютерами, потребуется выбирать адаптеры Marvell Yukon или NAT, в зависимости от изучаемых режимов.

На данном этапе устанавливаем режим «Not connected» и на этом заканчиваем создание виртуальной машины и переходим к установке Ubuntu в качестве гостевой ОС.

2.2. Установка Ubuntu 6.10 в качестве гостевой ОС

Запустите созданную виртуальную машину. До того, как она загрузится, нажмите на меню CD и выбирете там физический диск, если вы используете образ Ubuntu на CD, который вы вставили в привод, или выбирете «Capture ISO image» и укажите путь к ISO образу Ubuntu. Вы также можете поставить виртуальную машину на паузу, нажав RIGHT_ALT + P. При повторном нажатии пауза снимется.

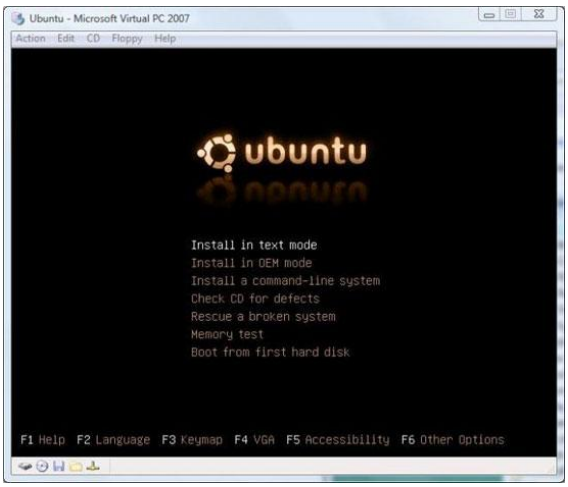

Рис. 2.9. Начальный этап установки Ubuntu 6.10.

Когда загрузится меню, нажмите F4 для настройки VGA. Выберите разрешение, которое хорошо работает на вашем мониторе (рис. 2.10), но убедитесь, что в нем 16-битная глубина цвета (например: 800 × 600 × 16). Затем выберите «Установить в текстовом режиме» и нажмите Enter.

Далее установщик системы будет последовательно выдавать на экран окна с вопросами, ответы на которые не составляют сложности. Вам будет необходимо, последовательно:

- выбрать язык установки и нажать Enter;
- указать страну, где вы живёте и опять нажмите Enter;
- когда появится окно автоматического обнаружения раскладки клавиатуры, выбрать “Нет” и нажать Enter;
- затем выбрать ваш тип клавиатуры и нажать Enter еще раз;
- указать раскладку клавиатуры и нажать Enter.

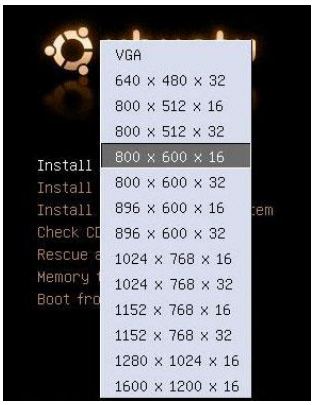

Рис. 2.10. Выбор разрешения монитора.

На следующем этапе Ubuntu проанализирует аппаратную часть и подгрузит необходимые драйверы. Далее появится диалоговое окно, где можно изменить имя хоста (рис. 2.11). Назовите хост как-нибудь типа «VirtualUbuntu» или «vmUbuntu06».

Это необходимо для того, чтобы при сетевых взаимодействиях с виртуальной машиной было понятно, к какому хосту мы обращаемся. Поменяв название, нажмите Enter.

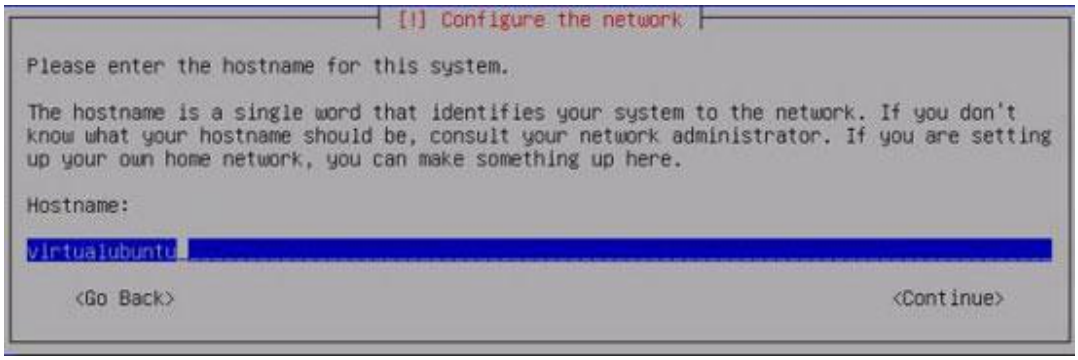

Рис. 2.11. Ввод имени хоста.

При выборе раздела диска, где и как требуется установить Ubuntu, выберите пункт «Erase entire hard disk» (рис. 2.12). Это предполагает, что будет использован весь виртуальный жесткий диск. Ubuntu автоматически разметит его, чтобы записать изменения. Для этого выберите Да и нажмите Enter.

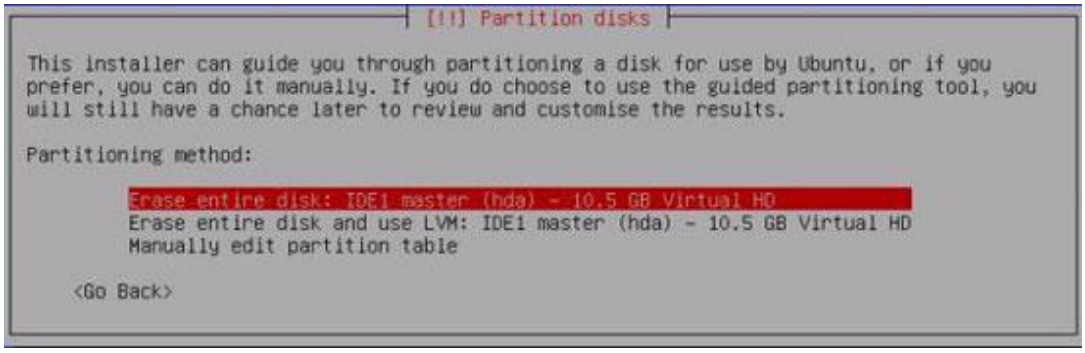

Рис. 2.12. Разметка диска — экран 1.

На следующих этапах будет необходимо:

- Указать нужный город, для настройки требуемого вам для работы часового пояса (рис 2.13).

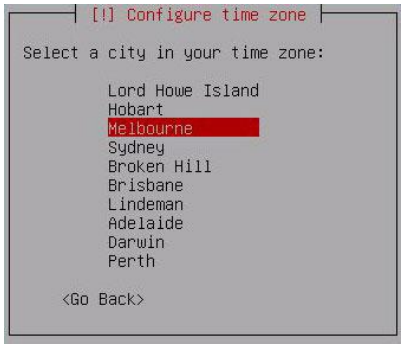

Рис. 2.13. Выбор часового пояса.

- При настройке часов выбрать «Нет» для UTC (Coordinated Universal Time — всемирное координированное время) и нажать Enter.
- ввести полное имя пользователя, который будет создан, и нажать Enter (рис 2.14).

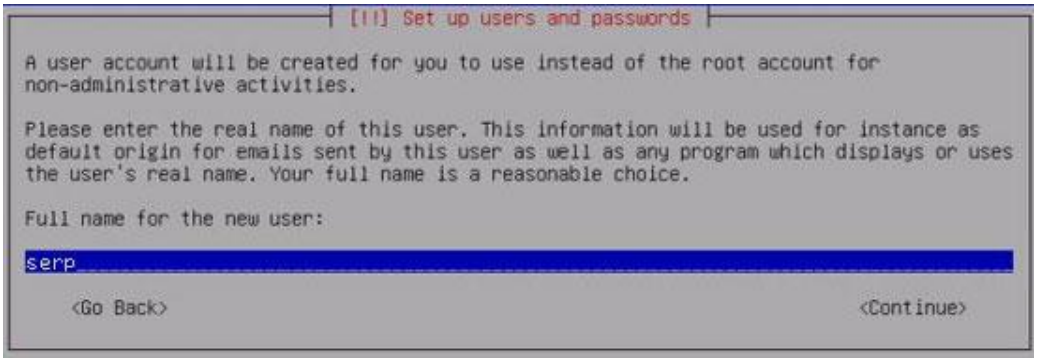

Рис. 2.14. Установка Ubuntu — Имя пользователя.

Затем будет необходимо либо подтвердить стандартный логин, который предложит Ubuntu, либо ввести собственный (без заглавных букв) и нажать Enter (рис. 2.15). Далее нужно будет задать пароль пользователя Ubuntu, нажать Enter, повторно ввести пароль и снова Enter.

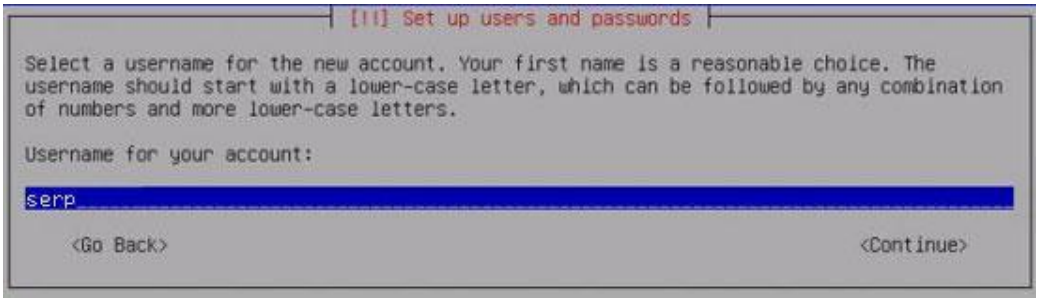

Рис. 2.15. Установка Ubuntu – Логин.

После этого Ubuntu начинает установку базовой системы, различного программного обеспечения и приложений (рис. 2.16).

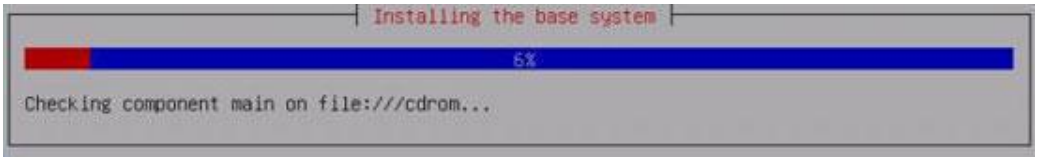

Рис. 2.16. Процесс установки Ubuntu.

Когда появится диалоговое окно для настройки графического xserver, выбирете приемлемое разрешение экрана: 640×480, 800×600 или 1024×768.

Рис. 2.17. Выбор разрешения экрана.

Установив требуемое разрешение, нажмите Enter, и установка продолжится. Это займет немного больше времени, можно успеть выпить чашечку кофе.

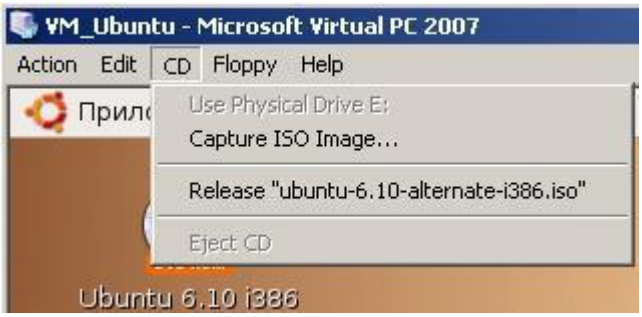

Рис. 2.18. Отключение (Release) ISO-диска в Virtual PC 2007.

Когда установка будет завершена, извлеките физический диск из привода или отмонтируйте виртуальный диск ISO в Virtual PC 2007 (рис. 2.18), после чего нажмите Enter, и начнется перезагрузка системы.

Если установка Ubuntu прошла успешно, то при загрузке вы увидите меню загрузчика GRUB. Если вы загрузите Ubuntu в обычном режиме, то получите искаженный экран, так как ОС будет пытаться работать в 24-битном режиме.

Чтобы это исправить, нажмите клавишу ESC, когда увидите меню GRUB, и выбирете в нем «recovery mode» для загрузки в режиме восстановления (рис. 2.19).

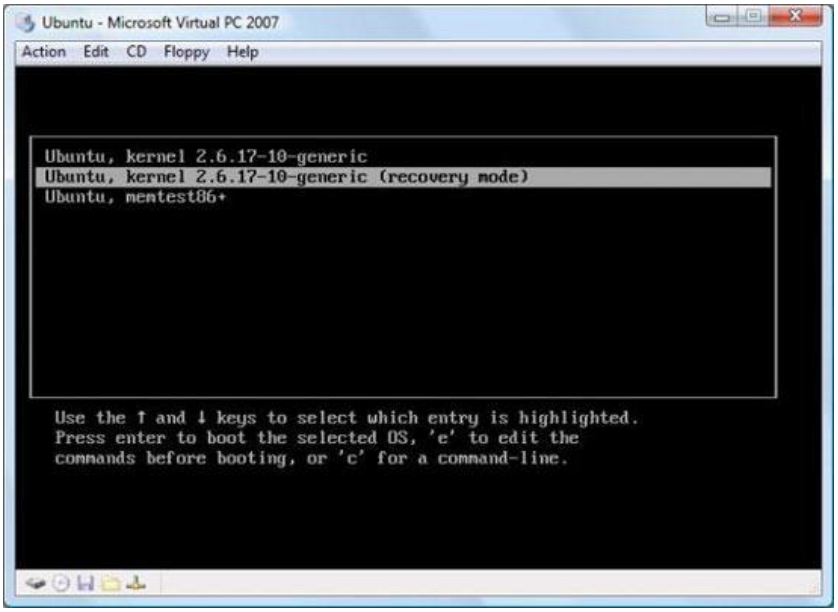

Рис. 2.19. GRUB, режим восстановления.

После загрузки мы можем спокойно работать, используя режим командной строки. Теперь нужно изменить конфигурацию и понизить глубину цвета. С этой целью выполним следующие действия:

➢ Во-первых, сделаем резервную копию конфигурационного файла. Для этого в командной строке введем (рис. 2.20):

```
sudo cp /etc/X11/xorg.conf /etc/X11/xorg.conf.backup
```

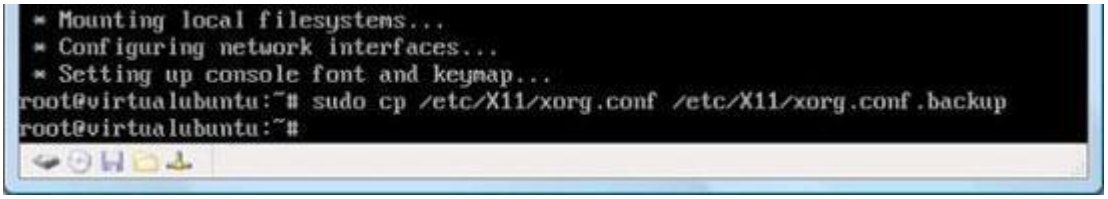

Рис. 2.20. Резервная копия xorg.conf.

Мы создали копию файла xorg.conf, которую назвали xorg.conf.backup. Ее можно будет использовать для восстановления настроек, если что-то будет сделано неверно.

Замечание.
Помните, что имена и команды в Linux чувствительны к регистру, поэтому обязательно надо вводить X11, а не x11.

➢ Во-вторых, отредактируем файл xorg.conf, используя входящий в состав Ubuntu текстовой редактор Nano, который вызовем командой:

```
sudo nano /etc/X11/xorg.conf
```

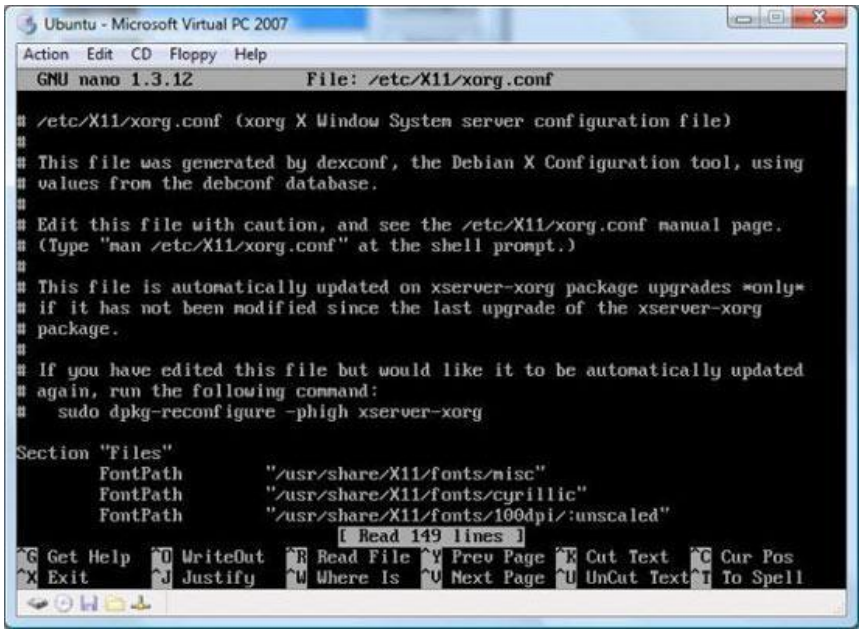

Рис. 2.21. Файл xorg.conf, открытый в текстовом редакторе.

В файле xorg.conf (рис. 2.21) найдем строку, содержащую параметр «DefaultDepth». Для этого надо нажать Ctrl+W и ввести DefaultDepth.

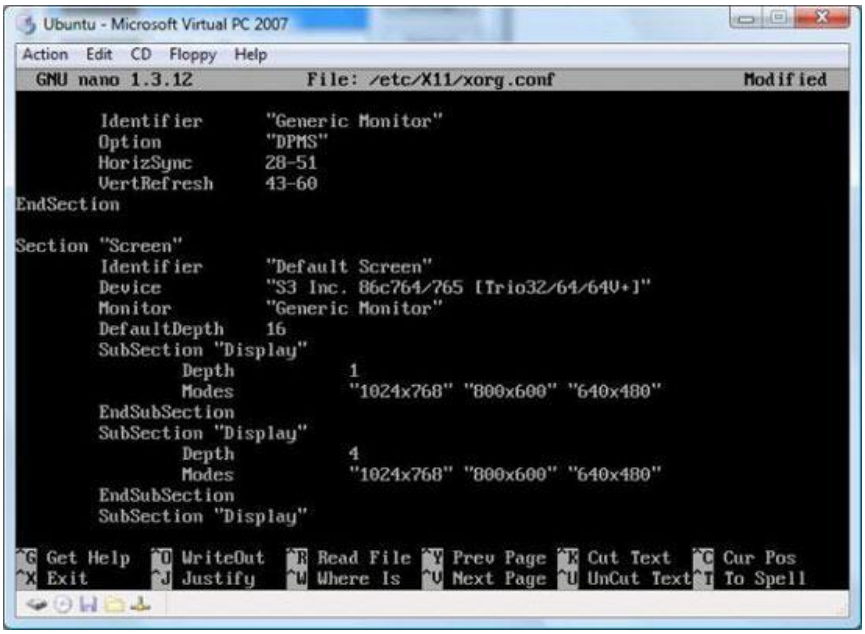

Рис. 2.22. Изменение параметра DefaultDepth.

Редактор Nano переместит курсор к нужной строке, и вы увидите, что у параметра DefaultDepth установлено значение 24. Используя кнопки управления курсором, сотрите 24 и напишите 16 (рис. 2.22).

➢ В-третьих, сохраним сделанные изменения, выйдем из редактора и перезагрузим Ubuntu.

- Для сохранения изменений в файле xorg.conf следует нажать CTRL+O.
- Затем нажать Enter, чтобы перезаписать существующий файл и CTRL+X, чтобы выйти из редактора.
- Для перезагрузки Ubuntu следует в командной строке (рис. 2.23) ввести команду reboot.

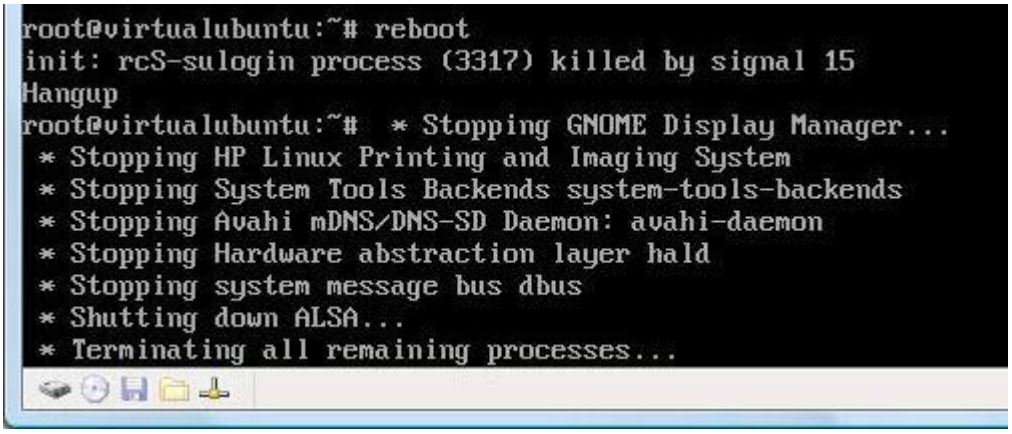

Рис. 2.23. Установка Ubuntu — Перезагрузка.

➢ Теперь загружаем Ubuntu в обычном режиме и можем начинать с ней работать. ОС Ubuntu будет использовать 16-битную глубину цвета.

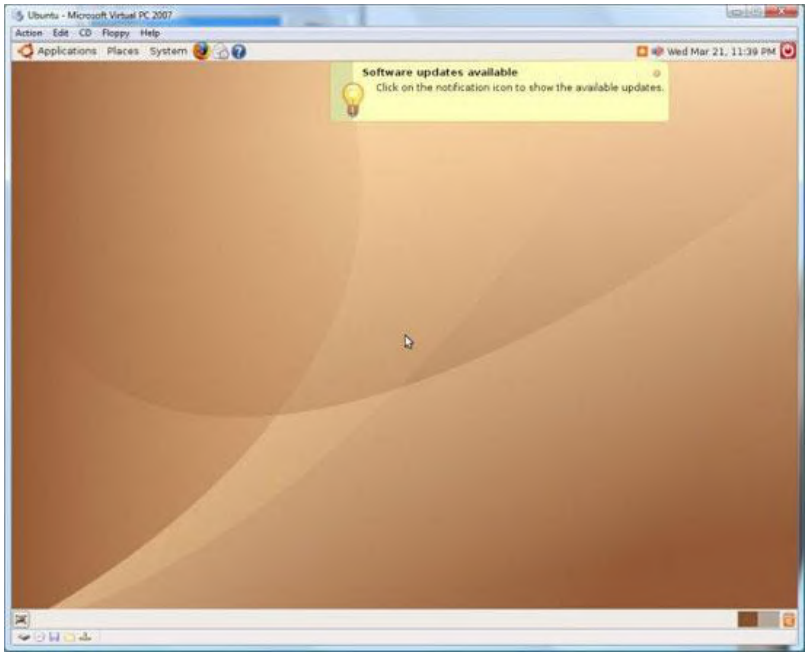

Рис. 2.24. Установка Ubuntu 6.10 завершена.

2.3. Установка Ubuntu 10.04 LTS в качестве гостевой ОС

Виртуализация Ubuntu 10.04 с использованием VMware, как правило, не вызывает затруднений. При использовании бесплатных продуктов Microsoft есть особенности, которые будут рассмотрены в данном разделе. Изложение базируется на статье «Installing Ubuntu 10.4 LTS on Windows Virtual PC on Windows 7» с блога Scott Hanselman.

Он предлагает для виртуализации Ubuntu 10.04 в Windows скачать ее ISO-образ с сайта http://www.ubuntu.com/desktop/get-ubuntu/download. При этом отмечается, что скачивать надо именно CD-образ (32-бит), так как при установке с DVD-образа возникает ошибка.

Для установки Ubuntu 10.04 надо создать новую виртуальную машину, а затем выполнить следующую последовательность действий:

- Запустить виртуальную машину. В ее основном меню перейти в режим CD -> Capture ISO Image -> и выбрать образ установочного диска.
- Перезапустить виртуальную машину (Action -> Reset). Загрузка пойдет с виртуального CD-привода. Как только появится экран, приведенный на рис. 2.25, нажать любую клавишу. Это прервет автоматическую установку.

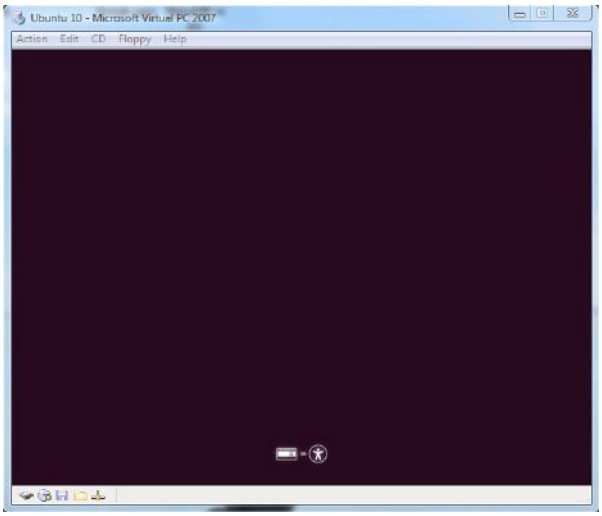

Рис. 2.25. Установка Ubuntu 10.04 на MS Virtual PC.

- На экране «выбора языка установки системы» выбрать язык русский и нажать Enter.
- На экране «меню установки» выбрать первый пункт — «Запустить Ubuntu без установки» (Enter не нажимать). После чего нажать клавишу F6, а затем ESC. Чуть ниже меню откроется строка запуска (рис. 2.26), в которой необходимо заменить параметр:

```
quiet splash   на   vga=791 noreplace-paravirt
```

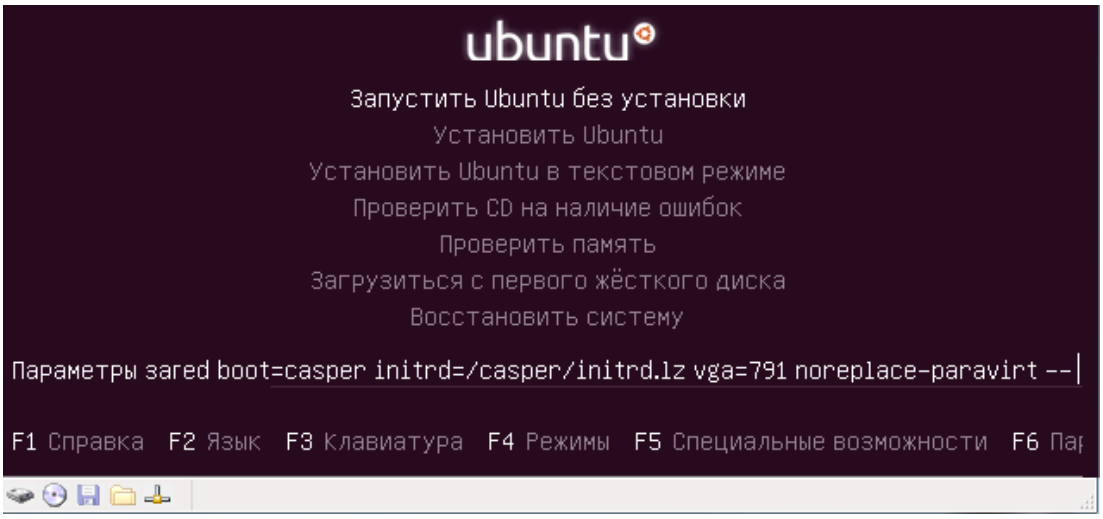

Рис. 2.26. Замена параметра в строке запуска.

Если использовать параметры по умолчанию, то установка будет прервана ошибкой. После исправления строки запуска нажить Enter и через некоторое время вы увидите рабочий стол Ubuntu (рис. 2.27).

Рис. 2.27. Рабочий стол Ubuntu.

- Далее запускаем процедуру установки Ubuntu 10.04 LTC на Virtual PC:
  - выбираем на рабочем столе ярлык – «Установить Ubuntu».
  - если имеется подключение к Интернету, то имеет смысл скачать обновления при установке.
  - на экране «Распределение места на жестком диске» — выбираем «Использовать весь диск».
  - а затем, так как диск у нас один и выбор невелик — нажимаем «Установить сейчас».

Замечание.
Напоминаю, чтобы освободить мышку из виртуальной машины следует использовать клавишу правый ALT.

- На последующих экранах будет необходимо выбрать часовой пояс, раскладку клавиатуры (переключается так же, как и в Windows — левый ALT + SHIFT), задать имя пользователя и пароль (рис. 2.28).

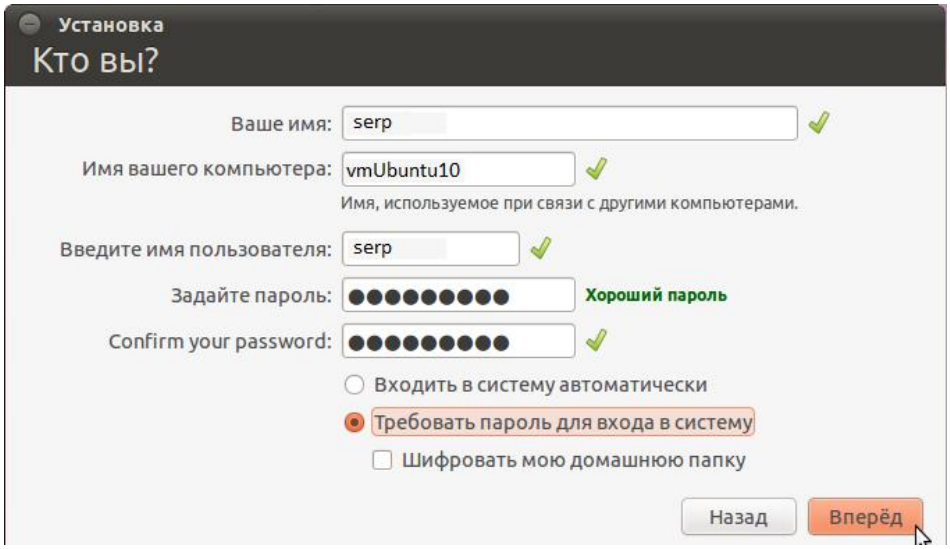

Рис. 2.28. Задание имени и пароля пользователя.

- По нажатию кнопки «Вперед», начнется процесс копирования файлов. По его окончанию появится сообщение (рис. 2.29).

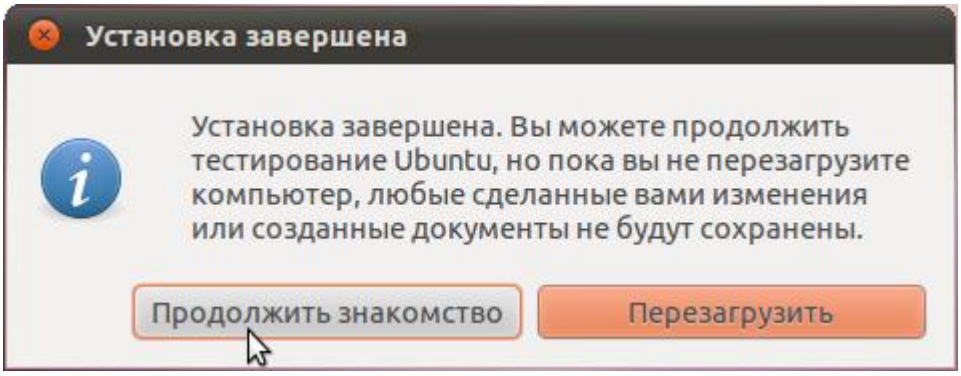

Рис. 2.29. Установка завершена.

Вы можете попробовать перезагрузить установленную систему, но, как уже упоминалось ранее, возможны сбои Ubuntu при работе в MS Virtual PC, и тогда процесс установки придется повторить.

Поэтому Scott Hanselman в своем блоге (http://www.hanselman.com/blog/InstallingUbuntu104LTSOnWindowsVirtualPCOnWindows7.aspx) предлагает более тонкую настройку Ubuntu 10.04 при виртуализации в MS Virtual PC. Рассмотрим этот процесс более подробно.

Внимание!!!
Не перезагружайте сейчас систему! До перезагрузки необходимо внести изменения в системные файлы.

Выбираем опцию «Продолжить знакомство», а затем, используя меню Places -> Filesystem (рис. 2.30), открываем окно файловой системы.

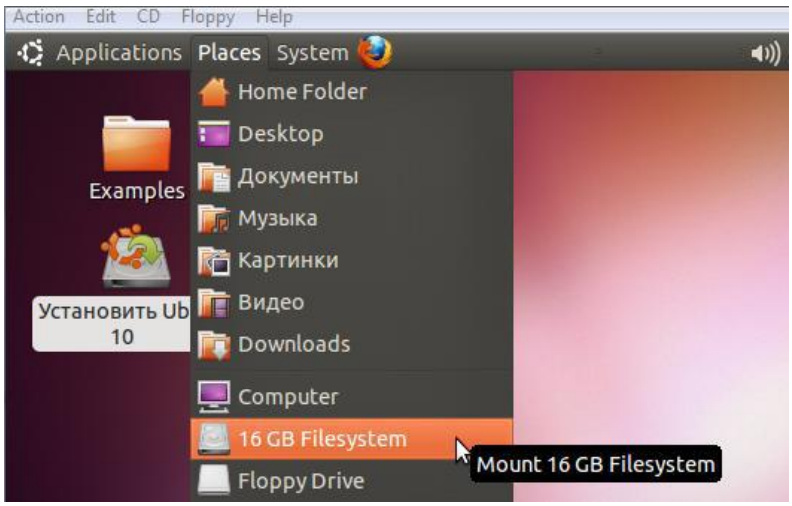

Рис. 2.30. Открываем окно файловой системы.

Далее, используя Applications -> Accessories -> Terminal, открываем окно терминала (рис. 2.31).

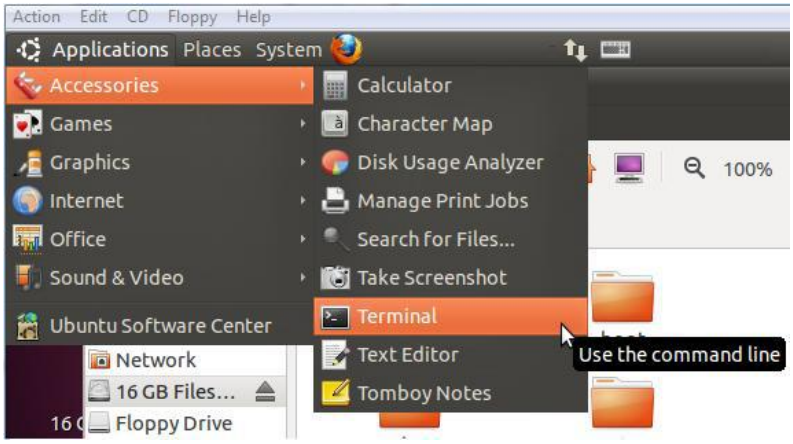

Рис. 2.31. Открываем окно терминала.

Разместите открытые окна на экране так, как показано на следующем скриншота (рис. 2.32) и выполните в терминале следующие команды:

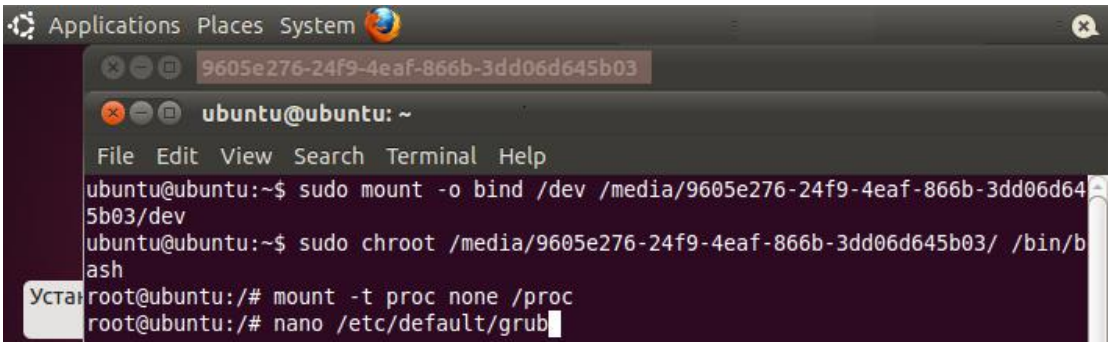

Рис. 2.32. Изменение конфигурации системы.

```
sudo mount -o bind /dev /media/{GUID}/dev
sudo chroot /media/{GUID}/ /bin/bash
mount -t proc none /proc
```

Обратите внимание, что вместо {GUID} надо вписать номер, который отображается в шапке окна Filesystem. В скриншоте на рис. 2.32 этот номер имеет значение 0605e276-24f9-4eaf-866b-3dd06d645b03. У вас, естественно, этот номер будет другой.

Затем, с правами root, запустите текстовый редактор Nano для редактирования конфигурационного файла /etc/default/grub.

```
sudo nano /etc/default/grub
```

В этом файле найдите и закомментируйте символом # строчку, содержащую параметр GRUD_HIDDEN_TIMEOUT, и замените значение параметра GRUB_CMDLINE_LINUX_DEFAULT на vga=791 (рис. 2.33).

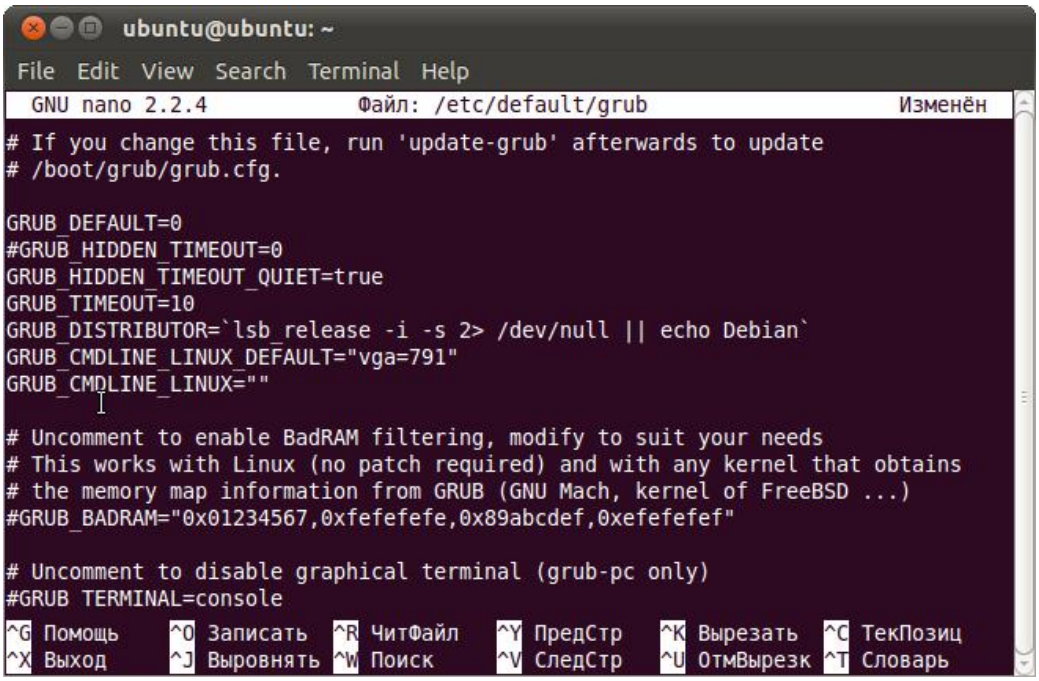

Рис. 2.33. Редактирование файла /etc/default/grub.

Вы можете для параметра «vga» выбрать и другое значение на основе приведенной ниже значений:

Depth	800/600	1024/768	1152/864	1280/1024	1600/1200
8 bit	vga=771	vga=773	vga=353	vga=775	vga=796
16 bit	vga=788	vga=791	vga=355	vga=794	vga=798
24 bit	vga=789	vga=792		vga=795	vga=799

Сохраняем изменения, нажав CTRL+X. Возвращаемся в терминал и вызываем на редактирование файл /etc/grub.d/10_linux, выполняя команду:

```
sudo nano /etc/grub.d/10_linux
```

В текстовом редакторе Nano откроется файл, в котором надо найти строчку с параметром args=″$4″. В этой строке следует после $4 через пробел дописать текст noreplace-paravirt (рис. 2.34).

```
ubuntu@ubuntu: ~
File  Edit  View  Search  Terminal  Help
  GNU nano 2.2.4            Файл: /etc/grub.d/10_linux

    GRUB_CMDLINE_EXTRA="$GRUB_CMDLINE_EXTRA crashkernel=384M-2G:64M,2G-:128M"
fi

linux_entry ()
{
  os="$1"
  version="$2"
  recovery="$3"
  args="$4 noreplace-paravirt"
  if ${recovery} ; then
    title="$(gettext_quoted "%s, with Linux %s (recovery mode)")"
  else
    title="$(gettext_quoted "%s, with Linux %s")"
  fi
  printf "menuentry '${title}' ${CLASS} {\n" "${os}" "${version}"
  cat << EOF
        recordfail
EOF
  save_default_entry | sed -e "s/^/\t/"

^G Помощь    ^O Записать  ^R ЧитФайл  ^Y ПредСтр   ^K Вырезать  ^C ТекПозиц
^X Выход     ^J Выровнять ^W Поиск    ^V СледСтр   ^U ОтмВырезк ^T Словарь
```

Рис. 2.34. Редактирование файла /etc/grub.d/10_linux.

Сохраняем файл (CTRL+X), а затем в терминале вводим команду:

```
sudo update-grub
```

Для перезагрузки Ubuntu в командной строке выполняем:

```
sudo shutdown -r now
```

После окончания перезагрузки система готова к работе (рис. 2.35).

Рис. 2.35. Окончание установки системы.

Система установлена, но хотелось бы предупредить пользователей Windows — графический интерфейс это только вершина айсберга. Вся магия Linux, а стало быть, и Ubuntu, в командной строке! И с этой магией специалисту в области информационных технологий надо быть знакомым.

3. ФАЙЛОВЫЙ МЕНЕДЖЕР И КОНСОЛЬ В UBUNTU

Начиная знакомство с Ubuntu, на первых шагах мы будем использовать ее графическую среду GNOME. Поэтому буквально два слова о терминологии.

GNOME (GNU Network Object Model Environment — сетевая среда объектной модели GNU) — это свободная среда рабочего стола для Unix-подобных операционных систем. Ее разработчики стремятся создать полностью свободную среду, доступную всем пользователям вне зависимости от их уровня технических навыков, физических ограничений и языка, на котором они говорят. В рамках проекта GNOME разрабатываются как приложения для конечных пользователей, так и набор инструментов для создания новых приложений, тесно интегрируемых в рабочую среду.

GNU (GNU's Not UNIX — «GNU — не UNIX») — свободная Unix-подобная операционная система, разрабатываемая Проектом GNU. А среда рабочего стола GNOME является частью этого проекта по созданию целостной Unix-совместимой программной системы.

А теперь перейдем непосредственно к Ubuntu. Как уже отмечалось во введении, основная наша цель — это знакомство с Ubuntu, как с операционной системой класса Linux, а также возможность на примере Ubuntu познакомиться с организацией сетевого взаимодействия Linux- и Windiws-компьютеров между собой.

Поэтому мы оставим в стороне вопросы системного, а тем более прикладного программного обеспечения, а познакомимся только с файловым менеджером и терминалом, которые используются для целей администрирования, как отдельного компьютера, так и сети в целом.

3.1. Использование файлового менеджера Nautilus

Большинство функций, которые используют пользователями при работе с файлами и файловой системой компьютера с ОС Ubuntu, доступны непосредственно из ее графической оболочки. Вызов соответствующих функций доступен из основного меню системы, путем выбора опции «Переход» (рис. 3.1).

Практически все опции меню «Переход» относятся к файловому менеджеру: вы можете перейти в домашний каталог или на рабочий стол, просмотреть список носителей данных (опция «Компьютер»), записать CD/DVD, создать Audio CD, подключиться к ресурсам Windows-сети, найти файлы и просмотреть список недавно использованных документов.

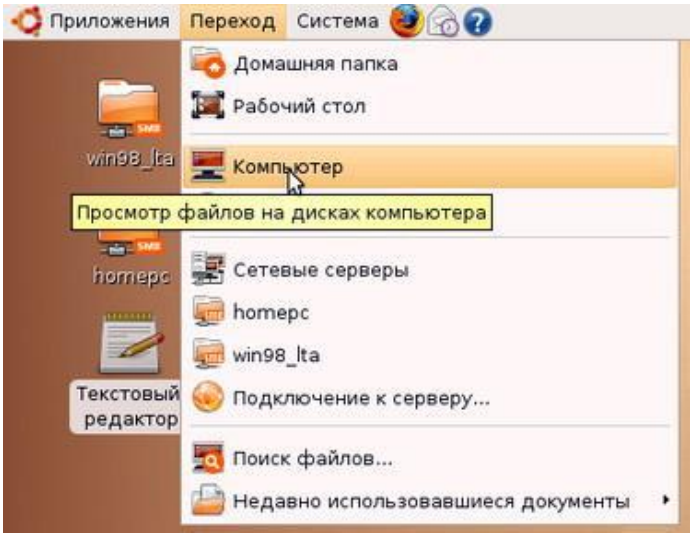

Рис. 3.1. Основное меню "Переход".

Для того, чтобы запустить «Обозреватель файлов» достаточно в основном меню «Переход» выбрать опцию «Компьютер», после чего экран дисплея будет иметь вид, аналогичный приведенному на рис. 3.2.

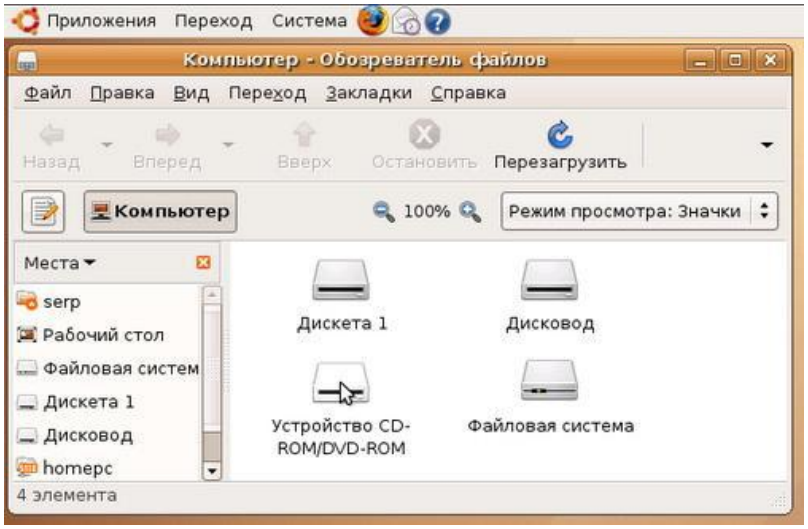

Рис. 3.2. Обозреватель файлов Ubuntu 6.10.

Использовать файловый менеджер очень просто: если вы умеете работать с Проводником Windows, тогда и с Обозревателем файлов вы тоже справитесь.

Все операции с файлами и каталогами, как в Проводнике Windows, так и в Обозревателе файлов Ubuntu производятся по принципу «выделил, скопировал (вырезал), вставил» (рис.3.3).

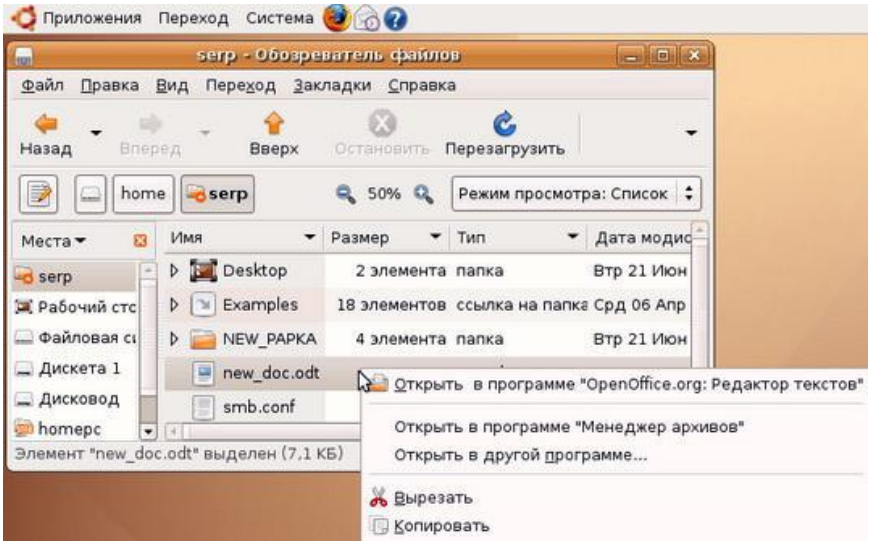

Рис. 3.3. Меню стандартных операции для конкретного файла.

Если щелкнуть на файле или каталоге правой кнопкой мыши, а потом выбрать команду «Свойства» и перейти на вкладку «Права», то вы сможете легко устанавливать права доступа к файлу или каталогу, не прибегая к использованию команды chmod (рис. 3.4).

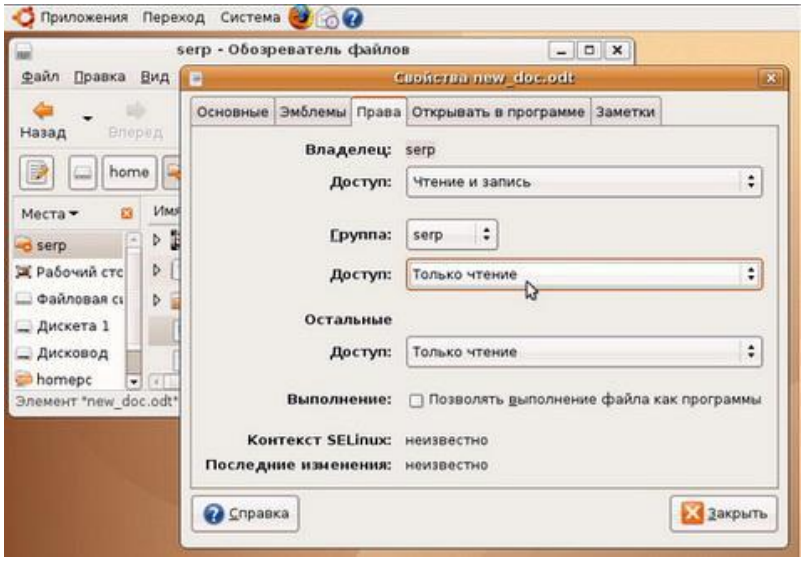

Рис. 3.4. Окно определения прав доступа к файлам и папкам.

Файловый менеджер это, пожалуй, самая главная программа в системе, поскольку она позволяет управлять данными на вашем компьютере. Интерфейс Nautilus (это название файлового менеджера Ubuntu), как следует из вышеизложенного, весьма понятен и прост. Этого было бы и достаточно, но можно отметить еще ряд сервисных возможностей файлового менеджера, которые могут оказаться полезными в работе.

- Боковое меню.

Обратите внимание на боковую панель Nautilus. В ней по умолчанию открыто меню "Места". Если вы внимательно присмотритесь к нему, то

обнаружите, что большинство пунктов совпадают с меню "Переход" системы. И это не случайно, так как оба меню связаны. Мало того, можно добавлять собственные пункты в боковую панель "Места" и, соответственно, в меню "Переход".

Для этого достаточно перетащить нужный каталог из основной области Nautilus в меню «Места» под горизонтальную черту. Если это сделать для некоторой папки с названием NEW_PAPKA, то выбрав в основном меню системы опцию «Переход», можно увидеть там эту папку (рис. 3.5). Сравните рис. 3.1 и рис. 3.5, и вы убедитесь в этом.

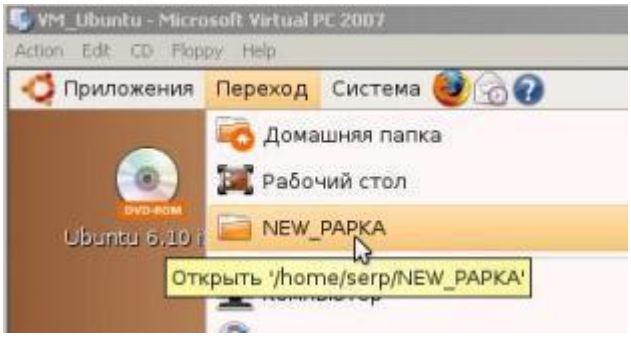

Рис. 3.5. Добавление новой опции в меню «Переход» системы.

➢ Кроме боковой панели «Места» в Nautilus доступно еще несколько других панелей (рис. 3.6).

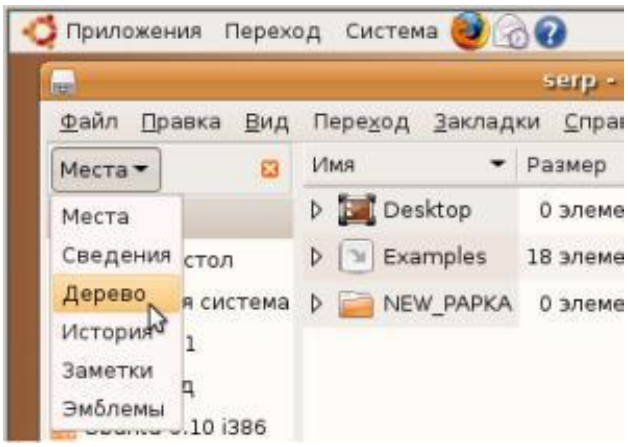

Рис. 3.6. Доступные боковые панели в Nautilus.

Обратим внимание на панель «Эмблемы». Эмблемы — это маленькие бирки, которые можно навешивать на файлы и папки для выделения их среди себе подобных. Для добавления эмблемы просто перетащите ее с боковой панели на файл или папку, для удаления перетащите еще раз. Управлять эмблемами также можно через свойства файла, которые можно изменить, нажав правой кнопкой мыши на нужном файле и выбрав пункт «Свойства».

➢ В Ubuntu, как и Windows предусмотрено несколько полезных сочетаний клавиш, используемых для управления файлами и папками:

Ctrl+C	– копировать выделенные объекты в буфер обмена.
Ctrl+V	– вставить объекты из буфера в текущую папку.
Ctrl+X	– вырезать выделенные объекты в буфер.
Ctrl+Shift+N	– создать новый каталог.
F2	– переименовать выделенный файл/каталог.
Del	– удалить выделенные объекты в корзину.
Shift+Del	– удалить выделенные объекты безвозвратно.

- Выделения нескольких файлов.

Для выделения нескольких файлов подряд надо, удерживая Shift щелкнуть левой кнопкой мыши по первому и последнему файлу, а для выделения файлов в разных местах текущего каталога надо, удерживая Ctrl, щелкнуть по каждому. Таким же образом можно выбирать файлы, используя не мышь, а пробел и стрелки на клавиатуре.

- Способ отображения содержимого папок.

Это достаточно полезная функция, про которую некоторые пользователи, даже не догадываются. Она позволяет изменять способ отображения содержимого папок. Для этого используют следующие сочетания клавиш:

Ctrl+1	– просмотр в виде значков.
Ctrl+2	– просмотр в виде списка.
Ctrl+3l	– просмотр в компактном виде.

Для того чтобы поменять способ отображения для всех папок, надо в меню «Правка» выбрать пункт «Параметры» и в открывшемся окне изменить вид по умолчанию.

- Начиная с Ubuntu 10.04, появился режим двухоконного просмотра.

Ранее этого режима недоставало многим пользователям. Теперь нажатие F3 разделит окно файлового менеджера на две части (рис. 3.7).

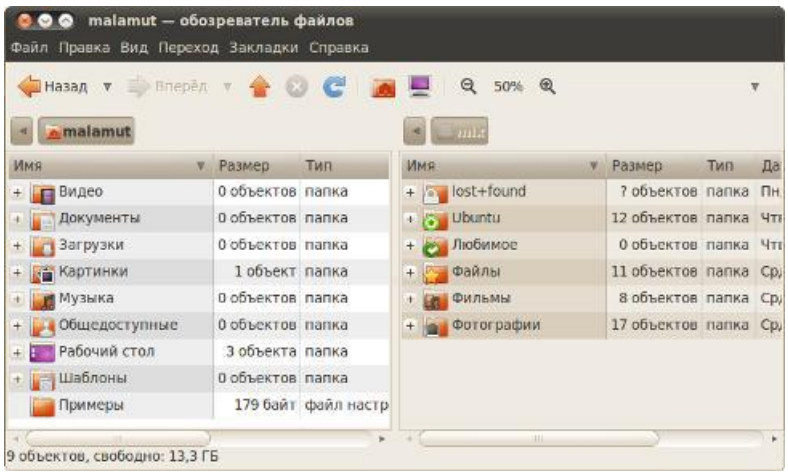

Рис. 3.7. Двухпанельный Nautilus в Ubuntu 10.04.

В Ubuntu, как и Windows, существуют скрытые файлы. Скрытые они потому, что по умолчанию не отображаются при просмотре папок. Нужны

они в основном для хранения различных пользовательских настроек, которые не должны мешаться при работе.

Замечание.
Чтобы сделать файл или папку скрытой надо всего лишь в начало названия добавить символ точки. А чтобы увидеть скрытые файлы, следует нажать Ctrl+H.

Можно заглянуть в свою домашнюю папку и посмотреть, сколько в ней скрытых элементов. Именно в этих элементах хранятся все ваши настройки операционной системы.

На этом закончим краткий обзор файлового менеджера Nautilus, рекомендуя более полное знакомство по литературным и Интернет-источникам, и переходим к вопросам управления системой и доступом к файловой системе из терминала и консоли.

3.2. Назначение и использование Терминала

Почему UNIX и Linux ранее отталкивали обычных пользователей? Потому что не было хорошего графического интерфейса и в Linux работали одни профессионалы. Сейчас все изменилось. В Linux очень удобный графический интерфейс, который с удовольствием используют и профессионалы, иногда забывая о командной строке. Дистрибутив Ubuntu вообще ориентирован на работу в графическом режиме, а в официальных руководствах, которые можно найти в Интернете, о консоли часто вообще не упоминается. А ведь она есть!

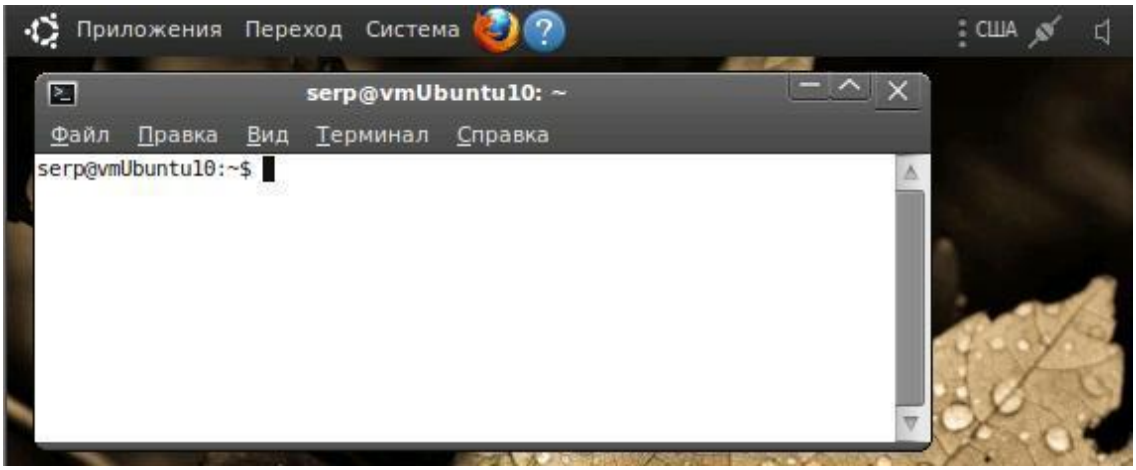

Рис. 3.8. Терминал в Ubuntu 10.04.

То, с чем мы познакомимся в этом разделе, на самом деле не консоль, и не терминал, а просто графический эмулятор. Но, если он так называется в меню, значит так пусть и будет. Давайте сначала посмотрим, что собой представляет этот самый терминал. Найти его можно в меню Приложения –> Стандартные –> Терминал. После того, как он будет запущен, на экране

появится окно, аналогичное тому, что приведено на рис. 3.8. В терминале отображается «приглашение командной строки» и мигающий курсор вслед за ним, приглашающий ввести команду. В приглашении командной строки:

```
serp@vmUbuntu10:~$
```

- Первым идет логин пользователя, который будет вводить, и запускать команду (в примере — это serp).
- Через символ @ от логина идет имя хоста, на котором работает этот пользователь. В приведенном примере — это имя созданной нами виртуальной машины vmUbuntu10.
- Далее через двоеточие указывается текущая директория, в которой находится пользователь, а после символ $ либо #. Второй вариант отображается в том случае, если вы работаете от имени суперпользователя.

При старте терминала текущая директория — это домашняя директория пользователя, которая обозначается символом «~».

Аналогом такой директории в Windows является папка C:\Documents and Settings\логин\My Documents. Если до этого вы работали только с Windows, то придется немного изменить «мировоззрение» относительно файловой системы.

В Linux-системах обычно личные файлы хранятся в домашней директории пользователя. Домашняя директория пользователя это такая директория, в которой пользователь является хозяином и может делать там все, что угодно. Путь к этой директории:

```
/home/логин/
```

Например, в рассматриваемом примере это директория: /home/serp/. Все остальные директории нужны для других целей. Например /bin/ — содержит исполняемые файлы, /root/ — является домашней директорией суперпользователя, /boot/ — используется загрузчиком и т. д.

Поэтому всякий раз, когда вы запускаете консоль, вы попадаете в домашнюю директорию. Мало того, вы не будете иметь доступ на запись к другим директориям, если только не будете действовать от имени суперпользователя.

Замечание.
Запомните, что символ "~" служит для обозначения домашней директории пользователя.

Работая в терминале, вы можете работать с разными каталогами своей операционной системы, писать управляющий скрипт, подключаться к удаленному компьютеру сети, администрировать сервер баз данных и многое другое. Но что делать, если все перечисленное, вы должны делать одновременно. С этой целью в терминале можно открыть сразу несколько

вкладок (рис. 3.9) и в каждой из них выполнять разные задачи с разными логинами и на разных хостах.

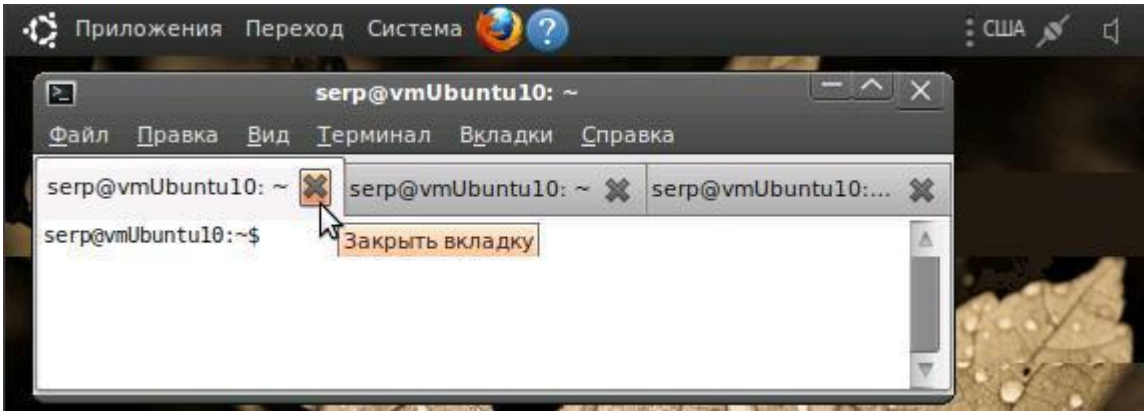

Рис. 3.9. Совокупность вкладок Терминала Ubuntu.

Заголовок каждой вкладки отображает логин, хост и текущую директорию каждого из профилей работы. Быстрый переход между вкладками доступен с помощью клавиш «Alt» + N, где N — номер вкладки, имеющий значение от 0 до 6.

Терминал создан для того, чтобы выполнять текстовые команды, поэтому можно отложить мышку в сторону и пододвинуть поближе клавиатуру. Предположим, что нам надо выполнить какую-нибудь команду, типа:

```
lsb_release -a 2> /dev/null | grep -P "(?<=Codename:)(.*)"
```

Набирать такие команды с клавиатуры посимвольно немного неудобно, поэтому есть резон познакомиться с основами управления терминалом.

Управление терминалом

Начнем знакомство с управлением терминалом с наиболее часто используемой операции – это операция копирования/вставки. Стандартные для Windows сочетания клавиш «Ctrl+C» и «Ctrl+V» в терминале не работают. Вместо них используется сочетание двух других клавиш, а именно «Ctrl+Insert» и «Shift+Insert» или сочетания «Ctrl+Shift+C» — для копирования и “Ctrl+Shift+V” — для вставки.

Но эти сочетания удобны для копирования и вставки некоторых фрагментов текста. Что же касается команд, то обычно команды набираются вручную, а не вставляются откуда-то. И тут на помощь может прийти великолепное свойство терминала, которое называется автодополнением.

Наберите в терминале символы apti, а потом нажмите клавишу Tab. Терминал автоматически дополнит за вас команду. Кстати, aptitude — это основная консольная утилита управления установкой и удалением приложений, но об этом после.

Если же набрать только apt и нажать Tab — ничего не происходит. Но если нажать Tab два раза подряд, то терминал выдаст список всех команд, начинающихся с apt (рис. 3.10).

Автодополнение удобно, если привыкнуть. Автодополнение в терминале работает практически везде, и не только для команд, но также для их аргументов и имен файлов. Поэкспериментируйте с ним, оно значительно сокращает время набора, да и вообще, терминал без автодополнения — это не терминал.

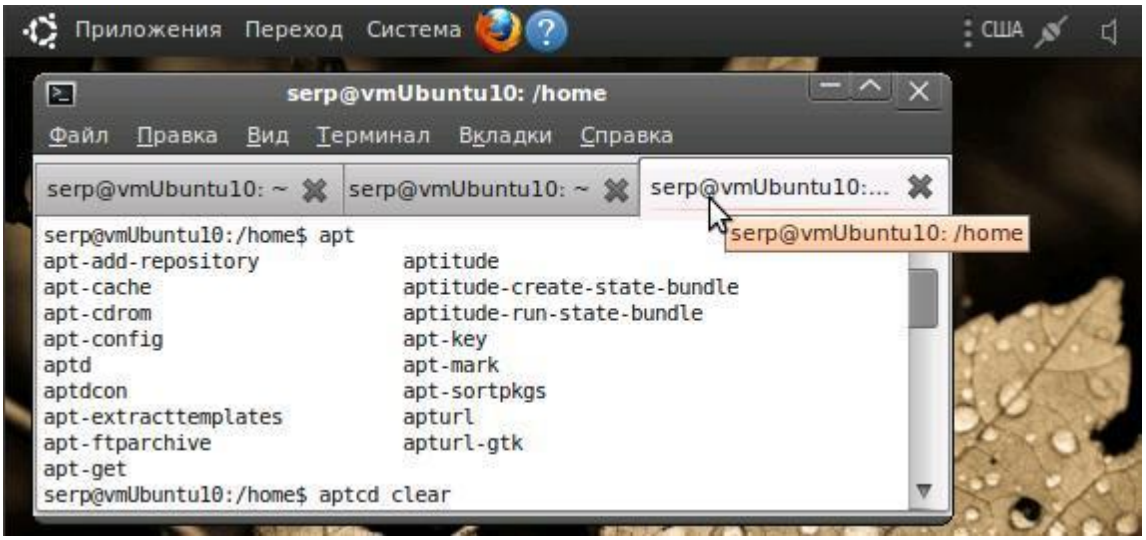

Рис. 3.10. Использование автодополнения при вводе команд.

Все вводимые команды терминал запоминает. При вводе новой команды можно, используя стрелки вверх и вниз, пролистать команды из буфера, выбрать наиболее подходящую, отредактировать ее и ввести как новую.

Посмотреть всю историю можно командой history. У каждой команды в истории есть номер, выполнить снова команду с определенным номером можно, набрав в терминале восклицательный знак и номер нужной команды. А повторить предыдущую набранную команду можно, просто написав два восклицательных знака:

```
!!
```

Бывает так, что вы что-то запустили в терминале и хотите прервать работу этого чего-то. Обычно это сделать очень просто, достаточно нажать на клавиатуре сочетание клавиш Ctrl+C.

Есть и другие управляющие сочетания, например Ctrl+D посылает сигнал конца файла запущенному приложению, а без запущенных утилит делает то же, что и терминальная команда exit.

Ну а если вы хотите более подробно управлять работающими программами, то посмотрите на системный монитор htop, который, правда, нужно в систему устанавливать отдельно.

Работа с файлами

При запуске терминала текущей директорией является домашний каталог пользователя, но потом, конечно, ее можно поменять. Узнать, в какой папке вы находитесь, очень просто. Достаточно посмотреть на приглашение терминала, то есть на символы, которые печатаются в начале каждой строки.

Имена директорий, как и файлов в Linux, чувствительны к регистру символов, то есть директории «Музыка» и «музыка» — это две совершенно разных директории.

Для смены директории предназначена команда cd (*Change Directory* — сменить директорию), а для создания новой — команда mkdir (*Make Directory* — создать директорию).

После команды cd можно указывать как полные пути относительно корня, так и относительные, отсчитывающиеся от текущего каталога. Заменитель адреса домашнего каталога «~» можно использовать при наборе путей, например:

```
cd ~/Музыка
```

Для перемещения непосредственно в домашний каталог достаточно просто набрать cd без аргументов, а для перемещения на каталог выше следует использовать команду:

```
cd ..
```

Две точки, указанные в команде через пробел от cd, обозначают всегда родительский каталог. Но добравшись до нужного каталога, хотелось бы узнать его содержимое.

Используя команду ls (*List* — список), можно просмотреть содержимое текущего каталога (рис. 3.11).

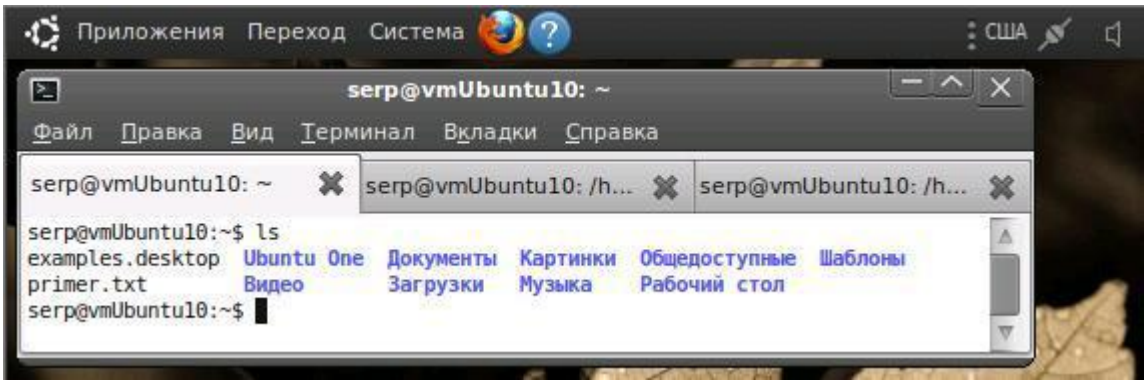

Рис. 3.11. Пример использование команды ls.

Обычно командам можно передавать различные модификаторы, а также указывать в командах дополнительные опции. Так, например, дополнительная опция в команде ls позволит получить более подробную информацию о содержимом текущего каталога (рис. 3.12).

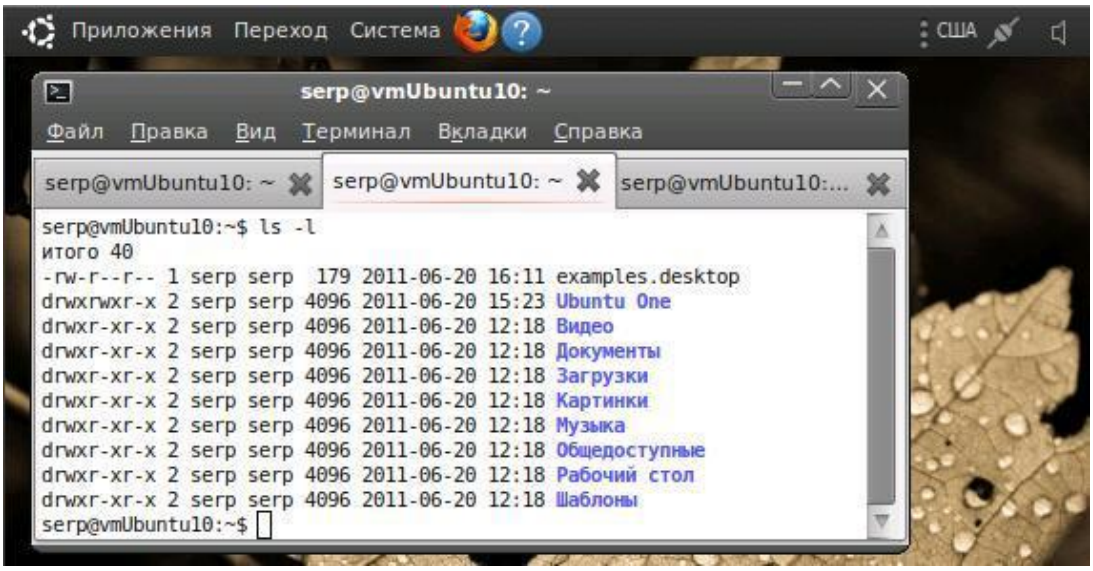

Рис. 3.12. Пример использование команды ls с параметром.

Очень часто параметрами команд являются имена файлов или папок. Например, команда cat показывает содержимое текстового файла. Если надо посмотреть содержимое файла primer.txt, находящего в домашнем каталоге, то надо выполнить команду:

```
cat ~/text.txt
```

Получение справки

Термин ман (мануал) берет начало от английского user manual, обозначающего документ, назначение которого состоит в предоставлении помощи по использованию некоторой системы и который входит в состав технической документации на эту систему.

Такая система помощи и справки предусмотрена и для пользователей Linux. Дело в том, что man — это система справки о командах Linux, которые можно вводить в терминале. Пользоваться ей очень легко, просто надо в терминале набрать:

```
man <Команда>
```

Например, если ввести man ls, то появится содержимое справки по команде ls, разбитое на отдельные разделы. Перемещаться по нему можно с помощью стрелок и клавиш PgUp и PgDown, а для выхода достаточно нажать клавишу Q. Кроме man-страниц у многих утилит есть встроенная справка, которую обычно можно посмотреть, запустив программу с ключом --help:

```
<Утилита> --help
```

Например:

```
ls -help
```

Есть и другие способы получения помощи, например, похожая на man утилита info. Но чаще всего наиболее полную информацию о программе можно получить именно из man-страниц, а краткую справку — указав ключ --help при вызове.

С непривычки все описанное может показаться дремучим лесом, а на самом деле это только самая верхушка айсберга, существуют еще тысячи полезных команд и интересных приемов работы в терминале. С помощью терминала можно редактировать файлы, слушать музыку, смотреть видео и выполнять массу сетевых и административных функций.

3.3. Текстовая консоль

Настоящий линуксоид должен уметь работать в консоли, а начинающий сетевой администратор хотя бы иметь о ней представление. Ведь когда появился Linux, о графическом интерфейсе не было и речи. Стоит вам поработать в консоли, и вы поймете все ее преимущества. В консоли можно выполнять те же операции, что и в графическом режиме, причем все намного быстрее.

На слабеньких компьютерах консоль позволяет эффективно использовать ресурсы компьютеров. Да, в графическом режиме на стареньком «Пентиуме» не поработаешь, зато в текстовом режиме его можно быстро превратить в очень полезный для всей сети компьютер — в шлюз, через который его более мощные собратья будут получать доступ к Интернету. А если в такой же компьютер вставить несколько сетевых плат, то можно получить управляемый маршрутизатор.

Кроме того, подключаясь удаленно к почтовым или Web серверам для их администрирования, консоль позволяет существенно экономить сетевой трафик и ускорять работу. Во многом консольный режим может помочь вам и при работе на локальном компьютере с Ubuntu.

Для перехода из графического режима в консольный режим достаточно нажать комбинацию клавиш:

```
Ctrl+Alt+Fn,
```

где Fn — одна из функциональных клавиш F1, F2, ..., F6. При вводе одной из этих комбинаций, экран монитора превратится в «черную дыру», которая запрашивает login пользователя (рис. 3.13).

Если в ответ на приглашение, ввести зарегистрированные в системе имя и пароль пользователя, то система выдаст некоторую справочную информацию и перейдет в консольный режим, отобразив «приглашение командной строки».

Так, например, при использовании Ubuntu, как гостевой системы в MS Virtual PC экран дисплея будет иметь вид, аналогичный рис. 3.13. В этом режиме допустим ввод любых консольных команд Ubuntu.

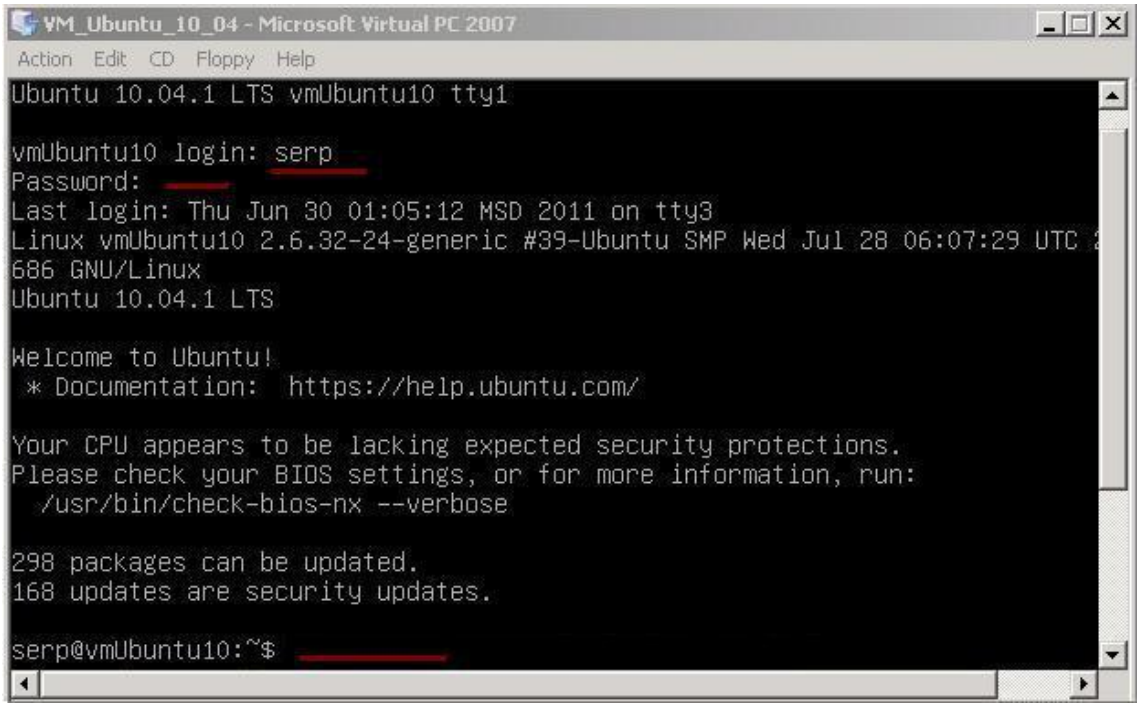

Рис. 3.13. Запуск консоли Ubuntu в Microsoft Virtual PC

Если у вас было открыто несколько консолей, то для переключения между ними следует использовать комбинации клавиш Alt+Fn.

Для возврата в графический режим работы Ubuntu из любой ее консоли используется комбинация клавиш Alt+F7.

Работа в консоли заключается в наборе нужной команды и нажатии Enter. В качестве примера на рис. 3.14 приведена последовательность из трех простейших команд:

- ввод текстовой информации во вновь создаваемый файл primer.txt;
- просмотр содержимого этого файла primer.txt;
- и просмотр содержимого домашнего каталога для контроля наличия нового файла в этом каталоге.

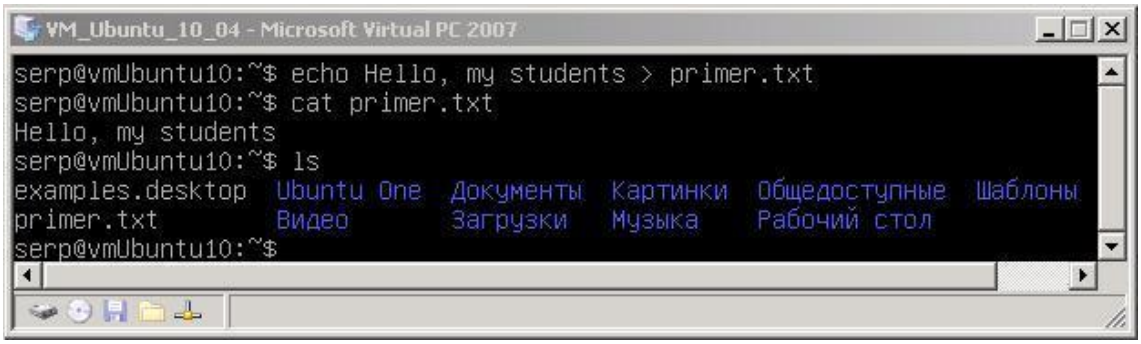

Рис. 3.14. Протокол выполнение трех команд в консоли Ubuntu

До появления Linux в основном использовали DOS. Она состояла из нескольких файлов, один из которых назывался command.com и загружался при старте системы.

Этот файл был командным интерпретатором и позволял выполнять инструкции, написанные пользователем. В число стандартных команд входили такие, как copy, dir и другие. Они предназначались для управления файлами.

Помимо стандартных команд можно было выполнить внешние, которые не входили в состав командного интерпретатора и по своей сути уже являлись программами с расширением com и exe. Например: FORMAT, CHKDSK и так далее.

Консоль или терминал Ubuntu, а точнее shell — это тоже командный интерпретатор. В разных дистрибутивах Linux поставляются различные интерпретаторы, но в большинстве случаев используется bash (рис. 3.15).

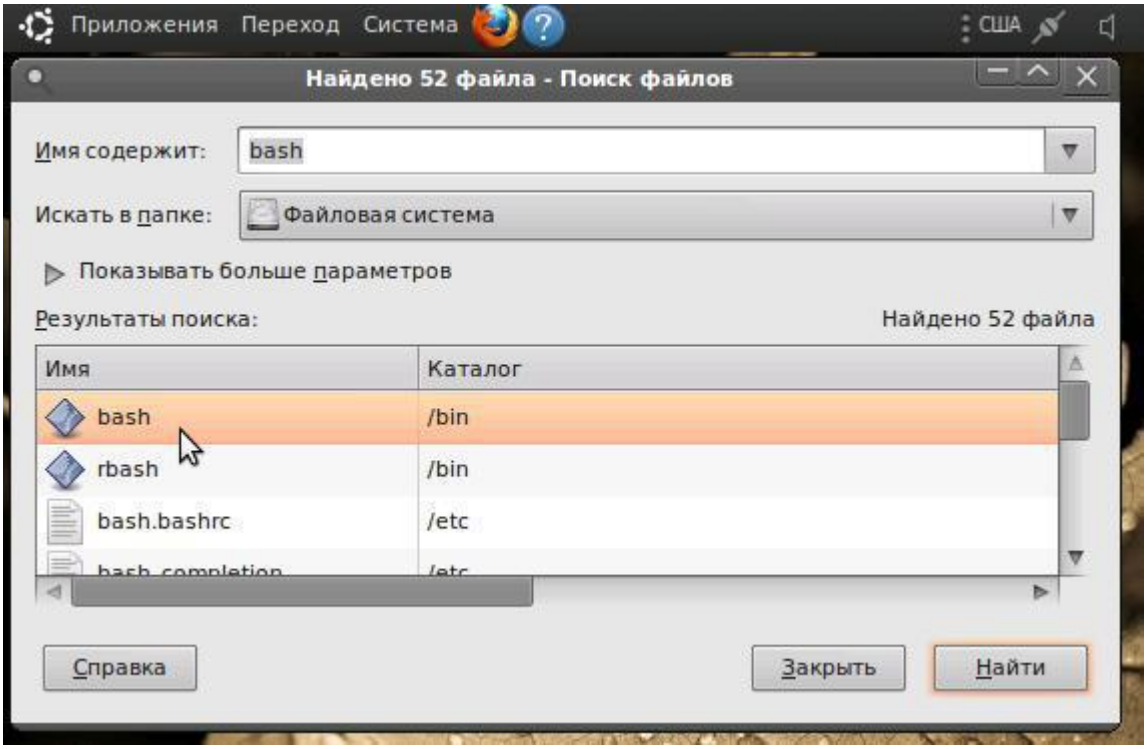

Рис. 3.15. Поиск файла bash в файловой системе Ubuntu.

С дистрибутивом ОС Ubuntu поставляется пакет core-utils (или gnu-core-utils) и другие. Именно они содержат набор программ, таких как cp, rm, mkdir, cd, ls, ln, uname, logname и так далее.

Список довольно внушительный. Убедиться в их существовании можно, перейдя в каталоги /bin или /sbin. Есть и другие каталоги, в которых они тоже могут располагаться, такие как /usr/bin или /usr/sbin.

Данные программы можно с легкостью заменить на свои собственные или переименовать их названия в более удобные для вас. Допустимо их переименование и на русский язык.

Все программы подряд начинающему пользователю изучать не стоит, но есть несколько из них, знание которых и применение которых весьма полезно:

➢ sudo

Команда предназначена для запуска другой программы с правами суперпользователя. Введите в консоли sudo и через пробел название программы. Например, введя

```
sudo nautilus
```

и нажав ENTER, вас попросят ввести пароль. Учтите, что при наборе он не отображается. После ввода пароля нажмите ENTER, и откроется файловый менеджер Nautilus, в котором можно будет создавать или редактировать какие-либо файлы там, где раньше это было запрещено.

➢ killall

Иногда некоторые нестабильные версии программ могут зависнуть, и при этом выполнить какое-либо действие в графической среде не представляется возможным.

Если в Windows что-либо зависает, то помогает только перезагрузка компьютера. В Ubuntu проблему можно решить, переключившись на консоль (Ctrl+ALT+F1), в которой после задания своего логина и пароля надо ввести killall и название программы. Например,

```
killall audacity
```

и нажать ENTER. Но как узнать какая именно программа зависла?

➢ top

Эта команда отображает список всех запущенных на компьютере процессов (рис. 3.16).

```
top - 06:55:39 up  6:54,  6 users,  load average: 0.42, 0.72, 0.57
Tasks: 133 total,   1 running, 131 sleeping,   1 stopped,   0 zombie
Cpu(s):  2.2%us,  6.9%sy,  0.0%ni, 90.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    509240k total,   418576k used,    90664k free,    60832k buffers
Swap:   747512k total,        0k used,   747512k free,   188712k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1895 serp      20   0  2544 1188  904 R  3.4  0.2   0:01.48 top
 1892 serp      20   0  2544 1184  904 S  3.1  0.2   0:04.89 top
 1209 serp      20   0 31784  18m 4164 S  0.9  3.7   3:16.85 ubuntuone-syncd
   19 root      20   0     0    0    0 S  0.6  0.0   1:38.03 ata/0
   36 root      20   0     0    0    0 S  0.6  0.0   1:06.76 scsi_eh_0
 1786 serp      20   0 46772  14m  10m S  0.6  2.9   0:29.62 gnome-terminal
 1089 serp      20   0 92544  28m  20m S  0.3  5.8   0:17.32 nautilus
```

Рис. 3.16. Результат выполнения программы top в консоли Ubuntu.

Обычно, программа, которая потребляет много ресурсов или та из них, которая зависла, отображается в самом верху. Именно эту программу и надо завершить.

Для этого надо выйти из списка процессов, нажав клавишу q, а затем для завершения требуемой программы воспользоваться командой killall.

➢ clear

Очищает окно консоли или терминала. Кроме этого, существует горячая клавиша Ctrl+L, которая выполняет то же действие независимо от того, находитесь ли вы в гноме, в терминале или в настоящей консоли.

➢ poweroff, halt, reboot, shutdown

Команда reboot используется для перезагрузки системы, halt — завершает работу системы, но не выключает ее питание, poweroff — завершает работу системы и выключает ее питание.

Команда shutdown позволяет завершить работу системы, перезагрузить систему, указать время завершения работы. Предположим, что вы хотите уйти, но сервер надо выключить в 19:30. Вам поможет:

```
shutdown -h 19:30 [сообщение]
```

Если нужно завершить работу системы прямо сейчас, то

```
shutdown -h now.
```

Для перезагрузки системы:

```
shutdown -r now.
```

Перечисленные выше консольные программы не входят в состав пакета core-utils, но они вам могут пригодиться. Напомним: чтобы вернуться в графический интерфейс, достаточно нажать Alt+F7.

И напоследок еще одно маленькое замечание. Если вы обратили внимание, то как в терминале (рис. 3.11 и рис. 3.12), так и в консоли (рис. 3.14) различные объекты выводились разным цветом. По умолчанию используются следующие цвета для разных объектов:

- папки — синие,
- ссылки — голубые,
- исполняемые файлы — зелёные,
- архивы — красные,
- видеофайлы, картинки и музыка — сиреневые.

Однако это зависит от версии дистрибутива и того, кто его настраивал. Если вы хоть немного познакомитесь с Ubuntu и его командным процессором bash, то без особого труда сможете не только настроить все эти цвета под себя, но и изменить системную подсказку в командной строке.

Изменить можно и наименование самих команд, создавая их псевдонимы. Для этого в файл .bash_profile добавьте строки вида:

```
alias псевмоним='команда'
```

Для того чтобы изменения вступили в силу, надо выйти из консоли, используя команду logout, и заново зарегистрироваться.

Следует отметить, что файловые операции можно выполнить при помощи файлового менеджера Nautilus. Установить программу можно при помощи менеджера пакетов synaptic, а информацию о запущенных процессах можно просмотреть при помощи системного монитора и так далее. Все это доступно и в графическом режиме Ubuntu.

3.4. Оболочка Bash и командные файлы Ubuntu Linux

Даже современные дистрибутивы, с изобилием графических утилит сложно представить без командного интерпретатора. В большинстве дистрибутивов это Bash, акроним от Bourne-Again Shell. Это командная Unix оболочка, написанная специально для проекта GNU. Также оболочку Bash зачастую называют консолью или терминалом. Обобщая все эти термины, мы далее будем их использовать как синонимы.

Консоль — это гибкое и мощное средство для работы, которое новички зачастую игнорируют. Бывает, что решить возникшую задачу проще именно с ее помощью. Я постараюсь показать, что это не так страшно, что у этого подхода есть преимущества. Также постараюсь дать элементарное представление о базовых консольных командах.

Для более подробного знакомства существует множество учебной и справочной литературы, а также специализированных сайтов в Интернете. Туда я и отправляю всех наиболее заинтересованных моих читателей, но и для них, я думаю, будет небесполезен тренинг по приводимому ниже материалу.

Итак, приступаем. Мы уже упомянули, что наша текстовая консоль — это окно командного интерпретатора GNU bash, включенного в поставку Ubuntu Linux. Чтобы посмотреть консольные команды, входящие в состав оболочки, достаточно в командной строке набрать:

```
[user]@[host]:~$ help
```

и на экране отобразятся эти команды. Если интересуют все доступные вам консольные команды, то надо перевести курсор в командную строку и нажать два раза клавишу TAB. Но будьте готовы к тому, что список будет состоять из двух-трех тысяч наименований. Вывод будет осуществляться постранично, с паузой для продолжения.

```
serp@vmUbuntu10:~$
Display all 2208 possibilities? (y or n)
...
abrowser
accept
...
--More—
```

При просмотре следует нажимать клавишу Пробел для вывода следующей страницы команд. Если считаете, что вам этого достаточно и Linux это не то, что вам нужно в этой жизни, то быстренько нажимайте клавишу Q, и этот кошмар закончится.

Но все не так страшно, как может показаться на первый взгляд. Зная даже небольшую долю из того, что вы только что наблюдали, вы получаете мощный инструмент администрирования как локального компьютера, так и удаленных компьютеров ЛВС.

Причем использовать командную строку вы можете как для выполнения отдельных команд, вводя их вручную, так и для выполнения некоторой последовательности команд, предварительно записав их в какой-то файл.

Особенность Linux как раз и состоит в том, что все настройки операционной системы, устройств и программных приложений находятся в открытых текстовых конфигурационных файлах, которые доступны любому пользователю, наделенному правами администратора. И этим мы будем неоднократно пользоваться в дальнейшем, а сейчас познакомимся с командой:

```
cat [имя_файла]
```

Эта команда обеспечивает вывод содержимого файла на стандартный вывод. По умолчанию — это экран дисплея. То есть используя эту команду, у нас появляется возможность просмотреть любой файл в текстовой консоли.

Но есть и другая очень привлекательная для нас особенность использования этой команды, когда мы стандартный вывод перенаправляем вместо дисплея в файл. Тогда у нас появляется возможность записать вводимый с клавиатуры текст в файл с заданным именем и месторасположением.

```
cat > [имя_файла]
```

Например, последовательность действий:

```
serp@vmUbuntu10:~$ cat > sample
строка 1
строка 2
строка 3
CTRL/d
```

позволит нам создать в текущем директории файл с именем sample, который будет содержать три строки. Вы можете убедиться в этом, используя для просмотра вновь созданного файла ту же команду cat.

Рассмотренный подход очень удобен при создании крохотных файлов. В любых других случаях удобнее пользоваться текстовым редактором,

простейшим из которых, включенных в поставку Ubuntu, является Nano. Для его вызова следует ввести команду:

```
nano [имя_файла]
```

В Linux, так же как и в операционных системах семейства Microsoft Windows, можно создавать командные файлы, которые содержат в себе набор команд интерпретатора shell. Такие файлы имеют свой синтаксис и позволяют даже оперировать такими структурами, как циклы и условия.

А теперь познакомимся с несколькими консольными командами и основами shell программирования Linux на примере командной строки Ubuntu 10.04 и интерпретатора Bash.

Создание файла, его просмотр и запуск на выполнение

Создаем новый файл, для чего вызываем текстовый редактор:

```
serp@vmUbuntu10:~$ nano sample
```

записываем три строки, содержащие различные Linux-команды:

```
pwd
ls
echo The END
```

сохраняем файл и выходим из редактора. Просмотрим содержимое вновь созданного нами файла sample. Для этого в командной строке введем

```
serp@vmUbuntu10:~$ cat sample
```

и на экране, если все было сделано правильно, должно появиться

```
pwd
ls
echo The End
```

А теперь выполним эти команды, запустив файл sample на выполнение:

```
serp@vmUbuntu10:~$ sh sample
```

```
/home/serp
examples.desktop
sample
Рабочий стол
The End
```

Таков общий принцип ...

Командный файл легко сделать исполняемым

В командной строке введем:

```
serp@vmUbuntu10:~$ chmod +x sample
```

И если теперь в командной строке мы просто укажем имя нашего файла

```
serp@vmUbuntu10:~$ /home/serp/sample
```

то на экране отобразится:

```
/home/serp
examples.desktop
sample
Рабочий стол
The End
```

Переменные, элементарные вычисления и комментарии

Вызываем редактор для редактирования файла sample:

```
serp@vmUbuntu10:~$ nano sample
```

Вводим новые три строки:

```
read x #вводим x
read y #вводим y
echo `expr $x '*' $y + 5`
```

Просмотрим содержимое отредактированного файла sample:

```
serp@vmUbuntu10:~$ cat sample
read x  #вводим x
read y  #вводим y
echo `expr $x '*' $y + 5`
```

Все, что находится за знаком #, это комментарий и интерпретатором не воспринимается. Запустим файл sample на выполнение:

```
serp@vmUbuntu10:~$ /home/serp/sample
2
3
11
```

Параметры $1, $2 ...$9

Эти параметры позволяют задавать аргументы командной строки для выполняемого файла. Вызовите редактор и отредактируйте файл sample так, чтобы при его просмотре на экране было:

```
serp@vmUbuntu10:~$ cat sample
echo Первый параметр: $1
echo Второй параметр: $2
echo Третий параметр: $3
echo Команда 'ls' $1 $2 $3
ls $1 $2 $3
```

Запустим файл sample на выполнение, задав еще три параметра:

```
serp@vmUbuntu10:~$ /home/serp/sample -l -s -r
```

и на экране, если все было сделано правильно, должно появиться:

```
Первый параметр: -l
Второй параметр: -s
Третий параметр: -r
Команда ls -l -s -r
4 drwxr-xr-x 2 serp serp 4096 2011-07-09 10:48 Рабочий
стол
4 -rwxr-xr-x 1 serp serp  165 2011-07-19 20:50 sample
4 -rw-r--r-- 1 serp serp  179 2011-06-20 16:11 examples
```

Параметры $# и $*

$# задает общее количество параметров, с которым вызывается командный файл:

```
serp@vmUbuntu10:~$ cat sample
echo Всего параметров = $#
```

```
serp@vmUbuntu10:~$ /home/serp/sample Par1 Par2 Par3
Всего параметров = 3
```

$* содержит сразу все параметры:

```
serp@vmUbuntu10:~$ cat sample
echo Все параметры команды:  $*
```

```
serp@vmUbuntu10:~$ /home/serp/sample All parameters are in
this variable
Все параметры команды: All parameters are in this variable
```

Переменные

Имя переменной может начинаться с буквы или символа подчеркивания. Знак равенства — это оператор присваивания. Например, _var=1, _word =slovo, value="t ak aia dli nna istr oka". Тип данных переменных shell это всегда строка символов!

```
serp@vmUbuntu10:~$ cat sample
echo Пример с переменной и параметром
_qq="HELLO, "
echo $_qq $1
```

```
serp@vmUbuntu10:~$ /home/serp/sample SerP
Пример с переменной и параметром
HELLO, SerP
```

Результат выполнения команды в переменную

Обратите внимание, что команда обрамляется обратными апострофами.

```
serp@vmUbuntu10:~$ cat sample
echo Результат команды в переменную
_date=`date`
echo $_date
```

```
serp@vmUbuntu10:~$ /home/serp/sample
Результат команды в переменную
Втр Июл 19 22:09:04 MSD 2011
```

Системные переменные

PATH — пути поиска исполняемых файлов.
HOME — домашний каталог.
MAIL — файл электронной почты.
SHELL — оболочка, в которой работаем.

```
serp@vmUbuntu10:~$ cat sample
echo $PATH
echo $HOME
echo $MAIL
echo $SHELL
```

```
serp@vmUbuntu10:~$ /home/serp/sample
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/b
in:/usr/games
/home/serp
/var/mail/serp
/bin/bash
```

Так как в пути поиска исполняемых файлов по умолчанию нет каталога /home/serp/, то для запуска sample на выполнение приходится постоянно писать полный путь.

Чтобы избавиться от этого, добавим в системную переменную PATH каталог HOME, и тогда не надо будет писать полный путь для запуска исполняемого файла sample.

```
serp@vmUbuntu10:~$ PATH=$PATH:$HOME
```

Просмотрим текущее состояние системной переменной PATH.

```
serp@vmUbuntu10:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/b
in:/usr/games:/home/serp
```

Отсюда видно, что наш рабочий каталог включен в список каталогов с исполняемыми файлами. А если это так, то и доступно

```
serp@vmUbuntu10:~$ sample
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/b
in:/usr/games:/home/serp
/home/serp
/var/mail/serp
/bin/bash
```

Теперь нет необходимости писать полный путь до команды на запуск файла sample.

Специальные символы

* — любая последовательность, любых символов.
? — один любой символ.
[...] — любой из символов диапазона.
"" — двойные кавычки.

```
serp@vmUbuntu10:~$ echo *
examples.desktop sample Рабочий стол
```

```
serp@vmUbuntu10:~$ echo "*"
*
```

Апострофы ", обратный апостроф ``, обратный слеш \.

```
serp@vmUbuntu10:~$ cat sample
text="Это текст"
echo '$text'
echo \$text
echo $text
text=`pwd`
echo $text
```

```
serp@vmUbuntu10:~$ sample
$text
$text
Это текст
/home/serp
```

Обратный слеш может быть полезен при вводе ESC-последовательностей.

```
serp@vmUbuntu10:~$ echo -e "Началот" "Табуляция" "\n"
"Новая строка" > sample
```

Обратите внимание, что команда echo используется с опцией -e. Для справки man echo.

```
serp@vmUbuntu10:~$ cat sample
Начало   Табуляция
Новая строка
```

Арифметические операции

Для выполнения арифметических операций в командном файле понадобится команда expr.

```
serp@vmUbuntu10:~$ expr 3 \* 2 + 8 / 2
10
```

Если при умножении не поставить \ или ", то * будет восприниматься как любой символ.

```
serp@vmUbuntu10:~$ expr 3 * 2
expr: синтаксическая ошибка
```

Стоит отметить, что \ — это не деление, а целая часть от деления, операция % дает остаток от деления.

Переменные окружения, команда export и unset

Для взаимодействия с другими процессами могут пригодиться переменные окружения. Их полный список можно посмотреть командой export:

```
serp@vmUbuntu10:~$ export
declare -x HOME="/home/serp"
declare -x LANG="ru_RU.UTF-8"
declare -x LESSCLOSE="/usr/bin/lesspipe %s %s"
declare -x LESSOPEN="| /usr/bin/lesspipe %s"
declare -x LOGNAME="serp"
...
```

Задать и очистить свою переменную

```
serp@vmUbuntu10:~$ myVar="Test"
serp@vmUbuntu10:~$ echo $myVar
Test
serp@vmUbuntu10:~$ unset myVar
serp@vmUbuntu10:~$ echo $myVar

serp@vmUbuntu10:~$
```

Оператор test

Этот оператор предназначен для проверки типов файлов и сравнения значений. Условные выражения, используемые в операторе test, строятся на основе проверки файловых атрибутов, сравнения строк, а также обычных арифметических сравнений. Рассмотрим ряд примеров использования этого оператора.

Для случая проверки типов файлов:

```
test -r file – Истинно, если файл file существует и
               доступен для чтения
test -w file – Истинно, если файл file существует и
               доступен для записи
test -x file – Истинно, если файл file существует и
               доступен для выполнения
```

Для случая сравнения значений:

```
test $x -eq $y   -  Истинно, если $x равен $y
test $x -en $y   -  Истинно, если $x не равен $y
test $x -ge $y   -  Истинно, если $x больше или равен $y
test $x -gt $y   -  Истинно, если $x больше $y
test $x -le $y   -  Истинно, если $x меньше или равен $y
test $x -lt $y   -  Истинно, если $x меньше $y
```

Остальные параметры можно посмотреть в man test.

Условный оператор IF

Конструкция условного оператора в упрощенном виде выглядит так:

```
if list1 then list2 else list3 fi
```

где list1, list2 и list3 — это последовательности команд, разделенные запятыми и оканчивающиеся точкой с запятой или символом новой строки. Кроме того, эти последовательности могут быть заключены в фигурные скобки.

Оператор if проверяет значение, возвращаемое командами из list1. При составлении условных выражений оператора if очень полезным является использование оператора test.

```
serp@vmUbuntu10:~$ cat > sample
if test -r sample
then
  echo "Доступен на чтение"
else
  echo "Не доступен"
fi
```

```
serp@vmUbuntu10:~$ sh sample
Доступен на чтение
```

В этом примере test -r sample — это условие, команда echo «Доступен на чтение» выполняется, если условие истинно, а команда echo «Не доступен» если условие ложно.

Цикл FOR

```
serp@vmUbuntu10:~$ cat sample
for x in 1 two 3
do
  echo $x
done
```

```
serp@vmUbuntu10:~$ sample
1
two
3
```

В этом цикле код между do и done выполнится 3 раза, при этом первый раз x=1, второй раз x=two и последний x=3.

Другой интересный пример:

```
serp@vmUbuntu10:~$ cat sample
for x in *
do
  echo $x
done
```

```
serp@vmUbuntu10:~$ sample
examples.desktop
sample
Рабочий стол
```

В списке переменных цикла символ * заставляет for использовать в качестве значения $x элементы текущего каталога.

Циклы WHILE и UNTIL

```
serp@vmUbuntu10:~$ cat sample
while test -r file
do
  sleep 10
  echo "Файл существует"
done
echo "Файла нет"
```

```
serp@vmUbuntu10:~$ touch file
serp@vmUbuntu10:~$ sample
Файл существует
Файл существует
```

```
^Z
[1]+  Остановлено  sample
```

```
serp@vmUbuntu10:~$ rm file
serp@vmUbuntu10:~$ fg
sample
Файл существует
Файла нет
```

Команда touch — создает новый файл с именем file, а команда fg — выносит на передний план последние задачи.

Аналогичный пример, использующий until, имеет вид:

```
serp@vmUbuntu10:~$ cat sample
until test -r file
do
  sleep 5
  echo "Файл не существует"
done
echo "Файл есть"
```

```
serp@vmUbuntu10:~$ sample
Файл не существует
Файл не существует
^Z
[1]+  Остановлено  sample
```

```
serp@vmUbuntu10:~$ touch file
serp@vmUbuntu10:~$ fg
sample
Файл не существует
Файл есть
```

4. ПОЛЬЗОВАТЕЛИ И ГРУППЫ В UBUNTU

В Windows мы привыкли, что нам разрешено все. Конечно, не всегда, но в большинстве случаев именно так. В Linux все несколько иначе. В процессе установки Ubuntu были заданы имя и пароль учетной записи пользователя, который должен быть администратором системы. Но при этом отмечалось, что использование учетной записи администратора не несет практически никакой угрозы безопасности системы. Попробуем разобраться в этом.

4.1. Суперпользователь

Во всех системах на базе Linux всегда есть один привилегированный пользователь, который зовется root или суперпользователь. Его полномочия ничем не ограничены. Он может делать в системе абсолютно все что угодно, и большинство системных процессов работают от имени root. Система полностью подвластна этому пользователю, и любая команда будет безоговорочно выполнена.

Поэтому работать под именем пользователя root надо с осторожностью. Всегда следует думать над тем, что собираетесь сделать. Если вы дадите команду на удаление корневой файловой системы, то система ее выполнит. Но если вы попытаетесь выполнить такое же действие, но с правами администратора, то система сообщит вам, что у вас нет полномочий.

Использование такого всемогущего пользователя крайне опасно, ибо любая ошибка может привести к катастрофическим последствиям, вплоть до полного уничтожения системы. Вот поэтому в дистрибутиве Ubuntu учетная запись root отключена. Вы не можете войти в систему, используя эту учетную запись.

Сделано это из соображений безопасности, то есть разработчики пытаются защитить систему от вас же самих, от ваших некорректных действий. Обычный пользователь в Linux, вообще говоря, никак не может повлиять на работоспособность системы. Он даже не может устанавливать и удалять программы, управлять системными настройками и изменять файлы вне своего домашнего каталога.

Грубо говоря, в Linux есть два типа пользователей: root и все остальные. Суперпользователь root может все, а все остальные только то, что им разрешено в настройках.

4.2. Администратор

Администратор в Ubuntu является обычным пользователем, но с правом при необходимости вмешиваться в работу системы, Он по умолчанию может по запросу делать все то же самое, что и суперпользователь. Однако случайно что-то испортить из-под администратора нельзя, так как перед выполнением каждого опасного действия система спрашивает у него его пароль.

Главное отличие администратора от суперпользователя как раз и заключается в необходимости вводить пароль для выполнения любого потенциально опасного действия.

Как это выглядит на практике? Зайдите в меню Система -> Администрирование и выберите пункт «Менеджер пакетов Synaptic». Это инструмент, с помощью которого можно устанавливать и удалять любые программы.

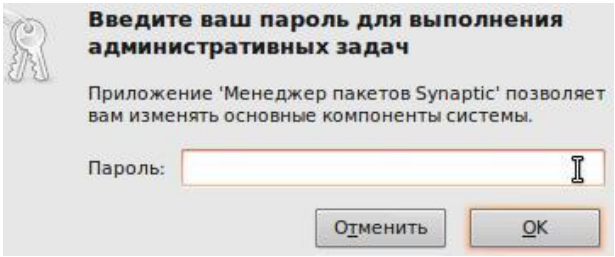

Рис. 4.1. Запрос пароля на выполнение действия с правами root.

Поэтому для запуска Synaptic нужны права администратора, и при попытке открытия этой программы система попросит вас ввести свой пароль (рис. 4.1). Если вы действительно являетесь администратором и правильно ввели пароль, то откроется сам Synaptic (рис. 4.2).

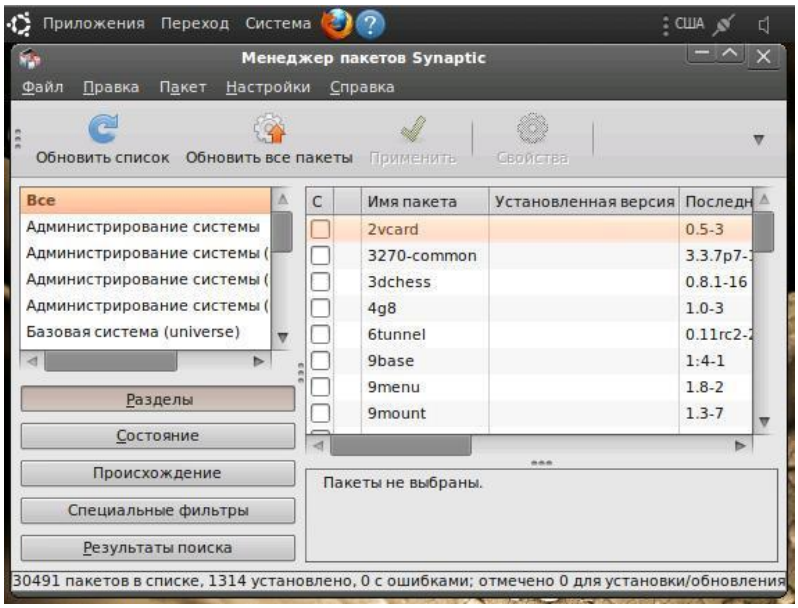

Рис. 4.2. Окно программы «Менеджер пакетов Synaptic».

Как работать с этой программой, узнаем позднее, а пока закройте ее, так как сейчас нас интересуют права доступа. Если в окне запроса пароля пароль будет введен неверно, то система просто закроет его, не выдавая никаких сообщений. Соответственно, и операция, для которой требовались права администратора, выполнена не будет.

Привилегии администратора нужны не только для запуска системных приложений. Откройте программу управления настройками времени, располагающуюся в меню Система -> Администрирование ->Дата и время. После того как откроется ее окно (рис. 4.3), вы увидите, что в нем ничего нельзя изменить, так как все поля заблокированы.

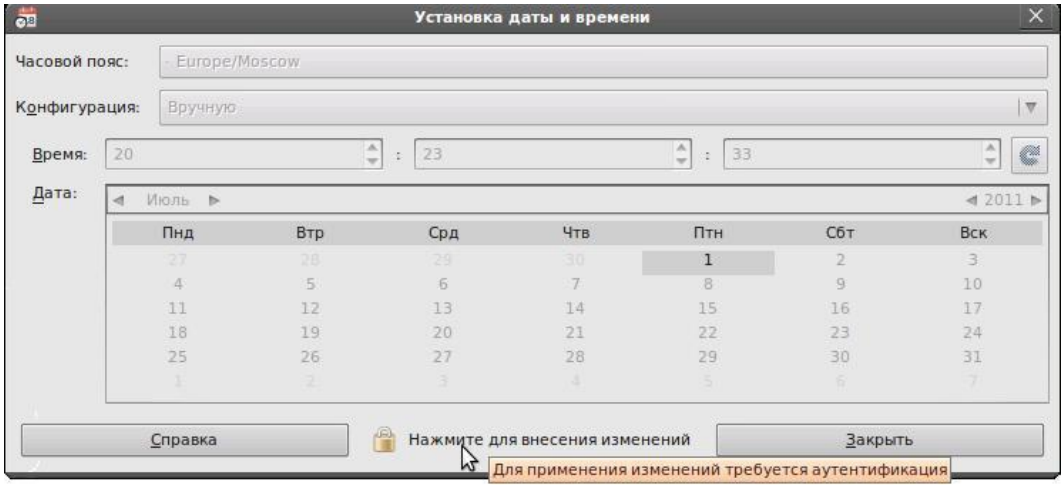

Рис. 4.3. Окно программы «Установка даты и времени».

Однако внизу есть кнопка с ключиком, рядом с которой написано «Нажмите для внесения изменений». Если ее нажать, то система снова спросит ваш пароль (рис 4.4). И, если у вас есть полномочия (а у администратора они, естественно, есть), то система откроет вам доступ к настройкам.

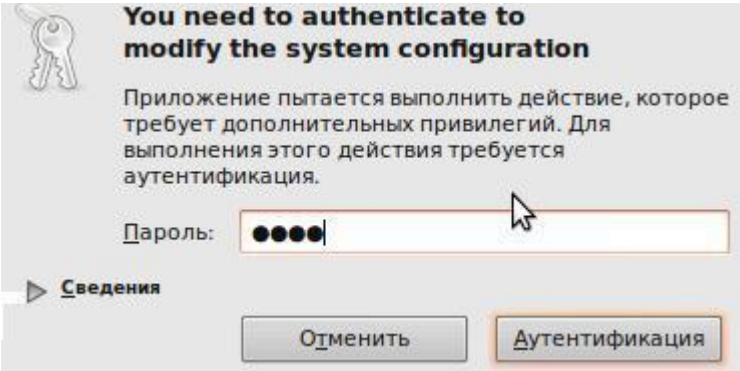

Рис. 4.4. Запрос пароля на изменение даты и времени.

Введенный при запуске любого графического приложения пароль хранится 15 минут. Поэтому спустя 15 минут, программа может повторно запросить пароль, если вы за это время не завершили работу программы.

4.3. Выполнение команд с правами root

То, о чем говорилось выше, относилось к работе в графической среде GNOME. Однако при системном конфигурировании и сетевом администрировании не всегда удобно ходить по опциям меню и подтверждать пароли во всплывающих окнах. Тем более, если предстоит выполнение большой последовательности команд, да еще при удаленном доступе. Как же нам поступать, работая в терминальном режиме?

Такие команды, как установка программного обеспечения, изменение конфигурационных файлов, сетевые настройки, требуют полномочий root. Чтобы администратору получить права root при выполнении каких-либо действий, нужно использовать команду sudo:

```
sudo «команда_которую_нужно_выполнить_с_правами_root»
```

Например, вам нужно изменить файл /etc/samba/smb.conf. Для этого следует использовать команду:

```
sudo gedit /etc/samba/smb.conf
```

Программа gedit — это текстовый редактор. Ему мы передали один параметр — имя файла, который нужно открыть. Если ввести эту же команду, но без sudo, то есть gedit /etc/samba/smb.conf, то текстовый редактор тоже запустится и откроет файл, но сохранить изменения вы не сможете, поскольку у вас не хватит полномочий. Программа sudo перед выполнением указанной вами команды запросит у вас пароль:

```
sudo gedit /etc/samba/smb.conf
Password:
```

Вы должны ввести пароль, который использовали для входа в систему.

Замечание.
Запомните, что введенный пароль хранится 15 минут.

Если из терминала нужно запустить графическую программу с правами root (например, gedit), желательно использовать не команду sudo, как было показано выше, а команду

```
gksudo «графическая_программа_с_правами_root»
```

Графические приложения с правами root проще запускать, используя главное меню. Но не все приложения есть в главном меню или не все приложения вызываются с правами root.

Например, в главном меню есть команда вызова текстового редактора, но нет команды для вызова текстового редактора с правами root. Поэтому намного проще нажать комбинацию клавиш Alt+F2. Появится окно «Выполнить программу» (рис. 4.5), в котором следует ввести нужную команду.

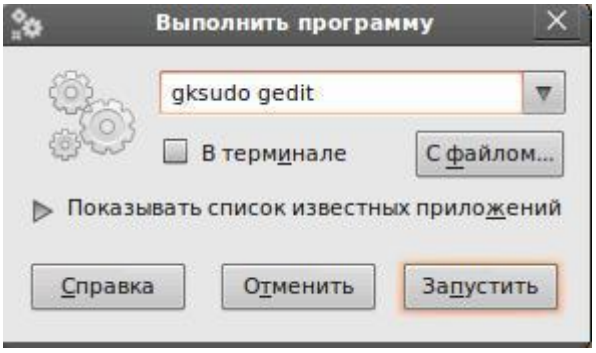

Рис. 4.5. Окно терминала при изменении прав доступа.

Иногда возникает необходимость выполнить подряд несколько команд с правами root. В этом случае можно использовать команду

```
sudo -s
```

После этого вы перейдете в режим суперпользователя (рис. 4.6). Об этом говорит login перед именем хоста, а также символ # в конце приглашения командной строки. Данная команда по действию похожа на

```
sudo -i
```

Но если sudo -i меняет домашний каталог на каталог /root, то sudo -s оставляет в качестве домашнего каталог пользователя. Система какое-то время помнит введенный пароль, то есть сохраняет открытой sudo-сессию. Поэтому при последующих выполнениях sudo ввод пароля может не потребоваться.

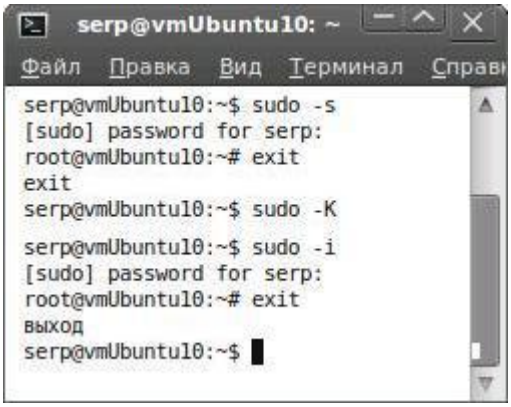

Рис. 4.6. Окно «Выполнить программу».

Для гарантированного прекращения сессии sudo надо в терминале набрать:

```
sudo -K
```

Для выхода обратно в режим обычного пользователя необходимо набрать exit или просто нажать Ctrl+D.

4.3. Создание учетных записей пользователей

Если вы не единственный пользователь ресурсов своего компьютера, тогда вам нужно создать дополнительные учетные записи. Простейший вариант создания новой учетной — это выбор в меню Система -> Администрирование -> Пользователи и группы (рис. 4.7).

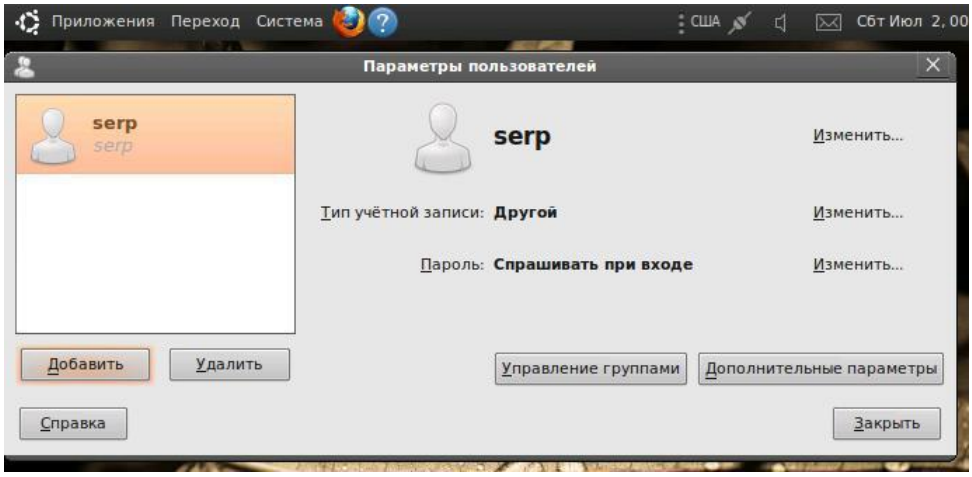

Рис. 4.7. Окно «Параметры пользователей».

Далее надо нажать кнопку «Добавить». После подтверждения пароля следует ввести имя пользователя и его пароль. Пароль можно ввести вручную, а можно сгенерировать произвольно (Пароль -> «Изменить ...»). Произвольно сгенерированный пароль будет сложнее для подбора, но и сложнее для запоминания. Поэтому решайте сами, что важнее — безопасность или комфорт.

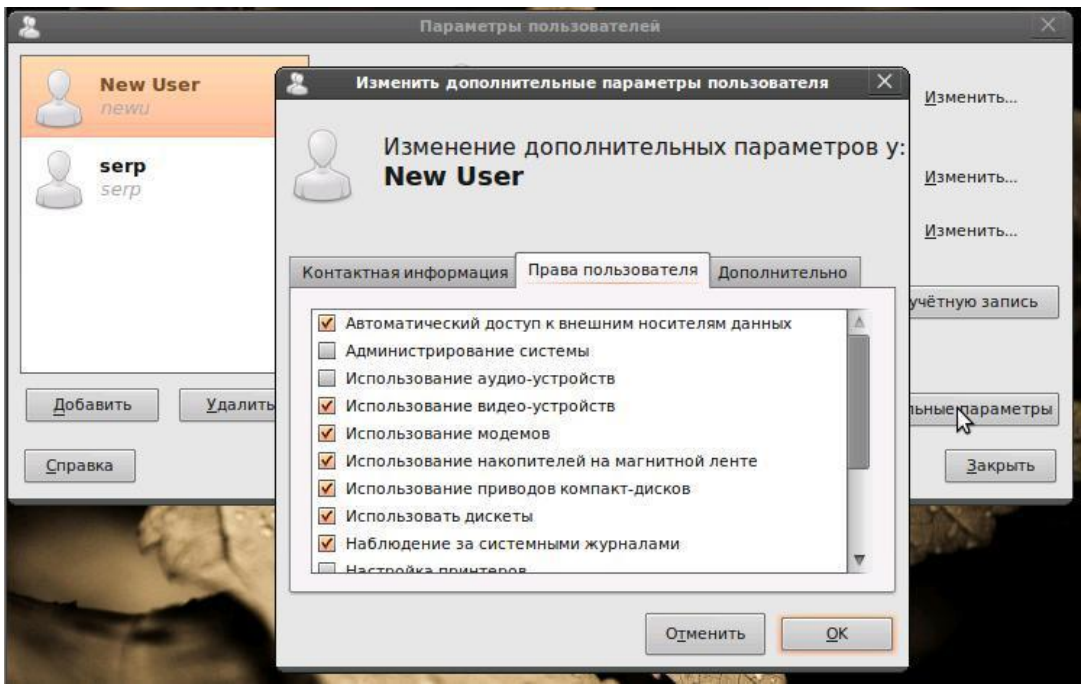

Рис. 4.8. Установка и изменение прав пользователя.

Затем надо определить тип учетной записи (кнопка «Изменить ...»): пользователь или администратор. Особенность Администратора в том, что он может использовать sudo.

Выбрав «Дополнительные параметры» -> «Дополнительно», следует задать, или принять по умолчанию для нового пользователя: его домашний каталог, используемый командный интерпретатор («Оболочка»), группу и ID.

На вкладке «Права пользователя» (рис. 4.8) можно определить привилегии пользователя. Как видно из этой вкладки, по умолчанию, обычному пользователю разрешено все, кроме выполнения задач по администрированию (использования sudo), подключения к беспроводным и проводным сетям, опубликования папок в ЛВС.

Замечание.
У Вас, как администратора системы, имеются права на изменение привилегий пользователей, установленных по умолчанию.

Давайте разберемся, что же происходит при создании новой учетной записи пользователя.

➢ Во-первых, формируется каталог /home/«имя_нового_пользователя», в который копируется содержимое каталога /etc/skel.

Каталог /etc/skel содержит «джентльменский набор» — файлы конфигурации по умолчанию, которые должны быть в любом пользовательском каталоге. Название каталога skel (от англ. skeleton) полностью оправдывает себя — он действительно содержит "скелет" домашнего каталога пользователя.

➢ Во-вторых, создается запись в файле /etc/passwd.

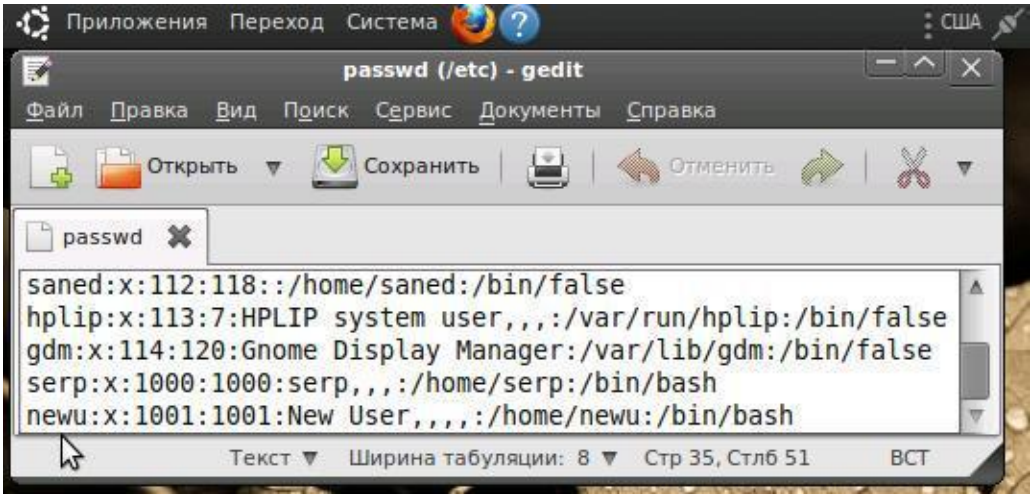

Рис. 4.9. Содержимое файла /etc/passwd после добавления пользователя.

Каждая запись этого файла содержит информацию о конкретном зарегистрированном пользователе в операционной систем, его логине,

пароле, принадлежности к определенной группе и ряд других параметров. Формат записи имеет вид:

```
«логин»:«пароль»:«UID»:«GID»:«имя»:«каталог»:«оболочка»
```

Рассмотрим последние две строки фрагмента файла /etc/passwd, приведенного на рис. 4.9. Эти строки соответствуют двум пользователям с логинами serp и new:

- Первое поле — это логин пользователя, который он вводит для регистрации в системе.
- Пароль сейчас в этом файле не указывается и второе поле с символом «x» осталось просто для совместимости со старыми системами. Пароли хранятся в файле /etc/shadow.
- Третье и четвертое поле — это UID (User ID) и GID (Group ID) — идентификаторы пользователя и группы соответственно. Идентификатор root всегда равен 0.
- Пятое поле — это настоящее имя пользователя. Может быть не заполнено и зависит от педантичности администратора системы. Если компьютер — сервер сети, то просто необходимо указать фамилию, имя каждого пользователя, а то когда придет время обратиться к пользователю по имени, вы его знать не будете. Попробуйте запомнить 200 или более фамилий и имен!
- Шестое поле содержит имя домашнего каталога. Обычно это каталог /home/«имя_пользователя».
- Последнее поле — это имя командного интерпретатора, который будет обрабатывать ваши команды, когда вы зарегистрируетесь в консоли.

Замечание.
Ubuntu, как и Linux, — системы многопользовательские, то есть на одном компьютере могут быть различные пользователи, со своими настройками, данными и правами доступа к различным системным функциям

4.4. Группы в Ubuntu

Кроме пользователей, в Linux для разграничения прав существуют группы. Каждая группа, так же как и отдельный пользователь, обладает неким набором прав доступа к различным компонентам системы. И каждый пользователь — член какой-либо группы — автоматически получает все права этой группы. То есть группы нужны для объединения пользователей по принципу одинаковых полномочий на какие-либо действия.

Каждый пользователь может состоять в неограниченном количестве групп. И в каждой группе может быть сколько угодно пользователей.

Каждый пользователь состоит всегда как минимум в одной группе. Это так называемая основная группа. По умолчанию она носит такое же имя, как и у самого пользователя.

Например, в Ubuntu есть одна очень полезная группа — admin. Любой член этой группы получает неограниченные административные привилегии. Создаваемый при установке Ubuntu пользователь автоматически становится членом группы admin. Именно поэтому, и только по этой причине, он и является администратором.

Группы позволяют более эффективно управлять правами пользователей. Например, есть три пользователя — serp, vambr и zam, которые должны совместно работать над проектом. Достаточно объединить их в одну группу, и пользователи будут иметь доступ к домашним каталогам друг друга. По умолчанию пользователь не имеет доступа к домашнему каталогу другого пользователя.

Создать группу, а также поместить пользователя в группу позволяют рассмотренные выше графические конфигураторы, вызываемые из меню Система -> Администрирование -> Пользователи и группы (рис. 4.10).

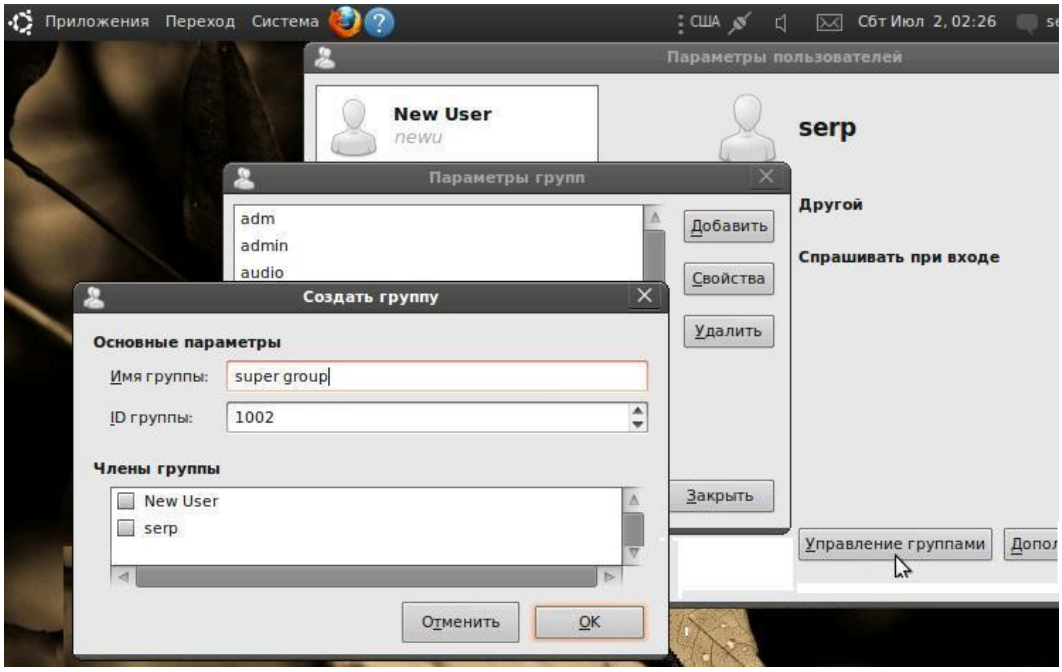

Рис. 4.10. Параметры, имена и члены группы.

Вы можете использовать графические конфигураторы — они очень удобные, но если вы хотите стать настоящим линуксоидом, то должны знать, что доступные в системе группы указываются в файле /etc/group. Добавить новую группу в систему можно с помощью команды

```
groupadd [-параметры] «имя_новой_группы»,
```

но, как правило, проще добавить в текстовом редакторе еще одну запись в файл /etc/group.

5. ФАЙЛОВАЯ СИСТЕМА UBUNTU

Linux поддерживает много файловых систем. В качестве корневой файловой системы можно выбрать: ext2, ext3, XFS, ReiserFS, JFS. Из них только ext2 не является журналируемой, все остальные ведут журналы своей работы.

Преимущество журналируемой файловой системы заключается в том, что, ведя журнал своей работы, она позволяет восстановить данные в случае сбоя. Перед тем как выполнить операцию, журналируемая файловая система записывает эту операцию в журнал, а после выполнения операции удаляет из журнала. Представим, что после занесения операции в журнал произошел сбой. Например пропало электропитание. Позже, когда сбой будет устранен, файловая система по журналу выполнит все действия, которые занесены в журнал. Конечно, и это не всегда позволяет уберечься от сбоя, но это все же лучше, чем ничего.

Файловые системы ext2 и ext3 совместимы. По сути, ext3 — это та же ext2, только с журналом. Раздел ext3 могут читать программы, рассчитанные на ext2. Например, Total Commander в Windows. В современных Linux-дистрибутивах по умолчанию используется файловая система ext3. Остальные файловые системы довольно специфичны.

5.1. Имена файлов в Linux

В Linux несколько другие правила построения имен файлов, чем в Windows. Прежде всего отметим три особенности этой файловой системы:

- во-первых, в Linux разделитель каталогов в путях — это символ «/» (прямой слеш), а Windows использует «\» (обратный слеш).
- во-вторых, в названиях файлов можно использовать все символы всех алфавитов мира, кроме символа «/»,
- в-третьих, в Linux нет такого понятия, как расширение файла.

В Linux можно назвать файл, например, «Jean Reno: Road to Leon», но надо иметь в виду, что имена файлов в Linux регистрозависимы. То есть, FILE.txt и file.Txt — это два совершенно разных файла, и они могут спокойно находиться в одной директории.

В Linux нет такого понятия, как расширение файла. Если для файла document1.doc в Windows: document1 является именем файла, а doc — его расширением, то в Linux document1.doc — это просто длинное имя файла.

Замечание.

Имя файла — до 254 любых символов, включая греческие буквы, китайские иероглифы, руны и т. д., в том числе и кириллицу.

Не использовать прямой слеш (/), не рекомендуются: '\', '?', '<', '>', '*', '|', а также символы переноса строки и табуляции.

Кириллицу в именах файлов используйте осмотрительно. Если вы не будете свои файлы передавать Windows-пользователям (на флешке или по Email) — используйте на здоровье. А если же нужно отправить файл по Email, лучше использовать латиницу в имени файла. Кодировка-то у всех разная, и поэтому вместо русскоязычного имени получатель может увидеть абракадабру.

5.2. Файлы и устройства

Сейчас Windows-пользователи будут вообще удивлены: в Linux есть файлы устройств, позволяющие обращаться с устройством, как с обычным файлом. Файлы устройств находятся в каталоге /dev (от англ. devices). Вот самые распространенные примеры файлов устройств:

/dev/hdx	–	файл устройства жесткого диска, где x — это тип подключения диска к шине;
/dev/hdxN	–	файл устройства раздела жесткого диска, где N — это номер раздела;
/dev/mouse	–	файл устройства мыши;
/dev/modem	–	файл устройства модема;
/dev/ttySn	–	файл последовательного порта, где n — номер порта (ttySO соответствует COM 1, ttyS1 — COM2 и т. д.).

Файлы устройств бывают двух типов: блочные и символьные. Обмен информации с блочными устройствами, например с жестким диском, осуществляется блоками информации, а с символьными — отдельными символами. Пример символьного устройства — последовательный порт.

5.3. Стандартные каталоги Linux

Файловая система любого дистрибутива Linux содержит следующие каталоги:

/	–	корневой каталог;
/bin	–	стандартные программы Linux (cat, cp, ls, login и т. д.);
/boot	–	каталог загрузчика, содержит образы ядра, конфигурационные файлы загрузчика;
/dev	–	файлы устройств;

/etc	–	конфигурационные файлы системы;
/home	–	домашние каталоги пользователей;
/lib	–	библиотеки и модули;
/lost+found	–	восстановленные после некорректного размонтирования файловой системы файлы;
/misc	–	может содержать все, что угодно, как и каталог /opt;
/mnt	–	обычно содержит точки монтирования;
/proc	–	каталог псевдофайловой системы procfs, дающей информацию о процессах;
/root	–	каталог суперпользователя root;
/sbin	–	каталог системных утилит, для пользователя root;
/tmp	–	каталог для временных файлов;
/usr	–	пользовательские программы, документация, коды;
/var	–	постоянно изменяющиеся данные системы: почтовые ящики, протоколы, пулы системы печати, замки и т. д.

5.4. Команды для работы с файлами

Основные команды Linux, предназначенные для работы с файлами представлены в табл. 5.1.

Таблица 5.1

Основные команды Linux для работы с файлами

Команда	Назначение
touch <файл>	Создание пустого файла
cat <файл>	Просмотр текстового файла
tac <файл>	Вывод содержимого текстового файла в обратном порядке
cp <файл1> <файл2>	Копирование <файл1> в <файл2>. Если <файл2> существует, программа попросит разрешение на его перезапись
mv <файл1> <файл2>	Перемещение <файл1> в <файл2>. Эту команду можно использовать также и для переименования файла
rm <файл>	Удаление файла
locate <файл>	Быстрый поиск файла
which <программа>	Вывод каталога, в котором находится программа, если она вообще установлена. Поиск идет в каталогах, указанных в переменной окружения path (это путь поиска программ)
less <файл>	Используется для удобного просмотра файла с возможностью скроллинга (постраничной прокрутки)

Упражнение.
В качестве примера работы в терминале Linux с файловыми командами выполните последовательность команд, приведенных в табл. 5.2, где поясняется назначение каждой из них.

Таблица 5.2

Пример последовательности команд, вводимых в терминале Linux

№	Вводимая команда	Описание назначения команды
01	touch file.txt	Команда touch создает в текущем каталоге новый файл file.txt
02	echo "some text" > file.txt	Команда echo записывает строку 'some text' в этот же файл. Обратите внимание: символ '>' — это перенаправления ввода/вывода
03	cat file.txt	Команда cat выводит содержимое файла. В файле должна быть записанная нами строка 'some text'
04	cp file.txt file-copy.txt	Команда cp копирует файл file.txt в файл с именем file-copy.txt
05	cat file-copy.txt	После этого мы опять используем команду cat для вывода содержимое файла file-copy.txt, чтобы убедиться, что файл действительно скопировался
06	rm file.txt	Команда rm удаляет файл file.txt. При удалении система спрашивает, хотите ли вы удалить файл. Если хотите, то нужно нажать клавишу 'Y', а если нет, то 'N'
07	cat file.txt	Точно ли файл удален? Убедимся в этом, введя команду cat file.txt. Система нам сообщает, что нет такого файла
08	mv file-copy.txt file.txt	Команда mv переименовывает файл file-copy.txt в файл file.txt
09	cat file.txt	Последняя команда выводит исходный файл file.txt

Думаем, особых проблем с этими командами у вас не возникло, тем более что принцип действия этих команд вам должен быть знаком по командам MS-DOS, которые, как квалифицированный пользователь Windows, вы должны знать наизусть.

Как в Windows, так и в Linux-системах вместо имени файла иногда очень удобно указать маску имени файла. Например, если у нас есть много временных файлов, которые заканчиваются строкой "tmp", то для их удаления нужно воспользоваться командой:

```
rm *tmp
```

Если же нужно удалить все файлы в текущем каталоге, можно просто указать звездочку:

```
rm *
```

Аналогично можно использовать символ «?», который в отличие от звездочки, заменяющей последовательность символов произвольной длины, заменяет всего один символ. Например, нам нужно удалить все файлы, имена которых состоят из трех букв и начинаются на d:

```
rm d??
```

Будут удалены файлы dl1, dbm, d78 и т. д., но не будут тронуты файлы, имена которых состоят более чем из трех букв и которые не начинаются на d. Следует отметить, что маски имен можно также использовать и при работе с каталогами.

Напомним, что при работе в терминале Linux справочную информацию по любой команде вы можете получить, набрав

```
man <имя команды>
```

При этом следует отметить, что все команды имеют большое количество различных опций. Используя команду man <имя команды>, вы можете познакомиться с ними и постепенно их все изучить.

5.5. Команды для работы с каталогами

Основные команды Linux, предназначенные для работы с каталогами приведены в табл. 5.3.

Таблица 5.3

Основные команды Linux для работы с каталогами

Команда	Назначение
mkdir <каталог>	Создание каталога
cd <каталог>	Изменение каталога
ls <каталог>	Вывод содержимого каталога
rmdir <каталог>	Удаление пустого каталога
rm -r <каталог>	Рекурсивное удаление каталога

При указании имени каталога можно использовать следующие символы:

- **.** – означает текущий каталог, если вы введете команду cat ./file, то она выведет файл file, находящийся в текущем каталоге;
- **..** – родительский каталог, например, команда cd .. перейдет на один уровень «вверх» по дереву файловой системы;
- **~** – домашний каталог пользователя.

Упражнение.
Теперь рассмотрим команды для работы с файлами на практике. Выполните команды, приведенные в табл. 5.4.

Таблица 5.4

Пример последовательности команд для работы с каталогами

№	Команда	Описание назначения каждой команды
01	mkdir directory	Первая команда (mkdir) создает каталог directory в текущем каталоге
02	cd directory	Вторая команда (cd) переходит (изменяет каталог) в только что созданный каталог
03	touch filel.txt	Эти две команды touch создают в новом каталоге
04	touch file2.txt	два файла — file1.txt и file2.txt
05	ls	Команда ls без указания каталога выводит содержимое текущего каталога
06	cd . .	Переходим в родительский каталог. Напомним, что '.' - текущий каталог. То есть, находясь в каталоге directory, мы можем обращаться к файлам file1.txt и file2.txt либо без указания каталога, либо ./file1.txt и ./file2.txt
07	ls directory	Поскольку мы находимся в родительском для каталога directory каталоге, то для того чтобы вывести содержимое только что созданного каталога, в команде ls нам нужно четко указать имя каталога
08	rm directory	Команда rm используется для удаления каталога. Но что мы видим: система отказывается удалять каталог!
09	rmdir directory	Пробуем удалить его командой rmdir, но и тут отказ. Система сообщает нам, что каталог не пустой, т. е. содержит файлы
10	rm -r directory	Для удаления каталога нужно удалить все файлы. Конечно, делать это не сильно хочется, поэтому проще указать опцию -r команды rm для рекурсивного удаления каталога. В этом случае сначала будут удалены все подкаталоги (и все файлы в этих подкаталогах), а затем будет удален сам каталог

Еще раз обратите внимание, что в Linux, в отличие от Windows, для разделения элементов пути используется прямой слеш (/), а не обратный (\). Кроме обозначений '. .' и '.' в Linux часто используется обозначение '~' — это домашний каталог. Предположим, что наш домашний каталог

/home/den. В нем мы создали подкаталог dir и поместили в него файл file1.txt. Полный путь к файлу можно записать так:

```
/home/den/dir/filel.txt
```

или так:

```
~/dir/filel.txt
```

Как видите, тильда '~' заменяет часть пути. Удобно? Конечно! Команды cp и mv работают аналогично. Для копирования, перемещения или переименования сначала указывается каталог-источник. а потом каталог-назначение. Для каталогов желательно указывать параметр -г, чтобы копирование или перемещение производилось рекурсивно.

5.6. Ссылки

В Linux допускается, чтобы один и тот же файл был в системе под разными именами. Для этого используются ссылки. Ссылки бывают двух типов: жесткие и символические. Жесткие ссылки жестко привязываются к файлу: вы не можете удалить файл, пока на него указывает хотя бы одна жесткая ссылка. А вот если на файл указывают символические ссылки, его удалению ничто не помешает.

Жесткие ссылки не могут указывать на файл, который находится за пределами файловой системы. Предположим, у вас два Linux-раздела. Один корневой, а второй используется для домашних файлов пользователей и монтируется к каталогу /home корневой файловой системы.

Так вот, вы не можете создать в корневой файловой системе ссылку, которая ссылается на файл в файловой системе, подмонтированной к каталогу /home. Это очень важная особенность жестких ссылок. Если вам нужно создать ссылку на файл, который находится за пределами файловой системы, вам нужно использовать символические ссылки. Для создания ссылок используется команда ln:

```
ln file1.txt link1
ln -s file1.txt link2
```

Первая команда создает жесткую ссылку link1, ссылающуюся на текстовый файл file1.txt. Вторая команда создает символическую ссылку link2, которая ссылается на текстовый файл file1.txt. Модифицируя ссылку (все равно какую — link1 или Iink2), вы автоматически модифицируете исходный файл — file1.txt.

Особого внимания заслуживает операция удаления. По идее, если вы удаляете ссылку link2, файл file1.txt также должен быть удален, но не тут-то было. Вы не можете его удалить до тех пор, пока на него указывает хоть

одна жесткая ссылка. При удалении ссылки link2 просто будет удалена символическая ссылка, но жесткая ссылка и сам файл останутся.

Если же вы удалите ссылку link1, будет удален и файл file.txt, поскольку на него больше не ссылается ни одна жесткая ссылка.

5.7. Перенаправление ввода/вывода при работе с файлами

Теперь, когда вы знаете, как работать с файлами, можно рассмотреть несколько полезных примеров по перенаправлению ввода/вывода. Рассмотрим следующую команду:

```
echo "some text" > file.txt
```

Символ '>' означает, что вывод команды, находящейся слева от символа, будет записан в файл, находящийся справа от знака, при этом файл будет перезаписан. Если вместо символа '>' указаны два символа '>>' то исходный файл не будет перезаписан, а вывод команды будет добавлен в конец файла.

```
echo "some text" > file.txt
echo "more text" >> file.txt
```

```
cat file.txt
some text
more text
```

Кроме символов '>' и '>>' для перенаправления ввода/вывода часто используется вертикальная черта (|).

Предположим, что мы хотим вывести содержимое файла big_text, но в нем очень много строк, и мы ничего не успеем прочитать. Поэтому целесообразно отправить вывод команды cat какой-то другой программе, которая будет выводить файл постранично, например

```
cat big_text | more
```

Конечно, это пример не очень убедительный, потому что для постраничного вывода гораздо удобнее использовать команду less:

```
less big_text
```

Вот еще один интересный пример. Допустим, мы хотим удалить файл file.txt без запроса, для этого можно использовать команду:

```
echo y | rm file.txt
```

В процессе своего выполнения команда rm запросит подтверждение на удаление (нужно нажать клавишу 'Y'), но за нас это сделает команда echo.

И еще один пример. Есть большой файл, и нам нужно найти в нем все строки, содержащие подстроку "Function". Чтобы не делать это вручную, можно воспользоваться командой:

```
cat file.txt | grep "Function"
```

Команда grep может использоваться и самостоятельно. Эта команда предназначена для вывода списка строк, содержащих нужный шаблон.

5.8. Права доступа. Команды chown и chmod

Для каждого каталога и файла вы можете задать права доступа. Точнее, права доступа автоматически задаются при создании каталога/файла, а вам при необходимости нужно их изменить. Какая может быть необходимость? Например, вам нужно, чтобы к вашему файлу-отчету смогли получить доступ пользователи — члены вашей группы. Или вы создали обычный текстовый файл, содержащий инструкции командного интерпретатора. Чтобы этот файл стал сценарием, вам нужно установить право на выполнение для этого файла.

Существуют три права доступа — чтение (r), запись (w), выполнение (x). Для каталога право на выполнение означает право на просмотр содержимого каталога. Вы можете установить разные права доступа для владельца, для группы владельца (то есть для всех пользователей, входящих в одну с владельцем группу) и для прочих пользователей. Пользователь root может получить доступ к любому файлу/каталогу вне зависимости от прав, которые вы установили. Чтобы просмотреть текущие права доступа, надо использовать команду вида:

```
ls -l <имя_файла/каталога>
```

Ниже приведен пример ввода такой команды и результат ее выполнения.

```
ls -l video.txt
-r--r----- 1 ppt group 300 Apr 11 11:11 video.txt
```

Самая левая группа символов r--r-----, полученных в результате выполнения команды, характеризует права доступа к файлу video.txt. Рассмотрим их подробнее:

Первые три левых символа (r--) определяют права доступа владельца файла или каталога. Первый из них соответствует чтению, второй — возможности записи, третий — запуску на выполнение. Как видно, владельцу разрешено только чтение этого файла, а запись и выполнение запрещены, поскольку в правах доступа режимы (w) и (x) не определены.

Следующие три символа (r--) задают права доступа для членов группы владельца. Права такие же, как и у владельца — можно читать файл, но нельзя изменять или запускать.

Последние три символа (-) задают права доступа для прочих пользователей. Прочие пользователи не имеют право ни читать, ни изменять, ни выполнять файл. При попытке получить доступ к файлу они увидят сообщение «Access denied».

Права доступа задаются командой chmod. Существуют два способа указания прав доступа: символьный (когда указываются символы, задающие право доступа — r, w, x) и абсолютный. Так уже заведено, что в мире Linux чаще пользуются абсолютным методом. Разберемся, в чем заключается этот метод. Рассмотрим следующий набор прав доступа:

```
rw-r-----
```

Данный набор прав доступа предоставляет владельцу право чтения и модификации файла (rw-), запускать файл владелец не может. Члены группы владельца могут только просматривать файл (г--), а все остальные пользователи не имеют вообще никакого доступа к файлу. Возьмем отдельный набор прав, например для владельца:

```
rw-
```

Чтение разрешено, значит мысленно записываем 1, запись разрешена, значит запоминаем еще 1, а вот выполнение запрещено, поэтому запоминаем 0. Получается число 110 в двоичной системе. Если число $(100)_2$ перевести в восьмеричную, получится число 6. Аналогично, произведем разбор прав (r--) для членов группы владельца. Получится 100, то есть 4. С третьим набором (---) все вообще просто — это 000, то есть 0. Записываем полученные числа в восьмеричной системе в порядке «владелец — группа — остальные». Получится число 640 — это и есть права доступа. Для того чтобы установить эти права доступа, надо выполнить команду:

```
chmod 640
```

Наиболее популярные права доступа:

644 – владельцу можно читать и изменять файл, остальным пользователям — только читать;

666 – читать и изменять файл можно всем пользователям;

777 – всем можно читать, изменять и выполнять файл;

Напомним, что для каталога право выполнения — это право просмотра оглавления каталога.

Иногда проще воспользоваться символьным методом. Например, у нас есть файл script, который нужно сделать исполнимым, для этого надо использовать команду:

```
chmod +x script
```

Для того чтобы снять право выполнения, задается параметр -x:

```
chmod -x script
```

Подробнее о символьном методе вы сможете прочитать в руководстве по команде chmod (man chmod).

Если вы хотите «подарить» кому-то файл, то есть сделать какого-то пользователя владельцем файла, то вам нужно использовать команду chown:

```
chown <пользователь файл>
```

Учтите, что , возможно, после изменения владельца файла вы сами не сможете получить к нему доступ, ведь владелец уже не вы.

5.9. Монтирование

Термин «дерево каталогов» используют для описания расположения файлов на компьютере практически во всех современных операционных системах. Отличие заключаются только в том, где находится корень этого дерева.

В Windows такими корнями являются так называемые логические диски: C:, D:, A: и т. д. Именно от них отсчитываются пути ко всем файлам. В Linux всегда только один корень — root (*англ.* — корень). Путь к любому файлу на компьютере отсчитывается относительно этого корня.

Особенность логических дисков Windows в том, что каждому из них соответствует раздел винчестера, диск, флешка или любое другое устройство хранения данных. В отличие от этого, Ubuntu содержимому любых устройств с данными отводит определенное место в существующем дереве каталогов. Операция присоединения устройства хранения данных к дереву каталогов называется монтированием, а место присоединения — точкой монтирования.

Для примера предположим, что на винчестере два раздела: для системы и для пользователя. Если используем операционную систему Windows, то она стояла бы в первом разделе (C:), а второй (D:) служил бы для хранения файлов пользователя. В Ubuntu это будет по-другому: первый раздел был бы корнем (/), а второй мог бы стать, например, /userdata/docum.

Это означает, что все содержимое второго раздела будет доступно внутри каталога /userdata/docum, и все файлы, сохраняемые в этот каталог, будут записываться во второй раздел жесткого диска. При этом сам каталог /userdata и все его содержимое находятся в первом разделе. На самом деле, в первом разделе находится даже каталог /userdata/docum, но он пустой, а вот все его содержимое уже находится во втором разделе.

Следует отметить, что не Ubuntu назначает точки монтирования разделов, а пользователь при установке системы. А вот подключаемые устройства, например флешки, Ubuntu монтирует сама, не спрашивая ни про какие точки монтирования. Она автоматически создает каталог внутри /userdata, в который и происходит монтирование, а после отключения устройства этот каталог автоматически же удаляется.

В любом случае, сколько бы у вас ни было разделов на жестком диске и сколько бы вы ни подключили внешних устройств, выглядеть это в Ubuntu будет всегда единообразно: единый корень, с которого начинаются пути ко всем файлам.

5.10. Доступ к файлам

Ubuntu позволяет ограничить доступ к файлу на редактирование или даже на просмотр для той или иной категории пользователей. Так, все системные файлы закрыты для редактирования обычному пользователю, а некоторые вы даже не сможете открыть и посмотреть их содержимое.

Ubuntu, как и любая Linux система воспринимает расширение файлов как часть его имени. Если у какого-либо графического файла удалить его расширение, то он спокойно откроется для просмотра. Конечно расширение нужно, и Ubuntu полагается в первую очередь именно на него, но у нее есть средства определения типа файла, не обращая внимания только на его имя. И это спасает от многих проблем. Не удивляйтесь, встретив в Ubuntu файлы вообще без расширения. С ними система спокойно работает, ничего при этом не спрашивая.

В Ubuntu есть мощная утилита определения типа файла, которая вообще не смотрит на имя и расширение, а пытается идентифицировать файл только по содержимому. Она полезна, когда файл не открывается, и надо убедиться, что в нем данные именно того типа, который вам нужен. Для вызова этой консольной утилиты следует в терминале ввести команду:

```
file <путь_имя_файла >
```

В отношении исполняемых файлов Ubuntu кардинально отличается от Windows. В Windows эти файлы имеют в основном расширение exe, а внутри нечитаемый набор байтов. В Ubuntu исполняемым может быть даже текстовый файл. Что же такое в Ubuntu исполняемый файл?

Фактически это любой файл, который помечен как исполняемый и который Ubuntu может запустить на выполнение. Другое дело, что не любой файл, помеченный как исполняемый, Ubuntu сможет выполнить, хотя в арсенале Ubuntu есть масса методов запуска файлов с совершенно разным содержимым.

Примером необычных исполняемых файлов Ubuntu являются скрипты. Скрипты — это текстовые файлы, содержащие набор инструкций для

программы-интерпретатора, которая занимается собственно выполнением скрипов. Если на скрипте установлено свойство исполняемости, то это программа, которую можно запустить обычным образом, а если не установлено, то это всего лишь текстовый файл.

Из этого вытекает весьма забавное свойство: один и тот же файл может быть одновременно и исполняемым, и не исполняемым, в зависимости от прав доступа. Например, пользователь хозяин файла сможет его запустить как программу, а для всех остальных он будет обычным файлом с данными.

Что касается текстовых файлов, то большинство из них откроется в стандартном текстовом редакторе Ubuntu без проблем, но все же иногда попадаются файлы, которые отображаются искаженно. В этом случае надо помочь Gedit и указать кодировку файла вручную. А вообще, рекомендуют перевести всю текстовую информацию на использование utf8, поскольку фактически только эта кодировка нормально распознается везде и всегда, а в будущем скорее всего только она и будет использоваться.

Если возникла необходимость открыть файл в программе, отличной от той, что запускается по умолчанию, то можно правой клавишей мыши на любом файле вызвать всплывающее меню и выбрать «Свойства» -> «Открывать в программе». В списке будут содержаться все приложения, которые сообщены Ubuntu о поддержке данного типа файлов.

Однако у этого механизма есть одна сложность: чтобы изменить привязку к приложению для определенного типа файлов, вам необходимо иметь файл нужного типа. Просто изменить список всех используемых в системе ассоциаций по умолчанию нельзя.

5.11. Установка программ

По мере работы в любой операционной системе у пользователя возникает потребность в дополнительном программном обеспечении. Что делать, если нужна новая программа для Ubuntu?

5.11.1. Установка программ из репозиториев

Репозиторий – это место централизованного хранения пакетов программного обеспечения. Использование репозиториев позволяет упростить установку программ и обновление системы. Пользователь волен выбирать, какими репозиториями он будет пользоваться, и даже может создать собственный.

Список используемых репозиториев содержится в файле /etc/apt/sources.list и в файлах каталога /etc/apt/sources.list.d/. Этот список проще всего просмотреть через специальное приложение, которое вызывается из главного меню: Система -> Администрирование ->

Источники Приложений (или с помощью Менеджер пакетов Synaptic). Если вы не добавляли локальные репозитории (например, CD/DVD диски), то для установки программ из репозиториев вам потребуется Интернет.

У такого метода масса преимуществ: это просто удобно, вы устанавливаете уже протестированные программы, которые гарантированно будут работать на вашей системе; зависимости между пакетами будут решаться автоматически; при появлении в репозитории новых версий установленных программ вас об этом проинформируют.

Установка с использованием графического интерфейса

Для этого выберите: Система -> Администрирование -> Менеджер пакетов Synaptic — и после ввода своего пароля вам станет доступен довольно функциональный инструмент по работе с пакетами. Используя его, у вас появляется возможность устанавливать программы частично, если вам, например, не нужна документация или еще что-то.

После запуска программы «Менеджер пакетов Synaptic» нажмите кнопку «Обновить» и подождите, пока система обновит данные о доступных программах. В списке доступных программ сделайте двойной клик на нужной программе либо клик правой кнопкой с выбором пункта «Отметить для установки». После того как все нужные программы помечены, нажмите кнопку «Применить». Подождите, пока необходимые пакеты будут скачаны и установлены.

Аналогичные функции доступны в программе «Центр приложений Ubuntu». Она вызывается из основного меню Приложения -> Центр приложений Ubuntu.

Установка с использованием командной строки

Установка из командной строки позволяет получить больше информации о процессе установки и позволяет гибко его настраивать, хотя и может показаться неудобной начинающему пользователю. Вся работа выполняется непосредственно в терминале, который запускается: «Приложение -> Стандартные -> Терминал». Обновить данные о доступных в репозиториях программах можно командой:

```
sudo apt-get update
```

Появится запрос на ввод пароля. Введите свой пароль и помните, что пароль в терминале никак не отображается ни звездочками, ни кружками. Для установки нужной программы следует ввести команду:

```
sudo apt-get install <имя-программы>
```

Например:

```
sudo apt-get install xrdp
```

Если нужно установить несколько программ, то их можно перечислить через пробел, например:

```
sudo apt-get samba samba-client
```

Но не все программы входят в основные репозитории Ubuntu. Поэтому могут быть случаи, когда придется вручную подключать необходимые репозитории с нужными программами или пакетами, или попытаться найти необходимую информацию по их установке на официальном сайте программы. Для поиска программы в списке доступных пакетов можно использовать команду:

```
sudo apt-cache search <ключевое_слово>
```

где <ключевое_слово> — название программы, часть названия программы или слово из ее описания.

5.11.2. Установка программ из deb-пакетов

Пользователя Windows, наверное заинтриговало само название deb-пакет, с которым в Windows он не встречался. Давайте разберемся с этим. Свое название эти пакеты получили от используемого ими расширения имен файлов. Расширение *.deb — это расширение имен файлов «бинарных» пакетов для распространения и установки программного обеспечения в ОС проекта Debian, и других, использующих систему управления пакетами dpkg.

Иконка deb-файлов в среде GNOME имеет вид, как на рисунке слева. Расширение deb — это часть слова Debian, которое образовано от слов Debra и Ian.

Debra — это имя подруги основателя компании «Дебиан» Яна Мердока, а Ian — его собственное имя. Но, вернемся к вопросу установки программ из deb-пакетов.

Итак, если нужной программы нет в репозиториях либо репозитории недоступны, то программу можно установить из deb-пакета, скачанного заранее или принесенного на USB накопителе.

Если deb-пакет есть в официальном репозитории, то его можно скачать без установки. Например, с сайта http://packages.ubuntu.com. Часто deb-пакеты лежат на сайтах разработчиков. Можно также воспользоваться поиском на сайте http://getdeb.net. Минус такого подхода состоит в том, что менеджер обновлений не будет отслеживать появление новых версий установленной программы.

Установка с использованием графического интерфейса

Для установки программы из deb-пакета в режиме графического интерфейса необходимо перейти в папку, где находится deb-пакет, навести

курсор мыши на пиктограмму нужного пакета и щелкнуть два раза. Откроется «Установщик программ GDebi», где надо просто нажать кнопку «Установить пакет». Далее будет выполнена автоматическая установка программы на ваш компьютер. Однако в ряде случаев процесс может быть прерван из-за возникших ошибок. Возможные ошибки:

- Пакет не может быть установлен. Например, он предназначен для другой архитектуры.
- В системе отсутствуют необходимые устанавливаемому приложению пакеты. В таком случае «Установщик программ GDebi» автоматически попытается получить нужные пакеты из репозиториев. Или же вы можете самостоятельно скачать требуемые пакеты и установить их.

Установка с использованием командной строки

Установка программ из deb-пакета выполняется с помощью утилиты dpkg, которая запускается из командной строки терминала Ubuntu. Например:

```
sudo dpkg -i /home/user/soft/telnetd_0.17-35ubuntu1_i386.deb
```

Обратите внимание, что при использовании dpkg нужно ввести полное имя файла, а не только название программы. Следить за ходом установки вы можете в терминале, где будут выводиться все шаги работы утилиты dpkg. Особо обратите внимание на окончательное сообщение. Там будет либо сообщение об успешной установке пакета, либо описание ошибки установки. Например, неудовлетворенные зависимости.

Можно одной командой установить сразу несколько пакетов. Например, следующая команда установит все deb-пакеты, находящиеся в одной директории:

```
sudo dpkg -i /home/user/soft/telnetd_*.deb
```

Это бывает полезно для установки пакета вместе с пакетами зависимостей.

Следует отметить, что пакет на Ubuntu можно установить, если только этот пакет сконфигурирован именно под вашу систему. Для установки используют либо командную строку, либо файловый менеджер Nautilus, где следует просто кликнуть мышкой на нужный пакет, так как ассоциации установки файлов в Ubuntu уже установлены по умолчанию.

5.11.3. Использование менеджера пакетов dpkg в Ubuntu

Утилита dpkg (сокращение от «Debian package») — это менеджер пакетов в Ubuntu Linux. Утилита предназначена для установки, сборки, удаления и менеджмента пакетов Debian.

Действия, выполняемые утилитой dpkg, однозначно определяются заданием опций и параметров, указанных при ее вызове.

Формат команды, использующий вызов утилиты dpkg, имеет вид:

```
dpkg [Опция] [Параметр...]
```

Назначение основных опций утилиты и примеры использования возможных параметров приведены в табл. 5.5.

Таблица 5.5

Назначение основных опций утилиты dpkg

Опция	Назначение
—i	Установка пакета
—I	Показать информацию о пакете
—R	Установка всех пакетов из директории рекурсивно
—r	Удаление установленных пакетов, оставляя конфигурационные файлы
—P	Удаление установленных пакетов вместе с конфигурационными файлами
—A	Обновление по информации из deb-архива
—c	Показать содержимое deb-пакета
—C	Поиск пакетов, которые были установлены в систему только частично
—l	Показать установленные пакеты с номером версии и коротким описанием
—L	Показать список файлов, установленных из пакета
—p	Показать детальную информацию о пакете
—s	Показать статус определенных пакетов
—S	Поиск по имени файла в установленных пакетах
--help	Показать краткую помощь
--unpack	Распаковать пакеты, но не конфигурировать их
--licence	Показать лицензию dpkg
--version	Показать информацию о версии dpkg
--configure	Переконфигурация всех распакованных пакетов
--merge-avail	Добавление информации из пакетов
--update-avail	Замена информации о доступных пакетах
--compare-versions	Сравнение версий пакетов (ver1 op ver2)

Чтобы собрать deb-пакет, следует использовать команду:

```
dpkg -b directory [filename]
```

Более подробную информацию по этой утилите вы можете получить, используя man. Основной и наиболее дружественной GUI «оболочкой» для dpkg является dselect, но она устанавливается отдельно от dpkg.

6. НАСТРОЙКА ЛОКАЛЬНОЙ СЕТИ

Цель данного раздела дать общие сведения по настройке подключения Ubuntu к локальной сети, используя как ее мощные графические средства, так и консольные команды. Но трудно изучать сетевое конфигурирование за одним компьютером. Поэтому еще ранее мы условились, что это изучение проводим на базе виртуальных машин. И в начале данного раздела определимся со структурой системы, на основе которой пойдет дальнейшее изложение.

6.1. Описание архитектуры виртуальной сети

Виртуальную сеть будем реализовывать на базовом компьютере, на котором установлена операционная система MS Windows XP. Этот компьютер физически имеет два сетевых интерфейса. Один из этих интерфейсом является беспроводным и подключен к точке доступа с выходом в Интернет. Второй – проводной и подключен к местной локальной сети.

На основном компьютере развернуто несколько виртуальных машин с различными гостевыми операционными системы. На начальном этапе ограничимся пока всего двумя: MS Windows 98 и Ubuntu 10.04. Общая схема сетевого взаимодействия будет иметь вид, приведенный на рис. 6.1.

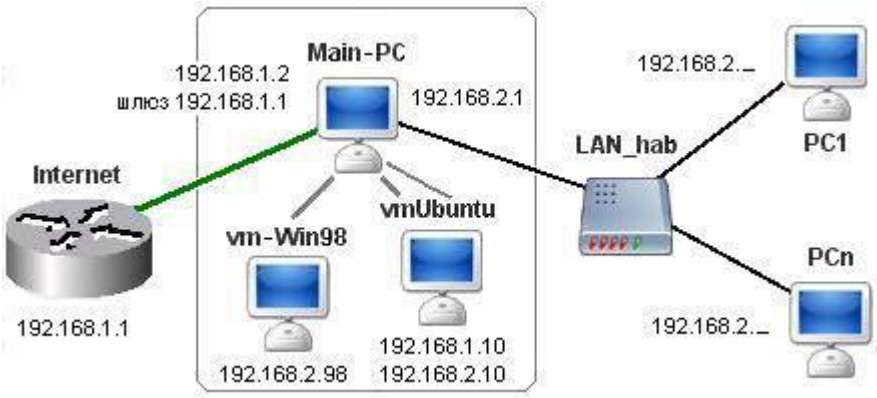

Рис. 6.1. Структура тестовой сетевой архитектуры.

Для возможности построения виртуальной сети на базе виртуальных машин в конфигурации последних должны присутствовать сетевые адаптеры. Их подключение выполняется в консоли Virtual PC отдельно для каждой из виртуальных машин.

Виртуальная машина с Windows 98 будет иметь всего один сетевой интерфейс, а машину с Ubuntu надо сконфигурировать с двумя сетевыми интерфейсами, чтобы была возможность познакомиться с ее работой в разных сетях. Для выполнения этих настроек используется опция Networking в режим Settings консоли Virtual PC (рис. 6.2).

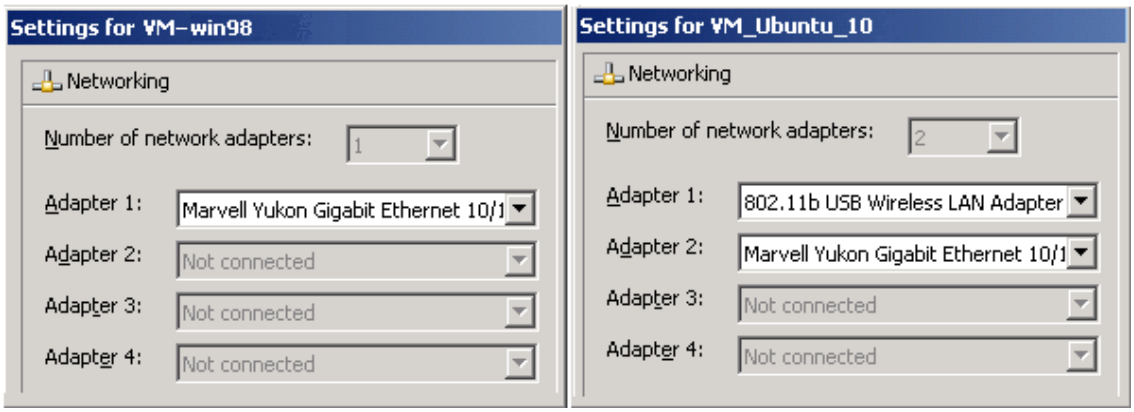

Рис. 6.2. Настройка сетевых интерфейсов виртуальных машин.

Учитывая, что сетевая настройка Windows 98 была выполнена ранее, а настройка Ubuntu по интерфейсу 192.168.1.10, была выполнена автоматически в процессе установки гостевой операционной системы, экран основного компьютера может иметь вид, аналогичный приведенному на рис. 6.3.

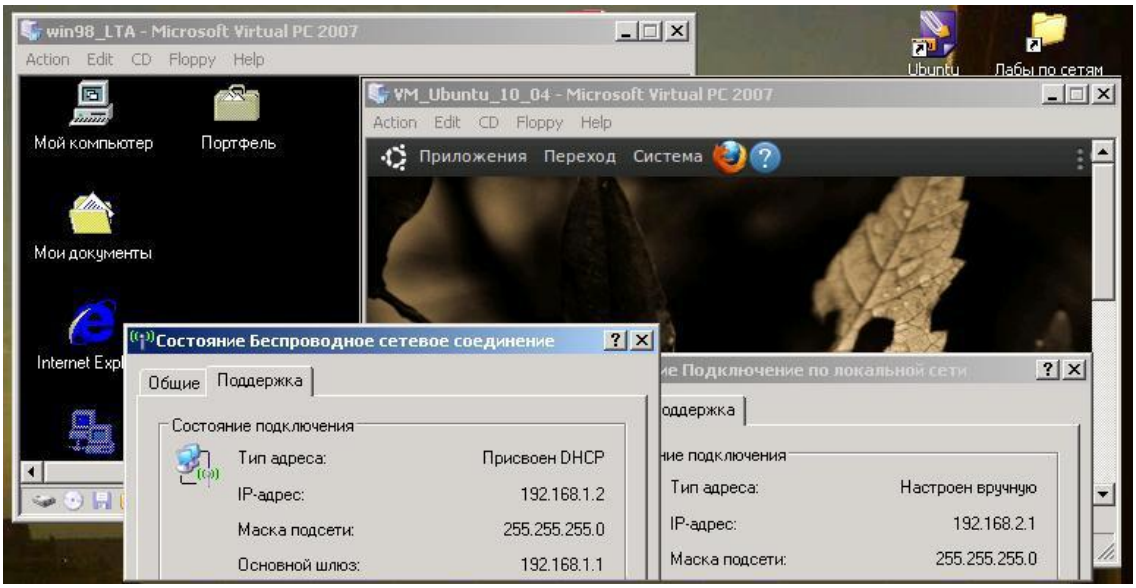

Рис. 6.3. Вид экрана с окнами двух гостевых ОС и параметров сетевых интерфейсов основного компьютера.

Далее будут изложены общие подходы к настройке Ubuntu по подключению к ЛВС, но примеры учитывают описанное выше состояние системы, когда второй интерфейс Ubuntu еще не настроен и не подключен.

6.2. Файлы конфигурации сети в Ubuntu

Приступая к настройке сети, надо иметь представление о файлах конфигурации сети, которые присутствуют в любом дистрибутиве Linux вне зависимости от его версии. Начинающим сетевым администраторам проще эти файлы редактировать с помощью графических конфигураторов — так удобнее. Но всегда полезно знать, где и какие параметры сетевых настроек хранятся.

Таблица 6.1

Файлы конфигурации сети

Файл	Описание
/etc/hosts	В этом файле можно прописать IP-адреса и имена узлов локальной сети, но обычно здесь указывается только IP-адрес узла localhost (127.0.0.1), потому что сейчас даже в небольшой локальной сети устанавливается собственный DNS-сервер
/etc/hosts.allow	Содержит IP-адреса узлов, которым разрешен доступ к сервисам данного узла
/etc/hosts.deny	Содержит IP-адреса узлов, которым запрещен доступ к сервисам данного узла
/etc/iftab	Содержит таблицу интерфейсов, т. е. соответствие имен интерфейсов и их MAC-адресов
/etc/motd	Файл задает сообщение дня (Message of the day). Данный файл используется многими сетевыми сервисами, например, FTP-, SSH-серверами, которые при регистрации пользователя могут выводить сообщение из этого файла
/etc/resolv.conf	Задает IP-адреса серверов DNS
/etc/services	База данных сервисов, задающая соответствие символьного имени сервиса (например, pop3) и номера порта (110/tcp)

6.3. Настройка сети с помощью конфигуратора

В более ранних версиях Ubuntu для запуска конфигуратора настройки сети надо было выбрать «Система -> Администрирование -> Сетевые настройки». Затем выделить «Соединение Ethernet» и нажать кнопку «Свойства» и выполнить настройки. Этого же эффекта можно было добиться, введя команду

```
network-admin
```

В дистрибутиве версии 10.04, с которым нам довелось работать, эта программа не была установлена по умолчанию. Зато в этой версии появился новый красивый инструмент, такой как апплет для управления сетевыми устройствами и соединениями – NetworkManager 0.8 фирм Red Hat и Novell. Он поддерживает практически все типы подключений.

О состоянии сетевых соединений и о своей работе Network Manager информирует пользователей пиктограммой в области уведомлений — правая часть верхней панели экрана. Вид пиктограммы меняется в зависимости от состояния соединения и его типа (рис. 6.4).

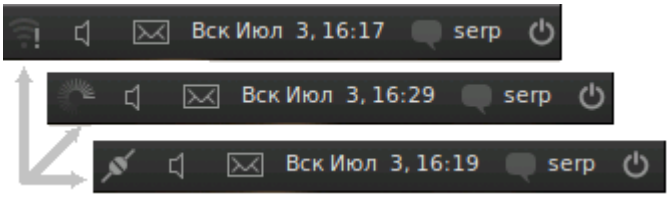

Рис. 6.4. Область уведомлений: соединение отсутствует, соединение устанавливается, соединение установлено.

NetworkManager поддерживает практически все существующие типы подключений:

- Wired — обычные соединения через кабель.
- Wireless — соединения через Wi-Fi адаптер.
- Mobile Broadband — соединения через сети GSM/3G с использованием мобильных телефонов в качестве модемов.
- VPN — зашифрованные соединения через VPN-туннели.
- DSL — PPPoE и модемные соединения.

Если нажать левой кнопкой мыши на пиктограмме NetworkManager в области уведомлений, то появится меню со списком доступных и активных подключений. В нем отображаются найденные беспроводные сети и установленные подключения. Можно активировать любое подключение, просто выбрав его из списка, или же, наоборот, прервать любое активное соединение, нажав на «Отключиться» под его названием.

Познакомимся с использованием приложения NetworkManager для подключения виртуальной Ubuntu-машины к ЛВС. С этой целью рассмотрим простую задачу.

Пусть требуется по одному из доступных сетевых интерфейсов, а именно по проводному интерфейсу, организовать для нашей виртуальной машины новое сетевое соединение со статическим IP-адресом 192.168.2.10.

Для реализации поставленной задачи нажмем правую кнопку мыши на пиктограмме NetworkManager в области уведомлений. Появится меню (рис 6.5), в котором выбраем опцию «Изменить соединения ...». Откроется окно редактора сетевых соединений (его можно открыть и из основного меню, выбрав Система -> Параметры -> Сетевые соединения).

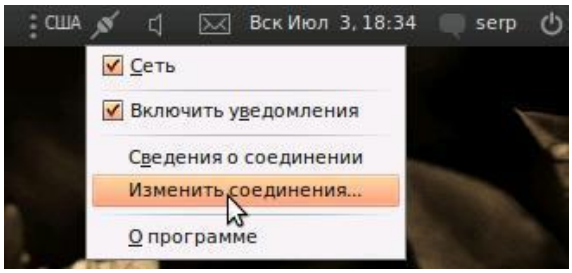

Рис. 6.5. Меню создание и изменение сетевых соединений.

В списке редактора соединений (рис. 6.6) по умолчанию выводятся все автоматически созданные подключения, по одному для каждого сетевого адаптера. Отметим, что сетевые адаптеры, так же как и разделы винчестера, имеют в Linux вполне определенные имена: eth0, eth1 и т. д.

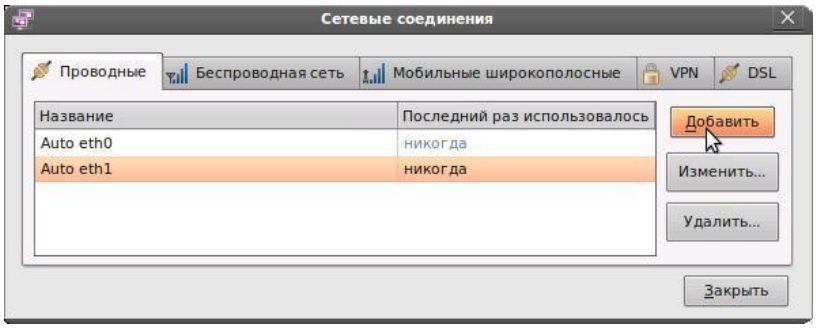

Рис. 6.6. Редактор соединений.

Из двух сетевых адаптеров, определенных для виртуальной машины выбираем тот, который будем настраивать, и нажимаем кнопку «Добавить». Откроется окно редактирования соединения.

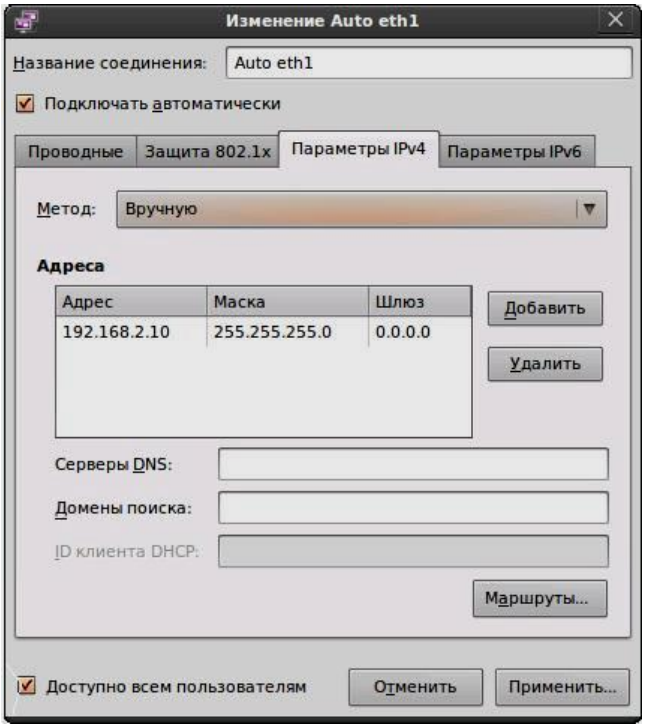

Рис. 6.7. Параметры сетевого соединения.

Первая вкладка содержит технические параметры, которые менять обычно не нужно. Для настройки IP-адресов следует перейти на вкладку «Параметры IPv4».

По умолчанию любое новое соединение настраивается на самый распространенный случай, то есть на получение IP-адреса и адресов DNS автоматически при подключении кабеля. Чаще всего такие соединения используются при выходе в сеть через различные роутеры и прочее сетевое оборудование. Поэтому, если вам нужен такой способ подключения, то вообще ничего не надо изменять.

Нам же требуется ручное указание IP-адреса. Для этого в выпадающем списке «Метод» выбираем «Вручную», устанавливаем требуемые параметры сетевого соединения (рис. 6.7) и нажимаем кнопку «Применить». Следует отметить, что адреса DNS, если их несколько, задаются через запятую.

Сетевое соединение установлено. Остается только его проверить. С этой целью на основном компьютере и виртуальной машине с Windows 98 открываем командную строку и выполняем, для нашего примера, команду

```
ping 192.168.2.10
```

Если настройка произведена правильно, то экран основного компьютера должен иметь вид, аналогичный рис. 6.8. То есть внутри основного компьютера у нас функционирует ЛВС из трех виртуальных компьютеров.

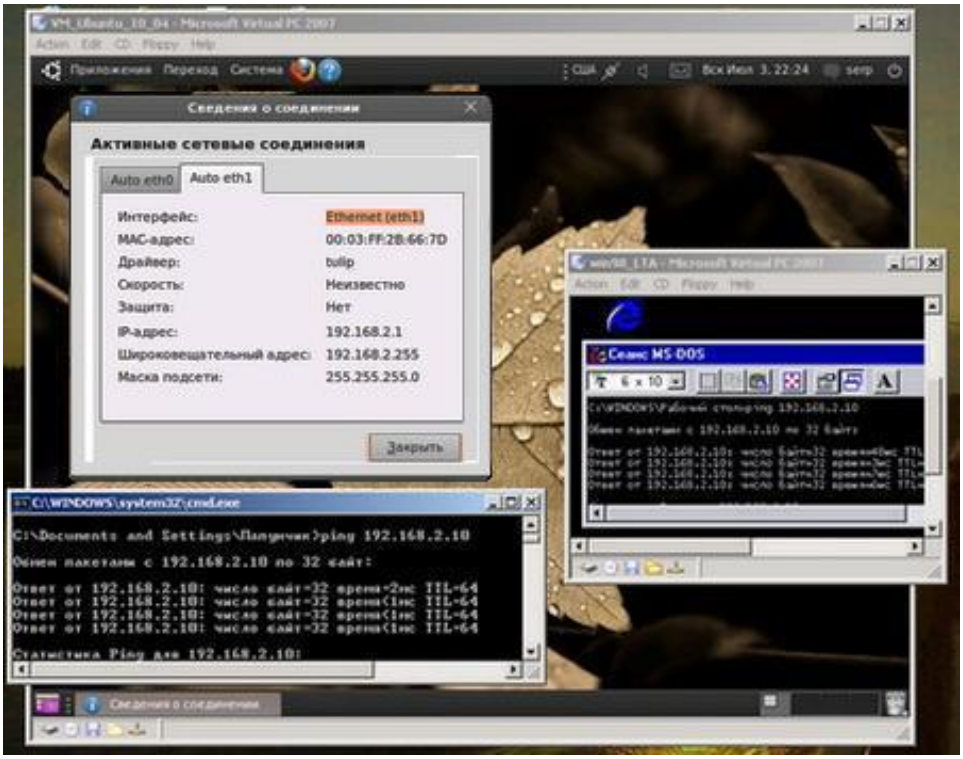

Рис. 6.8. Тестирование сетевого соединения.

6.4. Сетевые утилиты

Достаточно привлекательной в Ubuntu представляется утилита «Сетевые инструменты», которую можно вызвать из основного меню Система -> Администрирование -> Сетевые инструменты. В Windows такой программы не было. На некоторых сайтах подобные утилиты позиционируются как утилиты для взлома. На самом деле к взлому системы они имеют лишь косвенное отношение. Истинное их предназначение — это информирование администратора о состоянии его сети.

Запустите эту программу. На вкладке «Устройства» вы найдете исчерпывающую информацию о состоянии ваших сетевых устройств. Нужное устройство выбирается в раскрывающемся списке «Сетевое устройство». Полезной на этой вкладке является статистика интерфейса — информация о принятом/переданном трафике. Отметим, что информацию, приведенную на этой вкладке, можно получить с помощью команды ifconfig, которая работает в терминальном режиме. Назначение остальных вкладок «Сетевые инструменты»:

- Пинг — позволяет пропинговать заданный узел, то есть отправить определенное количество echo-запросов к этому узлу. Если будет echo-ответ — узел доступен, если нет — проблемы с сетью или узел выключен. Вкладку заменяет текстовая утилита ping.

- Сетевая статистика — получает общую информацию о настройках сети (таблицу маршрутизации, выводит активные сетевые сервисы, информацию о мультикасте — групповой отправке пакетов).

- Трассировка маршрута — позволяет вывести маршрут до заданного узла, то есть вывести адреса узлов, через которые проходит пакет до заданного вами узла. Вкладку заменяет утилита tracepath, а для редактирования таблицы маршрутизации используется утилита route.

- Сканирование портов — позволяет получить информацию о том, какие сетевые службы запущены на локальном компьютере. Если надо знать, какие службы запущены на удаленном компьютере, то введите IP-адрес этого компьютера или его доменное имя. Вкладка представляет собой простейший сканер портов.

- Просмотр — позволяет преобразовать IP-адрес узла в доменное имя и наоборот, используя DNS-серверы по умолчанию. Вкладка частично заменяет утилиту nslookup, возможности которой значительно шире.

- Пользователи (Finger) — позволяет получить информацию о компьютере и его пользователе с помощью службы Finger. Обычно такая служба отключена, поэтому вкладка сейчас бесполезна.

➢ Домены (Whois) — позволяет проверить, на кого зарегистрировано то или иное доменное имя, и зарегистрировано ли оно вообще.

Если в поле «Доменный адрес» ввести какой-либо адрес, например yandex.ru, и нажать Enter, то через несколько секунд будут отображены: тип домена, используемые серверы, организация (или частное лицо) кому принадлежит домен, номер телефона, Email, регистратор, дата создания и до какой даты оплачен домен.

В качестве примера используем утилиту «Сетевые инструменты» для проверки работоспособности настроенного нами сетевого соединения с основным компьютером (192.168.2.1) и машиной Windows 98 (192.168.2.98). Для этого откроем вкладку «Пинг» и в поле сетевой адрес, введем адреса одного узла, а потом — второго.

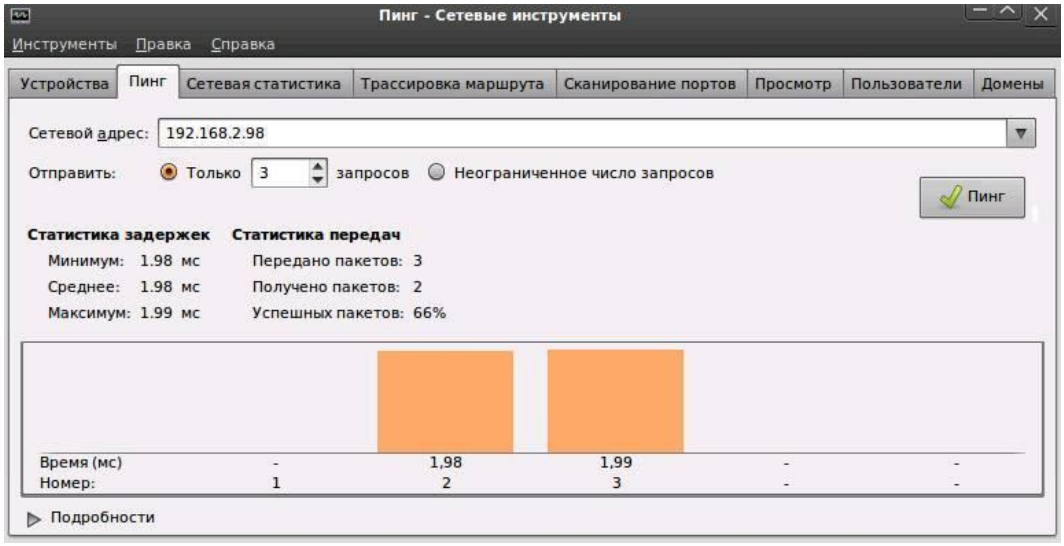

Рис. 6.9. Тестирование сетевого соединения из Сетевых инструментов.

Результат, приведенный на рис. 6.9, показывает, что связь виртуальной машины на базе Ubuntu с виртуальной машиной на базе Windows 98 — присутствует. То есть настройка сетевого соединения выполнена правильно, и виртуальная сеть внутри основного компьютера работает.

6.5. Доступ к общесетевым папкам ЛВС из Ubuntu

После того как вы убедились, что сетевые подключения выполнены правильно и у вас есть доступ к узлам ЛВС, следует проверить наличие доступа к общесетевым ресурсам этих узлов. Если на основном компьютере сформирована структура, аналогичная рис. 6.1, и на виртуальной машине с Windows 98 и основном компьютере есть, например, общедоступные папки, то для их просмотра с виртуальной машины с Ubuntu достаточно выбрать «Переход -> Сеть». При этом откроется обозреватель файлов (рис. 6.10), вид которого очень напоминает вид сетевого окружения в Windows.

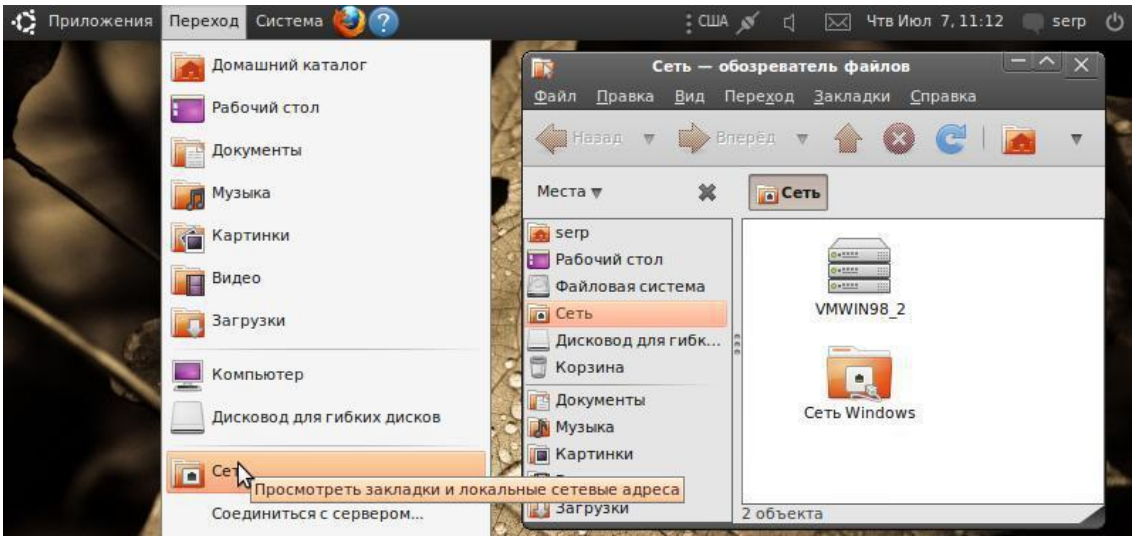

Рис. 6.10. Просмотр доступных общесетевых ресурсов.

Перемещаясь по пиктограммам окна «Обозревателя файлов», вы можете добраться до интересующей вас папки и получить к ней доступ. Если, конечно, имеете на это право. Так, например, не будет никаких проблем, если папка на виртуальной машине с Windows 98 открыта с полным гостевым доступом.

Возможен и иной подход на доступ к ресурсу ЛВС. Например, чтобы добраться до общедоступных ресурсов виртуальной машины Windows 98, можно с ней установить соединение. Для этого в основном меню выбираем Переход -> Соединиться с сервером. На экране появится окно «Соединение с сервером» (рис. 6.11).

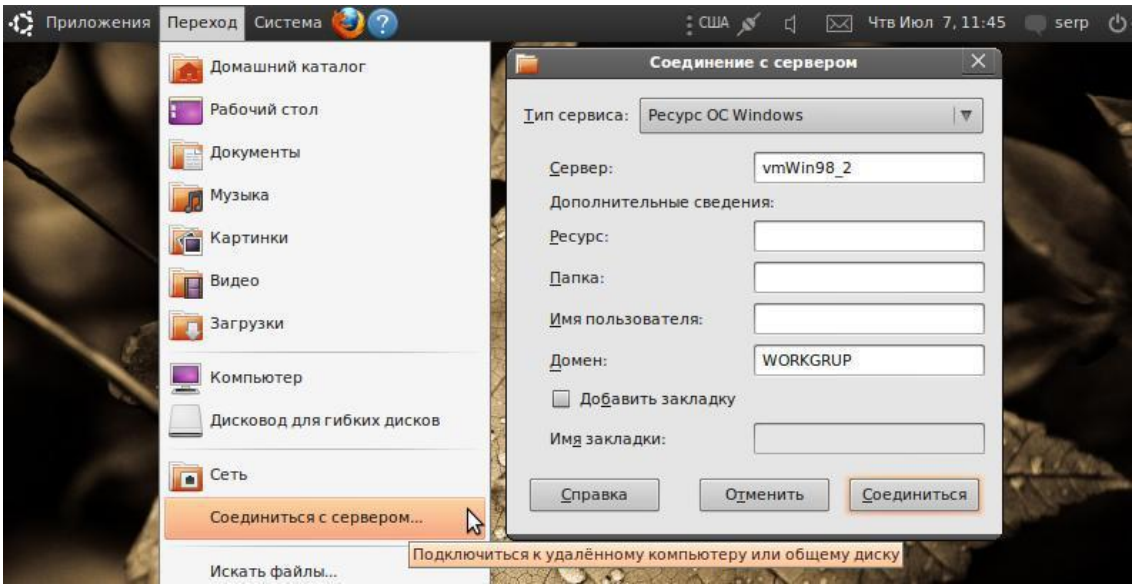

Рис. 6.11. Подключение к удаленному серверу.

В этом окне в выпадающем списке «Тип сервиса» выбираем «Ресурс ОС Windows», в поле «Сервер» указываем сетевое имя виртуальной машины с Windows 98, а в поле «Домен» — наименование рабочей группы,

в которую входит эта виртуальная машина. После этого нажмем кнопку «Соединиться» и откроется окно «Обозреватель файлов», с перечнем общедоступных ресурсов на машины с Windows 98. Вид его будет аналогичен скриншоту на рис. 6.10.

Следует отметить особенность данного режима доступа, которая заключается в том, что установленное соединение Ubuntu запоминает и создает указатель на это подключение. После выбора какого-либо ресурса на удаленном компьютере на своем рабочем столе Ubuntu создаст ярлык на доступ к этому ресурсу, а в основном меню «Переход» появится дополнительная опция со ссылкой на этот ресурс (рис. 6.12).

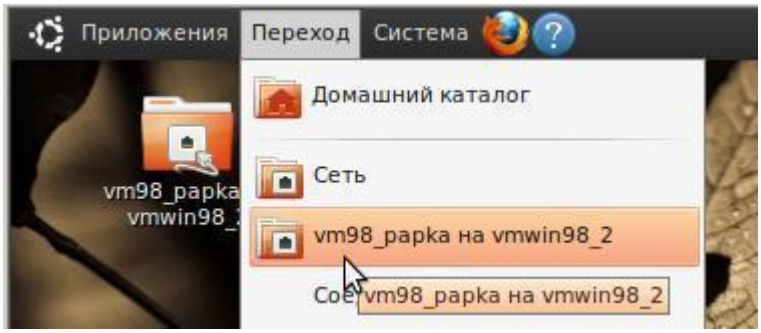

Рис. 6.12. Сформированные Ubuntu ссылки на удаленный ресурс.

Это позволяет в дальнейшем использовать удаленную папку, не устанавливая повторно сетевого подключения или поиска ресурса в локальной сети.

6.6. Доступ к ресурсам Интернет

Доступ к Интернету во многом определяется как каналом, который вы используете для подключения, так и типом провайдера, предоставляющим вам доступ. Поэтому настройки подключения могут быть различными. Если вы настраиваете доступ из локальной сети, то это во многом определяется структурой ЛВС и настройкой ее сетевым администратором.

В данном разделе мы познакомим вас с возможностью подключения виртуальной машины Ubuntu к Интернету на базе тестовой (рис. 6.1) архитектуры сети.

Как уже отмечалось, наш основной компьютер имеет IP-адрес 192.168.1.2 и беспроводным каналом связан с маршрутизатором Интернет, IP-адрес которого 192.168.1.1.

Но так как один из сетевых интерфейсов Ubuntu-машины, а именно eth0, имеет адрес 192.168.1.10, то она находится в одной логической сети с основным компьютером и маршрутизатором. Доступность соединения можно проверить, используя команды:

```
ping 192.168.1.2 , ping 192.168.1.1
```

Но если сейчас выполнить Приложения -> Интернет -> Firefox Web Browser, нам будет выдано окно, в заголовке которого будет:

```
Проблема при загрузке страницы - Mozilla Firefox
```

Дело в том, что запросы, которые идут от нашего компьютера, распространяются только внутри ЛВС, а интересующий нас ресурс находится в другой сети. Чтобы была возможность перенаправлять запросы из локальной сети в глобальную сеть, следует использовать маршрутизатор.

Но для этого нашей Ubuntu-машине надо указать, что маршрутизатор с IP-адресом 192.168.1.1 — это тот шлюз, через который наша ЛВС соединяется с глобальной сетью. Для этого в сетевых настройках eth0 (рис. 6.6, 6.7), адрес маршрутизатора должен быть указан в качестве шлюза для виртуальной Ubuntu-машины (рис. 6.13).

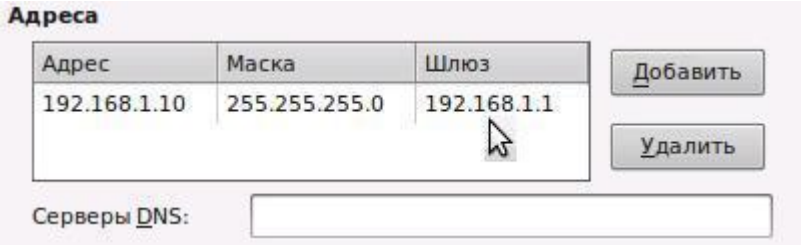

Рис. 6.13. Изменение настройки интерфейса eth0.

Если мы снова выполним Приложения -> Интернет -> Firefox Web Browser, а в строке ресурса введем www.ya.ru , то получим, то же самое сообщение о проблемах с загрузкой страницы.

Так, что же шлюз, не работает? Или мы его неправильно настроили? Не торопитесь. Попробуйте в Web-браузере в строке ресурса набрать следующий адрес:

```
87.250.251.3
```

И если шлюз настроен правильно, то на экране должно появиться окно, аналогичное тому, что приведено на рис. 6.14.

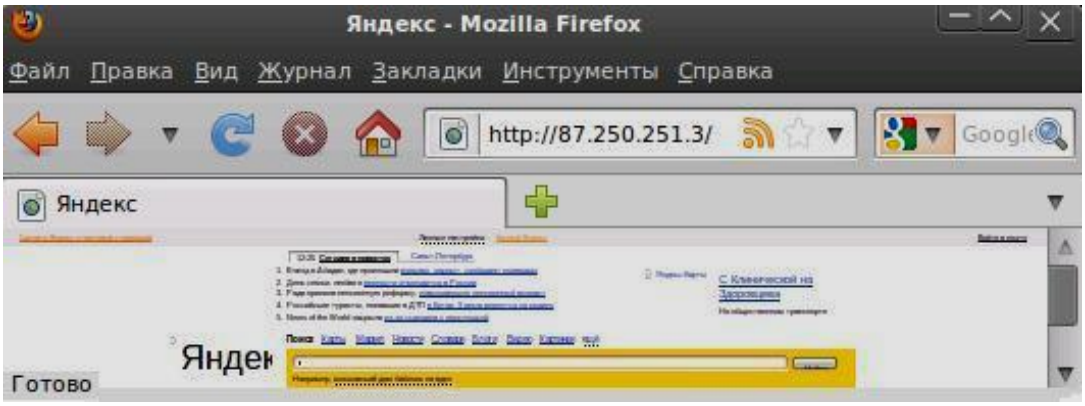

Рис. 6.14. Окно Web-браузера при доступе к ресурсу по IP-адресу.

То есть Ubuntu-машина имеет доступ из локальной сети в глобальную сеть, и шлюз настроен правильно. Только вы должны использовать IP-адреса ресурсов, а не их символические аналоги.

Вы спросите: и зачем мне это нужно? Дело в том, что если вы хотите иметь представление о Linux и сетевом администрировании, то со временем многие IP-адреса сетевых ресурсов вам будут хорошо знакомы.

Так, например, российское зеркало по Linux-системам (mirror.yandex.ru) — это не просто Интернет-сайт, но и FTP-сервер, где можно найти образы практически всех клонов Linux. А что касается Ubuntu, то все версии и обновления к ним.

Что бы установить соединение с этим FTP-ресурсом, следует выполнить Переход -> Соединиться с сервером. В окне «Соединение с сервером» выбрать тип сервиса и адрес сервера (рис. 6.15).

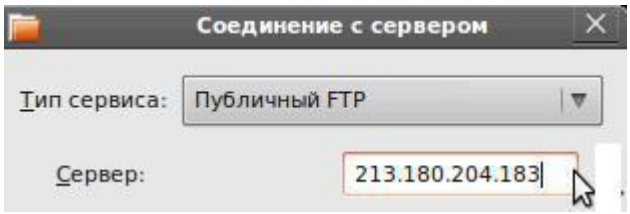

Рис. 6.15. Подключение к российскому зеркалу Linux-систем.

После нажатия кнопки «Соединиться» откроется окно «Обозреватель файлов», в котором будут представлены все ресурсы этого сервера, включая каталоги версий Ubuntu.

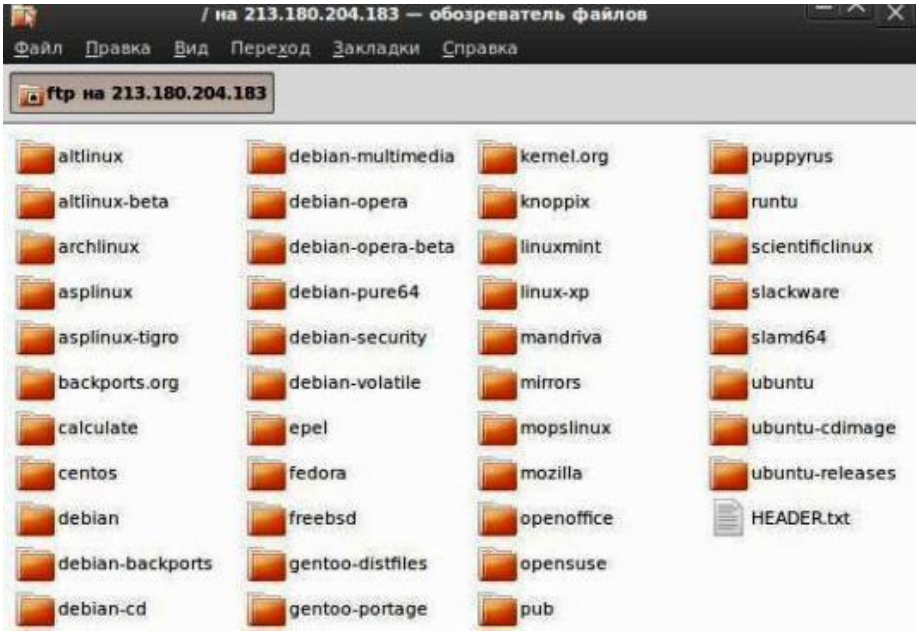

Рис. 6.16. Подключение к российскому зеркалу Linux-систем.

Но вернемся к установке на Ubuntu-машине доступа к Интернету. Все-таки, она пока не понимает символических имен ресурсов Интернет. За это в сети отвечают DNS-сервера. В нашей крошечной сети такого нет.

Поэтому попробуем воспользоваться внешними. Но если мы не знаем их адресов, то простейшим является в качестве DNS-сервера указать адрес маршрутизатора нашей сети. Он перешлет запросы на идентификацию имен какому-либо серверу DNS высшего уровня.

Для этого следует еще раз вернуться в редактирование сетевых соединений и для сетевого интерфейса eth0 указать адрес сервера DNS (рис. 6.17). Если к моменту настройки сети нам будут известны и адреса других DNS-серверов, то их можно указать в поле «Серверы DNS: » через запятую.

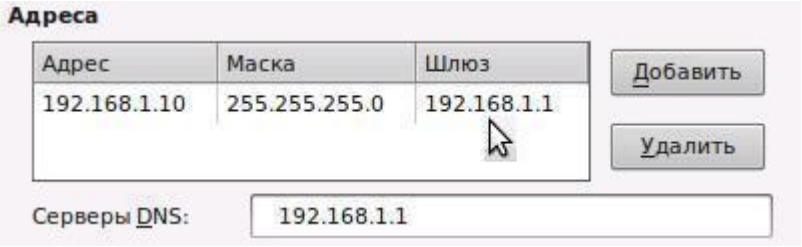

Рис. 6.17. Изменение настройки интерфейса eth0.

Если изменения были сделаны правильно и Ubuntu-машина была перезагружена, то выполняя Приложения -> Интернет -> Firefox Web Browser и вводя в строке ресурса www.ya.ru, система должна получить доступ к сайту Яндекс. При этом вид экрана будет аналогичен рис. 6.14.

Имея полноценно настроенный Интернет, вы можете вернуться в окно сетевых утилит «Сетевые инструменты» (рис. 6.9) и более полно познакомиться с их работой.

На вкладке «Просмотр», введя сетевой адрес www.ya.ru, можно получить IP-адрес этого ресурса, да еще и не один.

На вкладке «Трассировка маршрута», введя сетевой адрес www.ya.ru, можно получить список всех промежуточных узлов, через которые идет соединение вашего компьютера с сервисом Яндекс. Причем станет ясно, с каким именно из физических серверов установлена связь, так как одному доменному имени www.ya.ru может соответствовать несколько IP-адресов.

6.7. Разрешение имен в ЛВС

Все компьютеры нашей виртуальной сети, как и любой другой ЛВС, имеет два имени. Это IP-адрес и символическое имя. Как они согласуются между собой в нашей виртуальной сети? Ведь никаких серверов DNS в ней пока нет.

То, что у нас сетевые узлы доступны между собой, мы уже оттестировали в предыдущих разделах. Но проверка соединения выполнялась на основе IP-адресов отдельных узлов, но не их символических имен. Давайте немного раздвинем область наших экспериментов и расширим исходную сеть (рис. 6.1),

С этой целью подключим к ней еще две виртуальные машины. Естественно, что эти машины должны быть заранее созданы и на них должны быть установлены какие-либо Windows- или Linux-подобные ОС.

Без ущерба для общности рассмотрения вопросов взаимодействия Ubuntu- и Windows- компьютеров, такими ОС могут быть Ubuntu 6.10 и Windows XP, сконфигурированные в минимальном варианте для экономии памяти основного компьютера.

Задание.
Создайте в консоли MS Virtual PC две новые виртуальные машины vmUbuntu06 (192.168.2.6/24) и vm-WinXP (192.168.2.44/24) и выполните тестирование расширенной виртуальной сети.

После того как все будет настроено, проведем небольшой эксперимент, используя для него только основной компьютер Main-PC (192.168.2.2/24) и виртуальный vmUbuntu10 (192.168.2.10/24).

На первом этапе эксперимента на каждом из этих двух компьютеров войдем в командную консоль и выполним одну и ту же группу консольных команд, а именно:

```
ping 192.168.2.2
ping 192.168.2.6
ping 192.168.2.10
ping 192.168.2.44
ping 192.168.2.98
```

На этом этапе эксперимента сложностей не должно возникнуть. Если они проявились, то проверьте сетевые настройки на каждой из виртуальных машин и устраните выявленные недостатки. Только после этого переходим ко второму этапу эксперимента.

На втором этапе эксперимента, будем использовать те же два виртуальных компьютера (192.168.2.2/24) и vmUbuntu10 (192.168.2.10/24), но выполним другую группу команд, а именно:

```
ping Main-PC
ping vmUbuntu06
ping vmUbuntu10
ping vm-WinXP
ping vm-Win98
```

То есть будем тестировать сетевые подключения, обращаясь к узлам сети не по IP-адресам, а по символическим именам каждого из виртуальных узлов сети.

Выполняя эти команды на основном компьютере Main-PC, при доступе на компьютеры, на которых установлена какая-либо из версий Windows, вы должны были получить результат, аналогичный тому, что приведен ниже.

```
C:\ping vm-Win98
```

```
Обмен пакетами с vm-Win98 [192.168.1.98] по 32 байт:
Ответ от 192.168.1.98: число байт=32 время=7мс TTL=128
. . .
Статистика Ping для 192.168.1.98:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
(0% потерь)
```

А вот при доступе с того же самого основного компьютера Main-PC на компьютеры с любой версией Ubuntu результат будет иной, а именно:

```
C:\ping vmUbuntu06
При проверке связи не удалось обнаружить узел vmUbuntu06.
Проверьте имя узла и повторите попытку.
```

И совсем удручающий результат мы получим, выполняя эти команды на vmUbuntu10 (192.168.2.10/24). За исключением доступа к себе самой, результат будет аналогичен:

```
serp@vmUbuntu10:~$ ping vm-WinXP
ping: unknown host vm-WinXP
```

Работая в сети Windows даже на простейшем уровне — локальной рабочей группы, мы даже не задумывались о проблеме разрешения имен. Это связано с тем, что при настройке доступа к локальной сети в сетевых настройках мы указывали, что наш компьютер является клиентом для сетей Microsoft.

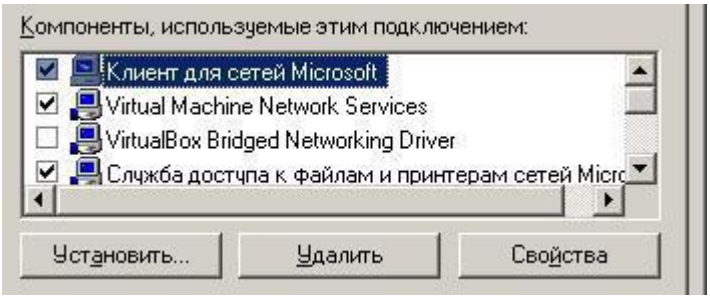

Рис. 6.18. Элемент настройки сети в Windows.

Тем самым мы указывали, что наш компьютер является узлом сетевой операционной системы SMB/CIFS, разработанной Microsoft. Именно она, помимо других сетевых функций, возложила на себе функции разрешения имен, используя для этого свои скрытые внутренние ресурсы.

Каким же образом в нашей Ubuntu-Windows сети решить проблему доступа к компьютеру по его символическому имени, то есть решить задачу разрешения имен узлов в ЛВС.

Самым простым способом для маленькой ЛВС является подход с использованием статических таблиц для поиска имён узлов, для хранения которых используется файл /etc/hosts.

Описание статической таблицы для поиска имен узлов доступно на Ubuntu-машине, при использовании команды:

```
man hosts
```

В ответ на эту команду на экране вашего компьютера появится страница руководства, которая в полном объеме приводится ниже.

hosts — Статическая таблица для поиска имен узлов

ОПИСАНИЕ

Данная страница руководства описывает формат файла /etc/hosts. Это простой текстовый файл, который ассоциирует IP адреса с именами узлов, по одному IP адресу в строке. Для каждого узла в одной строке должна быть представлена запись со следующей информацией:

```
IP_адрес каноническое_имя_узла псевдонимы
```

Поля записи разделяются пробелами и/или символами табуляции. Текст, начинающийся с символа «#» до конца строки считается комментарием и игнорируется. Имена узлов могут содержать только буквы, цифры, знак минус («-») и точку («.»). Они должны начинаться с буквы и заканчиваться буквой или цифрой.

Псевдонимы предоставляются для возможности выбора более одного имени, альтернативного произношения, сокращения имени узла или для указания наиболее общего имени узла (например, localhost). Формат файла описывается в RFC 952.

DNS сервер Berkeley Internet Name Domain (BIND) реализует сервер службы имен для UNIX систем. Он расширяет или замещает файл /etc/hosts при операциях поиска имени узла, а также освобождает от необходимости поддерживать актуальность и полноту /etc/hosts.

В современных системах, даже заданная в файле /etc/hosts информация может быть перекрыта информацией из DNS, это широко используется для следующих случаев:

- Начальная загрузка.
- Большинство систем имеют маленький размер файла /etc/hosts, который обычно содержит имена и адреса наиболее важных узлов локальной сети. Это полезно, когда служба DNS не запущена, например во время загрузки системы.
- NIS сайты,
- которые используют NIS. Хранят таблицу узлов в базе данных узлов NIS. Но даже при работе с NIS остается возможность использовать DNS, большинство NIS сайтов также используют для целей резервного копирования и файл /etc/hosts, где размещаются записи обо всех локальных узлах.

- Изолированные узлы.
- Маленькие сайты, которые являются изолированными от сети, используют файл /etc/hosts вместо DNS. Если локальная информация меняется редко и сеть не подключена к Интернет, DNS не дает ощутимых преимуществ.

ПРИМЕР

```
127.0.0.1       localhost
192.168.1.10    foo.mydomain.org    foo
192.168.1.13    bar.mydomain.org    bar
216.234.231.5   master.debian.org   master
205.230.163.103 www.opensource.org
```

ИСТОРИЧЕСКОЕ ЗАМЕЧАНИЕ

Перед появлением DNS, файл с таблицей узлов /etc/hosts был единственным способом определения имен узлов по IP адресам в развивающейся сети Интернет.

В самом деле, этот файл мог быть создан из официальной базы данных узлов, которая обслуживалась Центром управления сетевой информацией **(**Network Information Control Center (NIC)**)**, и далее с помощью локальных изменений, которые часто требовались, чтобы поддержать актуальность данных и чтобы учитывать неофициальные псевдонимы и/или неизвестные узлы. NIC больше не поддерживает файлы hosts.txt (приблизительно с 2000 года), но на их WWW существуют исторические файлы hosts.txt. Мы обнаружили три: от 92, 94 и 95 годов.

АВТОР

Данную страницу руководства написал Manoj Srivastava <srivasta@debian.org>, для системы Debian GNU/Linux.

ПЕРЕВОД

Перевел с английского В. Вислобоков <corochoone@perm.ru>

Обратите внимание в этом описании на пункт изолированные узлы, где подчеркивается мысль о том, что в небольших ЛВС не обязательна установка DNS-сервера, и подход с использованием /etc/hosts может оказаться совсем не устаревшим.

Упражнение.

Внимательно прочитав это описание, ваша задача настроить виртуальную сеть таким образом, чтобы был возможен доступ к любому узлу сети по его символическому имени.

Если вы сделаете все правильно, то второй этап рассмотренного выше эксперимента должен завершиться успешно.

Для сведения: точно такой же файл со статической таблицей для поиска имен узлов, существует и в Windows. В случае Windows XP этот файл находится в C:\WINDOWS\system32\drivers\etc\hosts.

Простейшие, но наиболее часто применяемые команды, используемые в сетевом администрировании, рассмотрены в Приложении к данному разделу, приведенному в конце книги. Следует отметить, что эти команды могут использоваться для сетевых настроек как в местном, так и удаленном терминальном доступе на Linux-компьютерах.

Кроме этого рекомендуем самостоятельно познакомиться с содержимым конфигурационных файлов, указанных в разделе 6.2. Изменения во всех конфигурационных файлах могут выполняться только от имени суперпользователя root.

7. УДАЛЕННЫЙ ДОСТУП В UBUNTU

Даже небольшая корпоративная сеть может содержать несколько серверов, находящихся в разных подразделениях предприятия. Странно будет выглядеть администратор этой сети, бегущий между этажами для локализации какого-либо сбоя. А что ему делать, если надо на всех компьютерах обновить какой-либо софт? Еще хуже ситуация, когда нет доступа к сетевому принтеру при формировании квартального отчета, а администратор взял отгул и занимается дома семейными делами. Ясно, что даже простейшее сетевое администрирование предполагает наличие удаленного доступа, как минимум к основным серверам и как максимум ко всем компьютерам ЛВС.

Исторически первым для удаленного доступа к консоли сервера использовался протокол telnet. Во всех сетевых операционных системах есть свой telnet-клиент. Программа так и называется — telnet. После подключения с ее помощью к удаленному компьютеру с ним можно работать, как обычно.

В окне telnet-клиента отображается как бы консоль удаленного компьютера — можно вводить команды и получать результат их выполнения. Все так, как если бы вы работали непосредственно за удаленным компьютером.

Со временем telnet устарел, и сейчас им практически никто не пользуется. Ему на смену пришел протокол SSH (Secure Shell), который, как видно из названия, представляет собой безопасную оболочку. Он отличается от telnet тем, что все данные, включая пароли доступа к удаленному компьютеру, передаются в зашифрованном виде. Во времена telnet были случаи перехвата паролей и другой информации, что и стало причиной создания SSH. Краткие сведения о протоколе SSH приведены в Приложении.

Протокол SSH для шифрования передаваемых данных использует следующие алгоритмы: BIowFish, 3DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm) и RSA (Rivest-Shamir-Adelman algorithm). Самыми надежными считаются IDEA и RSA. Поэтому для передачи действительно конфиденциальных данных лучше использовать один из этих алгоритмов. SSH-клиенты и SSH-серверы имеются для большинства операционных систем.

Технологии не стоят на месте, каналы связи становятся все более скоростными и у вас появляется возможность подключаться к удаленным

компьютерам с телефона или нетбука, да еще используя все красоту графических сред. Но об этом в следующем разделе.

А сейчас знакомство с удаленным доступом к Ubuntu-машине в сетях, содержащих Windows- и Linux-компьютеры начнем с наиболее простой, но эффективной технологии, использующей работу в удаленной консоли.

7.1. Установка SSH-сервера

Использование SSH — это легкий и безопасный способ получить доступ к удаленной Ubuntu-машине. Как правило, этот способ соединения используют для удаленного выполнения команд в консольном режиме. Для установки на Ubuntu-машине ssh-сервера набираем в консоли команду:

```
sudo apt-get install openssh-server
```

Если Ubuntu-машина, с уже настроенными сетевыми интерфейсами имеет доступ в Интернет, то проблем не будет. При отсутствии доступа в Интернет вам необходимо:

- На Ubuntu-машине, подключенной к локальной сети лаборатории, выбрать Переход -> Сеть.
- После того как откроется обозреватель файлов с доступом к обнаруженной Windows сети, добраться до общесетевого ресурса с программным обеспечением SSH-сервера (рис. 7.1).

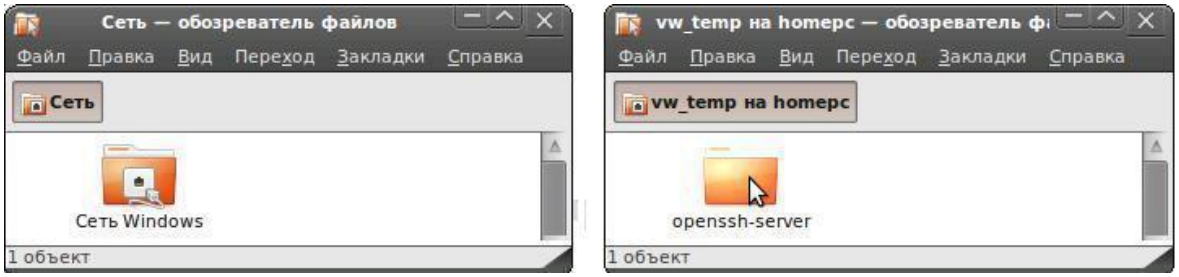

Рис. 7.1. Ресурсы локальной Windows сети лаборатории.

- На общедоступной папке openssh-server нажать правую кнопку мышки и выбрать «Копировать в ... » -> «Домашняя папка».
- Открыть терминал, перейти в каталог openssh-server вашей домашней папки и выполнить команду:

```
sudo sh setup-ssh
```

Примерный протокол действий, для пользователя serp и домашней папки с таким же именем, может иметь вид:

```
serp@vmUbuntu10:~$ cd openssh-server
serp@vmUbuntu10:~/openssh-server$ ls
  openssh-client_1%3a5.3p1-3ubuntu6_i386.deb
  openssh-server_1%3a5.3p1-3ubuntu6_i386.deb
```

```
  setup-ssh
serp@vmUbuntu10:~/openssh-server$ sudo sh setup-ssh
  . . .
  <- Serp -> End Install
```

- Удалите папку openssh-server из домашней папки, отключите сетевые ресурсы, которые могли появиться на рабочем столе.
- Очистите корзину. Не забывайте об этом и в дальнейшем. Помните, что вы работаете с виртуальной машиной, и она ест ресурсы основной.

Если вы внимательно следили за протоколом установки, то могли видеть, что в пакет SSH-сервера включены две программы SSH-клиент и SSH-сервер. То есть удаленно подключаться можно не только к вашей Ubuntu-машине, но и с нее можно подключаться к другим. При условии, что к ним возможен доступ по протоколу SSH. Конечно, если у вас на это будут права.

7.2. Тестирование и настройка SSH-сервера

Итак, сервер установлен и запущен. Протестируем его работу. Так как доступны и его клиентские возможности, то попробуем клиентом подключиться к серверу. То есть выполнить клиентский запрос к нашей же машине, которая выступит в качестве удаленного узла. Для этого в консоли надо набрать:

```
ssh localhost
```

Если сервер настроен правильно, то появится приглашение на ввод пароля. Вводим пароль на доступ к удаленной машине. Так как машина наша, то вводим, естественно, свой пароль. Далее можно вводить любые консольные команды, доступные на удаленном узле. Для завершения SSH-соединение используется команда:

```
exit
```

По умолчанию протокол SSH работает на 22 порту, хотя номер порта при желании можно и изменить.

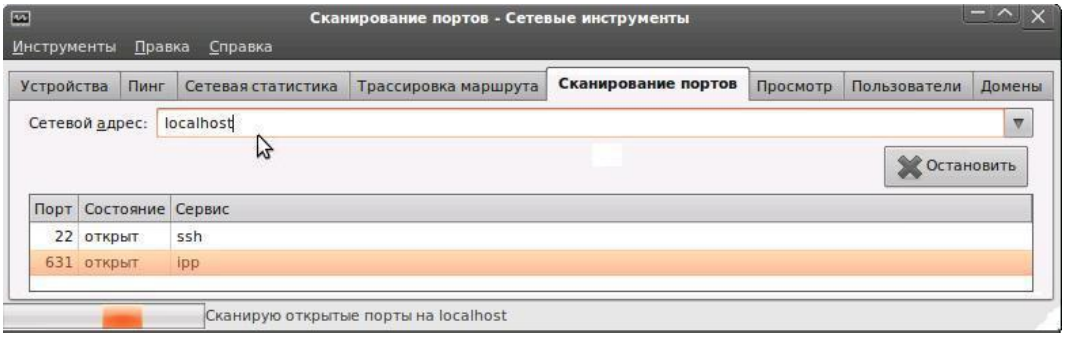

Рис. 7.2. Тестирование открытых портов на компьютере localhost.

Удостовериться в том, что этот порт доступен извне, а не закрыт, можно с использованием сетевых утилит. Для этого выберите: Система -> Администрирование -> Сетевые инструменты — и на вкладке «Сканирование портов» задайте имя или IP-адрес вашей Ubuntu-машины (рис. 7.2).

Для запуска SSH-сервера используют команду:

```
sudo /etc/init.d/ssh start
```

А для останова — ту же команду, но с параметром stop:

```
sudo /etc/init.d/ssh stop
```

Теперь о конфигурировании SSH-сервера. Если вы используете OpenSSH, то все настройки SSH-сервера хранятся в одном-единственном файле — /etc/ssh/sshd_config, а настройки программы-клиента — в файле /etc/ssh/ssh_config.

Настройки программы-клиента обычно задавать не нужно, поскольку они приемлемы по умолчанию. На вы загляните в файл /etc/ssh/ssh_config и разберитесь в его формате и опциях.

Значительно более нам интересен файл sshd_config, содержащий конфигурацию SSH-сервера. В приложении 1 к данному разделу приведен пример файла конфигурации SSH-сервера. Чтобы понять назначение директив, внимательно читайте комментарии, приведенные в этом примере.

В принципе сервер уже работает, но дополнительная настройка на тему безопасности еще никому не помешала. Что же, по нашему мнению, в файле /etc/ssh/sshd_config самое интересное?

Port 22	– Задание номера порта, на котором работает SSH-сервер. Рекомендуется изменить.
PermitRootLogin no	– Запрет подключения к SSH-серверу, используя логин Суперпользователь.
PermitEmptyPasswords no	– Запрет подключения пользователей, у которых пустые пароли. Очень рекомендуется, даже если вы единственный пользователь в системе.
AllowUsers ...	– Разрешение подключения только указанных пользователей. Логины пользователей разделяются пробелом. Рекомендуется при условии, что вы не единственный пользователь системы.

У вас может быть свое мнение по политике безопасности доступа к SSH-серверу, но опыт дело наживное. Самое главное, что после сохранения измененного файла конфигурации необходимо перезапустить SSH-сервер:

```
sudo /etc/init.d/ssh restart
```

7.3. Удаленное подключение к SSH-серверам

Чем подключаться к SSH-серверам? Вопрос не праздный, так как это зависит от среды, из которой осуществляется подключение. Если речь идет о Linux-системе, к которой относится и Ubuntu, то при установке сервера SSH-клиент установился автоматически. Работать с SSH-клиентом очень просто. Для подключения к удаленному компьютеру надо ввести команду:

```
ssh [опции] <логин>@<адрес_удаленного_компьютера>
```

В качестве адреса можно указать как IP-адрес, так и доменное имя компьютера. Наиболее часто используются следующие опции:

- c — Задает список шифров, в порядке предпочтения через запятые. Можно указать blowf ish, twof ish, arcfour, cast, des и 3des.
- f — Переводит ssh в фоновый режим после аутентификации пользователя.
- p — Определяет порт SSH-сервера (по умолчанию 22).
- q — Тихий режим — отображаются только сообщения о фатальных ошибках.
- i — Указывается пользователь, от имени которого нужно зарегистрироваться на удаленном компьютере.

Если подключение идет из Ububtu-машины, то можно ввести команду, например:

```
ssh serp@vmUbuntu
```

где serp — имя пользователя на удаленной системе, а vmUbuntu — имя узла, вместо которого можно указать его IP-адрес.

Если SSH-сервер находится на порту, отличном от стандартного, например 1010, то команду на доступ следует вводить, используя дополнительные опции:

```
ssh -p 1010 user@hostname
```

В том случае, если удаленное администрирование Ubuntu-машиной идет из среды Windows, то для этого случая опытные люди рекомендуют использовать программу PuTTY, которая легко находится по своему имени в поисковиках.

Существует множество и других программ, но мы остановимся на PuTTY, как достаточно простой и удобной в работе. И, что немаловажно в процессе обучения, абсолютно бесплатной.

7.3.1. Утилита PuTTY - клиент удаленного доступа

PuTTY — это клиент для различных протоколов удаленного доступа, включая SSH, Telnet, rlogin. Может работать и через последовательный

порт. Позволяет подключиться и управлять удаленным узлом, например сервером. В PuTTY реализована только клиентская сторона соединения — сторона отображения, а сама работа выполняется на стороне сервера.

Изначально разрабатывался для Microsoft Windows, позднее портирован на Unix. Сторонними разработчиками выпущены неофициальные версии на другие платформы, такие как мобильные телефоны под управлением Symbian OS и коммуникаторы с Windows Mobile. Программа выпускается под лицензией MIT.

К основным возможностям программы следует отнести:

- Сохранения списка и параметров подключений для повторного использования.
- Работа с ключами и версиями протокола SSH.
- Клиенты SCP и SFTP (соответственно программы pscp и psftp).
- Возможность перенаправления портов через SSH, включая передачу X11.
- Полная эмуляция терминалов xterm, VT-102, ECMA-48.
- Поддержка IPv6.
- Поддержка аутентификации с открытым ключом, в том числе и без ввода пароля.
- Поддержка работы через последовательный порт (начиная с версии 0.59).
- Возможность работы через прокси-сервер.

Естественно, что в рамках данного раздела мы не будем рассматривать все возможности и особенности данной утилиты. Тем более, что и такой цели перед нами не стоит.

Наша задача состоит в том, чтобы научиться из среды Windows удаленно администрировать Ubuntu-машины, на которой установлен SSH-сервер. Поэтому, на время забываем про Ubuntu и возвращаемся в Windows, чтобы познакомиться с утилитой PuTTY.

7.3.2. Как пользоваться утилитой PuTTY

Еще раз повторимся, что PuTTY — это популярный SSH-клиент для безопасного подключения к удаленному компьютеру. Утилита PuTTY ведет логи, позволяет настраивать шрифты, цвета и разрешение консоли, допускает сохранение в своей памяти ключей авторизации, поддерживает работу через прокси-сервер и является бесплатной в распространении.

Для того чтобы начать работу с PuTTY скачайте ее с официального сайта (http://www.chiark.greenend.org.uk/~sgtatham/putty/). Документацию по программе, правда, на английском языке, можно найти по адресу http://www.putty.nl/docs.html, ну а элементарные сведения изложены ниже.

Вернемся в Windows, где у вас уже есть скачанная утилита. Работает PuTTY без инсталляции, что конечно радует. Ведь для начала работы с ней

достаточно запустить файл PuTTY.exe, и перед вами появится окно, представленное на рис. 7.3.

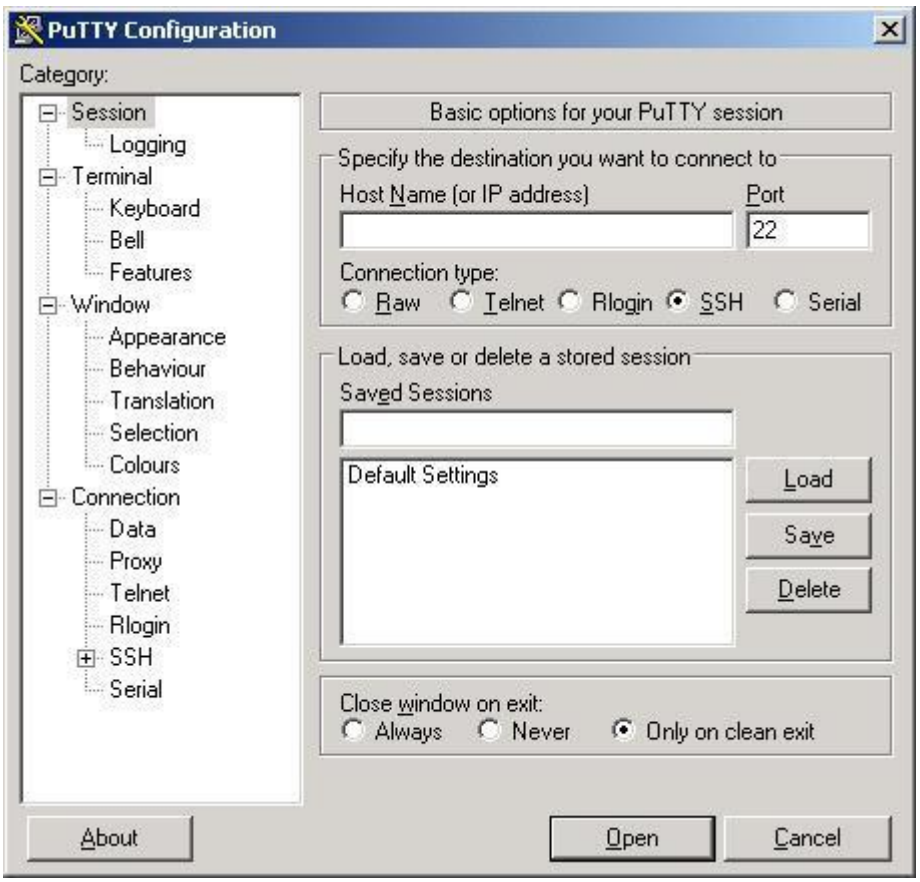

Рис. 7.3. Окно конфигурации утилиты PuTTY.

Последовательность действий по конфигурированию PuTTY для подключения к удаленному SSH-серверу будет следующей:

- В поле «Host Name (or IP address)» вводите имя или IP-адрес интересующего вас удаленного хоста, с которым вы собираетесь соединиться по SSH-протоколу.
- Для нашего примера следует ввести 192.168.1.10 или 192.168.2.10, в зависимости от текущей конфигурации вашей Ubuntu-машины.
- Порт оставляете по умолчанию 22 или вводите тот, на который настроен интересующий вас SSH-сервер.
- В меню слева активируете опцию SSH. Перед вами появится окно, как на рис. 7.4. Выберите протокол SSH версии «2 only».
- В меню слева выбираете опцию «Translation» и в выпадающем списке с кодировкой устанавливаете UTF-8. Вы собираетесь подключаться к Linex-системе, и для правильного отображения кириллицы надо использовать соответствующую кодировку.
- Возвращается во вкладку «Session» и в поле «Saved Sessions» вводим имя сессии (коннекта). Например, vmUbuntu_ssh.

- Для сохранения параметров сессии нажмите Save. Это позволит в дальнейшем из сохраненных сессий загружать нужную вам сессию (кнопка «Load»), не вводя заново требуемые параметры.

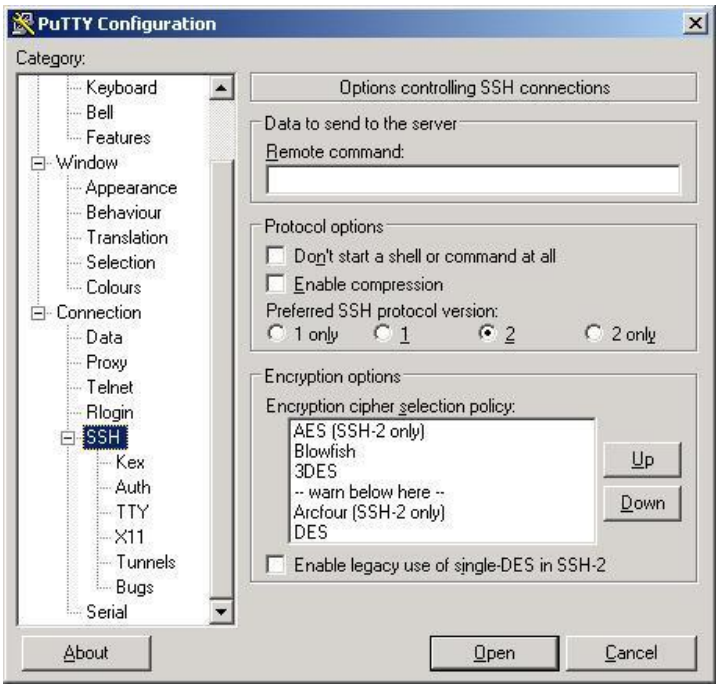

Рис. 7.4. Выбор версии протокола SSH для соединения.

- Чтобы соединиться с сервером, нажмите Open. Появится окно, аналогичное рис. 7.5, только абсолютно пустое, с единственной строкой сверху.

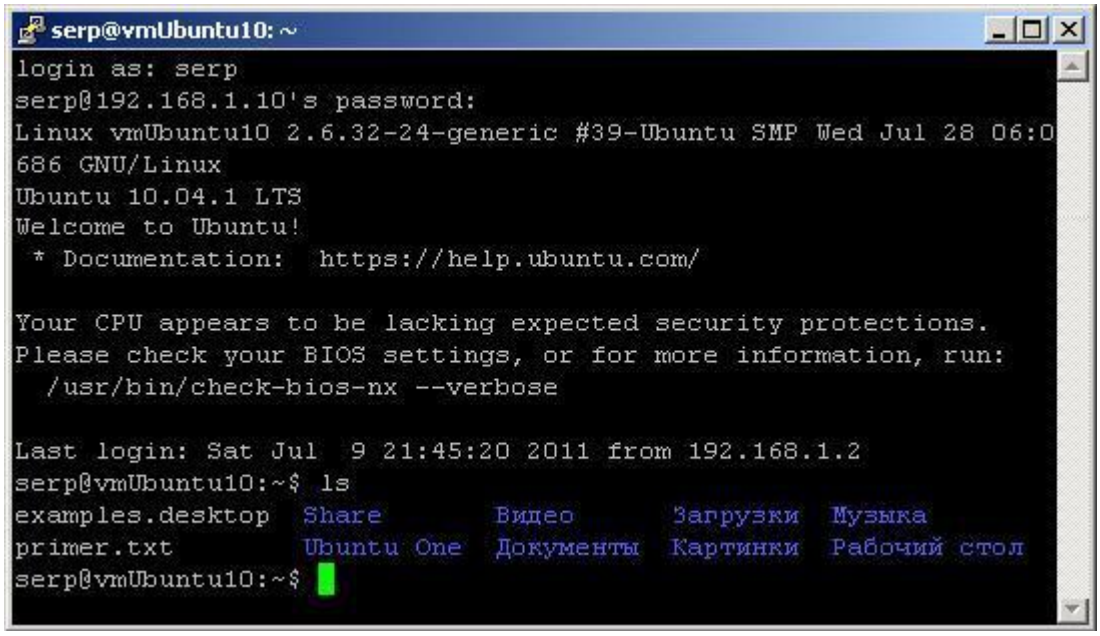

Рис. 7.5. Удаленная терминальная консоль узла 192.168.1.10.

- В строке «login as» введите логин для доступа по SSH и жмите Enter.

- После чего появится строка, сообщающая, что вы прошли на сервере аутентификацию как пользователь с веденным логином и требуется ввести пароль доступа. Не пугайтесь, что во время ввода пароля на экране ничего не отображается (ни звездочек, ничего подобного). Введите пароль и нажмите Enter.
- Если логин и пароль введены правильно, то выполнится подключение к серверу, и Вы попадете в командную строку Ubuntu Linux.
- Далее можно использовать любые консольные команды, которые будут выполняться на удаленном сервере.

Точно так же, как и на локальном терминале, вы можете использовать команду man <имя команды>, которая выдаст вам подробную информацию по любой команде. Дело за малым — знать администрирование и управление в Linux, а локально или удаленно, нам теперь безразлично. Для практики я рекомендую вам выполнить ряд упражнений:

Упражнения.

- В домашнем каталоге Ubuntu-машины создать текстовый файл, который затем из удаленной консоли Windows-машины отредактировать и результат посмотреть на Ubuntu-машине.
- Удаленно с Windows-машины создать файл на Ubuntu-машине, да еще с определенными правами доступа.
- И, наконец, так сказать на закуску, удаленно изменить порт SSH-сервера и повторно подключиться к нему, но уже с новыми параметрами. Если в процессе работы с сервером будете несколько раз менять параметры соединения, то будет больше вероятность того, что «враг» не пройдет.
- В качестве бонусной программы попробуйте в командной строке выполнить команду mc. Что это за команда и как заставить ее работать оставляем на вашей совести.

На этом знакомство с основами работы в удаленном терминальном доступе мы заканчиваем. Но технологии развиваются, каналы связи становятся все более скоростными, и графический интерфейс преследует нас во всех сетевых устройствах, от мощных компьютеров до iPad'ов и iFon'ов.

8. УДАЛЕННЫЙ РАБОЧИЙ СТОЛ В UBUNTU

Вы уже знакомы с опцией «Удаленный рабочий стол» по работе в Windows. Более того на одном из виртуальных или сетевых узлов у вас должна быть установлена Windows XP, на которой разрешен доступ на дистанционное управление рабочим столом. Если вы забыли, как это делается, то можете воспользоваться краткой справкой, приведенной в Приложении к этой главе.

До начала знакомства с этим разделом следует проверить, что реально существует доступ с основного Windows-компьютера к удаленному рабочему столу виртуальной машины с Windows XP для конкретного логина и пароля пользователя. Это нужно для того, чтобы не было проблем в процессе дальнейшего знакомства с использованием удаленного рабочего стола в среде Ubuntu-Windows систем.

Итак, в нашей виртуальной сети используются как Windows-машины, так и Linux-машины? Можно ли удаленно управлять Windows из Ubuntu или Ubuntu из Windows? Несомненно, да. Подобно тому, как используется подключение к удаленному рабочему столу между платформами Microsoft. Вы можете щелкать мышью на рабочем столе и запускать приложения точно так же, как если бы вы сидели прямо перед компьютером. Мы обсудим несколько различных возможностей, которые вы можете получить, используя подключение к удаленному рабочему столу.

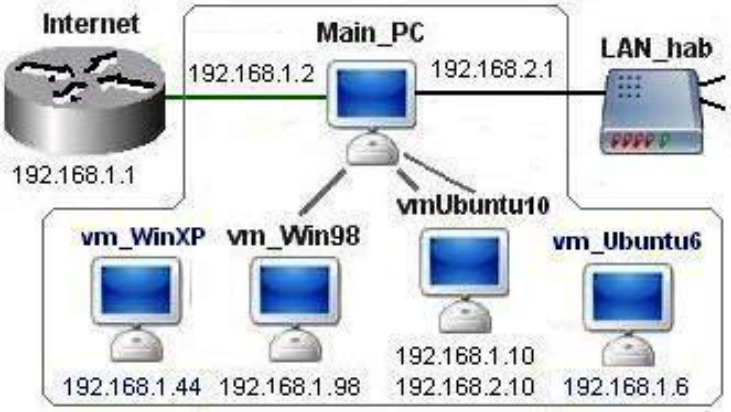

Рис. 8.1. Структура тестовой системы.

Структура тестовой системы (рис. 8.1) будет включать в себя основной компьютер, на котором установлены 4 виртуальных машины с Windows 98, Windows XP, Ubuntu версий 6.10 — Dapper Drake и 10.04 — Lucid Lynx. Если на вашем компьютере недостаточно рабочей памяти, то не

обязательно загружать для наших опытов все машины одновременно, часть из них может находиться в режиме сохранения состояния в Virtual PC Console.

Теперь необходимо разрешить удаленное подключение для вашей учетной записи на машине с Windows. Для этого откройте «Панель управления -> Система -> Удаленные сеансы» и поставьте галочку «Разрешить удаленный доступ к этому компьютеру». В списке разрешенных учетных записей для подключения добавьте ваш логин.

8.1. Выбор протокола удаленного рабочего стола

Приложения удаленного рабочего стола используются для протокола RDP (Remote Desktop Protocol — протокол удаленного рабочего стола), либо для протокола VNC (Virtual Network Computing — протокол виртуальной вычислительной сети).

Для удаленного подключения оба узла как сервер, так и клиент должны поддерживать один и тот же протокол. Проблема в том, что не все ОС используют по умолчанию одинаковые протоколы. При этом некоторые дистрибутивы Linux и некоторые редакции Windows не содержат в себе ни серверного, ни клиентского приложения удаленного рабочего стола, либо не содержат приложение удаленного рабочего стола вообще.

Вашей первой задачей должно стать определение протокола, который уже поддерживается на ваших компьютерах. Дополнительно к исследованию вашей ОС, поиску документации, вы должны понимать, что есть что и где. Это позволит вам обоснованно применять протокол для использования на всех ваших компьютерах.

Таблица 8.1

Используемые протоколы на различных платформах

Платформа	Протокол	Клиент	Сервер
Windows Home Edition	RDP	X	---
Windows XP Professional	RDP	X	X
Windows Vista Home Basic & Premium	RDP	X	---
Windows Vista Business & Ultimate	RDP	X	X
Большинство Linux систем	VNC	X	X

Обратим внимание, что на сегодняшний день удаленный рабочий стол VNC обычно медленнее, чем RDP соединения. Тем не менее VNC обычно легче реализовать на различных платформах. То есть вам надо точно знать, что существует два популярных метода получения доступа к рабочему столу другого пользователя.

Первый, и пожалуй наиболее популярный — это с помощью RDP, протокола прикладного уровня, разработанного компанией Microsoft.

8.2. Протокол RDP

Как уже было сказано, RDP — это протокол (http://www.rdesktop.org), разработанный корпорацией Microsoft для обеспечения доступа, контроля и управления ресурсами другого компьютера под управлением систем Windows. Он имеет ряд особенностей, преимуществ и оказался весьма успешным.

Особенности RDP:

- Поддержка 32-битного цвета.
- 128-битовое шифрование.
- Переадресация звука и видео (Web-камера, микрофон).
- Возможность обмена данными через буфер обмена.
- Позволяет использовать локальные ресурсы удаленного ПК (принтеры, сканеры, камеры).

Этот протокол обеспечивает полноценный менеджмент удаленного ПК и работает по стандартному принципу клиент-сервер.

- Сервер. Им является удаленный компьютер, с которым вы устанавливаете соединение. Это может быть либо стационарный компьютер в офисе, либо портативный компьютер, с которым вы соединяетесь удаленно, когда его владелец находится в поездке.
- Клиент. Это тот компьютер, с которого вы устанавливаете соединение с сервером. Например, переносной компьютер для работы во время путешествия, телефон, смартфон, нетбук и прочие «умные» устройства.

8.2.1. Практическое применение RDP в Ubuntu

Предположим, что вы администрируете несколько Web-серверов на Linux, а на работе вам приходится администрировать сеть компьютеров и серверов, на многих из которых установлена операционная система Windows. Чтобы успешно справиться с обеими задачами, вам хватит одного ноутбука с Ubuntu.

Для этих задач вам достаточно использовать консольную утилиту rdesktop, которая по умолчанию содержится в стандартной поставке любой версии Ubuntu Linux. Работает она предельно просто, если вам известно имя удаленного сервера, к которому вы должны подключиться (его IP-адрес или доменное имя).

Чтобы подсоединится к удаленному рабочему столу Windows, наберите в консоли:

```
rdesktop [опции] <адрес_удаленного_компьютера>
```

Получить все доступные опции можно, запустив эту утилиту без параметров. Наиболее часто используемыми являются:

—u – Учётная запись для аутентификации.
—d – Домен.
—p – Пароль, если опция не используется, то в этом случае rdesktop запросит пароль при запуске.
—k – Эмулируемая раскладка клавиатуры.
—g – Разрешение экрана, можно указать в процентах от всего экрана.
—f – Полноэкранный режим (можно переключиться с помощью комбинации клавиш Ctrl-Alt-Enter).
—s – Начальная оболочка пользователя (вместо Explorer).
—c – Начальный рабочий каталог пользователя.
—0 – Глубина цвета: 8, 16 или 24 бит. Основной для Ubuntu 24-битный цвет Windows XP/2003 не поддерживает.
—z – Активация сжатия передаваемых данных, актуально для медленных соединений.

Используя опции и флаги для утилиты rdesktop, вы можете модифицировать эту простую команду:

- Если вам необходимо указать определенную раскладку клавиатуры, то команда будет иметь вид

```
rdesktop -k en_us 192.168.1.44
```

Следует отметить, что это важно, так как входя в разные виртуальные машины под Windows, бывают ситуацию, когда при соединении, особенно с русифицированными версиями, rdesktop устанавливала кириллицу, и невозможно было ввести пароль латинскими буквами. Переключение клавиатуры в момент ввода пароля не работало.

- Указать размер открываемого окна в пикселях или в процентах можно командой:

```
rdesktop -g 60% -k en_us 192.168.1.44
```

- При этом 100 % – это не полный экран, а полное окно, в котором вы работаете. Если нужен полный экран, то следует использовать следующий формат команды:

```
rdesktop -f -k en_us 192.168.1.44
```

- Указать логин и пароль для удаленного компьютера, чтобы его не вводить при входе можно в команде.

```
rdesktop -u admin -p psw -k en_us 192.168.1.44
```

Естественно, что данный метод набора не безопасен, но в случае локальной работы, он более удобен и быстр, так как отсутствует необходимость постоянно вводить логины и пароли.

Если в структуре нашей тестовой системы активировать vmUbuntu10 и vm_WinXP, при условии что на последней установлен доступ к ее рабочему столу, а после этого в терминале vmUbuntu10 ввести команду:

```
rdesktop -g 60% -u serp -p serp 192.168.1.44
```

на удаленный доступ к рабочему столу vm-WinXP, то вид экрана основного компьютера будет иметь вид, приведенный на рис. 8.2.

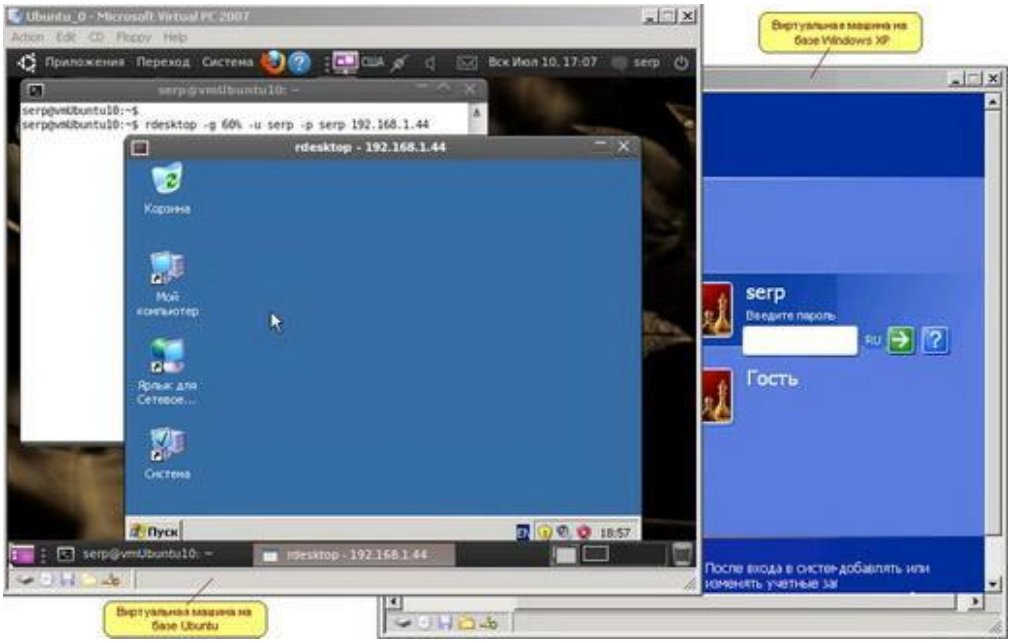

Рис. 8.2. Окна виртуальных машин основного ПК при доступе с Ubuntu-машины на Windows-машину.

При этом vm-WinXP — заблокируется, а в окне «rdesktop - 192.168.1.44» на vmUbuntu10, будут доступны любые действия с Windows XP. Естественно, в рамках тех прав, которыми наделен пользователь ее удаленного рабочего стола.

Но если ваша задача — администрирование компьютера с Windows XP, то, настраивая удаленное подключение к нему, вы себя, как пользователя ее рабочего стола, наделяете административными правами.

8.2.2. Графические клиенты RDP в Ubuntu

Утилита rdesktop является приложением командной строки. Это предполагает, что вы используете ее в консоли. Она имеет множество конструкций, что удобно для профессионального администратора. Более того, эту команду можно использовать в скриптовых файлах, автоматизирующих те или иные операции по управлению компьютером или сетью.

Но жизнь меняется, и менее профессиональным пользователям больше нравятся красивые окошки и возможность тыкать мышкой. Тем, кому из

вас это более по душе, Ubuntu-содружество предоставляет возможность использовать графические утилиты, поддерживающие доступ по протоколу RDP. К разряду наиболее востребованных на сегодня таких графических клиентов RDP можно отнести Gnome-RDP или Remmina. Все они поддерживают сессии, а также несколько открытых рабочих столов, что очень удобно.

- Установить Gnome-RDP очень просто:

```
sudo aptitude install gnome-rdp
```

Gnome-RDP поддерживает такие протоколы как: RDP, VNC, SSH. С помощью Gnome-RDP вы сможете настроить для RDP разрешение экрана, количество цветов, раскладку клавиатуры, вывод звука.

- Установить Remmina не сложнее:

```
sudo aptitude install remmina
```

Структура Remmina совершеннее , имеет больше опций, поддерживает протоколы: SSH, RDP, VNC, SFTP и обладает кучей опций и настроек.

Знакомство с этими или еще более продвинутыми программными продуктами дело вашей совести. Наша задача получить начальные сведения, а для этого мы познакомимся с входящим в состав стандартной поставки Ubuntu клиентом терминального сервера.

Выберите на компьютере vmUbuntu10 в основном меню Приложения -> Интернет -> Клиент терминального сервера. На экране появится окно, аналогичное тому, что приведено на рис. 8.3.

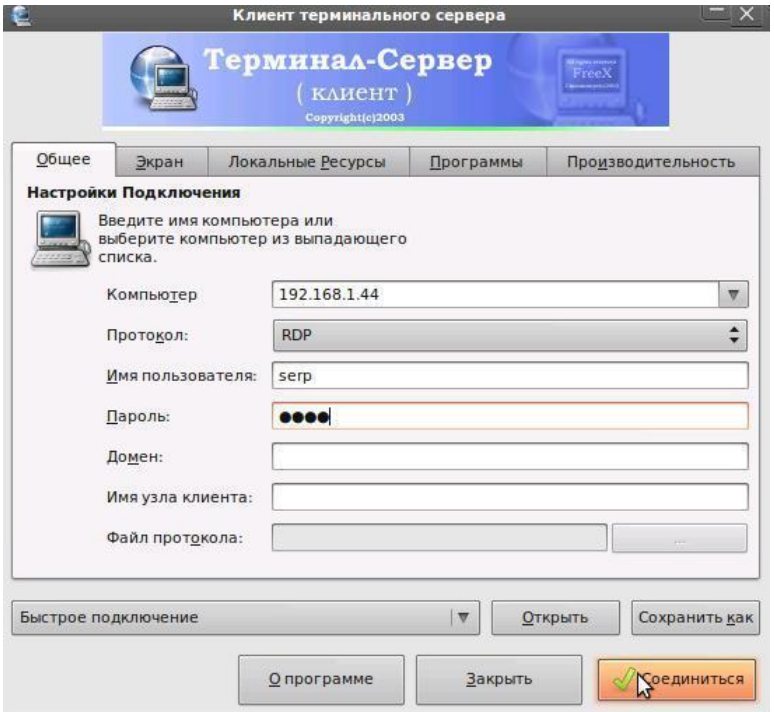

Рис. 8.3. Стартовое окно «Клиент терминального сервера».

В этом окне надо установить тип протокола, указать имя или адрес компьютера, удаленным рабочим столом которого надо воспользоваться. Логин и пароль устанавливаются по тем же соображениям, о которых говорилось при описании rdesktop.

Но не торопитесь нажимать кнопку «Соединиться», так надо перейти на вкладку «Экран», где следует установить размер отображения удаленного экрана и указать используемую глубину цвета. Но и этого мало. Перейдите на вкладку локальные ресурсы (рис. 8.4).

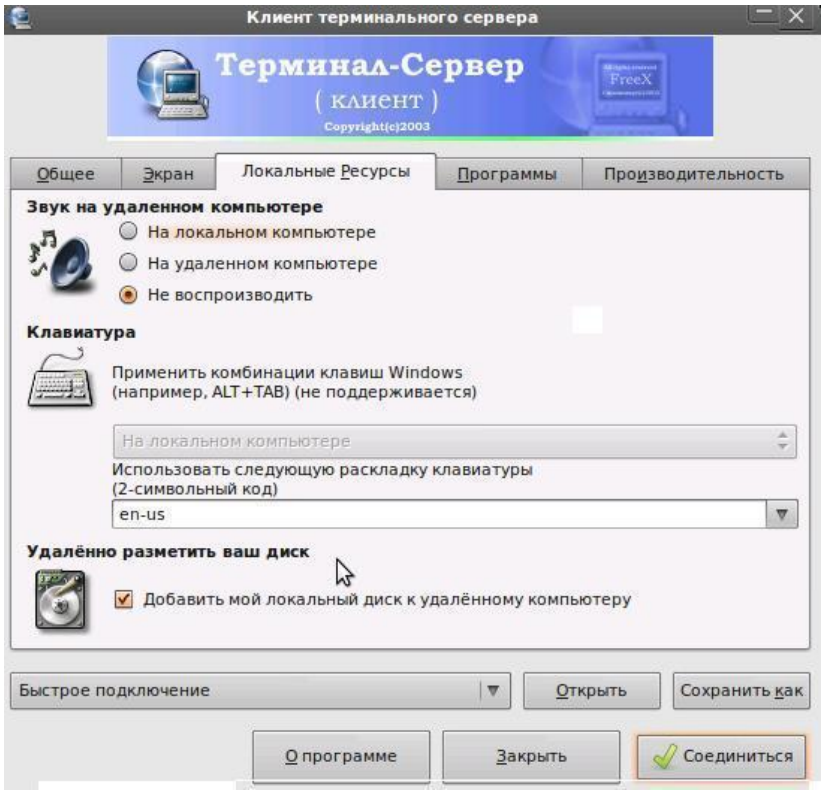

Рис. 8.4. Вклад «Локальные ресурсы» клиента терминального сервера.

Здесь следует указать стартовую раскладку клавиатуры в зависимости от того, в каком режиме вам необходимо вводить логин и пароль пользователя.

Если поставить птичку в поле «Добавить мой локальный диск к ... », то у вас появится возможность при работе на удаленном компьютере использовать ресурсы вашего локального компьютера.

Если в клиенте терминального сервера vmUbuntu10 был указан адрес vm-WinXP, то в vmUbuntu10 откроется окно, повторяющее рабочий стол указанного пользователя в vm-WinXP. На этом рабочем столе можно открыть папки как удаленного компьютера, так и локальные. Более того, можно стандартными методами переносить или копировать файлы между папками удаленного и локального компьютеров.

На рис. 8.5 приведен удаленный рабочий стол виртуальной машины с Windows XP, который открыт в виртуальной Ubuntu-машине. Показано, что

на этом столе стандартными методами Windows открыто два окна: одно — с содержимым папки на диске C:\ удаленной Windiws-машины, второе — с содержимым домашнего каталога локальной Ubuntu-машины. Вид папок соответствует состоянию, после того как файл primer из домашнего директория был скопирован в папку удаленной машины.

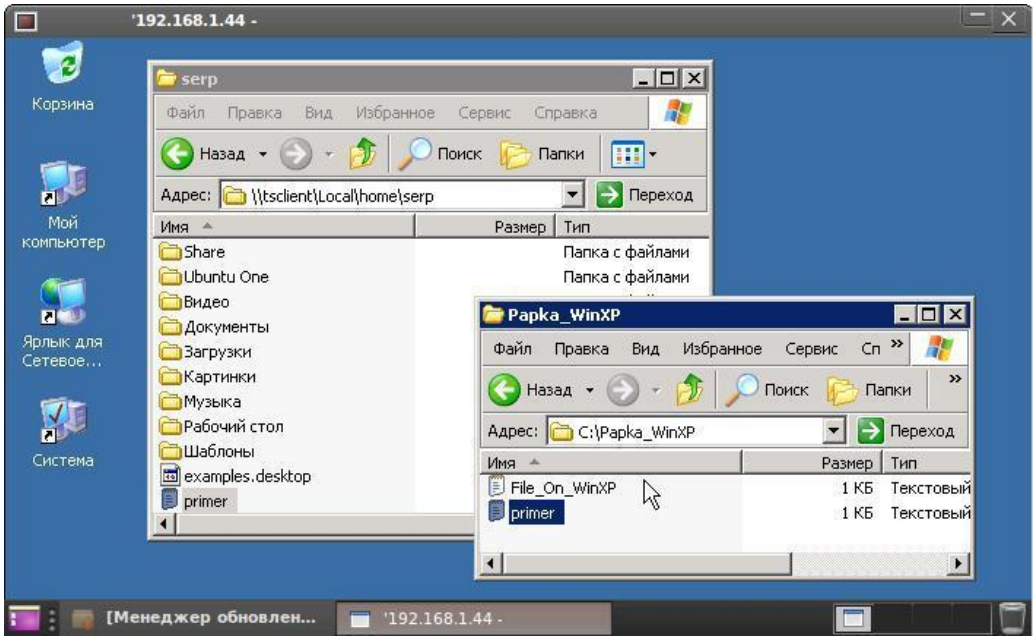

Рис. 8.5. Удаленный рабочий стол Windows XP на Ubuntu-машине.

8.2.3. Настройка удаленного рабочего стола в Ubuntu

Знакомые с Windows, могут предполагать, что и в Ubuntu где-то есть окно настроек удаленного рабочего стола. И, действительно, погуляв по меню, находим последовательность Система -> Параметры -> Удаленный рабочий стол. Откроется окно, а после того как установим птичку в первой строке, оно примет вид, как на рис. 8.6.

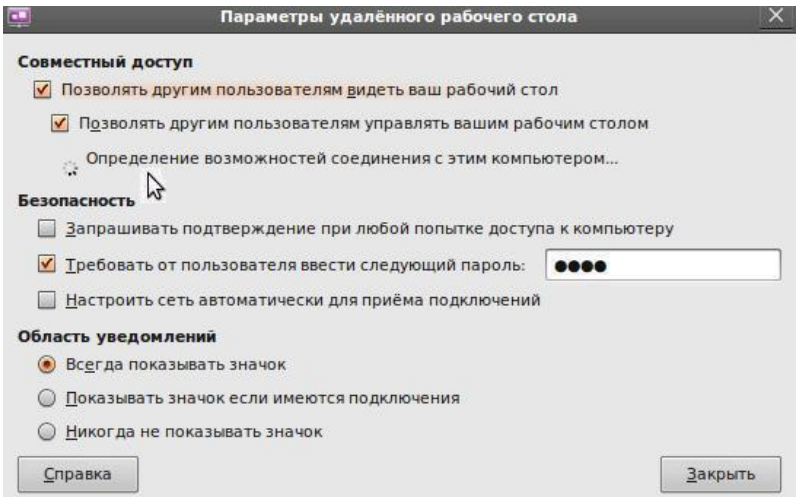

Рис. 8.6. Окно определения параметров подключения к рабочему столу Ubuntu.

Обратите внимание на анимированную иконку в начале третьей строки. Это операционная система определяет возможность работы этого компьютера в режиме удаленного рабочего стола. Если проверка пройдет штатно, то третья строка этого окна изменит свое содержание. В ней будет указан адрес, по которому можно обращаться извне к вашему рабочему столу (рис. 8.7).

Рис. 8.7. Окно параметры рабочего стола после проверки.

Приведенное на рис. 8.6 окно представляет собой интерфейс инструмента операционной системы, который позволяет обеспечивать нескольким пользователям доступ к сеансу работы в среде GNOME, а также настроить параметры такого доступа. Эти параметры непосредственно влияют на безопасность вашей системы.

Таблица 8.2

Параметры доступа к сеансу работы

Элемент диалогового окна	Описание назначения
Позволять другим пользователям видеть ваш рабочий стол	Удаленные пользователи могут ваш сеанс только просматривать. Все нажатия клавиш, щелчки мышью и события буфера обмена, порождаемые удаленным пользователем, будут игнорироваться
Запрашивать подтверждение	Для совместного использования вашего сеанса будет требоваться ваше разрешение, вы будете уведомлены при попытках других пользователей соединиться с вашим сеансом, можете выбирать подходящее время для соединения с вашим сеансом
Требовать ввести пароль:	Удаленные пользователи должны вводить пароль, если используется идентификация. Этот параметр повышает безопасность
Пароль	Введите пароль, который должен вводиться клиентом, желающим просмотреть или управлять вашим сеансом

Введя все параметры и сохранив их, мы уверены, что настроили удаленный доступ к рабочему столу vmUbuntu10. Но если теперь на

основном Windows-компьютере выполнить: Пуск -> Все программы -> Стандартные -> Связь -> Подключение к удаленному рабочему столу, — то нас ожидает глубокое фиаско с отказом в доступе.

Но это связь Windows с Ubuntu, а что будет при связи Ubuntu с Ubuntu? Если мы из vmUbuntu10, используя клиент терминального сервера, попытаемся связаться с vmUbuntu06, где также настроен удаленный рабочий стол, то получим результат как на рис. 8.8.

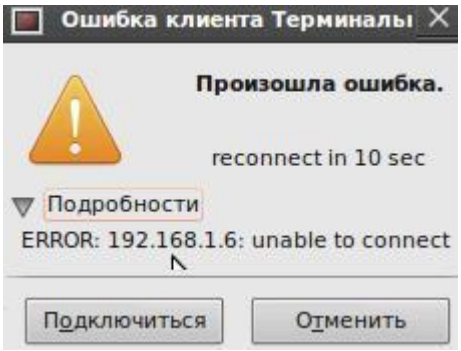

Рис. 8.8. Результат подключения клиента терминального сервера.

Вместе с тем, если мы из vmUbuntu10 попытаемся связаться vmUbuntu06, используя Приложение -> Интернет -> Просмотр удаленных рабочих столов, то получим положительный результат (рис. 8.9).

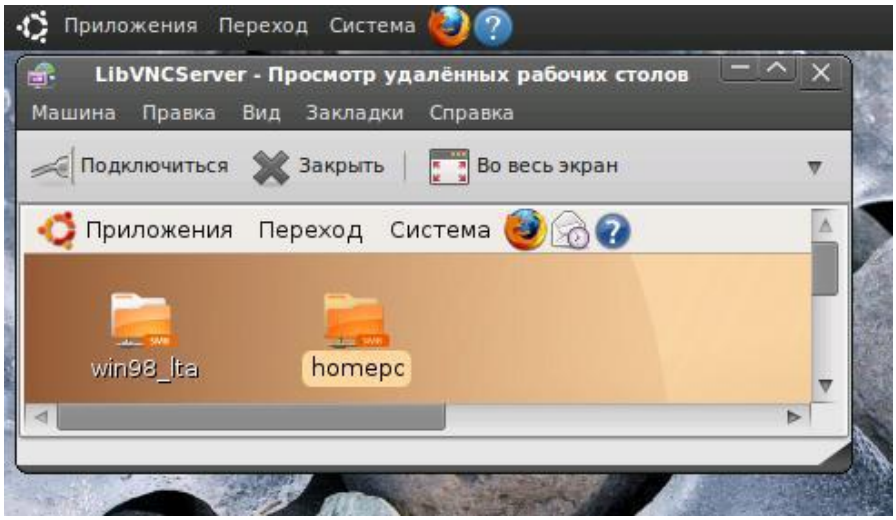

Рис. 8.9. Подключения vmUbuntu10 к vmUbuntu06 с помощью клиента удаленных рабочих столов.

Давайте разберемся, в чем причина этих явлений. Все дело в том, что серверы удаленных рабочих столов Ubuntu работают по протоколу VNC, а доступ к удаленным рабочим столам в Windows реализован на основе протокола RDP. О протоколе VNC речь пойдет ниже, а пока мы знакомимся с возможностями RDP. И тут возникает вопрос, а как обеспечить совместимость удаленных рабочих столов Windows-машин и Ubuntu-машин.

8.2.4. Совместимость удаленных рабочих столов Windows и Ubuntu по протоколу RDP

Удаленный рабочий стол в Ubuntu — это не более чем VNC. И, как отмечают многие пользователи, он является медленным и громоздким по сравнению с реальным Microsoft Windows Remote Desktop. Те из них, кто привык к скоростям Windows Remote Desktop, говорят, что VNC может просто свести с ума. Как же настроить удаленный рабочий стол Ubuntu, чтобы он использовал Windows Remote Desktop?

Вариантов может быть несколько. Но этот материал не научное эссе и мы не будем заниматься сравнительным анализом возможных подходов. Рассмотрим всего один из вариантов технологии объединения удаленных рабочих столов Windows- и Ubuntu-машин. Для этой цели будем:

- Для подключения с Ubuntu-машины на Windows-машину использовать rdesktop или клиента терминального сервера с протоколом RDP.
- Для подключения из Windows-машины к Ubuntu-машине использовать стандартные средства Windows по подключению к удаленному рабочему столу, установив для этого на Ubuntu-машине сервер RDP.

8.2.4.1. Установка RDP-сервера на Ubuntu-машину

Организовать RDP-сервер на Ubuntu-машине можно, например, с помощью пакета **xrdp**, который поддерживает протокол удаленного рабочего стола RDP для среды XWindows, то есть для класса Linux-машин. Именно его и следует установить на Ubuntu-машине. Рассмотрим разные варианты установок пакета **xrdp** на Ubuntu-машину.

Вариант 1. Если ваша Ubuntu-машина имеет доступ в Интернет, то для установки **xrdp** надо:

- В меню выбрать Приложения -> Центр приложений Ubuntu.
- В открывшемся окне в поле поиска ввести **xrdp** и нажать «Ввод». Ubuntu выполнит поиск пакета в репозиториях и выдаст окно с сообщением о результатах поиска (рис. 8.10).

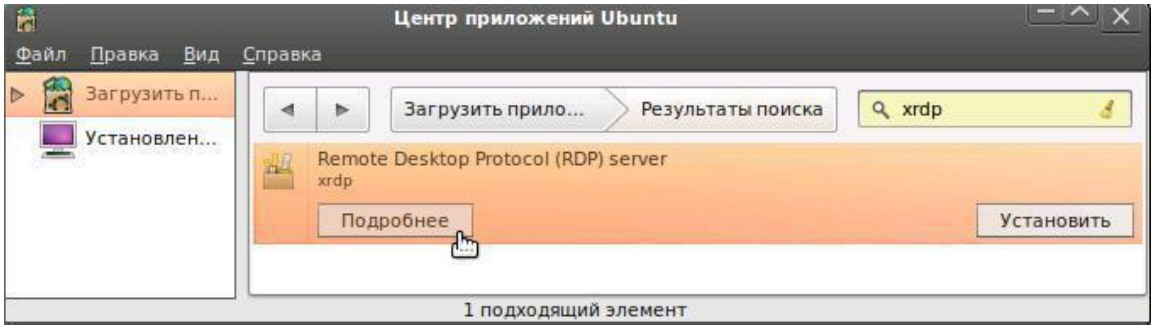

Рис. 8.10. Результат поиска пакета xrdp в репозиториях.

- Для ознакомления с назначением пакета и условиями поставки нажать кнопку «Подробнее». Будет выдано окно (рис. 8.11) с кратким сведением о пакете, лицензии и его версии.

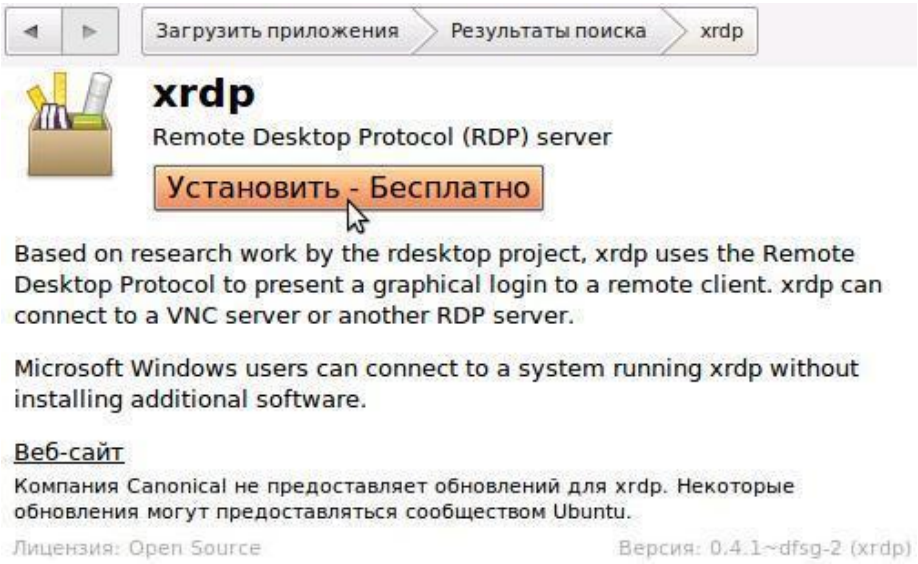

Рис. 8.11. Краткие сведения о пакете, его версия и лицензия.

- Обратите особое внимание на версию. Так, из рис. 8.11 видно, что для Ubuntu 10.04 в репозиториях зарегистрирована версия пакета 0.4.1~dfsg-2 (xrdp). Это нам пригодится, когда будем рассматривать случай установки пакета на Ubuntu-машине без доступа в Интернет или произойдет сбой при удаленной установке.
- Для установки пакета нажать кнопку «Установить — Бесплатно» и ждать окончания режима установки (рис. 8.12).

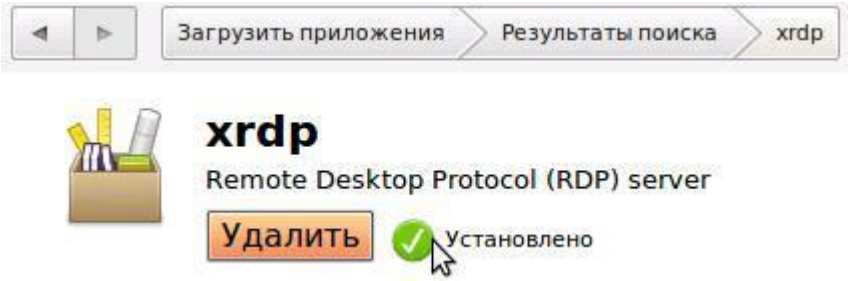

Рис. 8.12. Сообщение об окончании установки пакета xrdp.

Вариант 2. Если ваша Ubuntu-машина имеет доступ в Интернет, то достаточно войти в терминал и набрать команду:

```
sudo aptitude install gnome-rdp
```

Следует отметить, что такой подход доступен и в удаленном терминальном режиме, если мы воспользуемся протоколом SSH и утилитой PuTTY. Примерный вариант такой установки приведен в приложении 2 к данному разделу.

Привлекательным в этом подходе является то, что мы можем последовательно подключаться к удаленным узлам, куда у нас есть SSH-доступ, и там устанавливать нужное ПО. То есть, не вставая со своего любимого дивана, можем конфигурировать удаленные компьютеры.

Вариант 3. Если ваша Ubuntu-машина не имеет доступа в Интернет.

Следует найти в локальной сети или скачать у друзей файл xrdp_0.4.1~dfsg-2_i386.deb. Вспомните, чуть выше, был приведен разговор о версии пакета. В Интернете этот пакет можно найти на серверах: //launchpad.net, //opensource.telkomspeedy.com, //packages.ubuntu.com. Далее следует:

- Поместить этот файл в вашу домашнюю директорию на Ubuntu.
- Выделив этот файл, нажать правую кнопку мышки и в открывшемся меню выбрать «Установщик пакетов GDebi».

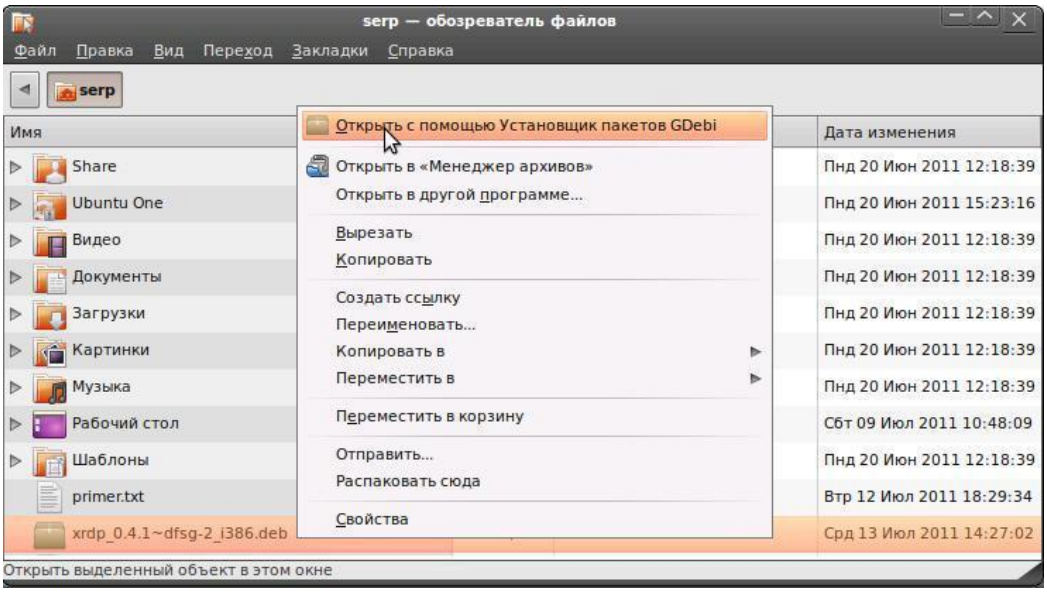

Рис. 8.13. Запуск установщика пакета xrdp.

Откроется окно «Установка пакета», об окончании которой система вас проинформирует (рис. 8.14).

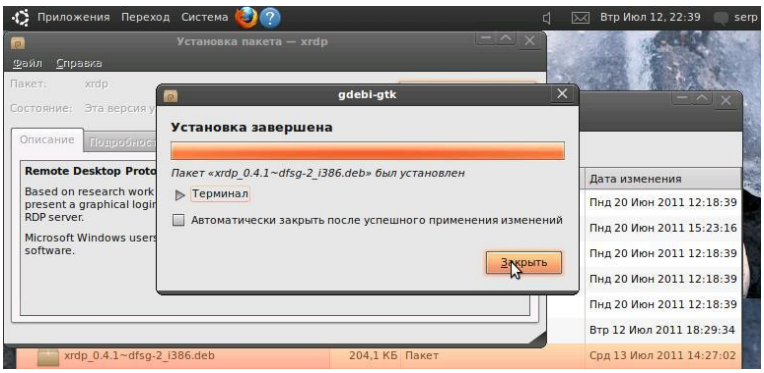

Рис. 8.14. Сообщение об окончании установки пакета xrdp.

Более того, кликнув мышкой на треугольник рядом с опцией «Терминал», можно получить протокол установки этого пакета. Обратите внимание, что он аналогичен тому, что был получен при вводе консольной команды. То есть все описанные выше окна — это не что иное, как графический интерфейс к консольной команде установки пакета.

После установки **xrdp** запустится автоматически. Если в вашей системе этого не произошло, запустите его вручную при помощи init-скрипта:

```
sudo /etc/init.d/xrdp start
```

Собственно, на этом установка сервера завершена. Отметим, что установить пакет можно и путем удаленного доступа с базового Windows-компьютера на vmUbuntu10, используя SSH и PuTTY. В приложении 8.2 приведен пример такой установки.

8.2.4.2. Доступ к RDP-серверу Ubuntu-машины

Итак, RDP-сервер на Ubuntu-машину установлен, можно возвратится к работе с удаленными рабочими столами. Теперь попробуем подключиться к RDP-серверу разными RDP-клиентами как из Windows, так и из Ubuntu.

Для чистоты эксперимента попробуем сделать это при помощи штатного mstsc.exe из поставки Microsoft Windows XP SP3. Для этого на основном компьютере с Windows XP следует:

- Выбрать Пуск -> Программы -> Стандартные -> Связь -> Подключение к удаленному рабочему столу (рис. 8.15).

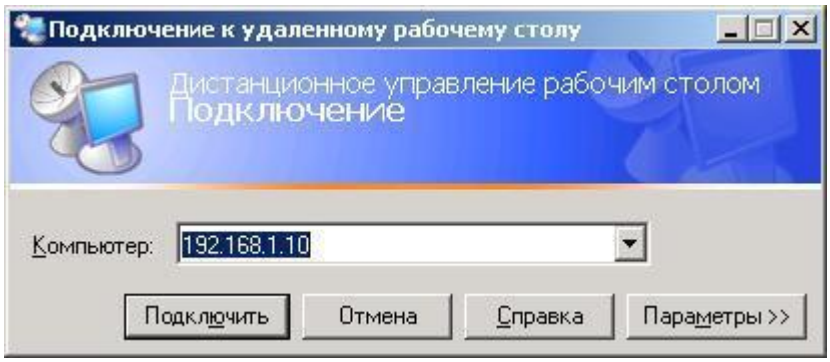

Рис. 8.15. Подключение из основного компьютера к рабочему столу vmUbuntu10.

- Ввести имя или адрес нашей виртуальной vmUbuntu10 (192.168.1.10), не забывая при этом, что есть еще вкладка «Параметры», которые можно установить и настроить.
- Нажать кнопку «Подключить».

В Windows-окне «Удаленный рабочий стол» основного компьютера появляется окно RDP-сервера vmUbuntu10 с запросом параметров доступа к ее удаленному рабочему столу. Введите имя пользователя, его пароль и нажмите «ОК» (рис. 8.16).

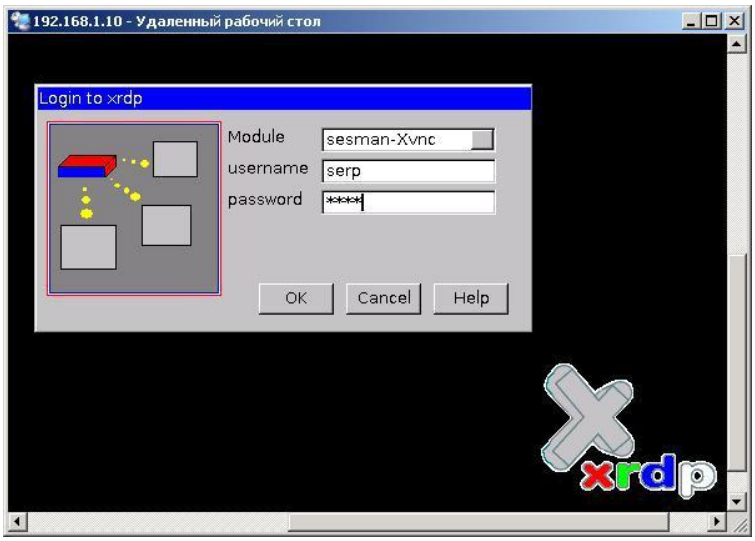

Рис. 8.16. Аутентификация доступа к рабочему столу vmUbuntu10.

Если аутентификации прошла успешно, то в Windows-окне основного компьютера откроется удаленный рабочий стол Ubuntu-машины, аналогично скриншоту на рис. 8.17.

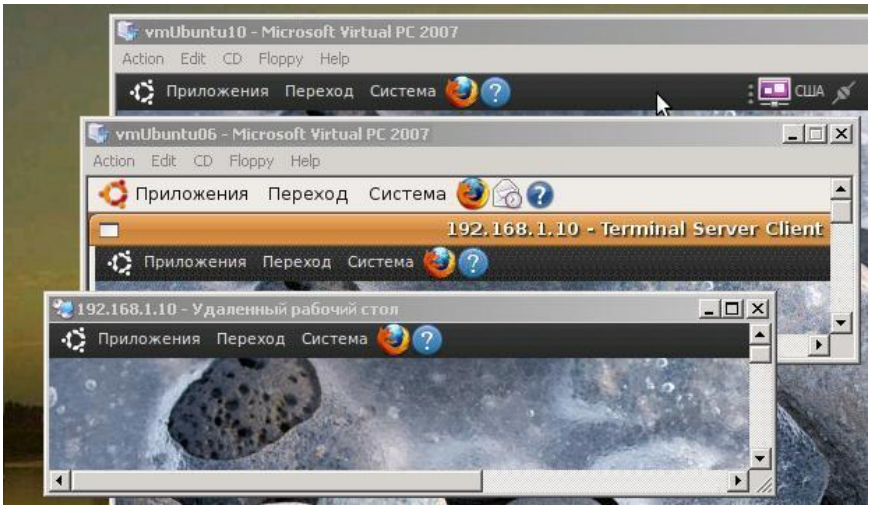

Рис. 8.17. Экран основного компьютера с подключенным к нему рабочим столом vmUbuntu10.

На рис. 8.17 приведен фрагмент экрана основного компьютера, на котором видно три окна:

- Верхнее — это виртуальная машина vmUbuntu10, на которой установлена Ubuntu 10.04.
- Среднее — это виртуальная машина vmUbuntu06, на которой установлена Ubuntu 6.04.
- Нижнее — это открытый из Windows XP основного компьютера, удаленный рабочий стол к vmUbuntu10.

Но с первым и третьим должно быть все ясно. О них чуть выше и шел достаточно подробный разговор. А зачем здесь третье окно и что автор этим хочет сказать? Попробуем разобраться и сделать какие-либо выводы.

Но для этого нам надо вернуться в раздел 8.2.3, где мы пытались организовать связь двух Ubuntu-машин, используя клиента терминального сервера. Потерпели фиаско, которое иллюстрировал рис. 8.8. То есть Ubuntu-машины не могли связаться по протоколу RDP. А что будет сейчас, когда мы на vmUbuntu10 установили RDP-сервер?

Давайте откроем виртуальную машину на базе Ubuntu 6.04 и в основном меню выберем: Приложения -> Интернет -> Клиент Терминального Сервера.

В открывшемся окне «Терминал-Сервер (клиент)», аналогичном рис. 8.3, 8.4, выберем для соединения протокол RDP, введем адрес нашей второй виртуальной машины vmUbuntu10 - 192.168.1.10 и нажмем кнопку «Соединиться». В этот раз, в отличие от рис. 8.8, окно vmUbuntu06 примет вид, аналогичный рис. 8.18.

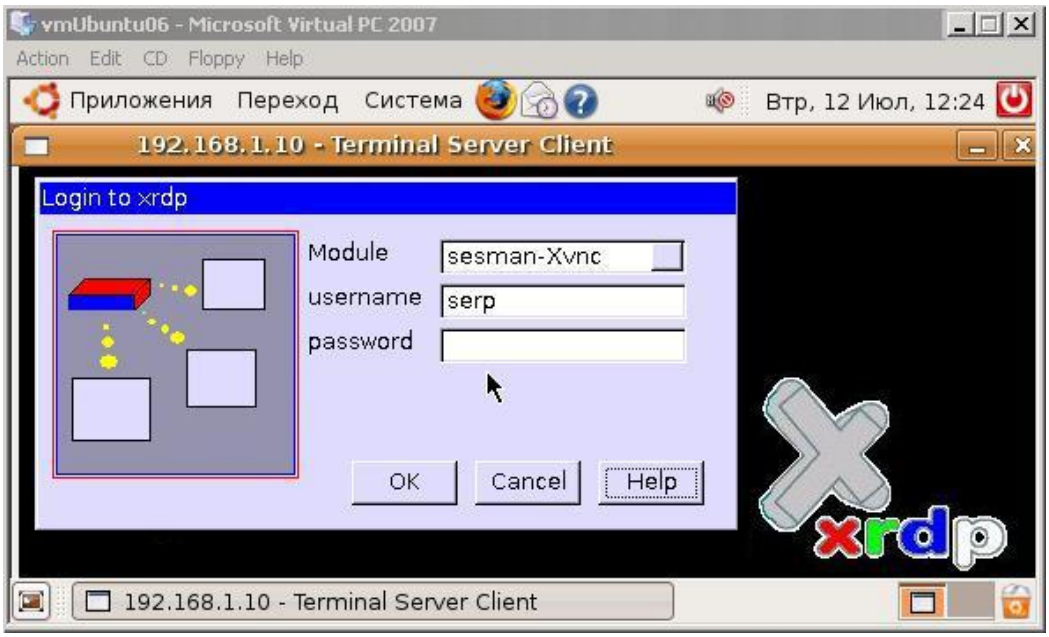

Рис. 8.18. Фрагмент экрана vmUbuntu06 с доступом на vmUbuntu10.

То есть Ubuntu-машины готовы установить связь по протоколу RDP. Клиент RDP машины vmUbuntu06, послав запрос на vmUbuntu10, был услышан ее RDP-сервером, который переслал обратно запрос с требованием выполнить аутентификации.

Если теперь введем имя и пароль пользователя, зарегистрированного на vmUbuntu10, то в окне виртуальной машины vmUbuntu06 нам будет доступен рабочий стол vmUbuntu10. Именно он и является вторым окном, приведенным на рис. 8.17.

Мы подходим к моменту, когда уже можно формулировать какие-либо выводы.

- Первый из них состоит в том, что мы можем объединить между собой работу удаленных рабочих столов по протоколу RDP как Ubuntu-Windows систем, так и Ubuntu-Ubuntu систем.

- Второй вывод заключается в том, что в Windows удаленный компьютер блокируется при удаленном входе на его рабочий стол. А в Ubuntu можно продолжать работать как на основном, так и на удаленном рабочем столе одновременно.

У вас есть три окна (рис. 8.17), и в каждом из них можно запустить различные приложения, конечно, если хватит мощности вашего основного компьютера. В одном окне — игру, в другом — терминал, в третьем — доступ в Интернет. И если ваш основной компьютер не достаточно мощный, то они будут работать с большой задержкой. Это будет вам действовать на нервы, и вы будете автору этих строк задавать вопрос: «И зачем мне все это нужно? »

А вы не забыли, что вся наша виртуализация, хотя она и хороша сама по себе, нужна нам была для того, чтобы смоделировать процессы, имеющие место в реальных компьютерных сетях. И все наши виртуальные машины — это реальные узлы вашей локальной или корпоративной сети. И здесь мы подходим к основным выводам, которых по философским, а не техническим меркам, может быть всего два, а именно, «один ко многим» и «многие к одному».

Другими словами, использование протокола RDP позволяет в Ubuntu-Windows системах:

- Проводить удаленное конфигурирование, настройку и администрирование множества узлов сети.
- Организовать удаленный терминальный доступ множества пользователей на один мощный сервер сети, используя доступ к его рабочему столу. При этом множество Windows-пользователей могут работать на своих компьютерах в среде Linux-систем. При условии, что именно она и установлена на этом сервере.

8.3. Доступ к удаленным рабочим столам по протоколу VNC

Выше мы рассмотрели работу компьютеров, использующих протокол RDP, являющийся основным в Windows-системах при организации доступа к удаленным рабочим столам.

Для взаимодействия Ubuntu-Windows систем нам пришлось устанавливать и запускать в работу RDP-сервер на Ubuntu-машине. Попробуем остановить работу этого сервера. Для этого в консоли vmUbuntu10 выполним команду:

```
sudo /etc/init.d/xrdp stop
```

И если состояние экрана основного компьютера было как на рис. 8.17, то мы увидим что окна удаленных рабочих столов закроются ввиду того, что потерялась связь основного компьютера и виртуальной машины vmUbuntu06 с виртуальной машиной vmUbuntu10. Если мы попробуем

повторно инициировать доступ с основного Windows-компьютера или виртуальной машины vmUbuntu06 к удаленному рабочему столу vmUbuntu10, то потерпим фиаско.

Как настраивать доступ к рабочему столу Ubuntu 10.04, мы уже знаем из п. 8.2.3. Попробуем сделать аналогичные действия на vmUbuntu06. Ничего особо нового (то есть наоборот, старого) мы на окне настройки не увидим (рис. 8.19), за исключением одной строки, которая важна для дальнейших наших рассуждений.

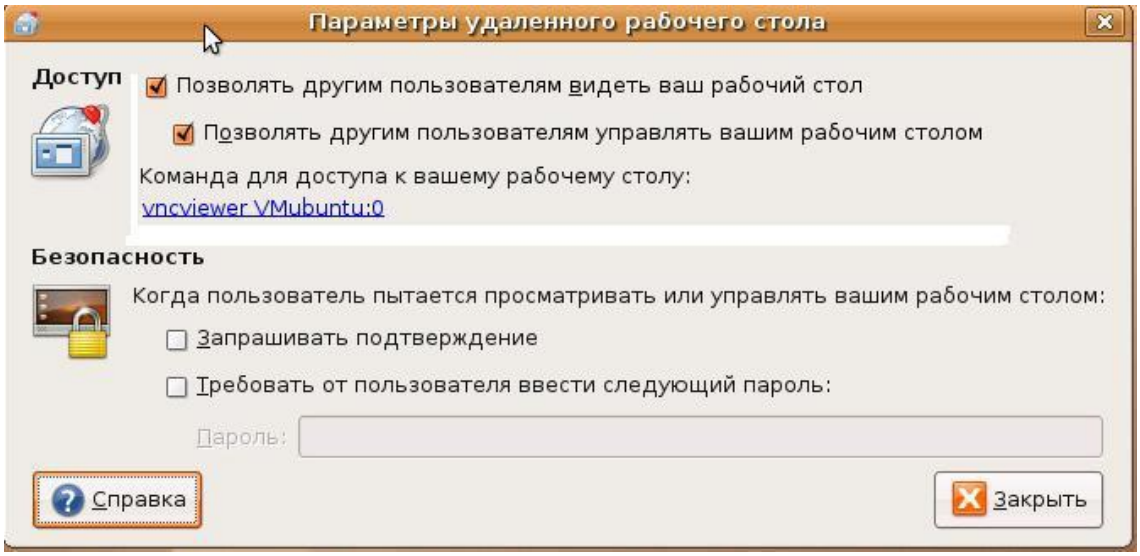

Рис. 8.19. Настройка доступа к рабочему столу в vmUbuntu06.

Ubuntu версии 6.04 предлагает нам для доступа к ее рабочему столу использовать команду:

```
vncviewer [имя или IP-адрес] : [номер дисплея]
```

Ключевыми являются первые три буквы этой команды, которые позволяют нам сделать вывод, что Ubuntu 6.04 организует свой рабочий стол так, что удаленный доступ к нему поддерживается протоколом VNC.

Работая в терминалах vmUbuntu10 или vmUbuntu06, при вводе команды vncviewer у нас появляется возможность получать доступ к удаленным столам противоположных Ubuntu-машин. Того же эффекта можно достичь если в основном меню выбирать:

- «Приложения» -> «Интернет» -> «Просмотр удаленных рабочих столов» (в Ubuntu 10.04)
- или «Приложения» -> «Интернет» -> «Клиент терминального сервера» -> «Протокол: VNC» (в Ubuntu 6.04).

То есть в Ubuntu-системах протокол VNC интегрирован в поставку операционной системы и является основным, по умолчанию, протоколом взаимодействия с удаленными рабочими столами Ubuntu-систем.

8.3.1. Общие сведения о VNC

VNC — это широко распространенный метод удаленного доступа к рабочему столу компьютера по сети. Данные о нажатии клавиш и движении мыши, выполняемых пользователем на собственном компьютере, передаются по сети на удаленный компьютер и воспринимаются им как действия с его собственной клавиатурой и мышью. Информация с экрана удаленного компьютера выводится на экране компьютера пользователя.

Работа по VNC через Интернет с удаленным компьютером, находящимся в противоположной точке мира, для пользователя выглядит так, как будто этот компьютер находится непосредственно перед ним. Особенно VNC протокол удобен при работе с графическим интерфейсом. Для начинающих администрирование удаленных серверов по VNC намного проще, чем через командную строку по SSH или панель управления с веб-интерфейсом.

- На удаленном компьютере должна быть запущена программа-сервер (VNC server), которая играет роль клавиатуры, мыши и монитора, и обменивается данными с компьютером пользователя. Доступ к VNC-серверу может быть защищен паролем.
- На компьютере пользователя должна быть запущена программа-клиент (VNC client, VNC viewer), которая передает на удаленный компьютер информацию о нажатиях на клавиши и движениях мыши, получает от него изображение и выводит его на экран.

Основной объем передаваемых по VNC данных приходится на графическую информацию, выводимую на экран. Для работы требуется канал, ширина пропускания которого от 32 Кбит/с до 2 Мбит/с. Для комфортной работы в полноцветном режиме при разрешении экрана 1024×768 скорость канала должна быть 1–2 Мбит/сек. При снижении качества графики, при уменьшении числа цветов и при некоторых дополнительных способах оптимизации приемлемое удобство может обеспечить скорость 128 Кбит/с.

Канал занимается полностью только при обновлении больших участков экрана, при печати текста трафик заметно меньше, а в остальное время канал практически не используется. Если при передаче по каналу возникают большие задержки передачи пакетов (медленные каналы, спутниковая связь, большие расстояния), это вызывает ухудшение времени реакции на нажатие клавиш и движение мыши, что значительно снижает комфортность работы.

VNC – система удаленного доступа к рабочему столу компьютера по протоколу RFB (Remote FrameBuffer). VNC является межплатформенным программным обеспечением. Реализация клиентов VNC существует даже

на JAVA, что позволяет использовать программное обеспечение VNC-клиента на телефоне или смартфоне.

К одному VNC-серверу может подключаться сразу несколько клиентов. VNC — это бесплатное программное обеспечение с открытым исходным кодом, что, без сомнения, является преимуществом. Для обеспечения безопасности устанавливать VNC-соединение возможно через SSL, VPN или SSH-туннель.

В некоторых случаях, когда сервер не снабжен GUI-интерфейсом, есть возможность установки программы Xming и вызов графического приложения на сторону клиента через SSH-туннель (см. п. 8.4). Более полная информация о протоколе VNC приведена в приложении 3 к данному разделу.

8.3.2. Настройка VNC-сервера в Ubuntu

Существует большое количество модификаций VNC-серверов и VNC-клиентов. Что касается Ubuntu, то наиболее популярным являются как раз те, что присутствует в сборке Ubuntu Linux по умолчанию, это:

- VNC-сервер для Linux — Vino,
- VNC-клиент для Linux — Vinagre.

Если по какой-то причине у вас в системе не оказалось ни сервера, ни клиента, или они перестали работать, то установить их заново можно, выполнив в консоли команды:

```
sudo aptitude install vino
sudo aptitude install vinagre
```

В состав пакета Vino входит множество файлов, но отметим два исполняемых файла, которые вам когда-либо потребуются при настройке VNC-соединений. Это файл /usr/bin/vino-preferences — автозапуск сервиса при старте системы и строка вызова /usr/lib/vino/vino-server в службах и приложениях. Еще одним вариантом запуска VNC-сервера Vino, является возможность использовать для этого команду:

```
sudo /usr/lib/vino/vino-server --sm-client-disable
```

При наличии на стороне VNC-сервера GUI, а Vino по сути и не работает без графической подсистемы, настроить его параметры можно выбирая: «Система» -> «Параметры» -> «Удалённый рабочий стол». И мы попадем в уже знакомое нам окно «Параметры удаленного рабочего стола» (рис. 8.6). То есть, устанавливая галочки или вводя пароль в этом окне, мы делаем не что иное, как настраиваем VNC-сервер Vino.

Есть еще один способ просмотра и изменения параметров VNC-сервера Vino. Правда, этот способ не для ламмеров. Нажмите Alt+F2, в

открывшемся окне введите gconf-editor. У вас откроется окно «Редактор конфигурации» (рис. 8.20).

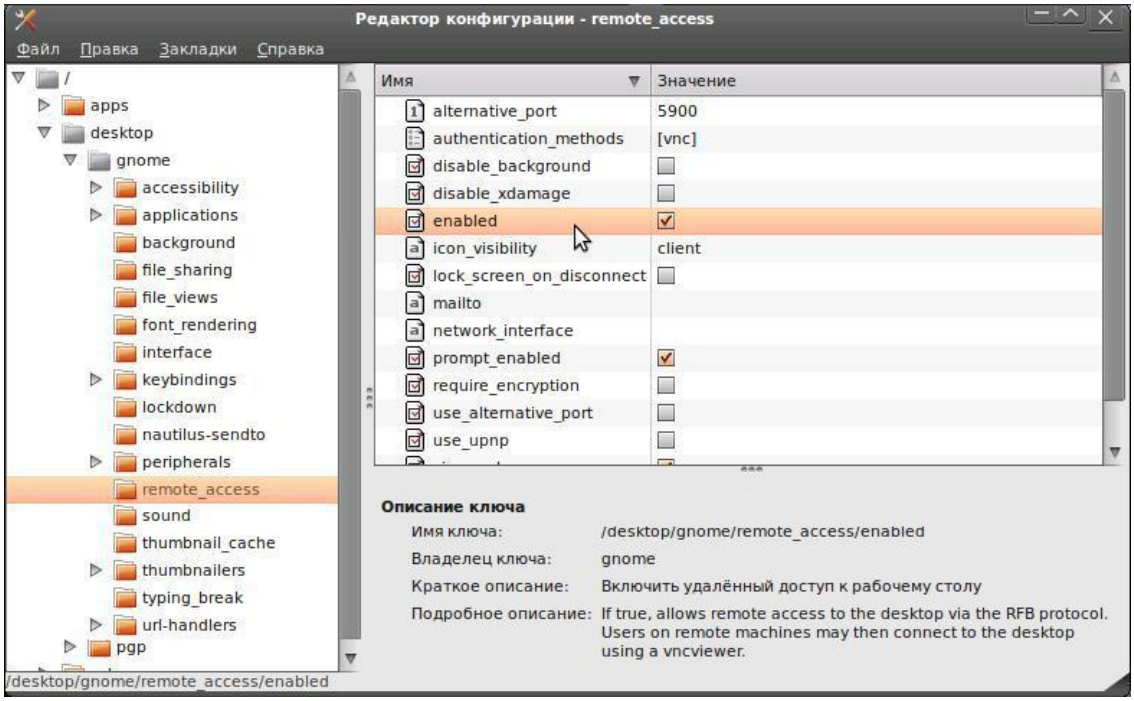

Рис. 8.20. Окно «Редактор конфигурации».

Следуя в левой части окна последовательно по опциям «desktop» -> «gnome» -> «remote_access», в правой части экрана у вас должно появиться окно с параметрами (ключами) настройки VNC-сервера Vino. В частности, если вам надо включить или отключить удаленный доступ к рабочему столу, достаточно установить или снять галку напротив пункта: enabled. Аналогично и с другими ключами, описание которых отображается в нижней части окна.

Если вам, по каким-то причинам не нравится Vino, вы можете использовать альтернативные серверы, такие как: tightvncserver, vnc4server, x11vnc. Установить их в Ubuntu так же просто, как и любой другой софт. Надо только подключиться к Интернет и ввести команду:

```
sudo aptitude install [имя VNC-сервера]
```

Не обязательно устанавливать сразу все, можно установить любой из них на выбор. В частности, отличие tightvnc от других серверов в том, что он не «расшаривает» ваш рабочий стол, а создает совершенно новую сессию.

То есть удаленный пользователь не увидит ваших открытых окон. В каких-то случаях это большой плюс и удобно для совместной работы. В каких-то, наоборот, не очень удобно, особенно когда необходимо кому-то помочь удаленно.

8.3.3. Настройка и работа с VNC-клиентом в Ubuntu

В этом разделе основное внимание уделяется утилите Vinagre, которая является VNC-клиентом Ubuntu. Именно она осуществляет подключение к удаленному рабочему столу другого компьютера и обеспечивает работу с ним. Для вызова Vinagre из терминала Ubuntu используется команда:

```
vinagre [Ключи] [Сервер][::Порт|:Дисплей]
```

Таблица 8.3.

Назначение основных параметров и ключей данной утилиты:

Параметры:		Назначение
Сервер		Имя или IP-адрес хоста, к удаленному рабочему столу которого выполняется подключение.
Порт		Номер VNC-порта, по умолчанию 5900. При значении ':0' будет ':5900', при ':1' - ':5901' и т.д. до 1024.
Дисплей		Идентификатор дисплея (туннеля) на котором Vinagre должен быть запущен (от 0 до 1024).
Ключи приложения:		Назначение
--fullscreen	-f	Открыть vingare в полноэкранном режиме
--new-window	-n	Создать новое окно верхнего уровня в текущем экземпляре
--file=имя файла	-F	Открыть файл, распознаваемый программой vinagre
--display=дисплей		Используемый дисплей X
Ключи GTK+		Назначение
--class=КЛАСС		Класс программы, используемый диспетчером окон
--name=ИМЯ		Имя программы, используемое менеджером окон
--screen=ЭКРАН		Используемый экран X
--sync		Включить синхронные вызовы X
--g-fatal-warnings		Сделать все предупреждения фатальными

Для справки. GTK+ (сокращение от GIMP Toolkit) — официальная библиотека создания графического интерфейса проекта GNU. На ее основе построены рабочие окружения GNOME — графическая среда Ubuntu.

Состоит из двух компонентов: GTK и GDK. Первый содержит набор элементов пользовательского интерфейса, или «виджетов» (таких, как кнопка, список и т. п.). GDK отвечает за вывод на экран и использует для этого X Window System.

Рассмотрим несколько примеров вызова VNC-клиента Vinagre из командной строки терминала.

Команда	Описание
~$ vinagre	– Загружает утилиту и открывает окно 'Просмотр удаленных рабочих столов'.
~$ vinagre --new-window &	– Загружает утилиту и открывает новое окно в 'фоновом режиме'
~$ vinagre 192.168.1.6::5900	– Подключение к серверу VNC с адресом 192.168.1.6 по порту 5900
~$ vinagre 192.168.1.6::5900:0	– Подключение к дисплею 0 сервера VNC с адресом 192.168.1.6 по порту 5900
~$ vinagre -F prim.vnc	– Подключение к VNC серверу, сконфигурированное в файле prim.vnc
~$ vinagre -f ::5900	– Подключение по 5900 порту к localhost в полноэкранной моде.

Однако знание этих команд необходимо, когда мы работаем с компьютером удаленно в консольном режиме или автоматизируем процесс администрирования с использованием скриптов. В большинстве случаев, работая за компьютером с Ubuntu локально, особенно пользователям непрофессионалам, оказывается значительно удобнее работать с утилитой Vinagre в ее графическом варианте.

Вызов VNC-клиента Vinagre осуществляется выбором в основном меню опций «Приложение» -> «Интернет» -> «Просмотр удаленных рабочих столов». На экране появится окно аналогичного названия (рис. 8.21).

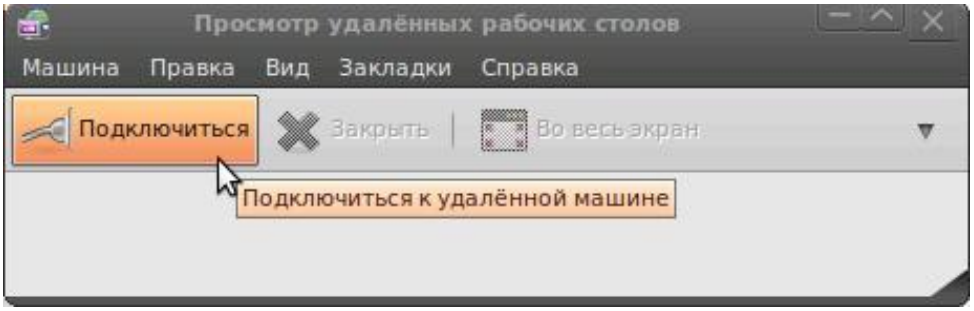

Рис. 8.21. Стартовое окно VNC-клиента Ubuntu — утилиты Vinagre.

Работа в графическом варианте утилиты Vinagre не вызывает никаких сложностей для человека, знакомого с компьютером. Все опции ее русифицированного меню достаточно понятны и есть подробная справка. Поэтому на всех возможностях утилиты останавливаться не будем, а попробуем подключить удаленный рабочий стол.

Для этого необходимо нажать кнопку «Подключиться». На экране появится окно (рис. 8.22), в котором необходимо определить параметры подключения к требуемому удаленному рабочему столу.

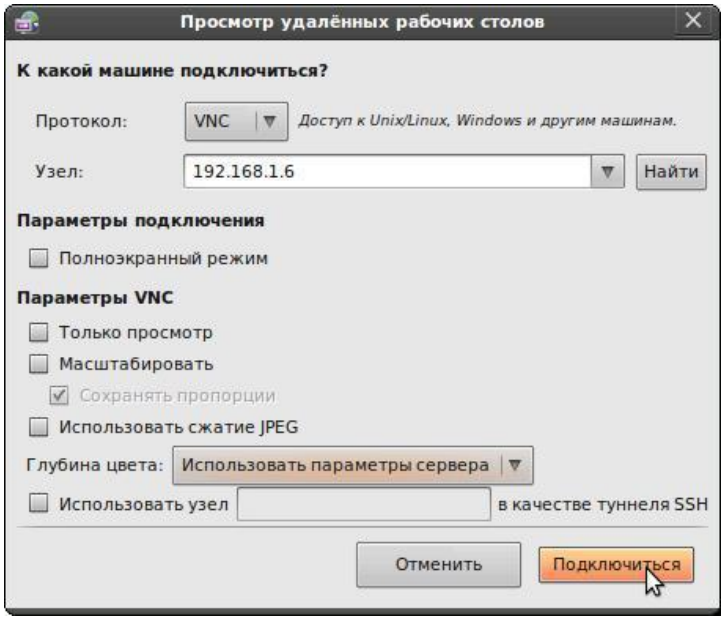

Рис. 8.22. Настройка параметров подключения к удаленному рабочему столу.

Выбор у нас небольшой, так как в нашей виртуальной сети всего две Ubuntu-машины. На одной из них, а именно vmUbuntu10, мы и работаем. Остается возможность подключиться только к vmUbuntu06.

Именно ее адрес мы набираем в поле «Узел» и, выбрав протокол VNC, нажимаем кнопку «Подключиться». Появится окно с требованием вашей аутентификация на компьютере vmUbuntu06. И если она пройдет успешно, то на экране виртуальной машины vmUbuntu10 появится окно, в котором будет отображен рабочий стол vmUbuntu06 (рис. 8.23).

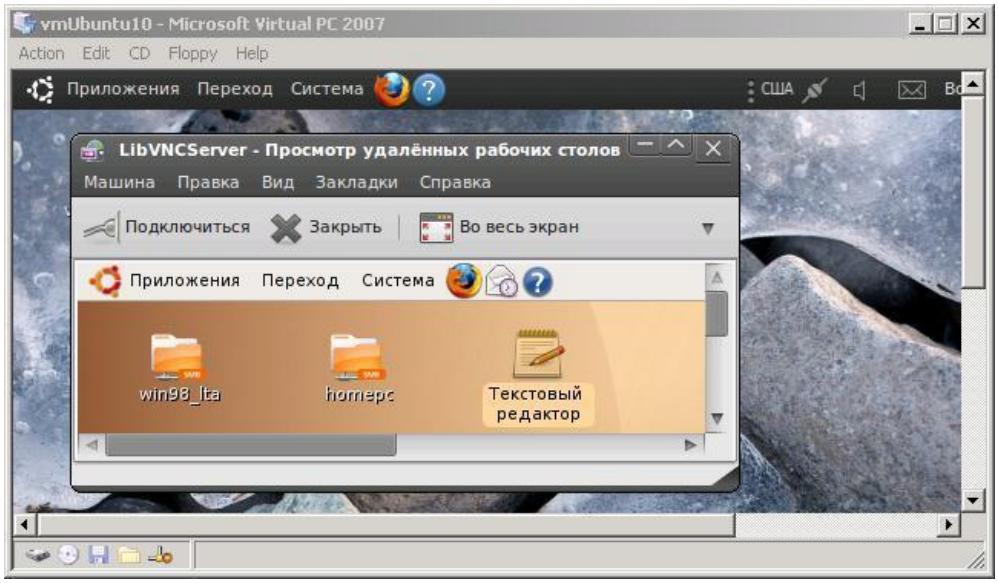

Рис. 8.23. Окно рабочего стола vmUbuntu06 на экране vmUbuntu10.

Причем вид или размер окна, его полноэкранный вывод, возможность только просмотра или допустимости действий внутри него мы могли определить на этапе подключения к удаленному рабочему столу (рис. 8.22).

Не будем подробно рассматривать возможности и параметры утилиты Vinagre, так как среди реализаций VNC существует масса классных бесплатных клиентов и серверов, таких, например, как x11vnc, TightVNC или UltraVNC.

Нам важна технология. И мы, как мне кажется, выяснили, что в среде Ubuntu-систем для дистанционного доступа к удаленным рабочим столам используется в качестве стандартного протокол VNC. И в стандартных поставках операционной системы Ubuntu присутствуют все необходимые программные компоненты, поддерживающие этот протокол. Причем большинство эти компонентов интегрированы непосредственно в графическую оболочку Ubuntu.

8.3.4. Совместимость удаленных рабочих столов Windows и Ubuntu по протоколу VNC

Говоря о совместимости удаленных рабочих столов и доступе к ним в Windows-Ubuntu системах, следует подчеркнуть еще раз, что в Windows для этого используют RDP, в Ubuntu — VNC. И когда мы говорили о совместимости по RDP, то для этого на Ubuntu-машину устанавливали RDP-сервер. Какую-то аналогичную технологию, но теперь по отношению к Windows, мы должны реализовать и сейчас. Почему по отношению к Windows? Да потому, что VNC — это стандарт Ubuntu Linux, и его лучше не менять.

В мире существует множество реализаций программных продуктов, поддерживающих протокол VNC:

- RealVNC — официальная версия, поддерживаемая командой AT&T Laboratories;
- TightVNC — альтернативная версия основанная на RealVNC;
- EchoVNC — реализация VNC под Microsoft Windows;
- UltraVNC — реализация VNC под Microsoft Windows.

Все эти продукты обладают своими достоинствами, и недостатки ориентированы на те или иные среды. Мы не будем проводить их сравнительного анализа. И это не наша задача. Наша задача показать принципиальную возможность реализации технологии доступа с Windows-машины на Ubuntu-машину. Почему именно с Windows-машины на Ubuntu-машину?

Да потому что RDP — стандарт рабочих столов Windows. А для доступа к ним, как было показано в п.8.2.1, в стандартной поставке Ubuntu есть утилита rdesktop.

Вместе с тем в стандартной поставке Ubuntu налицо наличие VNC-сервера. Так что единственное, чего нам не хватает — это VNC-клиента, который и следует попробовать установить на Windows-машину. Для использования на основном компьютере с Windows XP из множества возможных пакетов VNC-клиентов остановимся на RealVNC. Он имеет бесплатную версию, которую можно скачать с официального сайта http://realvnc.com/products/free/4.1/download.html и установить на основном компьютере. Причем при инсталляции, для наших целей можно ограничиться установкой, только VNC-клиента.

После запуска на Windows-компьютере программы VNC Viewer появится окно «Connection Details», где надо в поле «Server» указать адрес компьютера, к рабочему столу которого вы подключаетесь (рис. 8.24).

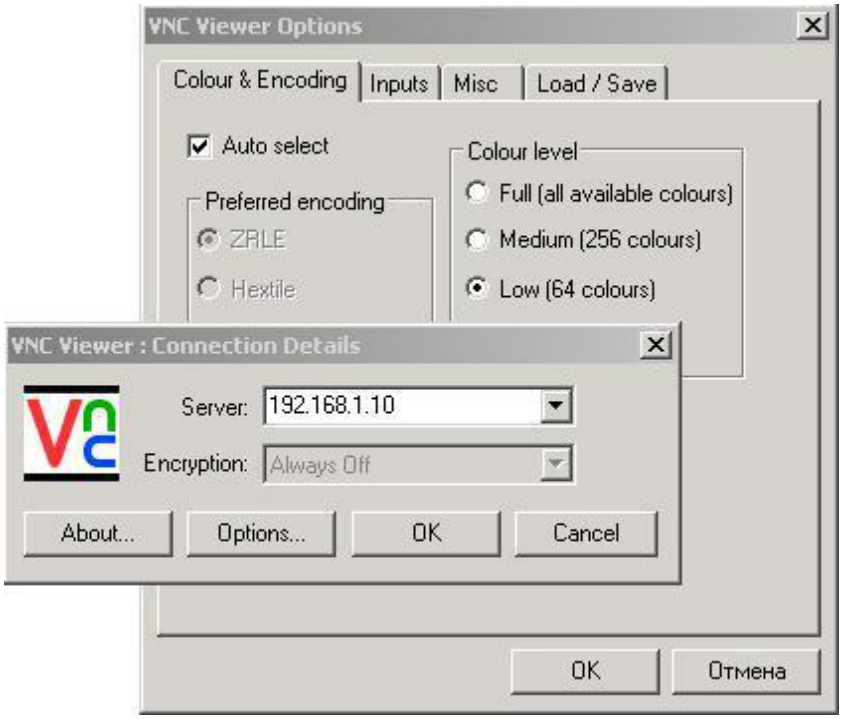

Рис. 8.24. Окно подключения и окно настройки параметров RealVNC.

При первом запуске VNC Viewer следует задать параметры подключения. Для этого нажмите кнопку «Options...». Откроется окно «VNC Viewer Options». Оно имеет три вкладки, на которых надо установить требуемые параметры подключения, а затем перейти на вкладку «Load/Save» для их сохранения.

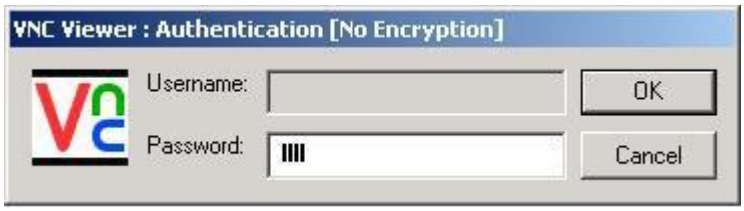

Рис. 8.25. Окно аутентификация подключения к удаленному столу.

После этого следует нажать кнопку «OK» и VNC-клиент начнет устанавливать соединение. Если соединение, возможно, то у вас будет запрошен логин и пароль пользователя удаленного рабочего стола (рис. 8.25). Если вы являетесь пользователем удаленного рабочего стола, то аутентификация выполнится успешно и на экране основного компьютера появится окно удаленного рабочего стола.

Так, например, используя нашу виртуальную сеть, мы можем с основного компьютера установить соединение одновременно с двумя удаленными рабочими стола виртуальных машин vmUbuntu06 и vmUbuntu10 (рис. 8.26).

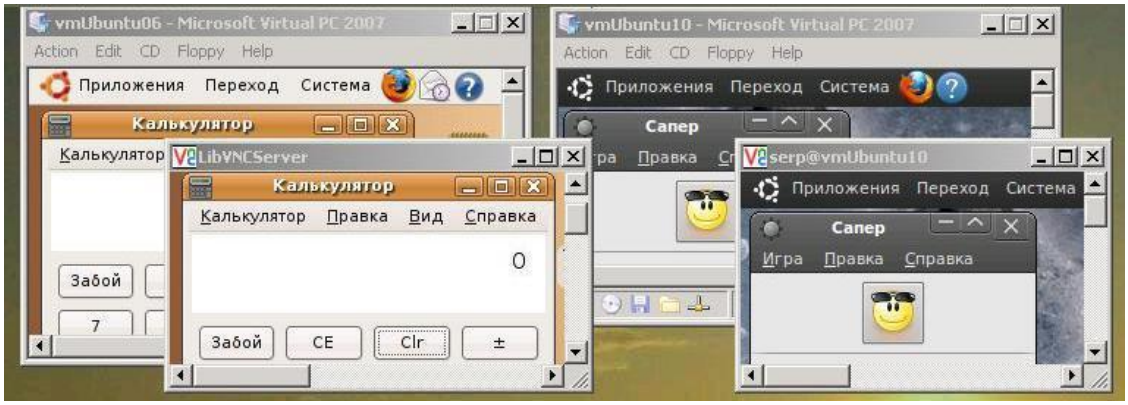

Рис. 8.26. Экран основного Windows-компьютера с удаленными рабочими столами двух Ubuntu-машин.

Таким образом, на экране одного Windows-компьютера у нас присутствуют окна удаленных рабочих столов двух Ubuntu-машин, на которых можно работать в операционной системе Linux Ubuntu, запускать различные приложения и игры, проводить расчеты. А если удаленный рабочий стол настроен только на режим просмотра, то просто наблюдать за происходящим на серверной стороне этого рабочего стола.

На этом знакомство с технологией доступа к удаленным рабочим столам в Ubuntu-Windows системах с использованием протокола VNC мы закончим. Но я хотел бы вас познакомить с еще одной технологией сетевого взаимодействия в Ubuntu-Windows системе по удаленному доступу к ее узлам на базе SSH-туннелей.

8.4. Удаленное подключение к Ubuntu из Windows с помощью Xming и SSH

В отличие от Windows, в Linux графическая оболочка не является частью ядра системы. Стандартная оконная система для Linux — это X Window System, или в разговорной речи «иксы». Краткую справку о ней вы можете найти в приложении 5 данного раздела. Но даже для поверхностного знакомства этого очень мало. Мы бы рекомендовали вам

углубиться в этот вопрос подробнее самостоятельно. Но вернемся к рассматриваемой теме.

Отметим, что X Window System берет на себя отрисовку графических элементов, взаимодействие с устройствами ввода-вывода и, самое главное, имеет прозрачную клиент-серверную архитектуру. Оконная система играет роль сервера, а графические приложения — роль клиентов. Как и положено клиентам, они подключаются к серверу и взаимодействуют с ним для отрисовки и для получения событий мыши и клавиатуры.

Но это еще не все! Дело в том, что оконная система может находиться на другом компьютере, а графическое приложение связываться с ней через сеть. Поэтому можно запустить приложение на удаленном компьютере, заставив его рисовать на том компьютере, за которым вы сейчас работаете. Можно и наоборот. Или вообще, можно запустить программу на одном удаленном компьютере с отрисовкой интерфейса на другом удаленном компьютере. Заманчивая возможность, не правда ли?

Попробуем, не вдаваясь в глубокую теорию, реализовать эту технологию применительно к удаленному доступу из среды Windows XP к Ubuntu-машине. Необходимый состав программных средств, который реализует эту сетевую технологию, будет следующим:

- На удаленной Ubuntu-машине (это vmUbuntu10 - 192.168.1.10).

Все, что нам будет нужно от Ububtu-машины - это его SSH-сервер. Через него мы будем удаленно подключаться и запускать нужные нам программы. Поэтому проверим, что он находится в активном состоянии и его сервис открыт по 22 порту. Если это не так, то повторно активируйте работу SSH-сервера, как мы это делали в п.7.2.

- На локальной Windows-машине (это основной ПК - 192.168.1.2).

Для подготовки к работе этого компьютера нам будут необходимы две программы:

- SSH-клиент, в качестве которого мы будем использовать уже установленную утилиту PuTTY.
- X Server для Windows, в его качестве можно использовать пакет Xming для Windows.

Необходимое программное обеспечение установлено, и можно перейти его настройке. Софт Ubuntu-машины настройки не требует. Основная работа заключается в настройке двух пакетов на Windows-машине.

Настройка PuTTY

Запускаем утилиту. Выполняем настройки аналогично п. 7.3.2 и переходим на вкладку Connection -> SSH -> X11, где дополнительно настраиваем перенаправление графического интерфейса (рис. 8.27).

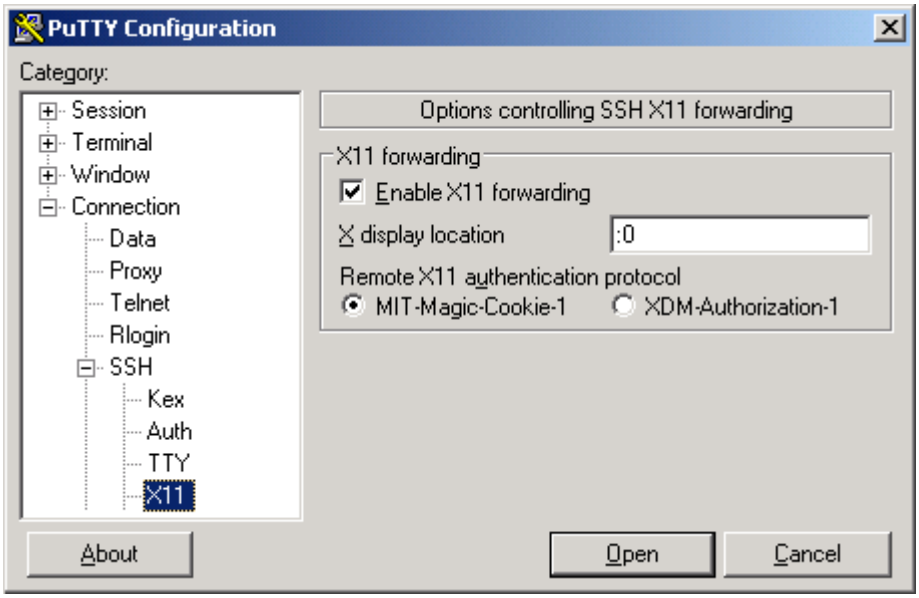

Рис. 8.27. Настройка перенаправления графического интерфейса в PuTTY .

На этой вкладке включаем перенаправление графического интерфейса. Указываем дисплей — :0. Возвращаемся на вкладку Session и сохраняем настроенную сессию под каким-либо именем для возможности загрузки при следующем запуске PuTTY.

Подключаемся к vmUbuntu10. После подключения вводим логин и пароль и видим текстовую консоль. В ней можно удаленно запускать консольные программы, но графические программы не могут рисоваться в консоли. Поэтому оставим на время наше подключение по SSH.

Настройка Xming

Для настройки Xming следует запустить программу XLaunch — это мастер настроек пакета Xming (рис. 8.28).

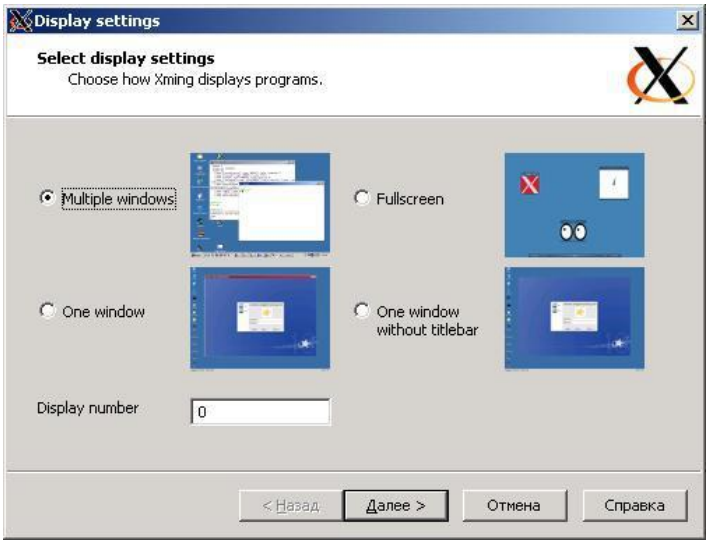

Рис. 8.28. Окно настроек пакета Xming.

В тестовом варианте, который мы используем для столь серьезного программного продукта, можно согласиться со всеми предлагаемыми по умолчанию опциями. Поэтому последовательно жмем «Далее» на всех экранах.

Тонкая настройка параметров Xming — отдельная песня. И мы предлагаем вам пропеть ее соло или в хоре с друзьями. Мы только зададим ее тональность.

- На первом шаге указывается способ интеграции графической оболочки Xming в графическое окружение Windows. По умолчанию каждое приложение Ubuntu будет в своем окне.
- На втором шаге предлагается указать приложение, которое будет автоматически запускаться вместе с иксами.
- На третьем шаге опция Clipboard позволяет интегрировать буфер обмена и есть возможность указать параметры запуска Xming. Строка параметров, например, может иметь вид:

```
-dpi 96 -xkblayout us,ru
```

В этой строке указан желаемый размер шрифтов и возможность работы с двумя раскладками клавиатуры.

И, наконец, на последнем шаге сохраняем настройки кнопкой «Save configuration» и запускаем X-сервер кнопкой «Готово». В системном лотке около часов появится иконка Xming. В дальнейшем запустить сервер с теми же настройками можно просто путем открытия сохраненного файла. Изменить настройки можно через контекстное меню файла.

Итак, X-сервер запущен. Запускаем PuTTY, попадаем в консоль, предоставленную соединением SSH. Ранее, в разделе 7.3.2, мы в ней удаленно вводили команды и запускали консольные приложения и в этой же консоли видели вывод этих приложений. Например, вводили команду ls или запускали текстовый редактор Nano.

А что теперь будет, если из этой текстовой консоли запустить графическое приложение?

До запуска X-сервера на основной Windows-машине мы бы получали ошибку, потому что подключались к удаленному компьютеру в консольном режиме, и рисовать окна было просто нечем. Но сейчас у нас включено перенаправление графики на нашу Windows-машину, на которой уже запущен свой X-сервер. Поэтому, если запустить оконное приложение в удаленном консольном терминале, то его окно нарисуется на компьютере Windows.

Например, если используя консольное подключение к vmUbuntu10, вызвать графическое приложение xeyes, то на экране основной Windows-машины появится новое графическое окно xeyes — это оконное приложение Linux, которое выполняется на Ubuntu-машине и отрисовывается на основном компьютере (рис. 8.29).

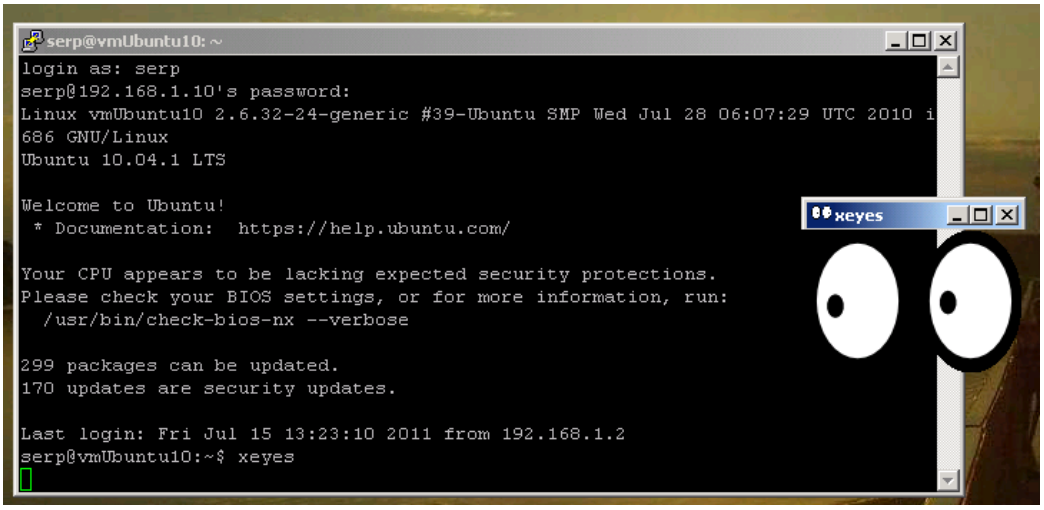

Рис. 8.29. Экран основного компьютера с запущенным на Ubuntu оконным приложением xeyes.

Вы можете попробовать ввести какую-либо другую команду, вызывающую оконное приложение Ubuntu. Например, текстовый редактор

```
gedit &
```

Амперсанд в конце команды указывает, что программу нужно запустить в фоновом режиме, чтобы во время ее работы консоль была доступна для других действий. Особый интерес представляет собой возможность удаленно запускать такое мощное оконное приложение Linux, как OpenOffice. Наберите в командной строке консоли команду

```
soffice -calc
```

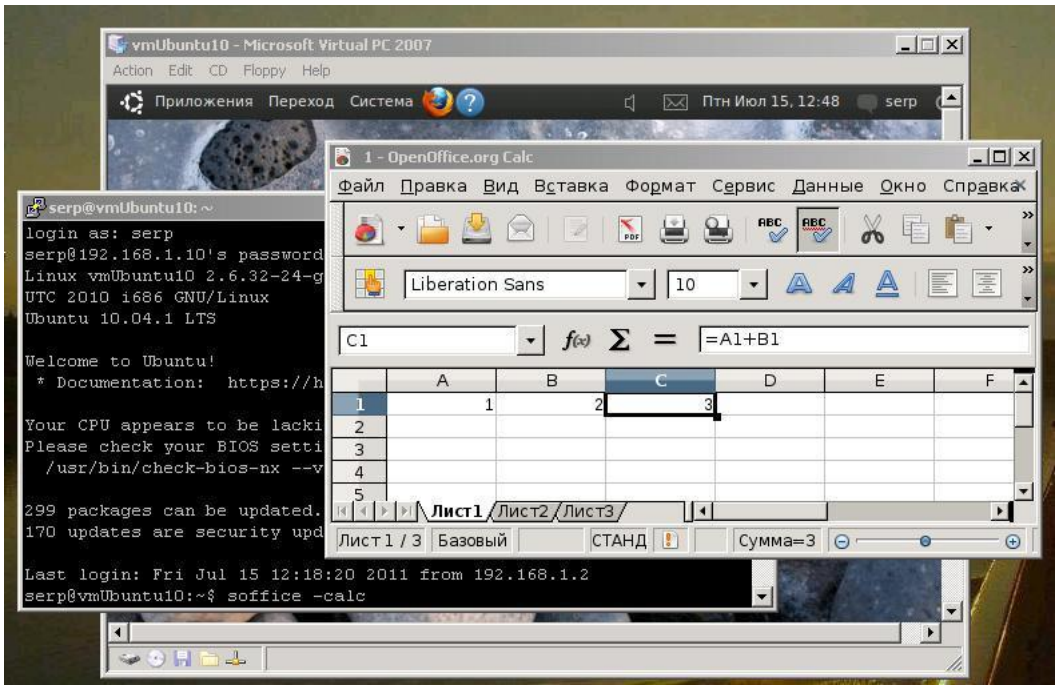

Рис. 8.30. Графическое окно OpenOffice.org Calc в Windows.

Экран основной Windows-машины примет вид, аналогичный тому, что приведен на рис. 8.30. На этом рисунке три окна: первое — виртуальная Ubuntu-машина vmUbuntu10, второе — консоль утилиты PuTTY, третье — графическое окно OpenOffice.org Calc, запущенного на vbUbuntu10 и отображенного на Windows-компьютере.

Надеемся, у вас все получилось и на вашем рабочем столе Windows красуются оконные приложения Linux, в том числе и аналог Excel.

Если же нет, а это может встретиться на некоторых версиях Ubuntu, то требуется более тонкая настройка, связанная с корректировкой пары параметров в файлах /etc/default/ssh и /etc/init/ssh.conf. Для тех, кого этот вопрос интересует достаточно подробно, приводим ссылки на форумы по этому вопросу:

- https://bugs.launchpad.net/ubuntu/+source/openssh/+bug/434799,
- http://forum.ubuntu.ru/index.php?topic=120715.15.

Для всех остальных может оказаться достаточно приведенного ниже листинга настройки SSH-сервера vmUbuntu06 (192.168.1.6) для доступа к ее рабочему столу, который будет иметь вид:

```
login as: serp
serp@192.168.1.6's password:
Linux vmUbuntu06 2.6.17-10-generic #2 SMP Fri Oct 13
18:45:35 UTC 2006 i686

serp@VmUbuntu06:~$ xeyes
Warning: locale not supported by Xlib, locale set to C
```

Из приведенного протокола выполнения команды xeyes видно, что обнаружена ошибка. Проведем дополнительный тест:

```
serp@VmUbuntu06:~$ echo $DISPLAY
localhost:10.0
```

Но при настройке в PuTTY проброса X11 мы для параметра X display location указывали дисплей :0. Для исключения этой ошибки отредактируем два файла.

На первом этапе правим файл /etc/default/ssh в части изменения параметра SSHD_OPTS, который должен принимать значение, равное 4:

```
serp@VmUbuntu06:~$ sudo nano /etc/default/ssh
. . .
# Options to pass to sshd
SSHD_OPTS=-4
. . .
```

Затем переходим к редактированию файла /etc/init/ssh.conf в части изменения строки exec /usr/sbin/sshd.

Следует отметить, что в Ubuntu версии 6 выполнение этих изменений может и не потребоваться.

```
serp@VmUbuntu06:~$ sudo nano /etc/init/ssh.conf
. . .
exec /usr/sbin/sshd -4
. . .
```

После выполнения этих изменений необходимо будет либо перезагрузить компьютер, либо перезапустить SSH-сервер:

```
# Перезапускаем SSH-сервер
serp@VmUbuntu06:~$ sudo /etc/init.d/ssh restart
```

В заключение данного раздела попробуем коротко сформулировать рассмотренную выше сетевую технологию, выделив ее основные узлы и процессы.

Итак, на Windows-машине мы используем SSH-клиент (PuTTY) для ввода консольных команд. Эти команды передаются по защищенному шифрованному каналу и принимаются SSH-сервером Ubuntu-машины, который транслируют их ядру операционной системы Ubuntu, а именно X Windows System.

Теперь уже Ubuntu, как клиент, запускает и выполняет соответствующую программу, а результаты ее выполнения, а именно графическую отрисовку, портирует своему серверу X Windows System.

Таким сервером у нас является установленная на Windows-машине программа Xming. Как порт сервера X Window System для операционной системы Microsoft Windows, она обеспечивает прием текущей сессии X11 с Ubuntu-машины и отрисовку ее графического окна на экране основной Windows-машины.

При этом поддержку приема шифрованной передачи сессии X11 с Ubuntu-машины по протоколу SSH осуществляет PuTTY, выполняя роль организации и поддержки защищенного канала между Windows- и Ubuntu-машинами.

8.5. Удаленное подключение к Ubuntu при отключенном GNOME

Этот раздел представляет собой бонусную программу для тех, кого заинтересовал предыдущий раздел. Давайте подумаем, а зачем вообще графическая среда Ubuntu, да еще в виртуальной сети, если отрисовкой графических окон у нас занимается Xming на основной Windows-машине.

На наш взгляд, это лишняя трата ресурсов основного компьютера при исследовании виртуальной сети. А если это так, то давайте попробуем устранить этот недостаток.

С этой целью немного отредактируем два конфигурационных файла Ubuntu Linux, а именно: /etc/init/gdm.conf и /etc/init/rc-sysinit.conf. Для редактирования файла /etc/init/gdm.conf выполним команду

```
serp@vmUbuntu10:~$ sudo nano /etc/init/gdm.conf
```

файл откроется для редактирования

```
# gdm - GNOME Display Manager
# The display manager service manages the X servers
running # on the system, providing login and auto-login
services

description    "GNOME Display Manager"
author         "William Jon McCann <mccann@jhu.edu>"

start on (filesystem
          and started dbus
          and (graphics-device-added fb0
PRIMARY_DEVICE_FOR_DISPLAY=1
               or drm-device-added card0
PRIMARY_DEVICE_FOR_DISPLAY=1
               or stopped udevtrigger))
stop on runlevel [016]
. . .
```

в этот файл добавим новую строку

```
. . .
start on (filesystem
          and runlevel[5]
          and started dbus
. . .
```

Сохраним сделанные изменения и закроем файл.

Откроем для редактирования второй файл, а именно /etc/init/rc-sysinit.conf, выполнив команду:

```
serp@vmUbuntu10:~$ sudo nano /etc/init/rc-
sysinit.conf
```

и в строке DEFAULT_RUNLEVEL заменим значение 2 на 5.

```
. . .
# Default runlevel, this may be overridden on the kernel command-line
# or by faking an old /etc/inittab entry
env DEFAULT_RUNLEVEL=5
. . .
```

Сохраним сделанные изменения и закроем файл.

После окончания редактирования двух конфигурационных файлов выполним перезагрузку Ubuntu-машины. Для этого введем команду:

```
serp@vmUbuntu10:~$ sudo shutdown -r now
```

После перезагрузки операционная система Ubuntu загрузится в текстом режиме, и вам будет предложено ввести имя пользователя, его пароль. Вид экрана виртуальной Ubuntu-машины примет вид, аналогичный тому, что представлен на рис. 8.31.

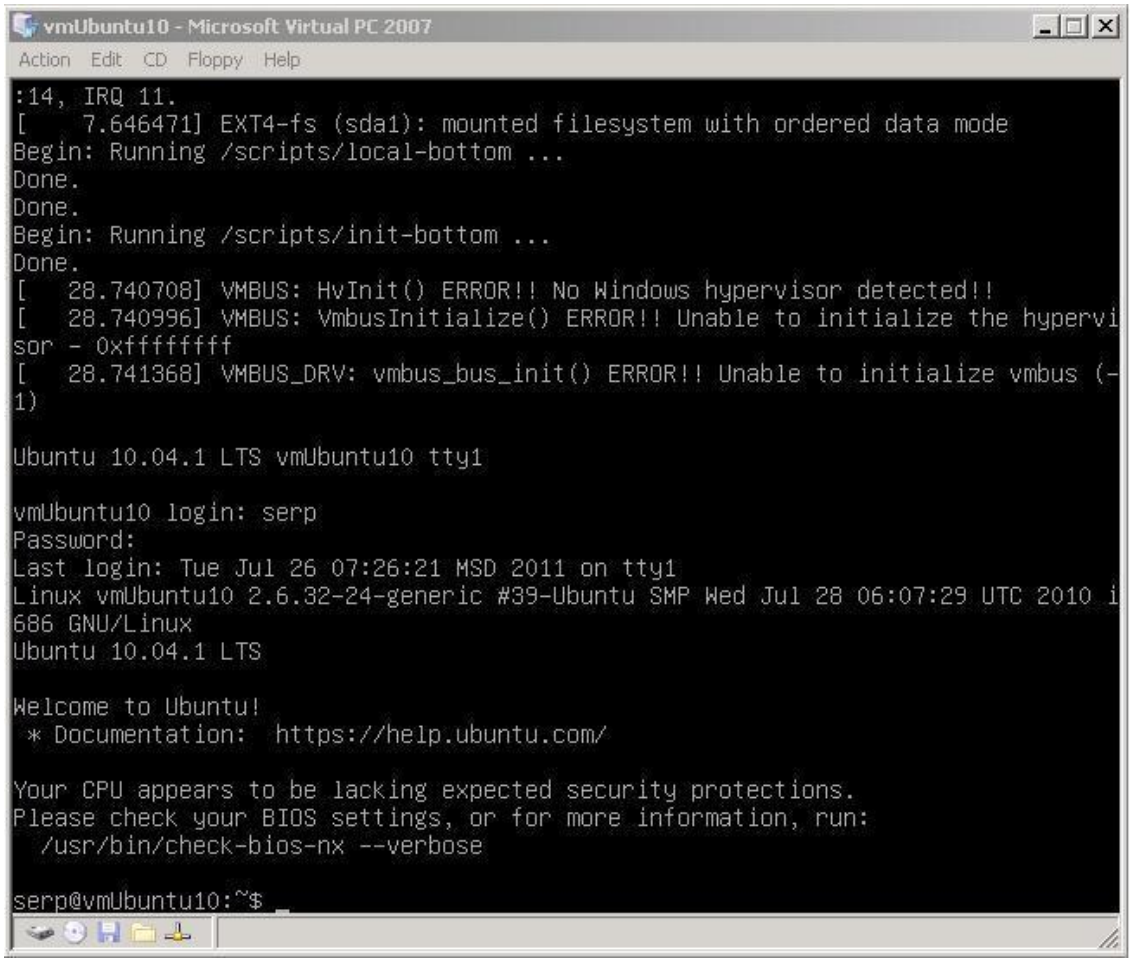

Рис. 8.31. Текстовый режим загрузки операционной системы Ubuntu.

То есть теперь будет доступна только текстовая консоль Ubuntu Linux, хотя для поставленной цели и она нам не очень-то нужна. Мы вообще можем забыть о том, что в нашей виртуальной сети есть окно vmUbuntu10 и его можно вообще свернуть, чтобы оно нам не мешало на экране.

Свернуть, но не закрывать, так как мощь и красота графических приложений Linux на базе библиотек Gtk+ или Qt — это совсем не то, от чего мы можем отказаться в нашей виртуальной среде.

Чтобы убедиться в этом, вы можете из основного Windows-компьютера, установив SSH-соединение с Ubuntu-машиной, проделать все то же самое, что делали в предыдущем разделе. И отрисовка Ubuntu-окон на Windows-

компьютере вам будет так же доступна, как и ранее, несмотря на измененный режим загрузки Ubuntu (рис. 8.32).

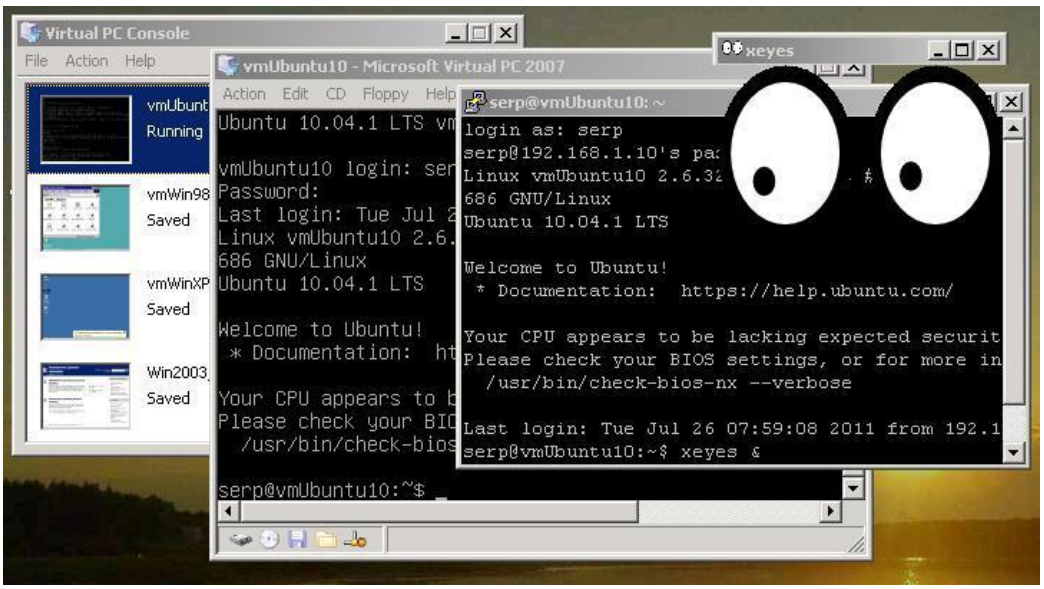

Рис. 8.32. Вид экрана основного Windows-компьютера с текстовым окном vmUbuntu10 и графическим Linux-окном eyes.

Если вы очень расстроились, что потерялся красивый графический интерфейс Ubuntu, то не расстраивайтесь. У вас есть возможность в любой момент войти в текстовую консоль vmUbuntu10 и набрать команду:

```
serp@vmUbuntu10:~$ startx
```

Все сразу будет хорошо, так как графическая среда Ubuntu на месте и можно сколько угодно ходить по меню и тыкать мышкой. Но посмотрите, сколько лишнего времени, работая в виртуальной сети на одном компьютере, вы будете тратить на перезагрузку операционной системы. Сколько дополнительных ресурсов требуется вашему основному компьютеру на поддержку графической оболочки. А ведь в вашей виртуальной сети такая виртуальная машина не одна. Решение за вами — каждый кузнец своего счастья.

9. ОБЩИЙ ДОСТУП В UBUNTU-WINDOWS СИСТЕМАХ

Изложение материала данного раздела будет базироваться на том, что у нас построена одноранговая виртуальная сеть из четырех виртуальных машин и одной основной. Структура тестовой системы имеет вид, приведенный на рис. 8.1. На всех Windows-машинах организованы полностью общедоступные папки: vmPC_share, vm98_share и vmXP_share. Логическая структура виртуальной сети на текущий момент:

- виртуальные машины vm_WinXP и vm_Win98 составляют рабочую группу Virtual-Net;
- основной компьютер Main-PC входит в рабочую группу Home-Net;
- виртуальные машины на базе Ubuntu пока логически не настраивались, но имеют сетевые подключения по IP-адресам;
- на всех компьютерах добавлен один и тот же пользователь с именем serp и таким же паролем.

Никаких дополнительных сетевых настроек на vmUbuntu10 и vmUbuntu06 после установки операционной системы не производилось.

Однако если выполнить из основного меню Переход -> Сеть или, находясь в файловом менеджере Nautilus, выбрать опцию Сеть, то на экране появится окно, аналогичное рис. 9.1.

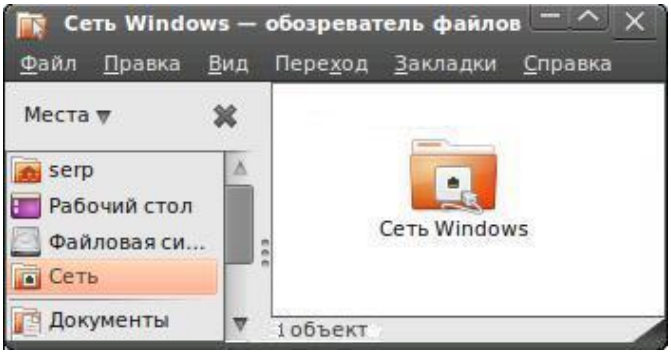

Рис. 9.1. Окно файловый менеджер Nautilus в режиме Сеть.

Если теперь сделать двойной клик на пиктограмме «Сеть Windows», то в этом окне отобразятся обе рабочие группы нашей виртуальной сети (рис. 9.2). Таким образом, файловый менеджер Nautilus позволяет, наряду с ресурсами локального компьютера, просматривать и общедоступные ресурсы сети, в том числе и на Windows-машинах. То есть он может выполнять те же функции, что и «Сетевое окружение» в Windows. Кликая

мышкой на пиктограммы в окне Nautilus, можно добраться до интересующих вас общедоступных папок Windows-машин, провести в них копирование, создание или удаление файлов. Естественно, если у вас есть на это права доступа.

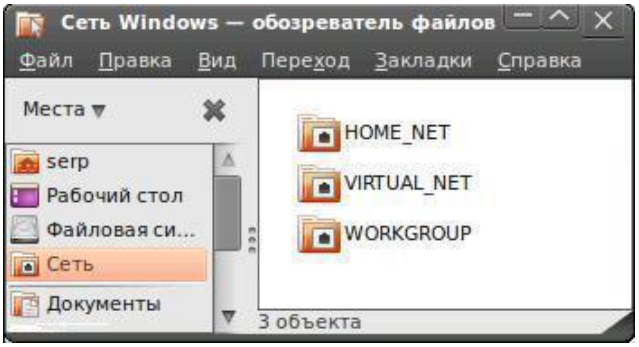

Рис. 9.2. Рабочие группы виртуальной сети в окне файлового менеджера.

Например, кликнув по пиктограмме Virtual-Net, можно убедиться в том, что нам доступны обе виртуальные Windows-машины (рис. 9.3).

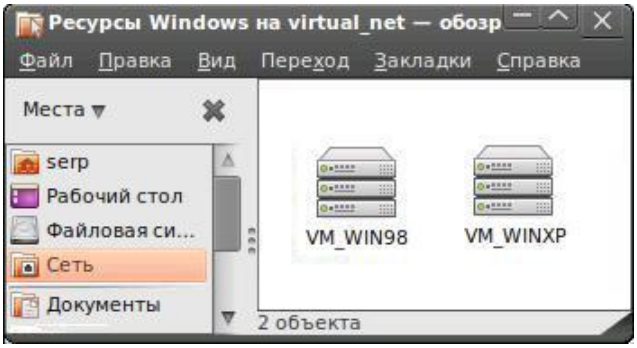

Рис. 9.3. Виртуальные Windows-машины рабочей группы Virtual-Net.

И тут возникает вопрос: «Что позволило компьютеру с иной ОС и другой файловой организацией получить сетевой доступ с общедоступным Windows-ресурсам? »

Но этот вопрос не единственный, потому что тут же возникает второй вопрос: «А почему мы не можем какую-либо папку Ubuntu-машины сделать общедоступной в сети, получая сообщение, аналогичное рис. 9.3 ?»

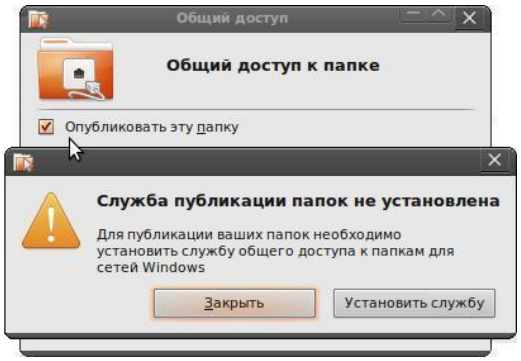

Рис. 9.4. Попытка открыть общий доступ к папке из домашнего директория.

В этой главе мы попробуем найти ответы на поставленные вопросы, а также познакомиться с организацией доступа к общесистемным ресурсам как со стороны Ubuntu к Windows, так и наоборот. А вопрос организации и администрирования файлового сервера Ubuntu рассмотрим уже в следующей главе.

9.1. Средства поддержки сетевого обмена в Ubuntu-Windows системах

9.1.1. Общие сведения о протоколе SMB/CIFS

Как мы видели на рис. 9.1, при входе в режим Сеть файловый менеджер Ubuntu отображает в окне пиктограмму с надписью не просто «Сеть», или как-либо иначе, а именно «Сеть Windows». То есть он понимает, что это сеть, в основе построения которой лежит протокол SMB/CIFS. Более подробно о нем можно узнать из описаний или в сокращенном виде на сайте http://ru.wikipedia.org/wiki/SMB. Некоторые общие сведения приведены в Приложение к данной главе. Здесь мы ограничимся только короткой справкой.

SMB (Server Message Block) — сетевой протокол прикладного уровня для удаленного доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессорного взаимодействия.

Первая версия протокола была разработана компаниями IBM, Microsoft, Intel и 3Com в 1980-х годах. Изначально SMB был реализован через NetBIOS поверх NBF, IPX/SPX или NetBIOS over TCP/IP по порту 139. Он использовался в сетях MS-NET и LAN Manager для DOS, а также в Windows for Workgroups.

В Windows 2000 впервые стал применяться SMB поверх TCP без NetBIOS, используя порт 445. В Windows Vista появился SMB 2.0, который был значительно упрощен. Если в SMB было более 100 команд, то в SMB 2.0 их осталось всего 19. При этом была повышена производительность, улучшена масштабируемость и добавлена возможность автоматического продолжения сеанса в случае временного отсоединения от сервера.

В 1996 году Microsoft, дополнив используемый в Windows NT 4.0 протокол, назвало его — CIFS (Common Internet File System). Новое имя прижилось, и теперь SMB и CIFS стали синонимами.

Протокол SMB основан на технологии клиент-сервер и соответствует прикладному и представительному уровням модели OSI. Он регламентирует взаимодействие рабочей станции с сервером и в его функции входят следующие операции:

- Управление сессиями. Создание и разрыв логического канала между рабочей станцией и сетевыми ресурсами файлового сервера.

- Файловый доступ. Рабочая станция может обратиться к файл-серверу с запросами на создание и удаление каталогов, создание, открытие и закрытие файлов, чтение и запись в файлы, переименование и удаление файлов, поиск файлов, получение и установку файловых атрибутов, блокирование записей.
- Сервис печати. Рабочая станция может ставить файлы в очередь для печати на сервере и получать информацию об очереди печати.
- Сервис сообщений. Протокол SMB поддерживает простую передачу сообщений со следующими функциями: послать простое сообщение, послать широковещательное сообщение, послать начало блока сообщений, послать текст блока сообщений, послать конец блока сообщений, переслать имя пользователя, отменить пересылку, получить имя машины.

Подводя некоторый итог, следует отметить, что в настоящее время протокол SMB связан главным образом с операционными системами Microsoft Windows, где используется для реализации «Сети Microsoft Windows» (Microsoft Windows Network) и «Совместного использования файлов и принтеров» (File and Printer Sharing).

В 1992 году появилась Samba — свободная реализация протокола SMB для UNIX-подобных операционных систем. Так как Microsoft не опубликовала документацию значительной части своих дополнений к SMB, то разработчикам Samba пришлось провести обратную разработку протокола.

Сейчас Samba работает на большинстве Unix-подобных систем, таких как Linux, POSIX-совместимых Solaris и Mac OS X Server, а также на различных вариантах BSD. В операционную систему OS/2 портирован Samba-клиент, являющийся плагином к виртуальной файловой системе NetDrive. Samba включена практически во все дистрибутивы Linux, в том числе и в Ubuntu.

И вот теперь мы подходим к ответу на именно те два вопроса, которые поставили себе в предыдущем разделе. То есть, чтобы получить доступ к общесистемным Windows-ресурсам, нам в Ubuntu необходима поддержка протокола SMB.

Войдите через основное меню в «Центр приложений Ubuntu» и, выбрав «Установленные приложения», выполните поиск чего-либо, начинающего с smb. После окончания поиска и нажатия кнопки «Подробнее», вид вашего экрана будет похож на рис. 9.5.

То есть при начальной генерации Ubuntu был установлен клиент доступа к серверам SMB/CIFS, которыми и являются наши Windiws-машины. Но если внимательно прочесть описание к smbclient, то становится очевидным, что это консольная утилита клиента SMB/CIFS.

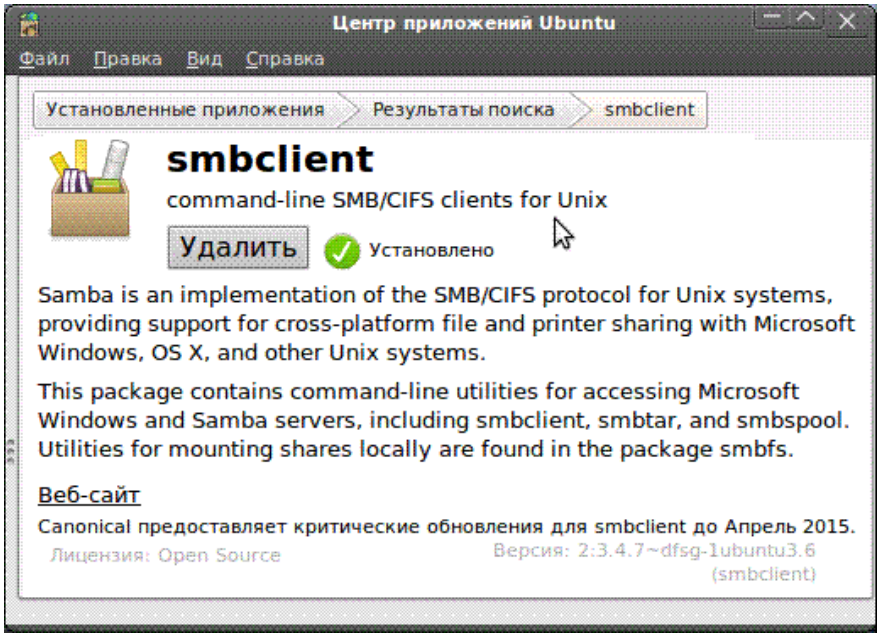

Рис. 9.5. Результаты поиска smbclient в установленных приложениях Ubuntu.

Стало быть, за отображение сетевых ресурсов в графической среде Nautilus она напрямую отвечать не может. А коль так, то ответа на первый из поставленных перед собой вопросов мы пока ответа не получили.

Если в Менеджере Пакетов Synaptic выполнить аналогичный быстрый поиск подстроки smb или samba, то можно увидеть список из достаточного большого числа пакетов, среди которых будут:

- smbclient (консольная утилита SMB/CIFS-клиента для Unix);
- samba (служба файлового обмена, печати и регистрации в системе, работающая по протоколу SMB/CIFS);
- smbfs (утилиты для файловой системы Samba);
- samba-tools (дополнительные консольные утилиты, такие как smbtorture для стресс-тестирования серверов и клиентов CIFS);
- samba-common (в этом пакете содержатся общие файлы, которые используются в Samba 3 и Samba 4) ;
- samba-comon-bin (аналогично предыдущему);
- samba-doc-pdf (документация в формате PDF);
- samba-doc (документация по Samba);
- samba-dbg (этот пакет предоставляет информацию об отладке программного обеспечения в Samba, его устанавливают, чтобы улучшить трассировку в случае сбоев);
- libwbclient0 (библиотека для написания клиентов для обмена данными с сервером Samba по протоколу winbind через программный канал);
- libwbclient-dev (расширение возможностей предыдущей библиотеки);
- winbind (сервис поддержки информации о пользователях и группах серверов Windows NT).

Существует множество других, особенно модных в последнее время графических пакетов для работы в протокольном стеке SMB поверх TCP/IP для операционной системы Linux. Но именно первые три из них являются теми пакетами и утилитами, которые наиболее часто используются в Linux-системах для поддержки протокола SMB и, как следствие, возможности взаимодействия Ubuntu- и Windows-машин.

9.1.2. Основные пакеты поддержки SMB /CIFS в Ubuntu 10.04 LTS

Samba — реализация протокола SMB/CIFS для систем Unix, которая обеспечивает совместное использование принтеров и обмен файлами для сетей с операционными системами Microsoft Windows, OS X и Unix.

➢ **smbfs —** утилиты для файловой системы Samba.

Deb-пакет: smbfs_3.4.7~dfsg-1ubuntu3.6_i386.deb

Данный пакет предоставляет утилиты для монтирования и размонтирования сетевой файловой системы cifs. Основа пакета smbfs — утилита smbmount, которая используется для монтирования совместно используемых ресурсов SMB в системе Linux. Она обеспечивает непосредственное монтирование удаленных SMB-ресурсов в Linux точно так же, как тома NFS монтируются в Linux.

Как и в команде mount, точка монтирования mountpoint представлена существующим каталогом в локальной системе, который должен быть пустым. В спецификации разделяемых SMB-ресурсов используются прямые косые черточки (/), вместо принятых в Linux обратных (\).

Следует обратить внимание, что smbmount не использует NetBIOS для поиска имени сервера. Если имя сервера SMB отличается от имени TCP/IP сервера, команда smbmount работать не будет. В такой ситуации используется хост-имя Unix для сервера.

➢ **smbclient —** консольная утилита SMB/CIFS-клиента для Unix.

Deb-пакет: smbclient_3.4.7~dfsg-1ubuntu3.6_i386.deb

Этот пакет содержит консольные утилиты для доступа к Microsoft Windows и Samba серверам. Включает в себя smbclient, smbtar и smbspool. Утилиты для монтирования общедоступных сетевых ресурсов на локальный узел находятся в пакете smbfs.

➢ s**amba —** служба файлового обмена, печати и регистрации в системе, работающая по протоколу SMB/CIFS.

Deb-пакет: samba_3.4.7~dfsg-1ubuntu3.6_i386.deb

Комплект ПО Samba — это набор программ, которые реализуют протокол SMB/CIFS в системах Unix, позволяя обслуживать запросы к файлам и принтерам клиентов Microsoft Windows, OS X и других систем Unix. Также Samba может выполнять функции контроллера домена NT4, и

интегрироваться в домены NT4 и области Active Directory в качестве сервера.

По своей сути Samba является межплатформенной сетевой файловой системой. В этом пакете есть все компоненты, необходимые для того, чтобы использовать Samba в качестве автономного файлового сервера или сервера печати. Для работы в домене NT4 или области Active Directory также потребуется установить пакет winbind. Этот пакет не нужен для подключения к имеющимся серверам SMB/CIFS (см. smbclie nt) или для монтирования удалённых SMB-ресурсов (см. smbfs).

Установить любой из перечисленных пакетов можно при наличии доступа в Интернет, либо из «Центр приложений Ubuntu», либо из «Менеджер Пакетов Synaptic», либо консольной командой:

```
sudo apt-get install [имя-пакета]
```

А в том случае, когда доступ в Интернет отсутствует, но у вас есть в наличии соответствующий deb-пакет, то можно использовать либо «Установщик программ GDebi», либо утилиту dpkg:

```
sudo dpkg -i [полный-путь-к-пакету-*.deb]
```

Однако на одном из серверов, который является хранилищем пакетов, дана следующая рекомендация: «Если вы работаете в Ubuntu, для загрузки и установки пакетов настоятельно советуем использовать менеджер пакетов, например aptitude или synaptic, а не делать это вручную через данный сайт». Используйте любой из серверов-зеркал, добавив его в свой файл /etc/apt/sources.list. Например, так: deb http://security.ubuntu.com/ubuntu lucid-security main.

Но мы с вами пока ничего устанавливать не будем и обойдемся тем, что уже есть, а это только smbclient. О возможности его использования мы поговорим отдельно, но несколько позднее. А сейчас мы можем ответить только на второй из поставленных перед собой вопросов, а именно: «Возможности по публикации общедоступных ресурсов Ubuntu в SMB-сети отсутствуют, так как не установлен пакет samba».

А как же быть с первым вопросом? Здесь все значительно сложнее и требует более глубокого подхода, схематичный путь к которому, мы постараемся, в меру сил кратко изложить.

9.1.3. Виртуальная файловая система в пользовательском пространстве — GVFS

Давайте вернемся к самому началу этой главы, когда мы рассматривали сетевой доступ. Обратите внимание, какой бы из режимов меню мы ни выбирали: «Сеть», «Соединение с сервером» и т. д., в этих действиях в той или иной мере участвовал Nautilus.

Nautilus — официальный файловый менеджер для рабочей среды GNOME. Он заменил Midnight Commander в GNOME 1.4 и стал, начиная с версии 2.0, файловым менеджером по умолчанию. На данный момент нас мало интересует, что он поддерживает закладки, фон окон, эмблемы, скрипты дополнений, просмотр текстовых файлов, изображений, звуковых или видео файлов. Больше интересует то, что он позволяет просматривать FTP-сайты, «расшаренные» Windows SMB-ресурсы, файловые системы мобильных телефонов по протоколу ObexFTP, а также HTTP-, WebDAV- и SFTP-сервера как локальные файловые системы.

Что же позволяет Nautilus выполнять такие функции? В более ранних версиях Ubuntu, в частности и во включенной в состав нашей виртуальной сети Ubuntu 6.10, это была среда GnomeVFS.

GnomeVFS — это виртуальная файловая система, которая являлась основой для файлового менеджера Nautilus. Она имеет модульную архитектуру и поставляется с различными модулями, реализующими поддержку файловых систем, http, ftp и других. Виртуальная файловая система обеспечивает программный интерфейс на основе URI, имеющий встроенную поддержку асинхронных файловых операций, библиотеку управления MIME-типами и другие возможности.

В последних версиях, к которым относится и рассматриваемая операционная система Ubuntu 10.04, Nautilus уже использует GVFS.

GVFS — виртуальная файловая система, созданная как альтернатива для GnomeVFS. Ее особенность в том, что она в отличие от GnomeVFS создает виртуальную файловую систему без создания пользовательского процесса. GVFS поддерживает различные интерфейсы, включая HAL-интеграцию, SFTP, WebDAV, SMB, ObexFTP, а также монтирование архивов (через libarchive).

В свою очередь GVFS использует библиотеку GIO и модуль GVFS-Fuse, с которыми взаимодействует по D-Bus:

• FUSE (Filesystem in USErspace — «файловая система в пользовательском пространстве») — это модуль для ядер Unix-подобных ОС, с открытым исходным кодом. Он позволяет пользователям без привилегий создавать их собственные виртуальные файловые системы без необходимости переписывать код ядра.

В отличие от традиционных, виртуальные операционные системы не хранят данные непосредственно. Они действуют как представление, трансляция существующей файловой системы или устройства хранения. В принципе, любой ресурс, доступный для использования FUSE, может быть экспортирован в файловую систему.

• GIO — это библиотека ввода/вывода с поддержкой GVFS. Она предоставляет высокоуровневый интерфейс для файлового ввода/вывода и

типов файлов, призванный заменить интерфейс, предоставляемый POSIX, а также интерфейс GnomeVFS.

• D-Bus — система межпроцессного взаимодействия, которая позволяет приложениям в операционной системе общаться друг с другом. Обладает высокой скоростью работы, не зависит от рабочей среды, есть версия для Windows и высокоуровневые библиотеки для фреймворков Qt, Java, GLib, C#, Python и библиотека для C++.

GVFS позволяет по желанию подключать виртуальные файловые системы, монтируя их через FUSE. Существует ряд программ командной строки, которые начинаются с «gvfs-», что позволяет выполнять такие команды, как cat, ls, stat, mount и т.д. над файлами примонтированными GVFS.

Используя библиотеку GIO, Nautilus отслеживает изменения локальных файлов в режиме реального времени, устраняя потребность вручную обновлять экран. Nautilus сохраняет историю посещенных папок, подобно многим веб-браузерам предоставляя простой доступ к ранее посещенным сетевым папкам.

И вот тут возникает вполне резонный вопрос: «Неужели в моей Ubuntu вся эта виртуализация реализована?». Чтобы это проверить и удовлетворить чисто человеческое любопытство, давайте зайдем в Центр приложений Ubuntu, установим режим «Установленные приложения» и выполним поиск подстроки gvfs. Результат будет удивительным, все о чем говорилось чуть выше, налицо (рис. 9.6). Осталось научиться это все использовать.

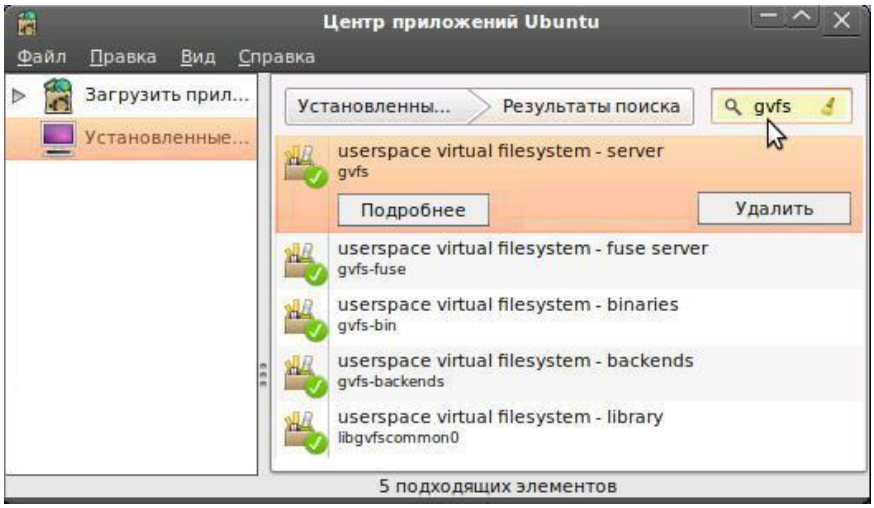

Рис. 9.6. Результаты поиска пакетов поддержки gvfs в установленных приложениях Ubuntu.

И вот здесь мы подошли к тому, чтобы наконец-то дать ответ на первый вопрос, поставленный в начале этой главы. Ответ на него будет такой: «Доступ и отображение удаленных общедоступных сетевых ресурсов

осуществляется с помощью виртуальной файловой системы GVFS, которая является основой для файлового менеджера Nautilus».

9.1.4. Основные пакеты поддержки GVFS в Ubuntu 10.04 LTS

GVFS — это виртуальная файловая система в пользовательском пространстве, в которой монтирование происходит как запуск отдельных приложений по команде D-Bus. В пакете также содержится gio модуль для добавления поддержки gvfs во все приложения, использующие gio API. Также поддерживается монтирование не gio приложений через fuse.

К основным пакетам поддержки виртуальной файловой системы в пользовательском пространстве относятся:

- **gvfs —** сервер gvfs, который предоставляет возможность монтирования всем gio приложениям, и минимальный набор утилит.

Deb-пакет: gvfs_1.6.0+git20100414-0ubuntu1_i386.deb.

Зависимости: libc6, libdbus-1-3, libgconf2-4, libgdu0, libglib2.0-0, libgvfscommon0, libudev0, policykit-1-gnome, dbus, gvfs-backends.

- **gvfs-fuse —** gvfs-fuse сервер, который экспортирует gvfs-монтировки на все приложения, использующие FUSE.

Deb-пакет: gvfs-fuse_1.6.0+git20100414-0ubuntu1_i386.deb.

Зависимости: fuse-utils, gvfs, libc6, libdbus-1-3, libfuse2, libglib2.0-0, libgvfscommon0.

- **gvfs-bin —** пакет программ поддержки.

Deb-пакет: gvfs-bin_1.6.0+git20100414-0ubuntu1_i386.deb.

Зависимости: gvfs, libc6, libglib2.0-0.

- **gvfs-backends —** обеспечивает поддержку afc, archive, burn, cdda, dav, dnssd, ftp, gphoto2, http, network, obexftp, sftp, smb and smb-browse.

Deb-пакет: gvfs-backends_1.6.0+git20100414-0ubuntu1_i386.deb.

Зависимости: gvfs, libarchive1, libavahi-client3, libavahi-common3, libavahi-glib1, libbluetooth3, libc6, libcdio-cdda0, libcdio-paranoia0, libcdio10, libdbus-1-3, libdbus-glib-1-2, libexpat1, libgconf2-4, libglib2.0-0, libgnome-keyring0, libgphoto2-2, libgphoto2-port0, libgudev-1.0-0, libgvfscommon0, libimobiledevice0 (Library for communicating with the iPhone and iPod Touch), libsmbclient, libsoup-gnome2.4-1, libsoup2.4-1, libxml2.

- **libgvfscommon0 —** пакет библиотек, используемых средствами протокольной поддержки.

Deb-пакет: libgvfscommon0_1.6.0+git20100414-0ubuntu1_i386.deb.

Зависимости: libavahi-client3, libavahi-common3, libavahi-glib1, libc6, libdbus-1-3, libexpat1, libglib2.0-0.

➢ **libgvfscommon-dev —** файлы проектов, необходимые для создания новых классов, использующих те же функции.

Deb-пакет: libgvfscommon-dev_1.6.0+git20100414-0ubuntu1_i386.deb.

Зависимости: libglib2.0-dev, libgvfscommon0.

➢ **gvfs-dbg —** пакет системы отладки gvfs.

Deb-пакет: 1.9.2-0ubuntu2_i386.

9.2. Сетевой доступ между Ubuntu-машинами

Перед тем как рассматривать межсистемное сетевое взаимодействие, давайте коротко остановимся на вопросе о доступе с одной Ubuntu-машины на другую. Рассматривая в п. 7.3 удаленный доступ по протоколу SSH, мы говорили, что для доступа между Ubuntu-машинами можно использовать консольную команду вида:

```
ssh [опции] <пользователь>@<адрес_удаленного_хоста>
```

Естественно, при условии, что пользователь является легальным пользователем удаленного компьютера, и известен пароль на доступ к этому компьютере. Напомню, что в нашей виртуальной сети имеются две виртуальные Ubuntu-машины. Попробуем осуществить удаленный доступ с vmUbuntu10 на vmUbuntu06, так как пользователь с именем serp имеет административные права на обоих машинах.

Для этого войдем в терминал vmUbuntu10. Терминал может быть как локальным, так и удаленным. Последнее возможно, если предварительно выполнено подключение к vmUbuntu10. Например, с помощью утилиты PuTTY с основного компьютера. Находясь в терминале vmUbuntu10, введем команду на доступ к vmUbuntu06:

```
serp@vmUbuntu10:~$ ssh serp@192.168.1.6
serp@192.168.1.6's password:
Linux vmUbuntu06 2.6.17-10-generic #2 SMP Fri Oct 13
18:45:35 UTC 2006 i686


 . . .
Last login: Fri Jul 29 08:53:05 2011 from 192.168.1.2
serp@vmUbuntu06:~$
```

Из приведенного листинга видно, что после того, как в ответ на запрос был введен истинный пароль пользователя, он получил доступ на удаленную машину.

Обратите внимание, как изменилась подсказка в командной строке. Мы находимся в домашней директории serp компьютера vmUbuntu06. И теперь можно выполнять на этом компьютере любые действия в рамках

полномочий, присвоенных пользователю serp на vmUbuntu06. Но это все справедливо для терминального режима, а как быть с графическим режимом. Ведь используя Nautilus в режиме просмотра сети, мы видели, что Nautilus не обнаружил компьютер vmUbuntu06, который не является smb-сервером.

И тут нам снова надо обратиться к протокольным стекам и вспомнить что такое SSH. Это для нас важно, так как SSHFS (Secure Shell File System) — это сетевая файловая система Linux, используемая для удаленного управления файлами по протоколу SSH (точнее, его расширению SFTP) таким образом, как будто они находятся на локальном компьютере.

Эту файловую систему может использовать не только Linux, но и любая другая операционная система, если для нее существует реализация FUSE. Используя в сети SSHFS, администратор может настроить ограниченный аккаунт на сервере для обеспечения большей безопасности, и пользователи смогут видеть только выделенную им область в системе.

Давайте воспользуемся этим подходом для доступа с vmUbuntu10 на vmUbuntu06. С этой целью в основном меню выберем Переход - > Соединиться с сервером. После этого появится диалоговое окно Соединение с сервером (рис. 9.7).

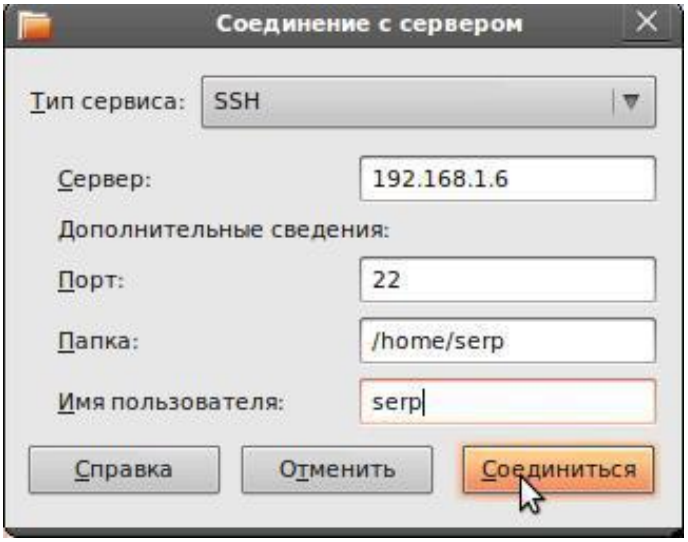

Рис. 9.7. Подключение к домашней директории vmUbuntu06 по протоколу SSH.

В этом окне в качестве типа сервиса выбираем SSH и настраиваем соответствующий порт. Если на удаленном узле порт SSH не был изменен, то по умолчанию для этого протокола установлен порт 22. Если на удаленном узле порт был изменен, то следует указать его новое значение.

После этого следует указать адрес сервера, имя пользователя и имя ресурса, доступного для этого пользователя, и нажать кнопку Соединиться. Открывается окно файлового менеджера с содержимым удаленной папки (рис. 9.8).

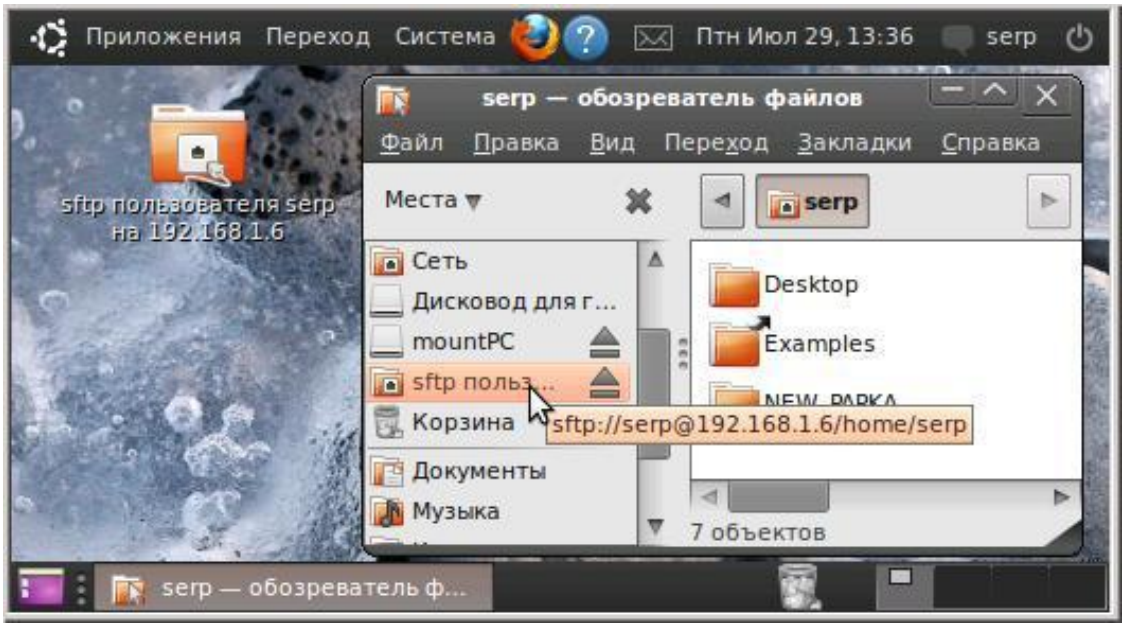

Рис. 9.8. Экран vmUbuntu10 после подключения в домашнюю директорию serp компьютера vmUbuntu06.

При этом одновременно формируются ярлык на рабочем столе и новые дополнительные опции в меню Места и меню Переход, которые ссылаются на удаленный ресурс. Щелкнув мышкой по тому или другому, мы тут же попадем в папку удаленного компьютера. Если, конечно, к этому моменту он еще не выключен или на нем не изменена политика доступа.

Обратите особое внимание на URI (URI=URL+URN) удаленного ресурса. В качестве протокола указан sftp, именно на его основе виртуальная файловая система обеспечивает программный интерфейс к этому ресурсу, а его монтирование осуществляется gvfs-fuse сервером на основе полного URI этого ресурса.

Именно возможность gvfs осуществлять монтирование через fuse и обеспечило нам возможность не только видеть на экране ярлык и опции меню со ссылкой на ресурс, но и при необходимости получить доступ к этому ресурсу.

Если кого-то и поставила в тупик последняя фраза, то хотелось бы напомнить, что основная цель данной книги — знакомство с сетевыми технологиями и протекающими при этом процессами, а не привитие навыков кликанья мышкой в нужном месте и в нужное время.

Хотя нельзя отрицать того факта, что это тоже очень важно и в ряде случаев дает определенный результат. Но давайте знакомство с процессами и технологиями оставим на конец главы, а сейчас продолжим кликать мышкой. При этом желательно, чтобы не оставалась в стороне суть выполняемых действий.

9.3. Доступ к общесистемным Windows-ресурсам из графической среды Ubuntu

Прежде всего, надо отметить достаточно широкий набор действий, которые доступны пользователю из файлового менеджера Nautilus по работе с общесистемными ресурсами. Но, так как работа в нем существенно не отличается от работы в «Сетевом окружении» Windows, то оставим это вам на самостоятельную практику.

Хотелось бы только обратить ваше внимание на одну возможность, которая может быть очень полезна на практике. Особенно если вам необходимо часто обращаться к какому-то конкретному общему ресурсу. Если, например, вам приходится часто обращаться к папке vmPC_share на основном компьютере, то вы можете поступить следующем образом:

- в файловом менеджере выбрать Сеть, а затем, последовательно раскрывая окна, дойти до нужной общедоступной папки;
- открыв правой кнопкой мышки всплывающее меню, выбрать опцию Подключить;
- при запросе о доступе ввести пароль на доступ к этому ресурсу.

Если аутентификация пройдет успешно, то вам откроется искомая папка и одновременно с этим на рабочем столе появится ярлык для доступа к этой папке и дополнительно к этому появится новая опция как в меню файлового менеджера, так и в основном меню Переход (рис. 9.9).

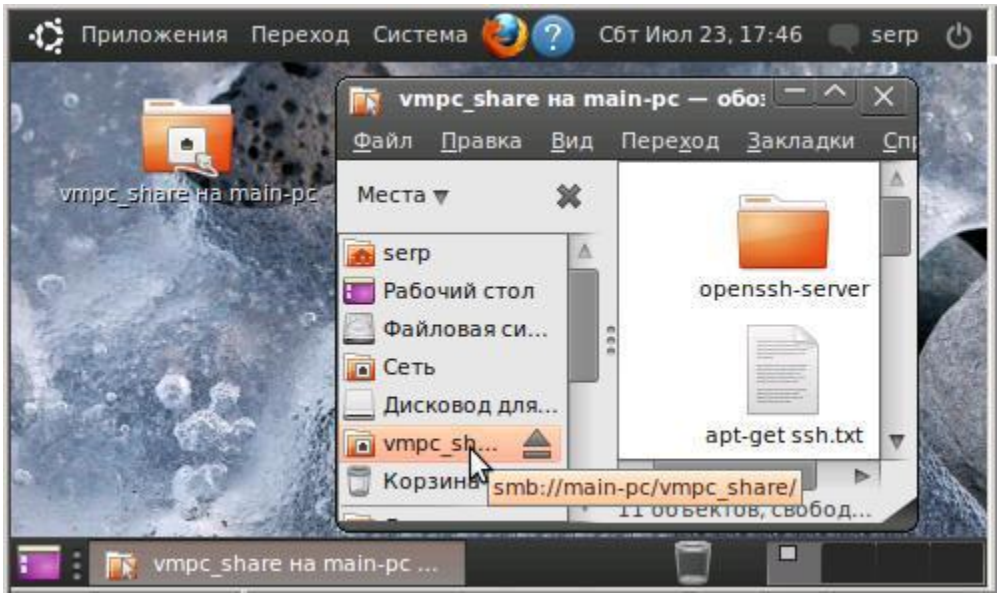

Рис. 9.9. Ярлык и новая опция меню файлового менеджера на доступ к папке vmPC_share на компьютере Main-PC.

Обратите внимание на всплывающую подсказку, которая выводится при наведении курсора на опцию меню, соответствующую подключенному URI ресурсу, в составе него указан протокол по которому осуществляется связь с этим ресурсом. И, как ни странно, этот протокол – smb, сетевой

протокол прикладного уровня для удаленного доступа к файлам, принтерам и другим сетевым ресурсам. Протокол, который является стандартом межсетевого доступа в сетях Microsoft, хотя мы осуществляем подключение к общесетевым ресурсам из Ubuntu.

Еще одним способом подключения удаленного общесетевого ресурса является выбор в основном меню «Переход» опции «Соединиться с сервером...» (рис. 9.10). Но какая связь между соединением с сервером и доступом к общедоступной папке?

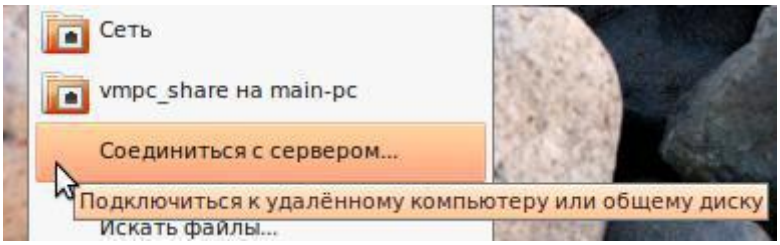

Рис. 9.10. Опция «Соединиться с сервером...» основного меню Переход.

На следующем примере покажем, что наши виртуальные Windows-машины являются полноценными SMB-серверами, которые поддерживаются Microsoft Windows Network. С этой целью выполним подключение папки vm98_share, находящейся на SMB-сервере vm-Win98, который в свою очередь входит в рабочую группу Virtual-Net нашей виртуальной сети (рис. 9.11).

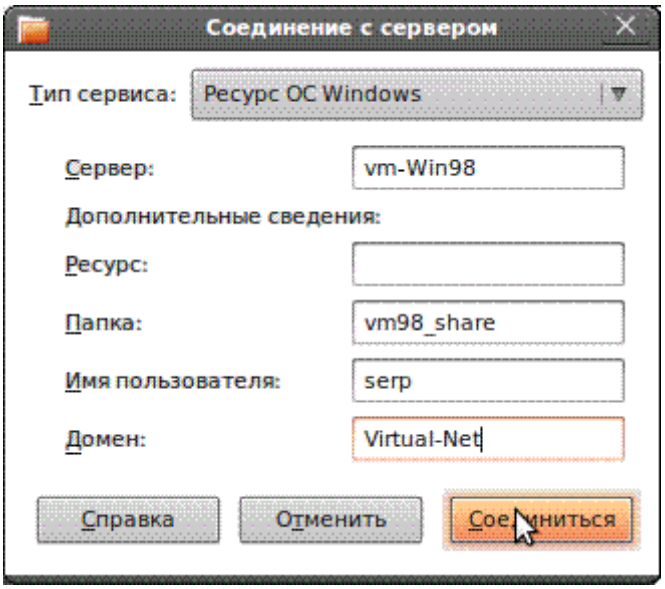

Рис. 9.11. Окно «Соединение с сервером».

Если аутентификация пройдет успешно, то откроется искомая папка, на рабочем столе появится новый ярлык, а в меню файлового менеджера и основном меню «Переход» новая опция со ссылкой на доступный общесетевой ресурс (рис. 9.12).

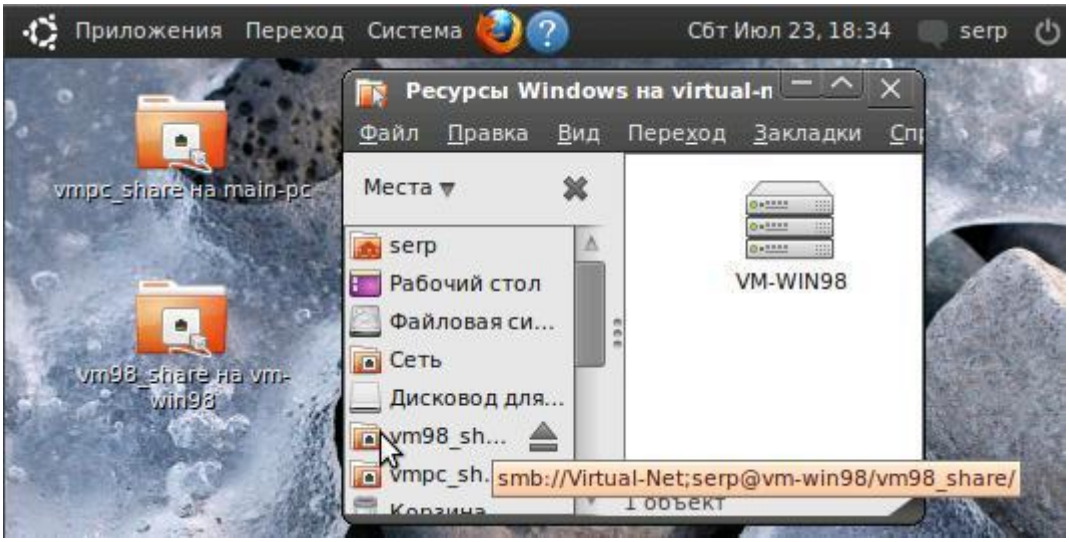

Рис. 9.12. Рабочий стол и окно файлового менеджера после подключения двух ресурсов.

Итак, у нас к Ubuntu-машине подключены общедоступные папки двух Windows-машин. Теперь мы можем выполнять доступные нам файловые операции между папками трех узлов нашей виртуальной сети. Однако ходить из папки в папку, кликать мышкой «копировать», а затем «вставить» — занятие не из приятных. И здесь, спасибо разработчикам, нам на помощь приходит Nautilus со своей волшебной клавишей F3. Окно файлового менеджера отображает общедоступные папки сразу двух удаленных Windows-машин с возможностью простого перетаскивания файлов из одной панельки окна в другую (рис. 9.13).

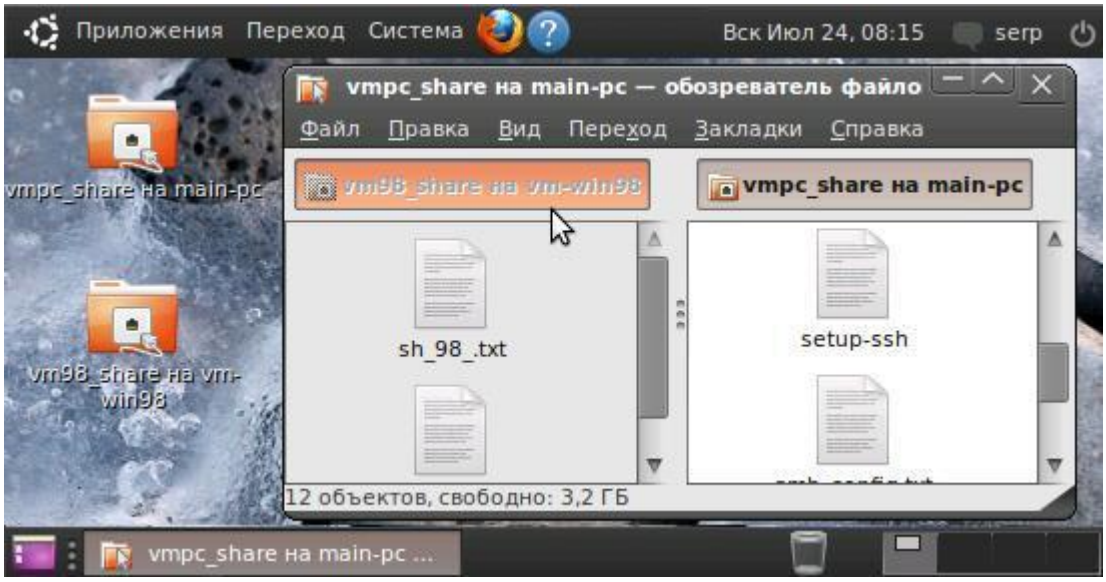

Рис. 9.13. Доступ к двум Windows-машинам с возможностью выполнения файловых операций.

Следует отметить, что никаких сложностей у нас не должно возникнуть, если внутри двух панелей Nautilus мы объединим удаленные

общедоступные ресурсы двух различных операционных систем. Например, мы можем, работая на vmUbuntu10, как пользовать serp, просматривать доступные этому пользователю ресурсы и на компьютере Main-PC и на виртуальной машине vmUbuntu06 (рис. 9.14).

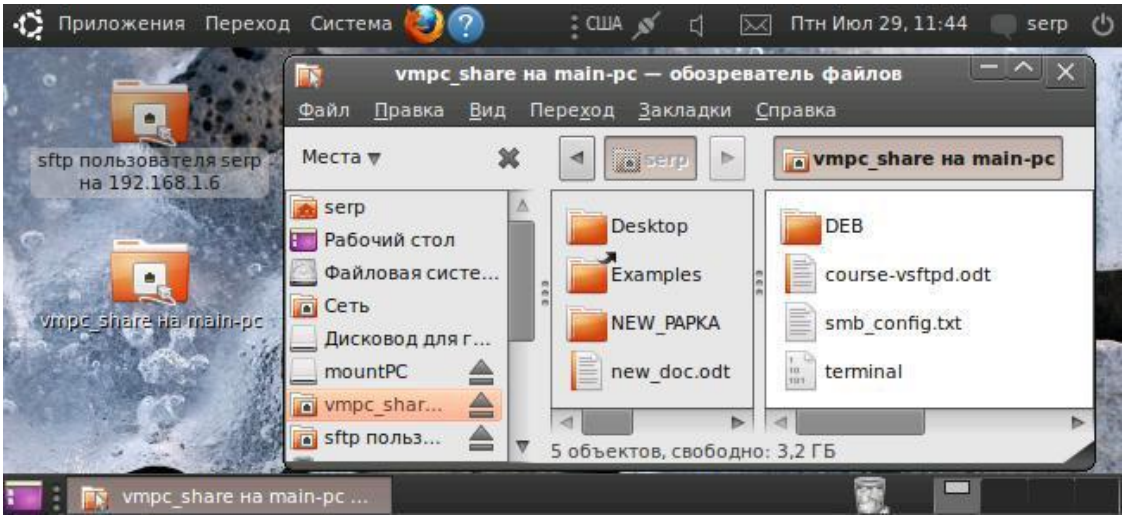

Рис. 9.14. Доступ к двум удаленным общедоступным ресурсам на Main-PC и vmUbuntu06.

Обратите внимание, что в разных панелях у вас ресурсы разных операционных систем с различной организацией файловых систем. Да что там различная организация файловых систем, если даже текстовые файлы и те могут иметь различную кодировку.

Но это не мешает вам выполнять файловые операции по созданию, перемещению, просмотру папок и файлов. И все это благодаря виртуальной файловой системе в пользовательском пространстве gvfs-fuse, работающей на вашем компьютере.

Отдельный интерес, но не очень связанный с рассмотрением взаимодействия Ubuntu-Windows машин, представляет собой подключение к удаленным серверным ресурсам. В том числе к iPhone, iPad или к ресурсам Интернет.

В частности, используя Переход -> Соединение с сервером..., можно без использования браузера, но при наличии доступа в Интернет, легко подключиться, например, к анонимному FTP-серверу российского зеркала Ubuntu, где представлены все версии этой операционной системы (рис. 9.15).

И опять-таки, хотелось бы акцентировать ваше внимание на всплывающей подсказке (URI), которая выводится при наведении курсора на опцию меню, соответствующую этому подключенному ресурсу.

У этого ресурса будет указан совсем другой протокол, по которому осуществляется связь с ним – этот протокол ftp. То есть в обозревателе файлов мы одновременно можем просматривать совсем разные типы сетевых ресурсов, организованных на различных сетевых технологиях.

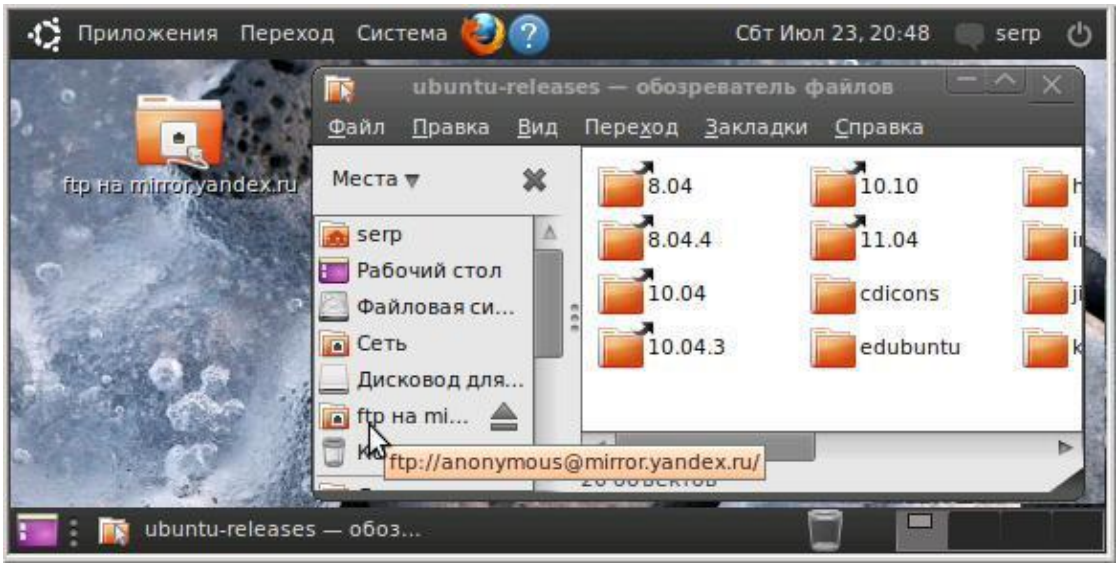

Рис. 9.15. Окно с ресурсом ubuntu-releases на сервере mirror.yandex.ru.

Но несмотря на все красоту и удобство графических утилит и оболочек, значительно большей функциональностью при профессиональной работе обладают консольные приложения.

Действительно, если перед вами стоит задача на два десятка компьютеров переписать по десятку файлов, да еще распределяя их по соответствующим папкам, то вы наверняка достаточно быстро устанете открывать окошки и тыкать то левой, то правой кнопкой мышкой. В конце концов, вы начнете ошибаться. А если такую работу вам необходимо делать несколько раз в неделю? А если компьютер к тому же удаленный?

9.4. Доступ к разделяемым ресурсам сети с использованием smbclient

При использовании Ubuntu, у нас есть возможность обеспечить SMB-доступ к разделяемым файлам и принтерам. Такой доступ можно осуществить несколькими способами. Самым простым из них является использование консольной утилиты smbclient, входящей в состав пакета Samba. Кстати, эта утилита, как мы выяснили ранее, уже установлена на нашем компьютере при начальной инсталляции Ubuntu.

У этого способа есть некоторые ограничения, особенно по доступу к файлам. Связано это с тем, что smbclient обеспечивает FTP-подобный доступ к совместно используемым удаленным файлам. При таком подходе, работая с файлами, нельзя использовать обычные команды Linux. Такие, как cp и другие. Доступ к совместным ресурсам из других приложений также ограничен.

Утилита smbclient используется для перемещения файлов, с разделяемых ресурсов SMB-сервера и обратно посредством FTP-подобного интерфейса, что требует до выполнения каких-либо файловых операций

установления соединения с удаленным SMB-сервером. Для этого надо использовать команду, которая в простейшем случае будет иметь вид:

```
smbclient //server-name-or-IP-address/resource-name
```

Конечно, в действительности все немного сложнее. Если необходимо указать имя конкретного пользователя и его пароль для получения доступа к защищенному ресурсу, то команда становится более громоздкой:

```
smbclient //host/resource -U user%password
```

В составе нашей виртуальной сети осталась еще одна неохваченная Windows-машина — это виртуальная машина vmWinXP (192.168.1.44/24).

Попробуем с использованием smbclient получить доступ к ее общедоступной папке vmXP_share и просмотреть содержимое этой папки.

```
serp@vmUbuntu10:~$ smbclient //192.168.1.44/vmXP_share -U
serp
Enter serp's password:
Domain=[VM_WINXP] OS=[Windows 5.1] Server=[Windows 2000]
```

```
smb: \> ls
  .                   D        0  Tue Jul 12 10:44:35 2011
  ..                  D        0  Tue Jul 12 10:44:35 2011
  test1.txt           A        9  Fri Jun 17 22:16:26 2011
  test2.txt           A     3765  Tue Jul 12 18:29:34 2011

      65522 blocks of size 1048576. 64276 blocks available
```

Из приведенного листинга видно, что в этой папке всего два файла, а для просмотра содержимого папки использовалась команда ls. Причем это команда не операционной системы Ubuntu, а аналогичная команда smbclient. Чтобы узнать список всех доступных команд этой утилиты, достаточно, находясь в ней, набрать команду help:

```
smb: \> help
?           allinfo        altname    archive    blocksize
cancel      case_sensitive cd         chmod      chown
close       del            dir        du         echo
exit        get            getfacl    hardlink   help
history     iosize         lcd        link       lock
lowercase   ls             l          mask       md
mget        mkdir          more       mput       newer
..          !
smb: \> exit
serp@vmUbuntu10:~$
```

Установив соединение с удаленным ресурсом, можно выполнять различные файловые операции, используя специальные команды из списка доступных в smbclient команд, который был перечислен выше при выводе по команде help. Рассмотрим назначение наиболее часто используемых команд:

Таблица 9.1

Основные команды утилиты smbclient

Команды	Описание
mkdir <directory>	Создание каталога на удаленном сервере (md)
rmdir <directory>	Удаление каталога с удаленного сервера (rd)
cd <directory>	Переход в другой каталог совместно используемого SMB-ресурса
dir,ls	Отображение содержимого текущего каталога сервера
get <file>	Получение указанного файла с удаленного сервера и сохранение его с тем же именем в текущем каталоге локальной системы. Можно задать другое имя для файла на локальной системе: get file localfilename
del <file>	Удаление указанного файла с сервера (rm)
lcd <directory>	Переход в указанный каталог на локальной системе
mget <filemask>	Получение всех файлов на удаленном сервере, удовлетворяющих указанной маске файла
prompt	Включение/выключение подсказки для операций с несколькими файлами (mput и mget). При задании значения on пользователи получают подсказку при копировании каждого файла
put <file>	Копирование указанного файла из текущего локального каталога в текущий каталог на удаленном сервере, имя файла остается прежним. Имя файла на удаленном сервере можно изменить: put file remote filename.
mput <filemask>	Копирование всех файлов локального каталога, удовлетворяющих указанной маске файла, в текущий каталог удаленного сервера.
recurse	Включает/выключает доступ к подкаталогам для операций с несколькими файлами (mput и mget). Когда задано значение on, команда при копировании файлов осуществляет поиск по всем подкаталогам текущего каталога.
quit, exit	Выход из программы smbclient

Утилита smbclient имеет несколько флагов, которые позволяют изменить характер ее соединения с SMB-серверами.

Таблица 9.2

Основные флаги утилиты smbclient

Флаг	Действие
-L <host>	Флаг выводит на экран список сервисов, доступных на сервере, заданном параметром host. При использовании этого флага нет необходимости указывать ресурс
-I <ip_address>	Флаг полезен, если не может быть найден адрес по имени. Утилита smbclient полагает, что компьютер расположен по указанному IP-адресу
-N	Флаг подавляет приглашение password. Особенно полезен, когда доступ к ресурсу осуществляется без пароля. Если этот флаг не установлен, а пароль не требуется, пользователь все же получает приглашение для ввода и должен нажимать клавишу Enter для ввода пустого пароля
-U <username>	Используя этот флаг, можно указать имя пользователя для установки соединения с ресурсом. Без этого флага сервер использует содержимое переменных среды USER или LOGNAME. Если они пустые, сервер не получает username. Отправить пароль к серверу можно, введя знак процента (%) после username, а затем введя пароль: -U username%password
-W <workgroup>	Определяет, какая рабочая группа используется при соединении с сервером
-T <tar-options>	Позволяет перемещать данные в tar-файл локальной системы Linux и обратно. Например, -Tx backup.tar восстанавливает файлы из backup.tar на удаленном ресурсе, в то время как -Tc backup.tar создает tar-файл с именем backup.tar, содержащий все файлы и каталоги удаленного ресурса
-c <list-cmd>	Передает строку list-cmd, состоящую из команд smbclient'a, разделенных точкой с запятой. Например, md new; cd new; put newfile; exit

Список всех доступных флагов утилиты достаточно большой и его можно получить, введя в командной строке

```
smbclient help   или  man smbclient help
```

Это позволит, достаточно подробно познакомиться с назначением флагов, а также с возможностями этой утилиты.

9.4.1. Пример работы с smbclient из командной строки

С целью более глубокого понимания происходящих процессов при работе с удаленными ресурсами, а также для знакомства с работой в среде утилиты smbclient рассмотрим небольшой пример. Он поближе познакомит вас с командами smbclient и практикой их использования.

Предположим, что на удаленной Windows-машине vm-WinXP (192.168.1.44/24) в ее общедоступной папке vmXP_share находятся два текстовых файла произвольного содержания с именами test1.txt и test2.txt (рис. 9.16).

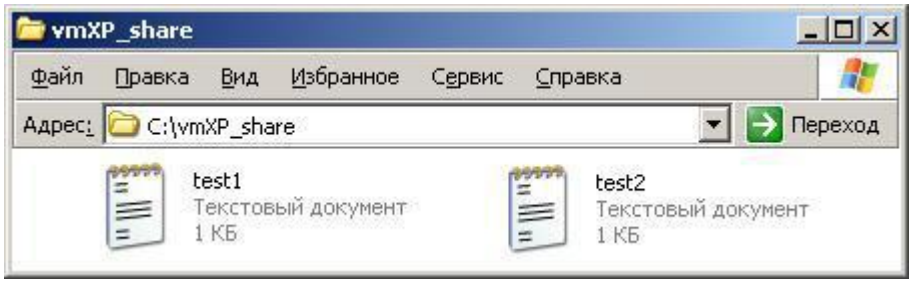

Рис. 9.16. Окно с ресурсом ubuntu-releases на сервере mirror.yandex.ru.

Находясь в текстовой консоли Ubuntu-машины, попробуем выполнить следующую последовательность действий:

- Создадим в рабочем директории Ubuntu-машины два новых каталога test1 и test2.
- В один из них, в каталог test1, скопируем файл test1.txt, изменив его имя на fromxp.txt.
- В другой, в каталог test2, скопируем все файлы с расширением *.txt из общедоступной папки виртуальной машины vm-WinXP.
- Находясь на Ubuntu-машине, создадим в папке vmXP_share новую папку с именем from_ubuntu.
- Во вновь созданную папку from_ubuntu скопируем из каталога test2 все текстовые файлы.
- В эту же папку, но из каталога test1, скопируем файл, не изменяя его имени.
- Повторим предыдущую операцию, изменив имя файла на новое, например, xp_ubuntu.txt.
- Проверим, как на Ubuntu-машине, так и Windows-машине наличие новых каталогов и содержание файлов в них.

Один из возможных вариантов последовательности выполнения этой задачи может иметь вид, который описан ниже.

Создаем на Ubuntu-машине в рабочей директории /home/serp два новых каталога:

```
serp@vmUbuntu10:~$ mkdir test1
serp@vmUbuntu10:~$ mkdir test2
```

Подключаемся к общедоступной папке удаленной Windows-машины vm-WinXP как к SMB-серверу:

```
serp@vmUbuntu10:~$ smbclient //192.168.1.44/vmXP_share -U
serp%serp
Domain=[VM_WINXP] OS=[Windows 5.1] Server=[Windows 2000
LAN Manager]
```

Устанавливаем на Ubuntu-машине текущим каталог test1:

```
smb: \> lcd /home/serp/test1
```

Получаем с vm-WinXP в текущий каталог Ubuntu-машины файл test1.txt под именем fromxp.txt:

```
smb: \> get test1.txt fromxp.txt
getting file \test1.txt of size 15 as fromxp.txt (0,5
KiloBytes/sec) (average 0,2 KiloBytes/sec)
```

Устанавливаем на Ubuntu-машине текущим каталог test2:

```
smb: \> lcd /home/serp/test2
```

Получаем из общедоступной папки vmXP_share в текущей каталог Ubuntu-машины все файлы с расширением *.txt:

```
smb: \> mget *.txt
Get file test1.txt? y
getting file \test1.txt of size 15 as test1.txt (0,3
KiloBytes/sec) (average 14,4 KiloBytes/sec)
Get file test2.txt? y
getting file \test2.txt of size 17 as test2.txt (0,6
KiloBytes/sec) (average 13,1 KiloBytes/sec)
```

Находясь на Ubuntu-машине, создаем в общедоступной папке vmXP_share новую папку from_ubuntu:

```
smb: \> mkdir from_ubuntu
smb: \> cd from_ubuntu
```

Копируем из текущего каталога Ubuntu-машине в общедоступную папку vmXP_share все текстовые файлы:

```
smb: \from_ubuntu\> mput *.txt
Put file test1.txt? y
putting file test1.txt as \from_ubuntu\test1.txt (0,2
kb/s) (average 0,2 kb/s)
Put file test2.txt? y
putting file test2.txt as \from_ubuntu\test2.txt (0,4
kb/s) (average 17,3 kb/s)
```

Устанавливаем на Ubuntu-машине текущим каталог test1:

```
smb: \from_ubuntu\> lcd /home/serp/test1
```

Копируем 2 раза файл fromxp.txt из текущего каталога в папку vmXP_share (без изменения и с изменением его имени) :

```
smb: \from_ubuntu\> put fromxp.txt
```

```
putting file fromxp.txt as \from_ubuntu\fromxp.txt (0,2
kb/s) (average 0,2 kb/s)
smb: \from_ubuntu\> put fromxp.txt xp_ubuntu.txt
putting file fromxp.txt as \from_ubuntu\xp_ubuntu.txt (0,0
kb/s) (average 0,1 kb/s)
```

Выходим из smbclient:

```
smb: \from_ubuntu\> exit
```

Проверяем наличие и содержимое двух папок на Ubuntu-машине :

```
serp@vmUbuntu10:~$ cd test1
serp@vmUbuntu10:~/test1$ ls
fromxp.txt

serp@vmUbuntu10:~/test1$ cd /home/serp/test2
serp@vmUbuntu10:~/test2$ ls
test1.txt  test2.txt
```

Для того чтобы убедиться, что все операции были выполнены правильно, выполним просмотр содержимого общедоступной папки vmXP_share, находящейся на Windows-машине vm-WinXP (рис. 9.17).

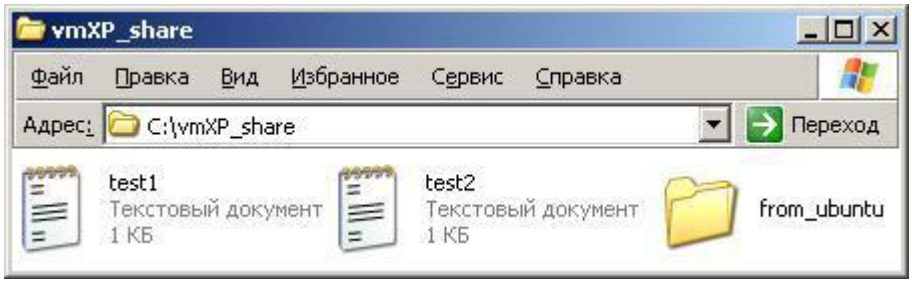

Рис. 9.17. Содержимое папки vmXP_share на машине vm-WinXP.

Из этого рисунка видно, что, работая на Ubuntu-машине с SMB-клиентом и используя доступ по протоколу SMB к vm-WinXP, мы действительно создали новую папку from_ubuntu.

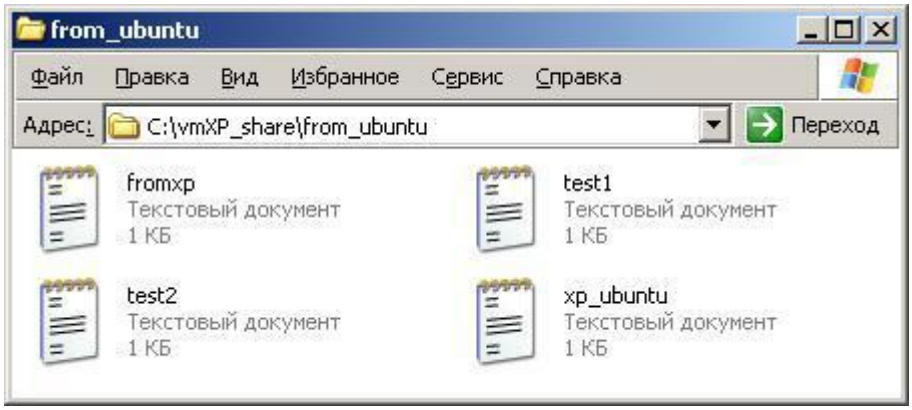

Рис. 9.18. Содержимое папки from_ubuntu на машине vm-WinXP.

Войдем во вновь созданную папку. Можно видеть, что все файлы, которые были получены с этого компьютера на удаленную Ubuntu-машину и с которыми там были произведены необходимые изменения (мы их не производили, а только переименовали) находятся снова на этом компьютере, причем уже в новой папке (рис. 9.18).

Опять вижим недоумение в ваших глазах. Зачем все так сложно, если в Nautilus это все можно сделать за пять минут.

Чтобы у вас не было желания высказывать свое недоумение, мы попросим вас все это выполнить не на паре компьютеров, как в примере, а хотя бы на трех или пяти. Сколько у нас есть виртуальных машин? А невиртуальных? И еще попросим прибавить к этому небольшую обработку файлов. Хотя бы предположим изменение текстового содержания переписываемых с компьютера на компьютер файлов.

Да забыли сказать, что все это вы должны сделать на Ubuntu-компьютере, который находится, пусть и не очень далеко, а в соседней комнате. Но и тут проблема, ключ от комнаты кто-то случайно утащил домой. У вас есть два варианта: либо ломать железную дверь, либо запускать Windows-машину и настраивать SSH-канал на админский доступ к Ubuntu-машине.

Вперед, желаем удачи! Только не забудьте, что это все надо будет повторить в нашем присутствии и желательно за пару минут. Так что без элементов shell-программирования едва ли успеете. Еще раз удачи.

9.4.2. Использование smbclient в Shell-скриптах

Вариантов использования может быть множество, в зависимости от задачи, которая стоит перед системным администратором. Мы же рассмотрим эту возможность только на одном небольшом примере.

Замечание.
Если с предыдущим заданием из п. 9.4.1 вы успешно справились, то можете пропустить этот раздел. Всех остальных, мы попробуем познакомить с основными подходами к решению таких задач.

Итак, суть задачи заключается в следующем:

- На множестве узлов сети имеется хотя бы по одному общедоступному ресурсу, полный доступ к которому возможен по известному нам стандартному имени и паролю.
- Необходимо периодически подключаться к каждому из этих ресурсов с целью выполнения на нем тех или иных действий. Например, либо обновлять какие-либо из файлов, либо изменять структуру каталогов, либо архивировать данные, хранящиеся на этом ресурсе, и т. д.

- Требуется автоматизировать процесс подключения к этим ресурсам для выполнения на каждом из них той или иной задачи, которая нужна на данный момент. Для этой цели необходим bash-скрипт, запускаемый на вашем компьютере для выполнения нужного задания на каждом из узлов сети.

Для решения поставленной задачи достаточно использовать следующую структуру вызова утилиты smbclient:

```
smbclient //<ресурс> -N -U <ЛогинПароль> -c <команды>
```

При использовании данной конструкции в составе bash-скрипта, следует принимать во внимание следующие факторы:

- Для возможности многократного обращения к данной процедуре все параметры ее вызова должны быть представлены не константами, а переменными, значения которых могут изменяться в процессе выполнения скрипта.
- Вызов этой процедуры следует выполнять в цикле столько раз, сколько у нас адресов общедоступных ресурсов. При этом адреса должны последовательно считываться из какого-либо справочного файла.
- Так как для одной и той же группы адресов в разные моменты времени могут быть разные задания, то есть разные списки smbclient-команд, то эти команды должны храниться в некотором файле команд.
- Что касается пользователя и пароля, то в целях безопасности они должны вводиться в командной строке при вызове процедуры автоматизации. Другими словами, реализуемый нами bash-скрипт должен воспринимать параметр командной строки.

Реализация поставленной задачи

➢ Базируясь на возможности нашей виртуальной сети, справочник удаленно доступных ресурсов сети, реализованный в виде текстового файла с именем data-address, может иметь вид:

```
serp@vmUbuntu10:~$ cat data-address
192.168.1.44/vmxp_share
192.168.1.2/vmpc_share
```

➢ Предположим, что текущей задачей является создание на каждом из удаленных ресурсов нового каталога и записи в него с компьютера администратора сети файла newfile.txt. Содержание этого файла может быть произвольным и состоять, например, всего из одной строки «File from Ubuntu PC».

Для реализации требуемых действий с использование команд утилиты smbclient, создадим файл data-command, содержание которого будет иметь вид:

```
serp@vmUbuntu10:~$ cat data-command
md NewFolder
cd NewFolder
put newfile.txt
```

➢ Создадим текстовый файл с именем sample для проверки того, что мы умеем работать с файлами, что файл data-command составлен правильно и что мы можем правильно из него сформировать текстовую строку <список-команд> для утилиты smbclient.

Эту строку в дальнейшем будем использовать в подстановке к параметру **-c** утилиты smbclient. Напомню, что все команды smbclient в этой строке должны разделяться точкой с запятой.

Одно из возможных содержаний файла sample может иметь вид, аналогичный нижеприведенному:

```
serp@vmUbuntu10:~$ cat sample
cmd=""                  #определяем переменную cmd, где
                        #будем формировать <список-команд>
while read str_cmd      #читаем текущую строку из файла
                        #data-command в переменную str_cmd
do
                        #считанную из файла команду
  cmd="$cmd$str_cmd; " #приклеиваем к текущему содержимому
                        #cmd, добавляя символ ; и пробел
done < data-command

echo -e "Список команд выполняемых удаленно: \n $cmd"
```

Оператор **read** читает данные из входного потока. В данном примере в этот поток перенаправлен файл data-command.

Опция **-e** в операторе **echo** позволяет ему воспринимать esc-последовательности, в качестве которой в данном примере используется символ **\n** — символ перевода строки.

➢ Если с содержимым файла sample все понятно, то установим ему атрибут выполняемого и запустим на выполнение:

```
serp@vmUbuntu10:~$ chmod +x sample
serp@vmUbuntu10:~$ /home/serp/sample
Список команд выполняемых удаленно:
 md NewFolder; cd NewFolder; put newfile.txt;
```

➢ Если результат такой же, как в рассмотренном выше примере, то переходим к написанию требуемого bash-скрипта, для которого будем использовать тот же файл sample, частично отредактировав его и добавив несколько строк:

```
serp@vmUbuntu10:~$ cat sample
# первый параметр запуска bash-скрипта из командной строки
# <логин-пароль> вида login%password сохраняем
# в переменной login_psw
login_psw="$1"

# читаем файл команд и формируем <список-команд> smbclient
cmd=""
while read str_cmd
do
  cmd="$cmd$str_cmd; "
done < data-command
echo -e "Список команд выполняемых удаленно: \n $cmd"

# данные входного потока читаем в addr, при вызове скрипта
# во входной поток перенаправим файл data-address
while read addr
do
  echo "-> Удаленный ресурс: $addr"
  smbclient //"$addr" -N -U "$login_psw" -c "$cmd exit"
done
```

В этом скрипте организован цикл вызова утилиты smbclient по количеству записей из файла входного потока. Если таким файлом будет являться наш data-address, то цикл будет выполнен для двух адресов, указанных в этом файле.

Организация скрипта с открытым входным потоком предполагает возможность без изменения кода скрипта использовать при его вызове разные справочные файлы адресов.

Обратите внимание на формирование параметра -c <список-команд> при обращении к утилите smbclient. Чтобы обеспечить автоматическое закрытие сессии подключения smbclient к удаленному ресурсу, сформированная ранее последовательность smbclient-команд добавляется командой exit.

➢ Протокол вызова и выполнения этого скрипта на локальной Ubuntu-машине будет иметь вид:

```
serp@vmUbuntu10:~$ /home/serp/sample serp%serp < data-
address
Список команд выполняемых удаленно:
```

```
 md NewFolder; cd NewFolder; put newfile.txt;
-> Удаленный ресурс: 192.168.1.44/vmxp_share
Domain=[VM-WINXP] OS=[Windows 5.1] Server=[Windows 2000
LAN Manager]
putting file newfile.txt as \NewFolder\newfile.txt (1,1
kb/s) (average 1,1 kb/s)
-> Удаленный ресурс: 192.168.1.2/vmpc_share
Domain=[VM-WINPC] OS=[Windows 5.1] Server=[Windows 2000
LAN Manager]
putting file newfile.txt as \NewFolder\newfile.txt (0,9
kb/s) (average 0,9 kb/s)
serp@vmUbuntu10:~$
```

➢ Если у вас на основном компьютере, помимо утилиты PuTTY, подключенной по SSH-каналу к vmUbuntu10, с помощью которой и выполнялись все описанные выше действия, были открыты еще два окна удаленных общедоступных папок, то в процессе выполнения этого скрипта вы могли наблюдать автоматическое их обновление (рис. 9.19).

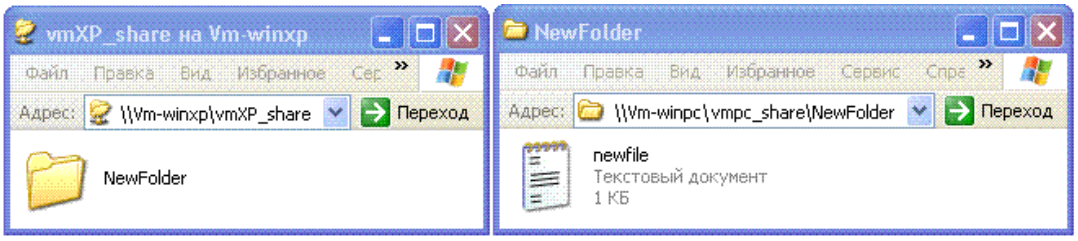

Рис. 9.19. Вид vmXP_share с новой папкой и вид новой папки с новым файлом.

При повторном запуске этого скрипта у вас будут появляться предупредительные сообщения, которые связаны с наличием указанных папок и файлов на удаленных ресурсах. Чтобы повторно проверить работу скрипта, надо эти папки и файлы удалить.

В качестве практики, мы попросим вас изменить скрипт таким образом, чтобы он предупреждал вас о таких ситуациях. Он должен предлагать возможные пути обхода этих ситуаций. Либо автоматически удалять папки и заменять их на новые. Либо отказаться от возможных изменений на каком-либо конкретном удаленном ресурсе.

Если вас заинтересовал рассмотренный пример, то следует отметить, что он может быть просто видоизменен для почтовой рассылки. если вместо обращения к утилите smbclient использовать конструкцию типа:

```
echo "email message" | mail -s Test Мой_ящик@yandex.ru
```

Но это разговор из другой темы, из другой сетевой технологии. А сейчас мы рассматриваем доступ к общесетевым ресурсам и возможность удаленного управления этими ресурсами и их модификацией.

Подводя некоторый итог данному разделу, мы можем констатировать, что, подключаясь удаленно к Ubuntu-машине, мы можем программно изменять содержимое общедоступных для нее ресурсов на удаленных Windows-машинах.

9.5. Монтирование удаленных сетевых ресурсов

Наверно, вы не обратили внимания, что во всех предыдущих примерах мы не использовали vmUbuntu06. А почему? Все дело в том, что на ней не установлен smb-сервер, и использовать для доступа к ней smb-клиента просто не представляется возможным. Но даже когда можно использовать smbclient, то существует ряд ограничений:

- Мы не можем использовать стандартные команды shell-оболочки для выполнения файловых операций.
- У нас нет возможности непосредственно переписывать файл с одной Window-машины на другую. Мы должны сначала получить файл (get) на свой компьютер, а потом скопировать его (get) на другой компьютер.
- Для любой модификации файла надо сначала получить файл с удаленного узла, модифицировать его, а потом скопировать на удаленный узел, предварительно удалив старый с тем же именем.
- Мы не можем осуществить непосредственный просмотр файла удаленного узла.

Вместе с тем все эти операции за два клика выполняются в Nautilus. Так что, значит, smbclient плох, а Nautilus — хорош? Не спешите с выводами. Это две совершенно разные сетевые технологии, каждая из которых имеет свои области применения.

Но если с первой мы уже познакомились, то со второй это еще только предстоит. И для знакомства с ней нам надо разобраться с различными вариантами монтирования общедоступных удаленных ресурсов. Здесь тоже есть разные подходы, определяемые теми задачами, которые стоят перед вами.

9.5.1. Автоматическое монтирование сетевых ресурсов средой gvfs-fuse в Nautilus

При работе с сетевыми ресурсами в Nautilus мы отмечали возможность свободного доступа к папкам и файлам различных операционных систем. Более того, Nautilus при первом подключении создал на рабочем столе ярлыки, сохранив нам ссылки на доступные сетевые ресурсы. Это позволяет легко выполнять повторный доступ к этим ресурсам.

А что это такое, если не точки монтирования. Но где они зарегистрированы, как к ним обращаться, чтобы, не используя Nautilus,

иметь возможность доступа к сетевым ресурсам для выполнения тех или иных файловых операций. Где находятся сетевые папки, ярлыки на которые Nautilus вывел на рабочий стол?

Вспомним уже установленный нами факт, что основой для файлового менеджера Nautilus является gvfs-fuse — виртуальная файловая система в пользовательском пространстве, в которой монтирование происходит как запуск отдельных приложений.

А что такое нажатие кнопки Подключить в среде Nautilus, как не запуск gvfs-приложения? А если так, то именно gvfs-fuse должна отвечать за процесс монтирования. Попробуем исследовать точки монтирования нашей Ubuntu системы:

```
serp@vmUbuntu10:~$ mount -i
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
. .
gvfs-fuse-daemon on /home/serp/.gvfs type fuse.gvfs-fuse-
daemon (rw,nosuid,nodev,user=serp)
serp@vmUbuntu10:~$
```

Из приведенного листинга видно, что за одну из точек монтирования как раз и отвечает gvfs-fuse сервер, который экспортирует gvfs-монтировки на все приложения, использующие fuse. И этой точкой монтирования является /home/serp/.gvfs, то есть папка .gvfs в нашей домашней директории.

А если это так, то в обозревателе файлов откроем домашний каталог, у нас это serp. Затем в меню файлового менеджера выбираем Вид и в открывшемся меню отмечаем опцию Показывать скрытые файлы.

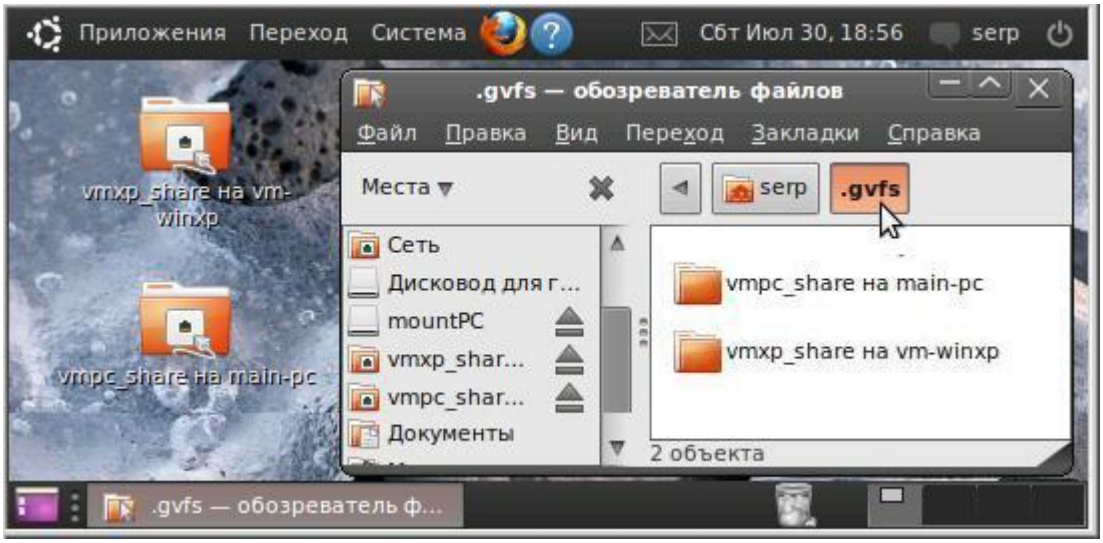

Рис. 9.20. Папки примонтированных сетевых ресурсов в /home/serp/.gvfs.

В достаточно большом списке ресурсов вашего домашнего каталога находим скрытую папку .gvfs. Входим в нее и видим папки

примонтированных сетевых ресурсов (рис. 9.20). Зная точки монтирования, можно непосредственно выполнять любые файловые операции, используя для этого консольные команды Ubuntu.

Рассмотрим простейший пример. Поставим перед собой задачу скопировать файл test1.txt из общедоступной папки vmXP_share виртуальной машины vm-WinXP в общедоступную папку vmpc_share на основном компьютере Main-PC, изменив имя файла на new.txt. А затем модифицируем содержимое этого файла.

С учетом промежуточных тестовых операций, листинг консоли vmUbuntu10, на которой выполняются команды, будет иметь вид:

```
~$ cd .gvfs
~/.gvfs$ ls
vmpc_share на main-pc  vmxp_share на vm-winxp
~/.gvfs$ ls 'vmxp_share на vm-winxp'
from_ubuntu  test1.txt  test2.txt
~/.gvfs$ cd 'vmxp_share на vm-winxp'
```

```
~/.gvfs/vmxp_share на vm-winxp$ cp test1.txt
../'vmpc_share на main-pc'/new.txt
```

```
~/.gvfs/vmxp_share на vm-winxp$ cat ../'vmpc_share на
main-pc'/new.txt
Это тестовый файл
строка1
строка2
```

Если откроем папку vmpc_share непосредственно на основном компьютере, то найдем в ней файл new.txt, который можно открыть в Блокноте (рис. 9.21).

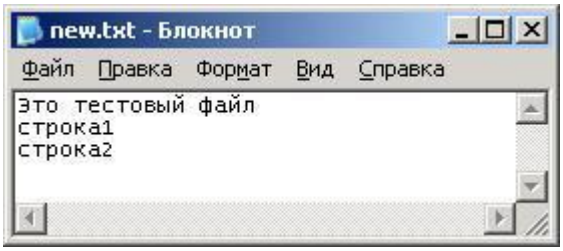

Рис. 9.21. Файл, скопированный с vm-WinXP на Main-PC командой cp на vmUbuntu10.

Это подтверждает тот факт, что работая на Ubuntu-машине и используя автоматически смонтированные с помощью Nautilus и gvfs-fuse сетевые ресурсы, можно обеспечить выполнение файловых операций непосредственно между Windows-машинами.

9.5.2. Монтирование сетевых ресурсов с использованием gvfs-fuse

Возможности, которые представляет Nautilus по доступу к сетевым ресурсам, достаточно широкие, но вместе с тем утомительно каждый раз ходить по опциям меню и окошкам, чтобы подключить тот или иной сетевой ресурс. Когда этих ресурсов три-пять, это сносно, а когда значительно больше, то хочется применительно к Nautilus вспомнить поговорку: «Его бы энергию да в мирных целях».

Вспомни два теоретических положения, а именно:

- gvfs — это виртуальная файловая система в пользовательском пространстве, в которой монтирование происходит как запуск отдельных приложений по команде D-Bus;
- монтирование осуществляется gvfs-fuse сервером на основе полного URI этого ресурса.

А если теперь к этим положениям добавить очевидный факт, что Nautilus — это тоже отдельное приложение, запуск которого возможен, то напрашивается очевидный вывод: «Почему бы не заставить Nautilus выполнять автоматическое монтирования, но по нашему командному требованию». Именно в этих целях мы можем использовать консольную команду

```
nautilus <URI сетевого ресурса>
```

Рассмотрим небольшой пример. Отключим все сетевые подключения, которые ранее были установлены с vm-WinXP (192.168.1.44/24) и Main-PC (192.168.1.2/24). Затем войдем в терминал vmUbuntu10 и выполним две команды (рис. 9.22).

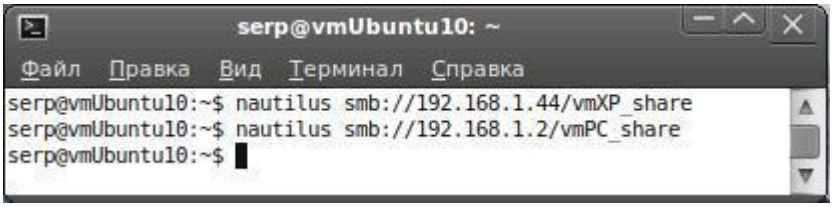

Рис. 9.22. Подключение двух сетевых ресурсов, путем вызова Nautilus из терминала.

Обратите внимание, насколько быстрее устанавливается подключение к Windows-ресурсу, чем работа с Сеть Windows в окнах самого Nautilus. Но это уже другой рассказ, сродни использованию Сетевого окружения Windows путем кликанья мышкой или указанию прямого адреса ресурса.

Если все было сделано верно, то, зайдя в папку /home/serp/.gvfs, вы должны получить результат, аналогичный рис. 9.20. Выполняя данный пример, вы наверняка заметили, что в процессе его выполнения открывались окна Nautilus и отображалось содержание сетевых ресурсов. В каких-то случаях это просто здорово, но, рассматривая процесс

монтирования сетевых ресурсов, такое использование Nautilus — это «стрельба из пушки по воробьям».

Монтирование на лету — это только одна из множества других функциональностей Nautilus. И выполняется эта функциональность путем обращения Nautilus по команде D-Bus к gvfs-fuse серверу, который экспортирует gvfs-монтировки на все приложения, использующие fuse.

И здесь появляется желание для выполнения подключения сетевых ресурсов использовать gvfs-монтирование, которое можно было бы осуществлять непосредственно из командной строки. Этот подход реализуем средствами файловой системы в пользовательском пространстве путем выполнения команды:

```
gvfs-mount <URI сетевого ресурса>
```

Рассмотрим пример, аналогичный вышерассмотренному. Для этого отключим все ранее установленные сетевые подключения, а затем выполним ряд команд, аналогичных листингу терминала, приведенного на рис. 9.23.

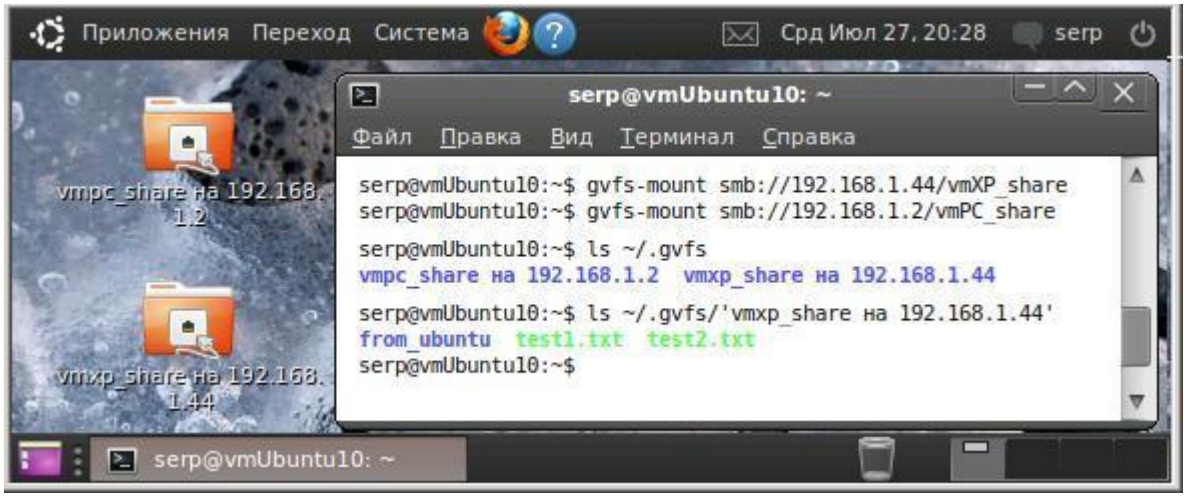

Рис. 9.23. Gvfs-монтирование сетевых Windows-ресурсов на Ubuntu-машине.

Из рисунка видно, что при монтировании сетевых ресурсов на рабочем столе появляются ярлыки на доступ к этим ресурсам, точкой их монтирования является папка /home/serp/.gvfs, а доступ к конкретному каталогу или файлу этого ресурса возможен из папки ~/.gvfs/<имя gvfs-монтировки ресурса>, причем <имя gvfs-монтировки ресурса> система назначает сама.

После выполнения gvfs-монтирования у нас появляется возможность выполнения любых файловых операций с удаленными сетевыми ресурсами. После их окончания мы можем провести размонтирование этих ресурсов, с автоматическим удалением ярлыков рабочего стола (рис. 9.24).

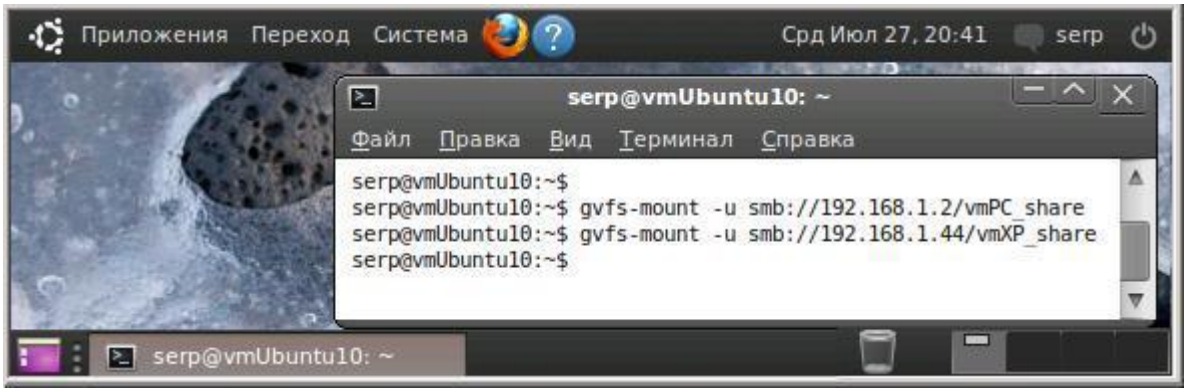

Рис. 9.24. Размонтирование gvfs-монтирововок сетевых ресурсов.

За возможность gvfs-монтирования бэк-эндов разных протокольных стеков отвечает пакет gvfs-backends, который включает в себя в качестве зависимостей большое количество библиотек (см. п. 9.1.4). Именно они поддерживают монтирование на лету различных сетевых устройств, включая фото и видеотехнику, iPhone, iPod Touch и т. д.

Из технической документации следует, что пакет gvfs-backends_1.6.0+git20100414-0ubuntu1_i386.deb — обеспечивает поддержку afc, archive, burn, cdda, dav, dnssd, ftp, gphoto2, http, network, obexftp, sftp, smb и smb-browse протоколов.

В нашей виртуальной сети с экзотическими сетевыми устройствами туго, но вместе с тем есть один не smb-ресурс, которым является виртуальная машина vmUbuntu06. Именно на ее примере мы и покажем возможность gvfs-монтирования не smb-устройств.

```
# Монтирование и размонтирование сетевого ресурса по
протоколу SSH
serp@vmUbuntu10:~$ gvfs-mount ssh://192.168.1.6/home/serp
Введите пароль
User: serp
Password:
serp@vmUbuntu10:~$ gvfs-mount -u
ssh://192.168.1.6/home/serp
```

```
# Монтирование и размонтирование сетевого ресурса по
протоколу SFTP
serp@vmUbuntu10:~$ gvfs-mount sftp://192.168.1.6/home/serp
Введите пароль
User: serp
Password:
serp@vmUbuntu10:~$ gvfs-mount -u
sftp://192.168.1.6/home/serp
```

```
# Монтирование и размонтирование сетевого ресурса по
протоколу SSH для конкретного пользователя
serp@vmUbuntu10:~$ gvfs-mount
ssh://serp@192.168.1.6/home/serp
Введите пароль
Password:
serp@vmUbuntu10:~$ gvfs-mount -u
ssh://serp@192.168.1.6/home/serp
```

Из приведенного листинга ясна сетевая технология gvfs-монтирования ssh и sftp ресурсов. Что же касается возможности манипуляций с этими ресурсами после установления с ними соединения, с этим вы разберетесь и без моего участия. Это же относится и к возможностям gvfs-монтирования других устройств с отличными от рассмотренных протоколов.

Для практики мы попросим вас написать скрипт, аналогичный тому, который вы составляли при рассмотрении smbclient, но теперь с использованием gvfs-монтирования. Дело за вами, успехов!

9.5.3. Монтирование сетевых ресурсов с использованием mount и fstab

Рассмотренная технология всем хороша, но пока мы работаем в консоли локально, а вот если удаленно, то возникают затруднения. По крайне мере на момент написания данной книги:

- либо у авторов не хватает квалификации,
- либо в этом виновата FUSE (Filesystem in USErspace — «файловая система в пользовательском пространстве»).

Если пока gvfs-монтирование из удаленной консоли для нас лично недоступно, то это не повод отказываться от поставленной цели — непосредственного манипулирования сетевыми ресурсами.

Тем более что существуют и старые, давно проверенные технологии монтирования сетевых ресурсов. Мы не будем подробно останавливаться на технологии монтирования сетевых ресурсов с использованием команды mount, так как существует бесчисленное множество ресурсов и форумов Интернет, где эта тема смакуется уже с десяток лет. Кроме этого вам всегда доступна команда man mount.

Рассмотрим только небольшой пример применительно к нашей виртуальной сети, чтобы на его основе получить представление о данной технологии. Итак, у нас в наличии:

- Виртуальная сеть из Ubuntu- и Windows-машин.
- Желание, используя Ubuntu-машина, переписать файл с одной Windows-машины на другую.
- Удаленное подключение с основного компьютера к Ubuntu-машине по SSH-каналу с помощью утилиты PuTTY.

Для возможности монтирования сетевых ресурсов необходимо прежде всего определить точки их монтирования. Это могут быть любые каталоги

в доступной нам области файловой системы Ubuntu. Единственное требование, чтобы они были пустыми. Для простоты, создадим два новых каталога в пользовательской рабочей директории.

В нашем примере рабочая директория — serp, а точки монтирования (новые каталоги) — mountPC и mountXP. Их имена могут быть и любыми другими.

```
serp@vmUbuntu10:~$ mkdir mountPC
serp@vmUbuntu10:~$ mkdir mountXP
```

Используя mount с правами root, выполняем монтирование (подключение) сетевых ресурсов по протоколу cifs (smb/cifs) к соответствующим папкам.

```
serp@vmUbuntu10:~$ sudo mount -t cifs
//192.168.1.2/vmPC_share  ~/mountPC -o
user=serp,password=serp,iocharset=utf8
serp@vmUbuntu10:~$ sudo mount -t cifs
//192.168.1.44/vmXP_share ~/mountXP -o
user=serp,password=serp,iocharset=utf8
```

Просматривая состав свой рабочей директории, видим, что после монтирования хозяином папок mountPC и mountXP стал root.

```
serp@vmUbuntu10:~$ ls  -l
drwxr-xr-x 2 serp serp 4096 2011-06-20 12:18 Видео
-rwxr-xr-x 1 serp serp   45 2011-07-19 22:12 sample
. . .
drwxr-xr-x 1 root root    0 2011-06-17 14:14 mountPC
drwxr-xr-x 1 root root    0 2011-07-28 15:58 mountXP
```

Теперь нам доступны сетевые ресурсы, просмотр которых можно выполнить, используя точки монтирования mountPC и mountXP.

```
serp@vmUbuntu10:~$ ls ~/mountXP
from_ubuntu  test1.txt  test2.txt

serp@vmUbuntu10:~$ ls ~/mountPC
setup-ssh    smb_config.txt
```

Выполним копирование файла test1.txt с виртуальной машины vm-WinXP на основной компьютер Main-PC, изменив имя файла на new.txt.

```
serp@vmUbuntu10:~$ sudo cp ~/mountXP/test1.txt
~/mountPC/new.txt

serp@vmUbuntu10:~$ ls ~/mountPC
setup-ssh    smb_config.txt     new1.txt
```

Выполним размонтирование (отключение) сетевых ресурсов и просмотром убедимся, что точки монтирования свободны

```
serp@vmUbuntu10:~$ sudo umount ~/mountPC
serp@vmUbuntu10:~$ sudo umount ~/mountXP

serp@vmUbuntu10:~$ ls ~/mountPC
serp@vmUbuntu10:~$ ls ~/mountXP
```

То есть поставленную перед собой цель, а именно возможность использования файловых операций непосредственно между удаленными сетевыми ресурсами мы выполнили. Но следует напомнить, что для этого надо выполнять монтирование сетевых ресурсов. А если в вашей корпоративной сети эти ресурсы более-менее постоянные, и для выполнения стандартных ежедневных операций вам каждый раз придется проводить монтирование, то здесь вы скажете все, что думаете о сетевых технологиях.

А совершенно зря, так как помимо команды mount, которая работает только на один текущий сеанс и не сохраняет выполненные монтировки после перезагрузки операционной системы, существует еще и системный файл /etc/fstab, который отвечает за монтирование вашей операционной системы. Работать с ним надо очень осторожно, чтобы не грохнуть всю вашу Ubuntu. Вот посмотреть на него можно совершенно свободно, используя команду:

```
cat /etc/fstab
```

Содержание этого файла имеет вид, аналогичный приводимому ниже:

```
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally
# unique identifier for a device; this may be used with
# UUID= as a more robust way to name devices that works
# even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>               <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid     0       0
# / was on /dev/sda1 during installation
UUID=ff1ad4ea-c50a-43f7-bd28-9c46b / ext4 errors=remount-ro  0       1
# swap was on   /dev/sda5 during installation
UUID=6b23e475-8de7-45d2-96c9-6e69b346e none    swap    sw      0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto,exec,utf8  0       0
```

В нашу задачу сейчас не входит подробный разбор этого файла и его содержания. У нас интерес проще — вставить в этот файл всего одну строку:

```
//192.168.1.2/vmPC_share  /home/serp/mountPC  cifs
_netdev,user=serp,password=serp,iocharset=utf8
```

Вызовите текстовый редактор с правами root, и аккуратно, не испортив других строк, добавьте требуемую строку в конец файла. Обратите внимание, что она очень похожа на формат команды mount, которую вы использовали ранее. Для вызова редактора можно использовать команду:

```
sudo nano /etc/fstab
```

После этого следует перезагрузить операционную систему. После ее загрузки на вашем рабочем столе будет пиктограмма с именем mountPC, кликнув по которой откроется соответствующий сетевой ресурс (рис. 9.25).

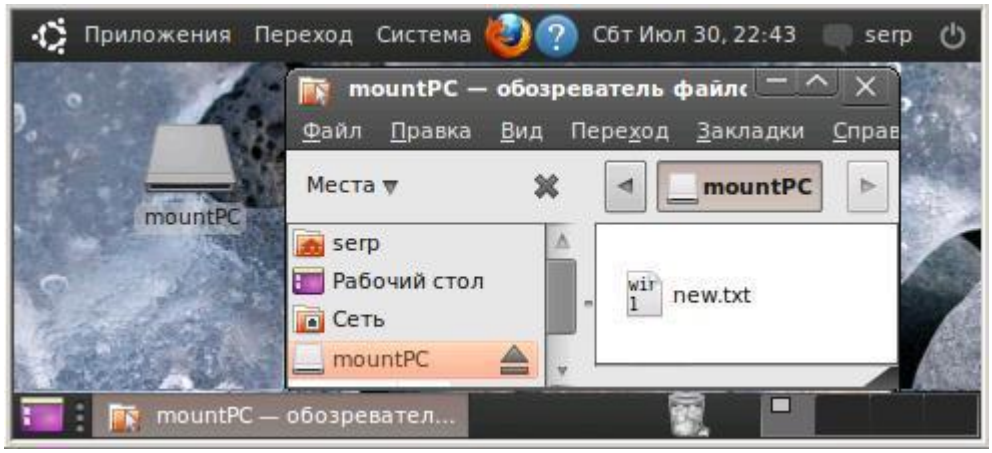

Рис. 9.25. Вид рабочего стола после перезагрузки с измененным файлом /etc/fstab.

О том, что наша монтировка сохранилась, мы увидим не только по рабочему столу, но и при выполнении команды:

```
mount -i
```

Результат выполнения этой команды будет аналогичен тому, что приведен ниже:

```
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
. . .
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc
(rw,noexec,nosuid,nodev)

//192.168.1.2/vmPC_share on /home/serp/mountPC type cifs
(rw,_netdev,user=serp,password=serp,iocharset=utf8)

gvfs-fuse-daemon on /home/serp/.gvfs type fuse.gvfs-fuse-
daemon (rw,nosuid,nodev,user=serp)
```

Откуда видно, что нам доступен не только сетевой ресурс, примонтированный статически к /home/serp/mountPC, но в этом сеансе работы операционной системы доступны и динамические gvfs-монтировки. При этом следует отметить: чтобы размонтировать статическую

монтировку на следующие сеансы запуска Ubuntu следует удалить добавленную нами строку из файла /etc/fstab. Все это хорошо, но описанное в этой главе относилось в первую очередь к доступу из Ubuntu к удаленным Windows-ресурсам. Пора бы рассмотреть и обратный процесс, обратив свое внимание на Samba-сети в Ubuntu.

9.6. Команды файловых операций в Gnome Virtual File System (GVFS)

Виртуальная файловая система Gnome (gvfs), хоть и является виртуальной, но все же это файловая система. И как любая файловая система, она обладает командным языком манипулирования файлами. Только особенность этого языка заключается в том, что в качестве файлов могут выступать любые ресурсы, как локальные, так и удаленные.

Основное требование, чтобы к этим ресурсам можно было обращаться по URI. В общем случае формат вызова команд gvfs имеет вид:

```
gvfs-<команда> [-опция] <URI, где URI=URL+URN>
```

Используя man этих команд (Written by Alvaro Lopez Ortega, Sun Microsystems Inc., 2008) и информацию сайта http://www.unix.com/man-page/All/1/gvfs-copy/, приведем краткое описание основных команд и их назначение:

- **gvfs-cat <URI>** --- Вывод файла на стандартный вывод, используемый gvfs, по умолчанию.

Пример использования:

```
#просмотр файла удаленного ресурса.
~$ gvfs-cat smb://192.168.1.2/vmpc_share/new.txt
#просмотр файла с веб-сервера
~$ gvfs-cat http://www.ya.ru
#просмотр файла локального ресурса.
~$ gvfs-cat file:///home/user/readme.txt
```

- **gvfs-copy** <URI источника> <URI получателя> --- Копирование файла с одного ресурса на другой.

Пример использования:

```
~$ gvfs-copy  http://www.habarov.spb.ru/foto/petrovich.jpg
                        file:///home/serp/foto.jpg
~$ gvfs-copy  smb://192.168.1.2/vmpc_share/new.txt  fromPC
```

- **gvfs-info** [-w] [-f] [-a] [-n] <URI> --- Вывод информации о файлах и директориях, используя Gnome vfs.

Опции:

—w --- Список атрибутов типа writeable.

—f --- Получить информацию о файловой системе.

—a --- Список получаемых атрибутов.

—n --- Не следовать за символические ссылки

Пример использования:

```
~$ gvfs-info file:///etc/fstab
~$ gvfs-info http://www.sun.com/
```

➢ **gvfs-less <URI>** --- Выполняет команду less, используя gvfs как препроцессор. В отличие от more позволяет листать файл не только вниз, но и вверх.

Пример использования:

```
~$ gvfs-less http://www.sun.com/
~$ gvfs-less file:///etc/release
```

➢ **gvfs-ls** [-a] [-h] [-l] [-c] <URI> --- Отображает содержимое директория, используя виртуальную файловую систему.

Опции:

—a --- Действует как фильтр атрибутов, определяя какие атрибуты должны отображаться.

—h --- Показывать скрытые файлы.

—l --- Использовать подробный формат вывода.

—c --- Показывать дополнения.

Пример использования:

```
~$ gvfs-ls file: ///usr/sbin/
~$ gvfs-ls -l smb://server/resource/
```

➢ **gvfs-mkdir** <URI> --- Создание нового директория, определяемого посредством URI, с использованием Gnome vfs.

Пример использования:

```
~$ gvfs-mkdir file:///home/mountPC
~$ gvfs-mkdir sftp://192.168.1.6/home/serp/share_dir
```

➢ **gvfs-mount** [-m] [-u] [-l] [-i] <URI> --- Подключение, отключение и просмотр файловых систем с использованием Gnome vfs.

Опции:

—m --- Монтирование файловой системы.

—u --- Размонтирование файловой системы.

—l --- Список подключенных драйверов, томов и точек монтировки.

—I --- Используется с —l для вывода дополнительной информации.

Пример использования:

```
# выводит информацию о точках монтирования и драйверах.
~$ gvfs-mount –i
 # выполняет отключение подключенной ранее файловой системы.
~$ gvfs-mount -u /home/serp/mountPC
```

➢ **gvfs-move** <URI источника> <URI получателя> --- Перемещает файлы с одного URI-ресурса на другой, используя Gnome vfs.

Пример использования:

```
# перемещает файл с WEB-сервера на локальный ресурс.
~$ gvfs-move  http://www.myserver.ru
file:///home/user/index.html
```

➢ **gvfs-open** <URI> — Открывает URI-ресурс в соответствующем приложении.

Пример использования:

```
# открытие локального image-файла.
~$ gvfs-open file:///tmp/example.jpg
# открытие удаленного ресурса.
~$ gvfs-open http://www.habarov.spb.ru/
```

➢ **gvfs-rename** <URI источника> <новый URI> --- Изменяет имя файла или каталога, используя Gnome vfs.

Пример использования:

```
~$ gvfs-rename file:///home/user/p.html
               file:///home/user/p.html.old
```

➢ **gvfs-rm** <URI> --- Удаляет информацию о файле или каталоге, указанным посредством URI, используя Gnome vfs.

Пример использования:

```
~$ gvfs-rm file:///tmp/example
~$ gvfs-rm smb://server/dir/example
```

➢ **gvfs-save** [-b] [-c] [-a] [-p] [-v] [-e] <URI> --- Сохраняет информацию в файле, используя Gnome vfs.

Опции:

—b --- Создавать копию перед записью файла.

—c --- Только создать файл, если не существовал.

—a --- Добавить в конец существующего файла.

—p --- При создании файла, ограничить доступ только для текущего пользователя

—v --- Записывать в конце файла уникальный идентификатор (etag).

—e --- Перезаписать уникальный идентификатор (etag) файла.

Пример использования:

```
 # добавление строки в конец существующего локального файла.
~$ echo "hello" | gvfs-save file:///tmp/example
# добавление файла в конец файла с сохранением копии.
~$ gvfs-cat file:///tmp/file | gvfs-save -b smb://server/dir/file
```

➢ **gvfs-trash** <URI> --- Перемещает файл или папку в мусорную корзину (trash) средствами Gnome vfs.

Пример использования:

```
~$ gvfs-trash file:///tmp/example
~$ gvfs-trash smb://server/dir/example
```

➢ **gvfs-tree** [-h] [-l] <URI> --- Выводит список папок и файлов в виде подчиненного дерева

Опции:

—h --- Показывать скрытые файлы.

—l --- Отображать ссылки, точки монтировки как директории.

Пример использования:

```
~$ gvfs-tree file:///usr/bin/
~$ gvfs-tree smb://server/resource/
```

10. ФАЙЛОВЫЙ СЕРВЕР SAMBA

Эта глава, являясь продолжением предыдущей, также будет посвящена вопросам общедоступных ресурсов в Ubuntu-Windows системах. С возможными подходами по доступу из Ubuntu к Windows-ресурсам мы познакомились в предыдущей главе. Но там же мы отметили отсутствие доступа из Windows к ресурсам Ubuntu и невозможность публикации их папок в качестве общедоступных.

При попытке публикации какой-либо папки выдавалось сообщение, что служба публикации папок не установлена (рис. 9.4). Для ее функционирования необходимо, чтобы на компьютере был установлен пакет Samba. Однако эта служба только одна из составных частей достаточно мощного пакета Samba, поддерживающего сетевую файловую систему в гетерогенных smb-сетях.

Samba — это программа, которая позволяет обращаться к сетевым дискам на различных операционных системах по протоколу SMB/CIFS. Имеет клиентскую и серверную части.

Она является свободным программным обеспечением, выпущенным под лицензией GPL. Копия этой лицензии идет вместе с этим пакетом. Вам разрешается распространять копии пакета Samba, но пожалуйста, распространяйте их целиком.

Последняя версия пакета Samba может быть получена с ftp-сервера samba.anu.edu.au из каталога pub/samba/. Она также доступна на различных сайтах-зеркалах в Internet.

Пакет Samba работает на большинстве Unix-подобных систем. Он включен практически во все дистрибутивы GNU/Linux, в том числе и в Ubuntu. Для установки пакета Samba можно воспользоваться Менеджером пакетов, Центром приложений или открыть терминал и ввести:

```
sudo apt-get install samba
```

При отсутствии прямого доступа в Интернет, вам может помочь установка из deb-архива:

```
samba_2%3a3.4.7~dfsg-1ubuntu3.7_i386.deb
```

При установке пакета Samba будут дополнительно установлены или обновлены пакеты samba-client и samba-common.

Если установка произведена успешно, то, выбрав «Переход» -> «Сеть», можно убедиться, что файловый менеджер Nautilus стал воспринимать vmUbuntu10 в виде узла smb-сети, наряду с другими Windows-машинами (рис. 10.1).

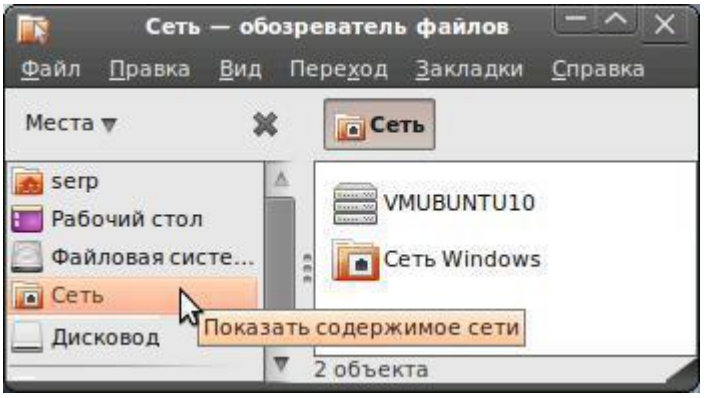

Рис. 10.1. Окно файловый менеджер Nautilus в режиме Сеть.

Чтобы убедиться в том, что Ubuntu-машина теперь доступна в Windows-сети, следует на основном компьютере войти в сетевое окружение. Затем выбрать режим просмотра «Вся сеть» и обнаружить, что появилась новая рабочая группа с именем Workgroup. При входе в эту рабочую группу вам будет доступен компьютер vmUbuntu10 (рис. 10.2).

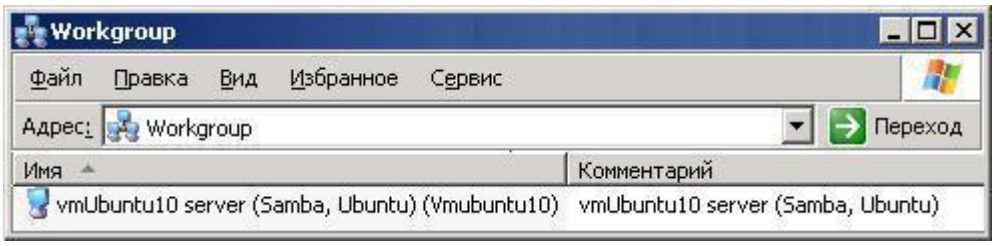

Рис. 10.2. Окно файловый менеджер Nautilus в режиме Сеть.

Из рис. 10.2 следует, что установленный на Ubuntu-машине пакет Samba выполняет для нее роль smb-сервера, который доступен из сети Windows. Но почему Ubuntu-машина входит в рабочую группу Workgroup? Почему она таким образом идентифицируется в Сетевом окружении Windows? С этим разберемся немного позднее.

10.1. Общедоступные папки в Ubuntu Desktop

Создадим на vmUbuntu10 в рабочей директории папку с именем share. Чтобы сделать эту папку общедоступной, достаточно щелкнуть правой кнопкой мыши на этой папке и выбрать пункт меню «Опубликовать папку». Появится окно, в котором следует ввести и установить необходимые параметры и нажать кнопку «Изменить ресурс» (рис. 10.3).

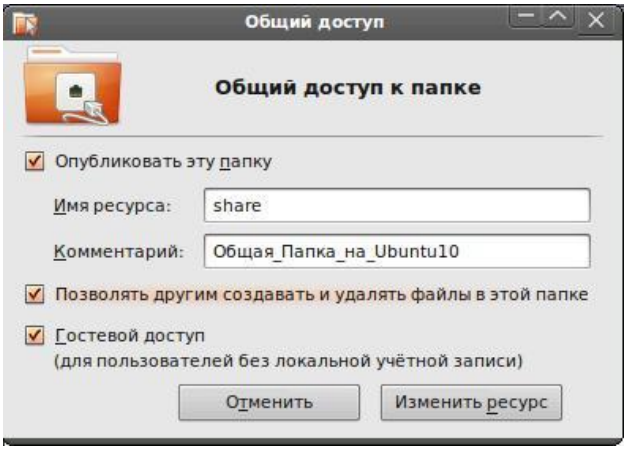

Рис. 10.3. Настройка параметров общего доступа для папки share.

Никаких дополнительных действий и правки конфигурационных файлов при этом выполнять не надо. Войдя в сетевое окружение основного компьютера, мы увидим, что теперь возможен доступ из Windows к общедоступной папке на vmUbuntu10 (рис. 10.4). Возможен и прямой доступ к общедоступной папке, если использовать URN вида:

```
\\vmUbuntu10\share.
```

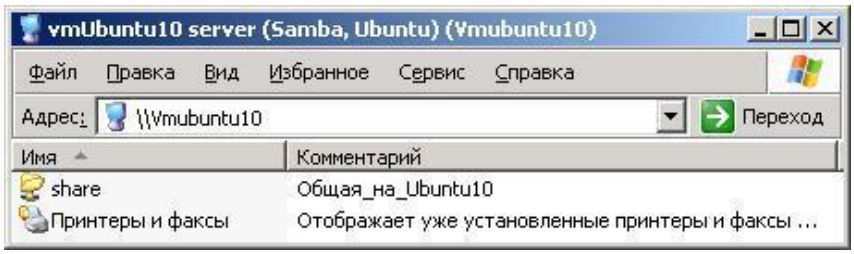

Рис. 10.4. Просмотр ресурсов vmUbuntu10 из сетевого окружения основного компьютера.

Более того, так как при настройке параметров доступа был выбран гостевой доступ, все пользователи сети будут иметь свободный доступ к этой папке. Но при этом полезно помнить, что папка share создавалась в рабочей директории пользователя, которым у нас являлся serp, а раз так, то он и является владельцем этой папки. В этом можно убедиться, вызвав на папке правой кнопкой режим Свойства папки (рис. 10.5).

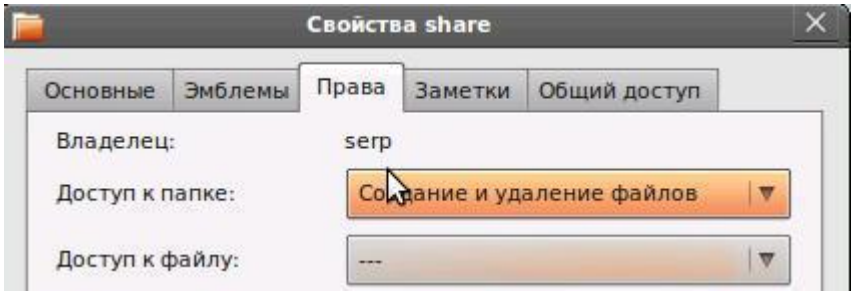

Рис. 10.5. Фрагмент окна просмотра свойств папки share.

Если отключить гостевой доступ к папке share (рис. 10.3), то и при доступе к ней из Windows свободно войти в нее не удастся.

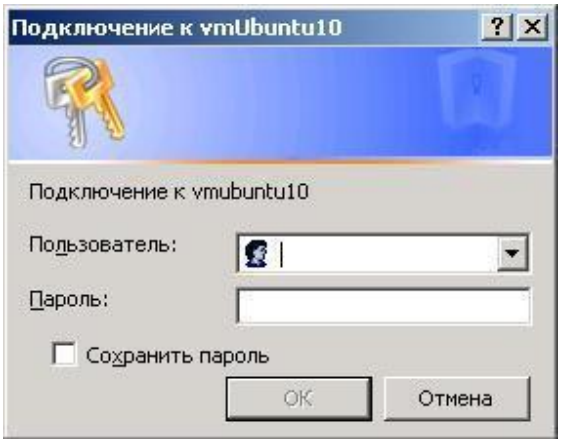

Рис. 10.6. Окно аутентификации на доступ к папке share.

Появится окно «Аутентификация», в котором надо ввести логин и пароль хозяина этой папки (рис. 10.6), и только в этом случае может быть осуществлен доступ к папке share.

Если эта или любая другая общедоступная папка будет опубликована в smb-сети без гостевого доступа, а Windows-пользователи будут иметь логин и пароль, аналогичный владельцу этой папки, то окно аутентификации появляться не будет, и Windows-пользователи будут свободно попадать в эту папку аналогично папкам на своих компьютерах.

В том случае, если Windows-пользователи не являются прямыми владельцами ресурсов, доступных им на Ubuntu-машине, то хотелось бы сделать несколько важных, по нашему мнению, замечаний:

- Следует предостеречь вас, что, выполнив парольный вход на общедоступную папку, Windows запоминает этот вход, и любое повторное обращение к папке share из сетевого окружения не будет требовать аутентификации в процессе текущей сессии.

Так что, отключившись от общего ресурса, Windows-пользователь не должен торопиться бежать пить кофе, не позаботившись о своем компьютере. В его отсутствие, даже не зная логина и пароля и не проходя сетевую аутентификацию, любой «случайно» подошедший к Windows-компьютеру через сетевое окружение этого компьютера будет иметь доступ к удаленным, имеющим парольный доступ, ресурсам.

- Доверяться флажку «Сохранить пароль» тоже особо не следует, так как он действует на следующие сессии работы Windows-компьютера, но не на текущую сессию.

- И, наконец, самое важное. Вы создали общедоступную папку в своей директории, определили к ней доступ по паролю, сообщили пользователям логин и пароль на доступ к ней. Теперь из Сетевого окружения Windows

удаленные пользователи действительно могут добраться только до общедоступного ресурса вашей Ubuntu-машины.

Но подумайте, что, сообщив логин и пароль, вы тем самым дали логин и пароль для входа на вашу Ubuntu-машину. А если у вас для себя открыт удаленный доступ к Ubuntu, например по SSH, то, зная сообщенный вами логин и пароль, любой другой также сможет войти на вашу Ubuntu-машину, а не просто в ее общедоступную папку.

После этого, мы надеемся, вы наверное задумаетесь об организации доступа к общесетевым ресурсам и начнете понимать разницу между пользователями компьютера и пользователями общесетевых ресурсов. А отсюда уже всего один шаг к организации файлового сервера.

10.2. Графический интерфейс для настройки ресурсов SMB

Если у вас небольшая ЛВС и нет особых требований по политике безопасности, то вы можете, не заморачиваясь с конфигурационными файлами Samba, консольными командами или скриптами, воспользоваться графическими утилитами конфигурации файлового сервера Samba.

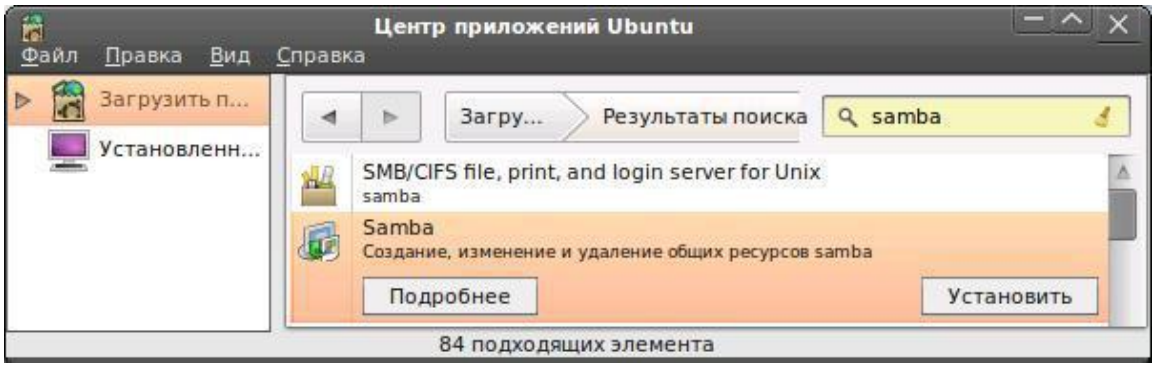

Рис. 10.7. Пакет графической утилиты конфигурации файлового сервера Samba.

Существует большое количество таких утилит, включая возможность Web-доступа к настройкам Samba.

Мы же коротко рассмотрим одну из наиболее простых и распространенных утилит, которую можно установить, например, из Центра приложений Ubuntu (рис. 10.7), из deb-архива system-config-samba_1.2.63-0ubuntu4_all.deb или, введя в консоли, команду:

```
sudo apt-get install system-config-samba
```

После установки этого пакета в основном меню Система -> Администрирование появится новая опция Samba (рис. 10.8), при выборе которой будет доступно графическое окно с возможностью простейших настроек Samba сервера.

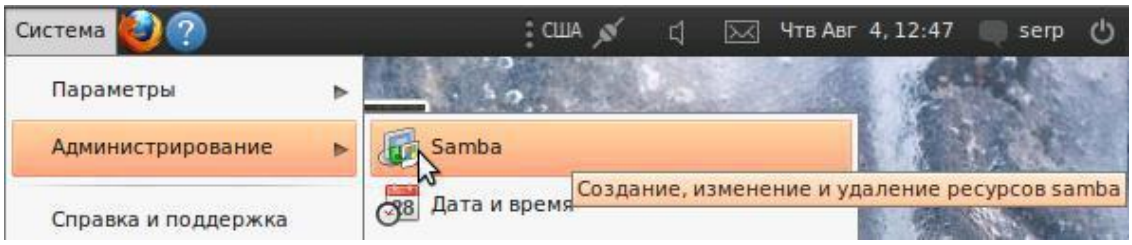

Рис. 10.8. Новая опция меню вызова графического конфигуратора Samba сервера.

10.2.1 Назначение Samba Server Configuration Tool

Данная утилита, которая в русифицированном варианте называется «Настройка сервера Samba», в исходном варианте Samba Server Configuration Tool и которую для простоты мы будем называть просто конфигуратор Samba, предназначена для управления общим доступом к ресурсам Samba сервера, а именно распределенным доступом к его файлам и принтерам. Это графическое приложение предназначено для управления общими ресурсами, пользователями и основными настройками сервера Samba. Оно изменяет файлы конфигурации в каталоге /etc/samba/. При этом сохраняются изменения, внесенные за рамками этого приложения.

Для запуска конфигуратора Samba следует выбрать Система -> Администрирование -> Samba, и у вас откроется окно Настройка сервера Samba (рис. 10.9).

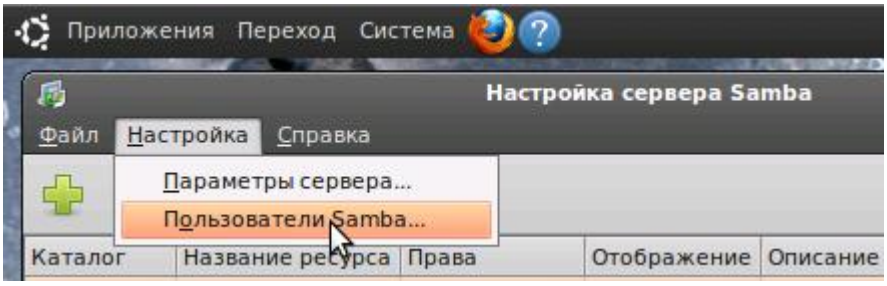

Рис. 10.9. Окно конфигуратора сервера Samba.

Замечание.
Утилита «Настройка сервера Samba» не показывает общих принтеров и стандартной строфы конфигурации, позволяющей пользователям обращаться к своим домашним каталогам на сервере Samba.

10.2.2. Настройка параметров сервера

Первым шагом настройки сервера Samba является настройка основных параметров сервера и нескольких параметров безопасности. Запустив приложение, выберите в выпадающем меню Настройка -> Параметры сервера. На экране появится вкладка Основной, вид которой приведен на рис. 10.10.

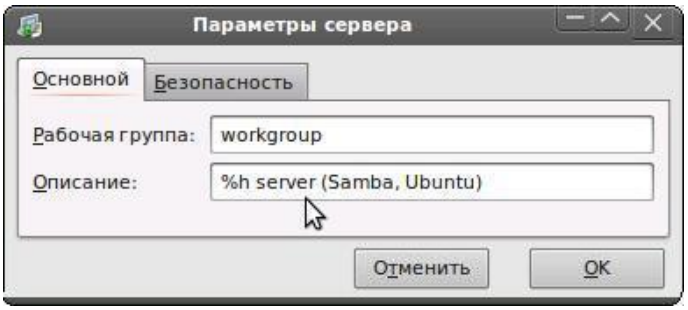

Рис. 10.10. Параметры сервера — Основные.

На вкладке Основной задается, к какой рабочей группе будет относиться компьютер, а также его краткое описание. Этим полям соответствуют параметры workgroup и server string в файле smb.conf.

Обратите внимание на параметры, заданные по умолчанию. Именно эти параметры и отвечали за тот вид, который имел vmUbuntu10 в сетевом окружении Windows на рис. 10.2. Вполне естественно, что эти параметры могут быть изменены в соответствие с вашими требованиями.

В строке «Описание» (server string) вкладки Основной возможно использование макроопределений. Так, используемая в строке Описание (рис. 10.10) переменная % h определяет DNS имя сервера. Именно поэтому на рис. 10.2 имя ресурса начинается с vmUnuntu10, как макроподстановка переменной %h. Полный список, возможных к использованию переменных, приведен в приложении 2 данной главы. Основными из них являются:

- % S — имя ресурса,
- % v — версия Samba,
- % T — текущая дата и время и т.д.

Вкладка Безопасность (Security), приведенная на рис. 10.11, содержит следующие параметры:

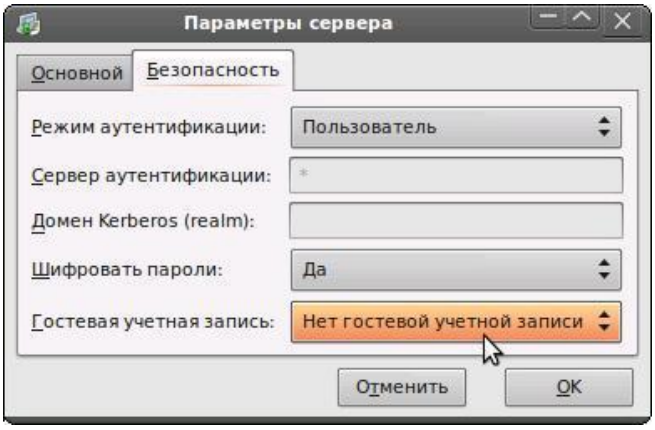

Рис. 10.11. Параметры сервера - Безопасность.

➢ Режим аутентификации. Этой настройке соответствует параметр security. Он может соответствовать одному из следующих типов проверки подлинности.

- ADS. Используется, когда сервер Samba работает как участник домена в сфере домена Active Directory (Active Directory Domain, ADS). Для этого на сервере должен быть установлен и настроен Kerberos, а Samba должна стать участником сферы ADS с помощью команды net, включенной в пакет samba-client.
- Домен (Domain). При этом сервер Samba проверяет пользователя, полагаясь на первичный и резервный контроллер домена Windows NT. Сервер контроллеру передает имя и пароль пользователя, и ждет от него ответа. В этом случае надо в поле «Сервер аутентификации» указать NETBIOS-имя первичного или резервного контроллера домена и параметр «Шифровать пароли» должен иметь значение Да.
- Сервер. Сервер Samba проверяет имя пользователя и пароль с помощью другого сервера Samba. Если это не удается, сервер пытается проверить вход в режиме проверки подлинности пользователя. В поле “Сервер аутентификации” надо указать NETBIOS-имя другого сервера Samba.
- Ресурс (Share). Сервер Samba не спрашивает имя и пароль, пока пользователи не попытаются подключиться к определённому общему каталогу этого сервера.
- Пользователь (User). Устанавливается по умолчанию. Пользователи должны вводить свои имя и пароль для сервера Samba. Этот вариант поддерживает работу параметра «Имя пользователя Windows». Подробности в разделе 10.2.3.

➢ Шифровать пароли (Encrypt Passwords). В этом случае пароли между клиентом и сервером передаются не открытым текстом (при этом их можно перехватить), а в зашифрованном виде

➢ Гостевая учетная запись (Guest Account). Когда обычные или гостевые пользователи подключаются к серверу Samba, им должен сопоставляться подходящий пользователь сервера. Выберите одного из существующих пользователей системы, который станет гостевой учетной записью Samba. Когда к серверу Samba подключаются гости, они получают те же привилегии, что и данный пользователь. Этой настройке соответствует параметр guest account.

После того как все параметры установлены и нажата кнопка OK, изменения сохраняются в файл конфигурации и «демон» перезапускается. Это приводит к тому, что изменения вступают в силу немедленно.

10.2.3. Управление пользователями Samba

Конфигуратору Samba требуется, чтобы перед добавлением пользователя Samba в системе, играющей роль сервера Samba, уже

существовала учетная запись пользователя. Пользователь Samba сопоставляется с существующей учетной записью пользователя.

Для входа в режим управления пользователями сервера Samba надо в окне конфигуратора (рис. 10.9) выбрать Настройка -> Пользователи Samba. Откроется окно, аналогичное окну, расположенному на заднем плане рис. 10.12. В левой части этого окна выводится список всех уже зарегистрированных пользователей сервера Samba.

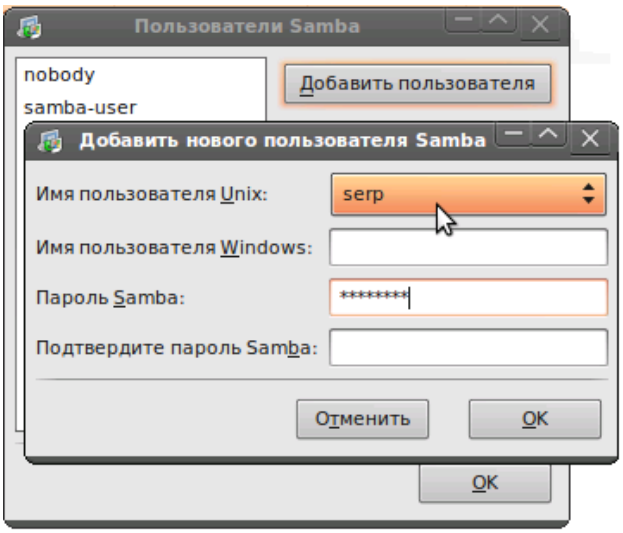

Рис. 10.12. Окно управления пользователями Samba.

Чтобы зарегистрировать нового пользователя, надо нажать кнопку Добавить пользователя. Появится новое окно, аналогичное тому, что приведено на переднем плане рис. 10.12.

В окне «Добавить пользователя» выберите «Имя пользователя Unix» из списка существующих в локальной системе пользователей. Если в системе Windows пользователь имеет другое имя и ему нужно подключаться из своей системы к серверу Samba, укажите «имя пользователя Windows» в поле «Имя пользователя Windows». Чтобы этот вариант работал, параметр Режим аутентификации на вкладке «Доступ окна Параметры сервера» должен иметь значение «Пользователь».

Также настройте Пароль Samba этого пользователя Samba и подтвердите его, набрав его еще раз. Даже если вы решили использовать для Samba зашифрованные пароли, рекомендуется назначать для всех пользователей Samba пароли, отличающиеся от их паролей в системе.

Чтобы поменять свойства существующего пользователя, выберите его из списка и нажмите кнопку «Изменить пользователя». Чтобы удалить существующего пользователя Samba, выберите его и нажмите кнопку Удалить пользователя.

При удалении пользователя Samba соответствующая учетная запись пользователя в системе не удаляется. Внесенные изменения вступают в силу сразу после нажатия кнопки OK.

10.2.4. Добавление ресурса

Чтобы создать новый общесетевой ресурс Samba, нажмите в основном окне конфигуратора Samba кнопку Добавить. Откроется окно «Создать ресурс Samba» (рис. 10.13). Для создания и описания нового ресурса требуется задать, выбрать или ввести ряд параметров.

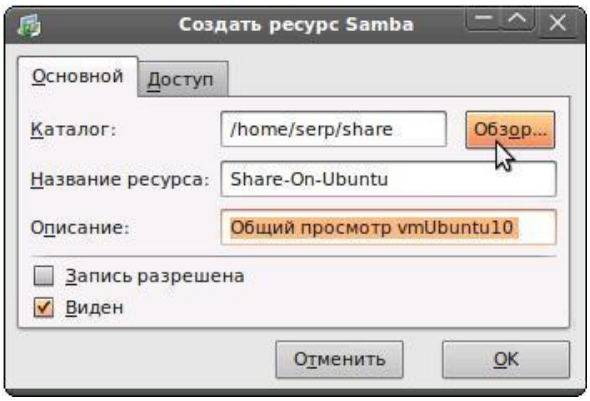

Рис. 10.13. Добавление нового Samba-ресурса.

На вкладке «Основной» при создании нового ресурса настраиваются следующие основные параметры:

- Каталог. Это каталог, который станет общим ресурсом Samba. Здесь указывается имя уже существующего каталога.
- Имя ресурса. Имя, с которым ресурс будет виден с удаленных компьютеров. По умолчанию это значение совпадает со значением поля Каталог, но его можно изменить.
- Описание. Краткое описание ресурса.
- Запись разрешена. Параметр, который определяет, будут ли пользователи иметь право только на чтение файлов в общем каталоге или смогут и читать, и писать файлы.
- Виден. Параметр, который определяет, будут ли ресурс виден в сетевом окружении Windows.

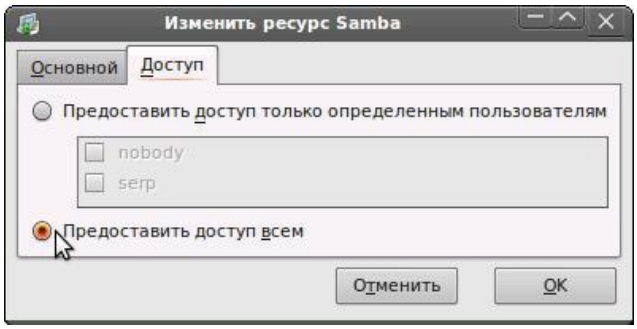

Рис. 10.14. Права пользователей на доступ к ресурсу.

На вкладке «Доступ» (рис. 10.14). следует определить, будет ли доступ к ресурсу разрешен только выбранным или всем пользователям Samba.

Если вы хотите разрешить доступ только определенным пользователям, выберите их из списка доступных пользователей Samba.

Ресурс добавляется сразу после нажатия кнопки OK и отображается в таблице конфигуратора Samba (рис. 10.15).

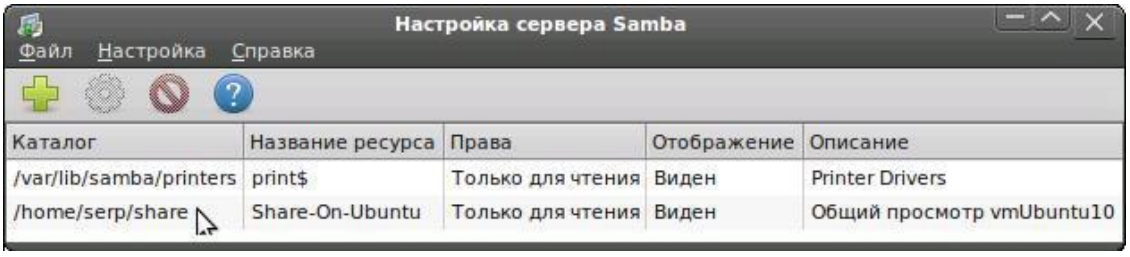

Рис. 10.15. Список всех ресурсов сервера Samba.

Выполним на основном компьютере просмотр ресурсов сети с помощью Сетевого окружения. При входе на vmUbuntu10 мы обнаружим два общесетевых ресурса, один из которых был создан средствами публикации папок, а второй является только что созданным ресурсом файлового сервера Samba (рис. 10.16).

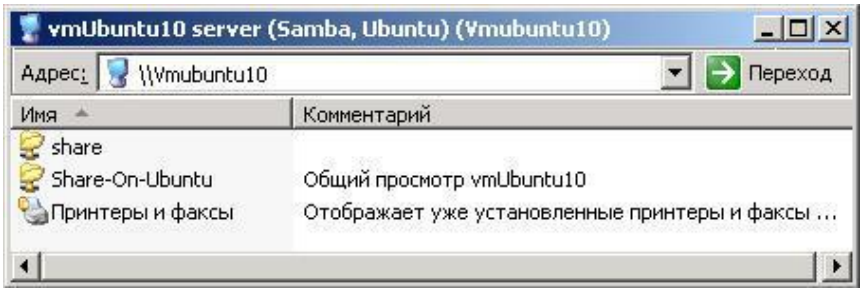

Рис. 10.16. Вид общедоступных ресурсов vmUbuntu10 с основного ПК.

10.2.5. Изменение параметров сервера

Просматривая параметры сервера (рис. 10.10) и Сетевое окружение Windows (рис. 10.2), видно, что vmUbuntu10 находится рабочей группе Workgroup. Однако все другие виртуальные машины — в группе Virtual-Net. Переведем vmUbuntu10 в эту же группу и изменим ее описание, чтобы познакомиться с использованием макроопределений (рис. 10.17).

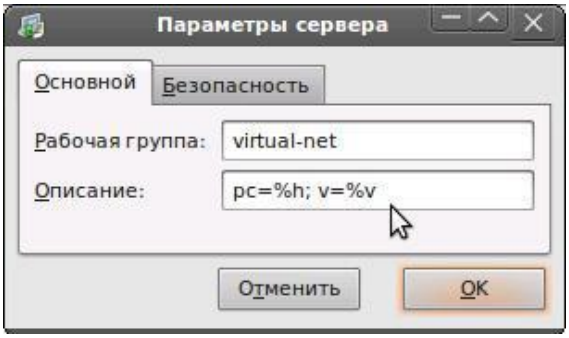

Рис. 10.17. Изменение параметров идентификации сервера Samba.

Как показала практика, эти изменения автоматически не обновляются. В Сетевом окружении Windows остается старая рабочая группа. Чтобы обновления вступили в силу, следует выполнить рестарт сервиса Samba. Для этого используется команда:

```
sudo service smbd restart
```

Практика работы с Ubuntu 10.04 в виртуальной сети показала, что в ряде случаев этого оказывалось недостаточно и приходилось перегружать Ubuntu-систему. После этого желаемый результат был налицо (рис. 10.18).

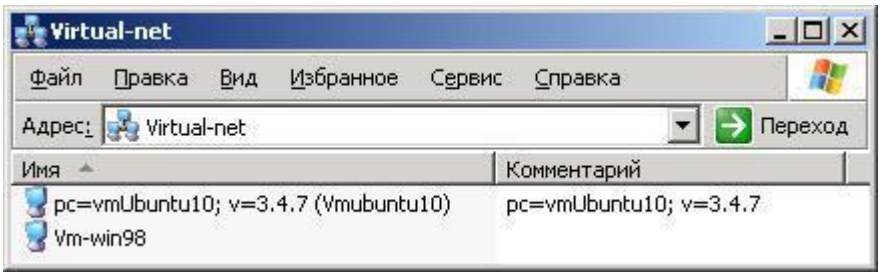

Рис. 10.18. Вид измененных параметров Samba в Сетевом окружении Windows.

В дальнейшем изложении будем использовать более простое описание сервера Samba для его идентификации. Поэтому, оставив рабочую группу Virtual-Net, в качестве описания примем SambaServer и перегрузим систему.

10.3. Базовая настройка файлового сервера Samba

Файловый сервер Samba, а тем более сама Samba — это сложная программная среда. Рассмотренная выше графическая утилита обладает ограниченными возможностями по конфигурации Samba-сервера и может работать только в графической среде Ubuntu. Но если организуется, даже в небольшой ЛВС, выделенный файл-сервер на базе Ubuntu Linux, то встают вопросы:

- о целесообразности использования графической среды Gnome,
- о возможности удаленного администрирования файл-сервера.

Рассмотрим одну из возможных сетевых технологий, которая позволит решить задачу построения выделенного файл-сервера Samba с возможность его удаленного администрирования.

Во-первых, эта технология предусматривает автоматическую загрузку ядра Linux Ubuntu-машины по включению питания. При этом будет отсутствовать запуск графической среды и какой-либо, административный или пользовательский, вход на Ubuntu-машину.

Во-вторых, организуется удаленное подключение по telnet или SSH с любого другого компьютера к Ubuntu-машине, на которой реализован файл-сервер. В случае нашей виртуальной сети это может быть, например,

вход на vmUbuntu10 с основного компьютера Main-PC по протоколу SSH с использованием Windows-утилиты PuTTY (рис. 10.19).

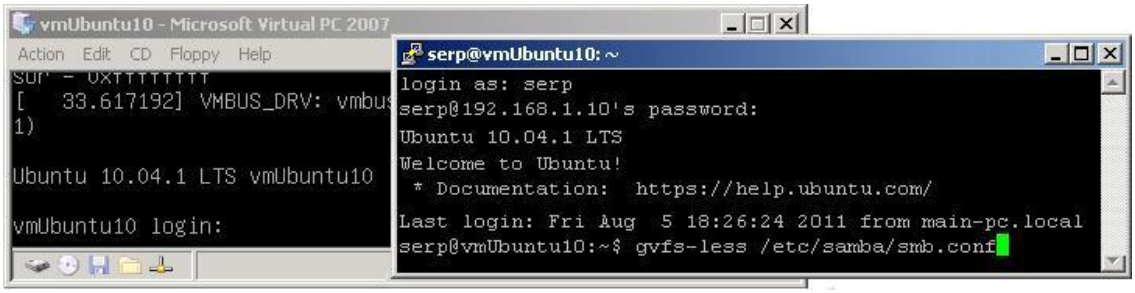

Рис. 10.19. Удаленный доступ с Main-PC через PuTTY к vmUbuntu10, загруженной без Gnome и без входа пользователя.

Следует отметить, что, работая на одном компьютере с несколькими виртуальными машинами, отсутствие загрузки графической среды на одном из них (например, vmUbuntu10), существенно повышает скорость работы по конфигурированию как всей виртуальной сети, так и файл-сервера Samba, в частности. Это возможно в том случае, если удаленная работа с vmUbuntu10 будет организована в терминальном доступе с использованием утилиты PuTTY.

При этом дополнительное удобство можно получить, используя утилиту Midnight Commander, которую можно заранее установить на vmUbuntu10. Тогда при работе внутри PuTTY будет доступна консольная команда **mc** (рис. 10.20).

Рис. 10.20. Удаленный доступ к vmUbuntu10 с основного Main-PC через PuTTY с запуском Midnight Commander.

Однако мы не будем уделять время на возможные технологии доступа к файловому серверу Samba. Основная цель данного раздела — дать начальное представление о создании, настройке и администрировании сервера Samba. Поэтому далее вы можете использовать любую удобную вам технологию: работать в графической или текстовой среде vmUbuntu10, работать удаленно в командной строке или в среде Midnight Commander.

Дальнейшее изложение будет базироваться на просмотре и редактировании конфигурационных файлов Samba, выполнении тех или иных команд. В какой среде вы будете их выполнять, значения не имеет, важно познакомиться с базовыми настройками файлового сервера Samba.

10.3.1. Основы настройки сервера Samba

Конфигурационный файл для сервера Samba ver.3 находится в папке /etc/samba и называется smb.conf. Он там появился после установки пакета Samba на вашем компьютере. Исходный вид этого файла имеет вид, приведенный в приложении 1 к данной главе.

Он имеет несколько секций, каждая из которых отвечает за конкретный набор параметров настройки файлового сервера:

```
#
#================= Global Settings ===================
[global]
## Browsing/Identification ###
 . . .
####      Networking       ####
 . . .
#### Debugging/Accounting ####
 . . .
####### Authentication #######
 . . .
########## Domains ###########
 . . .
########## Printing ##########
 . . .
############ Misc ############
 . . .
#================= Share Definitions ==================
[ . . . ]

[ . . . ]
```

Секция [global] определяет общие настройки серверной части Samba в целом для всех ресурсов.

- workgroup: имя рабочей группы;
- security: уровень определения прав доступа на уровне пользователей;
- os level: приоритет данного сервера среди других компьютеров рабочей группы: определяет, кто именно будет главной машиной, отвечающей за отображение ресурсов сети, для сравнения, у Win9X os level = 34, а у NT4 os level = 64;

• domain master: определяет мастер-сервер домен, в данном случае директива отключена;

• domain logons: если вы не планируете вводить вашу Ubuntu-машину в уже существующий домен Windows, то надо использовать domain logons = no;

• wins support: обычно в простейшей сети WINS не нужен, мы его отключаем и у себя тоже;

• comment: комментарий, видимый в сети как комментарий к ресурсу;

• path: путь к каталогу ресурса;

• public: отметка о доступе на чтение всем авторизованным пользователям (в том числе и гостевым, если они определены);

• writable: запрещение работы на запись всем пользователям;

• write list = @staff: разрешение работы на запись всем пользователям, входящим в системную группу staff.

С основными секциями и параметрами мы будем постепенно знакомиться. Нам придется в дальнейшем вносить изменения в этой файл, но мы рекомендуем вам править не этот файл, а создать для себя свой новый, аналогичный исходному.

Сохраните файл /etc/samba/smb.conf в качестве резервной копии /etc/samba/smb.conf.default. Это нужно на случай, если все станет очень плохо и придется откатиться в начало. Кроме того, там есть много интересного, что следует почитать. И вообще, заведите себе привычку перед любым изменением любого конфигурационного файла делать его резервную копию.

А сейчас давайте внимательно посмотрим на два файла /etc/samba/smb.conf. На тот, который приведен в Приложении 10.1 и на тот, который находится сейчас на виртуальной машине vmUbuntu10.

Если смотреть внимательно, то уже в начальных секциях можно обнаружить расхождения:

Текущий файл /etc/samba/smb.conf :

```
#================= Global Settings =================
[global]
## Browsing/Identification ###
# Change this to the workgroup/NT-domain name your Samba
server will part of
   workgroup = virtual-net
# server string is the equivalent of the NT Description
field
   server string = SambaServer
```

Исходный файл /etc/samba/smb.conf из Приложения 10.1:

```
#================= Global Settings =================
```

```
[global]
## Browsing/Identification ###
# Change this to the workgroup/NT-domain name your Samba
server will part of
   workgroup = WORKGROUP
# server string is the equivalent of the NT Description
field
   server string = %h server (Samba, Ubuntu)
```

Из этого сравнения становится ясно, почему после установки Samba компьютер vmUbuntu10 по умолчанию являлся членом рабочей группы WORKGROUP и имел описание (server string), как на рис. 10.2 и 10.10.

Но на данный момент, как следует из текущего файла /etc/samba/smb.conf, компьютер vmUbuntu10 — член рабочей группы virtual-net и имеет описание SambaServer. То есть, работая с графическим конфигуратором Samba, мы занимались не чем иным, как изменением параметров файла /etc/samba/smb.conf, который во многом и отвечает за работу файлового сервера Samba.

Сравним самый конец этих двух файлов. Здесь явно видно различие, которое заключается в том, что в текущем файле /etc/samba/smb.conf, в отличие от исходного файла, появилась новая секция, которой нет в исходном файле конфигурации, а именно:

```
[Share-On-Ubuntu]
    comment = Общий просмотр vmUbuntu10
    path = /home/serp/share
;   writeable = No
;   browseable = yes
    guest ok = yes
```

Сравните содержание этой секции с рис. 10.13 и 10.14. Из сравнения вам должно быть ясно, что и здесь графический конфигуратор, а точнее утилита «Настройка сервера Samba», изменила вид /etc/samba/smb.conf для описания нового общего ресурса сервера Samba, который:

- Имеет название [Share-On-Ubuntu] и дополнительное описание «Общий просмотр vmUbuntu10» (см. рис. 10.15 и 10.16).
- Соответствует каталогу /home/serp/share на Ubuntu-машине vmUbuntu10 (см. рис. 10.13 и 10.15).
- Запись в этот ресурс не разрешена (см рис. 10.13), и это определяется строкой параметра:

```
;       writeable = No
```

Точка с запятой перед строкой параметра writeable обозначает, что это комментарий, то есть не используется при чтении файла /etc/samba/smb.conf сервисом Samba при его старте или рестарте.

Указанное в этой строке значение параметра No является значением, которое устанавливает сервис Samba параметру writeable по умолчанию. Переприсваивать параметру значение, которое и так устанавливается по умолчанию — бессмысленно, поэтому и комментарий, то есть точка с запятой.

• Ресурс является видимым с других узлов сети (см. рис. 10.13), что определяется соответствующим значением параметра browseable.

• И, наконец, этот ресурс доступен по гостевому доступу (см. рис. 10.14).

Другими словами, мы убедились, что графический конфигуратор, есть не что иное, как окно к файлу /etc/samba/smb.conf. Причем это окно очень маленькое, и чтобы успешно работать по настройке и администрированию сервера Samba надо, как минимум, достаточно хорошо разбираться во всех секциях файла /etc/samba/smb.conf.

Чтобы выполнять те или иные изменения в файле /etc/samba/smb.conf, достаточно его открыть любым из тактовых редакторов с правами root. При начальном знакомстве с этим файлом хотелось бы обратить внимание еще на несколько глобальных параметров:

• Параметр security. Если у вас сеть строиться по принципу клиент/сервер, то нужно установить значение параметра server, при использовании одноранговой сети, то есть сети без выделенного сервера, следует для параметра security выбрать значение user или share.

По умолчанию это значение:

```
# security = user
```

• Обратите внимание и на настройки кодировок для правильного отображения кириллицы:

```
client code page = 866
character set = utf8
```

• Параметр interfaces указывает интерфейсы, на которых должен работать сервис Samba. Следует указать тот интерфейс, который связывает вашу машину с Windows-сетями. В примере нашей сети это:

```
interfaces = 192.168.1.10/24
```

• Параметр socket options позволяет изменять сетевые настройки работы файлового сервера. В надежной высокоскоростной сети можно оптимизировать работу Samba-сервера используя следующий формат

```
socket options = TCP_NODELAY IPTOS_LOWDELAY SO_KEEPALIVE
SO_RCVBUF=8192 SO_SNDBUF=8192
```

В качестве тренировки попробуйте ваш текущий файл /etc/samba/smb.conf привести к исходному состоянию. Но при этом не

забудьте сохранить копию этого файла. Протестируйте сеть и посмотрите, к чему это привело.

Далее рассмотрим два примера по настройке различных конфигураций сервера Samba. В основе данного раздела лежит материал из книги Джона Терпстры «Samba-3 в примерах» — «John Terpstra. Samba-3 by Example». Эту книгу можно найти на сайте разработчиков Samba www.samba.org или в русском переводе на http://samba-doc.ru/.

10.3.2. Простой Samba-сервер: доступ всем на всё

В этом примере речь пойдет о том, как организовать простой файловый сервер с беспарольным доступом к общим ресурсам. То есть можно будет «всем всё» на основе материалов книги Samba-3 by Example.

Предположим, что в небольшой рекламной фирме идет совместная работа по разработке новых буклетов, содержащих графические баннеры и фотоматериалы. Требуется организовать файловый сервер Samba с двумя общими ресурсами:

- foto — в котором будут храниться графические материалы всех пользователей нашего файлового сервера;
- общие проекты — где сотрудники могут централизованно обмениваться информацией по проекту.

При конфигурировании данного сервера будет использован параметр force user. Использование это параметра гарантирует, что файлы будут принадлежать одному и тому же идентификатору пользователя (user identifier UID), и поэтому никогда не будет проблем с доступом к файлам.

Другими словами, это значит, что мы заведем на файл-сервере одного пользователя, который автоматически будет становиться владельцем всех файлов и папок общесистемных ресурсов. Именно имя и пароль этого пользователя и будут служить всем пользователям правами доступа к файловому серверу Samba.

При таком подходе последовательность организации и настройки сервера может быть аналогичной описанной ниже.

Все последующие операции должны будут выполняться от имени суперпользователя, то есть с правами root. Чтобы перейти в этот режим, выполняем команду:

```
serp@vmUbuntu10:~$ sudo -i
root@vmUbuntu10:~#
```

Замечание.
Напомним: чтобы вернуться из сессии суперпользователя в сессию текущего пользователя, надо использовать команду exit.

1. Создание владельца общих ресурсов

Таким пользователем будет являться, например, пользователь с именем — samba-user. А пароль этого пользователя пусть будет samba-psw.

При создании нового пользователя зададим каталог пользователя (ключ -m) и его пароль (ключ -p):

```
~# useradd -m samba-user -p samba-psw
```

Убедимся, что пользователь samba-user добавился в систему, и для этого пользователя создан его домашний каталог:

```
root@vmUbuntu10:~# cat /etc/passwd | grep samba-user
samba-user:x:1001:1001::/home/samba-user:/bin/sh
```

```
root@vmUbuntu10:~# ls -l  /home | grep samba-user
drwxr-xr-x  2 samba-user samba-user 4096 2011-08-06 17:38
samba-user
```

2. Создание папки для общесистемных ресурсов

Теперь создадим папку, например server, в которой будут находиться наши будущие общие ресурсы. Можно обойтись и без этой папки, но удобнее, если все общие папки будут лежать в отдельном каталоге.

```
~# mkdir /home/server
```

Теперь создадим папки для общесистемных ресурсов:

```
~# mkdir /home/server/{foto,"Общие проекты"}
```

3. Назначение владельца общесетевых ресурсов

В качестве владельца папок для общесетевых ресурсов назначим пользователя с именем sumber-user из группы users.

```
~# chown -R samba-user:users /home/server
```

и изменим этому пользователю разрешения на доступ к папке общесетевых ресурсов:

```
~# chmod -R ugo+rwx samba-user:users /home/server
```

В итоге разрешения на общие ресурсы будут иметь следующий вид:

```
root@vmUbuntu10:~# ls -l  /home/server
  итого 8
  drwxrwxrwx 2 samba-user users 4096 2011-08-06 18:12 foto
  drwxrwxrwx 2 samba-user users 4096 2011-08-06 18:12
Общие проекты
```

Если вы работаете удаленно и в своей работе используете Midnight Commander, то вид его на текущий момент может иметь вид, аналогичный рис. 10.21.

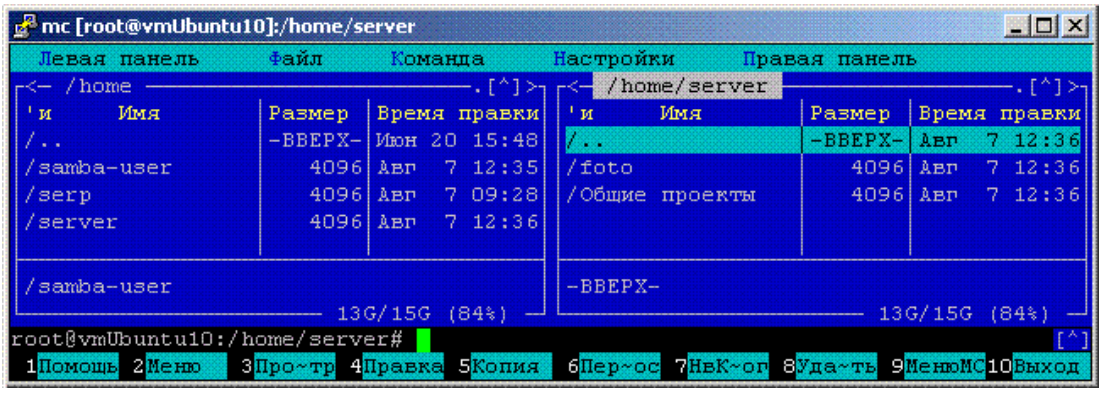

Рис. 10.21. Окно PuTTY с Midnight Commander и вновь созданные каталоги.

4. Конфигурирование сервера Samba

Если вы еще не сохранили исходный конфигурационный файл, то выполните команду

```
~# mv /etc/samba/smb.conf /etc/samba/smb.conf.default
```

Следует отметить, что разработчики Samba настоятельно рекомендуют иметь файл smb.conf как можно меньшего размера, а также не иметь внутри файла комментариев. Они же рекомендуют, как можно более тщательно, комментировать все изменения, которые вносятся в файл.

Данные противоречивые рекомендации выполним следующим образом: создадим файл smb.conf.comment, в котором подробно опишем все, что мы делаем, а потом получим из него рабочий файл smb.conf:

- Создаем файл /etc/samba/smb.conf.comment, используя команду

```
~# touch /etc/samba/smb.conf.comment
```

- Заполним файл smb.conf.comment примерно таким содержимым:

```
[global]
#название рабочей группы, в windows по умолчанию WORKGROUP
workgroup = Virtual-Net
#NetBIOS-имя, под которым файл-сервер будет отображаться в
сетевом окружении
netbios name = vmUbuntu10
#режим безопасности - вход на файл-сервер доступен Ubuntu-
пользователям
security = USER
#строка описания файл-сервера в сетевом окружении
server string = SambaServer

#описание общесетевых ресурсов
[foto]
#строка описания этого ресурс в сетевом окружении
comment="Фото и графика"
```

```
#путь к этой общедоступной папке
path=/home/server/foto
#эти параметры необязательны, для чего - будет ясно ниже
create mask = 0777
directory mask = 0777
#принуждаем быть владельцем общего ресурса
#пользователя samba-user
force user=samba-user
#указываем, что можно не только читать, но и записывать
read only=No

[Общие проекты]
comment="Новые буклеты"
path="/home/server/Общие проекты"
create mask = 0777
directory mask = 0777
force user=samba-user
read only=No
```

➢ Используя команды testparm, получим рабочий файл smb.conf, а заодно проверим, нет ли в нем ошибок:

```
~# testparm -s /etc/samba/smb.conf.comment >
/etc/samba/smb.conf
Load smb config files from smb.conf.comment
rlimit_max: rlimit_max (1024) below minimum Windows limit
(16384)
Processing section "[foto]"
Processing section "[Общие проекты]"
Loaded services file OK.
 . . .
```

➢ Создан файл smb.conf, в котором нет ни одной строки с комментариями. Утилита testparm проверяет в основном опечатки, правильность заданных параметров она почти не контролирует. Например, если в написании параметра «workgroup» будет допущена ошибка — «workgroup», то сообщение утилиты testparm будет выглядеть следующим образом:

```
root@vmUbuntu10:/etc/samba# testparm -s smb.conf.comment >
smb.conf
Load smb config files from smb.conf.comment
rlimit_max: rlimit_max (1024) below minimum Windows limit
(16384)
Unknown parameter encountered: "wrkgroup"
Ignoring unknown parameter "wrkgroup"
 . . .
```

➢ После исправления всех опечаток итоговый файл smb.conf будет иметь вид:

```
[global]
        workgroup = VIRTUAL-NET
        server string = SambaServer
        security = USER

[foto]
        comment = "Фото и графика"
        path = /home/server/foto
        force user = samba-user
        read only = No
        create mask = 0777
        directory mask = 0777

[Общие проекты]
        comment = "Новые буклеты"
        path = "/home/server/Общие проекты"
        force user = samba-user
        read only = No
        create mask = 0777
        directory mask = 0777
```

5. Запуск сервера Samba

Чтобы все настройки файлового сервера Samba вступили в силу, необходимо перезапустить сервис Samba, используя в Ubuntu 10.04 для этого команду:

```
#...service smbd restart
root@vmUbuntu10:/home# service smbd restart
smbd start/running, process 1144
```

В более ранних версиях Ubuntu для этой цели использовалась команда

```
#.../etc/init.d/samba restart
```

6. Локальная проверка работы сервера Samba

Для проверки используем smbclient и подключимся сами к себе, используя парольный доступ к файловому серверу:

```
root@vmUbuntu10:~# smbclient -L //192.168.1.10 -U samba-
user%samba-psw
Domain=[VIRTUAL-NET] OS=[Unix] Server=[Samba 3.4.7]
```

```
      Sharename       Type      Comment
      ---------       ----      -------
      foto            Disk      Фото и графика
      Общие проекты   Disk      Новые буклеты
      IPC$            IPC       IPC Service (SambaServer)
      share           Disk
Domain=[VIRTUAL-NET] OS=[Unix] Server=[Samba 3.4.7]

      Server               Comment
      ---------            -------
      VMUBUNTU10           SambaServer

      Workgroup            Master
      ---------            -------
      HOME-NET             MAIN-PC
      VIRTUAL-NET          VMUBUNTU10
```

7. Проверка работы сервера Samba с Windows-машины

При первом подключении с Сетевого окружения Windows к файловому серверу Samba буден выдан запрос на аутентификацию пользователя аналогично рис. 10.22.

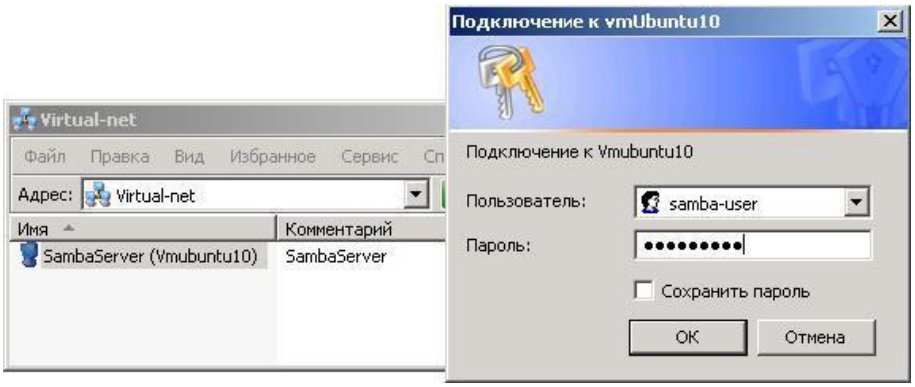

Рис. 10.22. Доступ к файловому серверу Samba из сетевого окружения Windows.

После успешной аутентификации на Windows-машине будут доступны общесистемные ресурсы vmUbuntu10 (рис. 10.23).

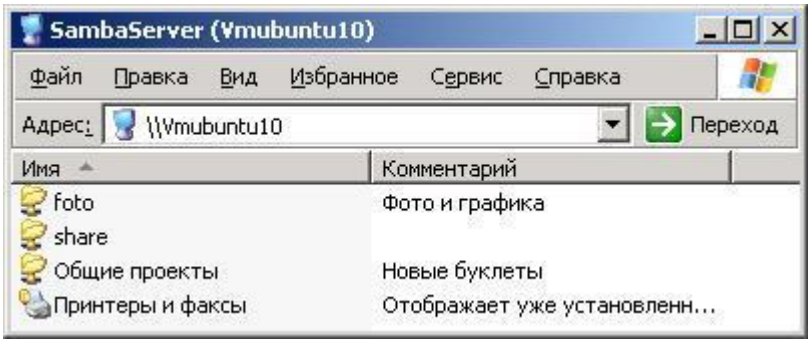

Рис. 10.23. Общие ресурсы vmUbuntu10, доступные с Main-PC.

Аналогично, к ресурсам vmUbuntu10 будет доступ и в Сетях Windows на виртуальной машине vmUbuntu06 (рис. 10.24).

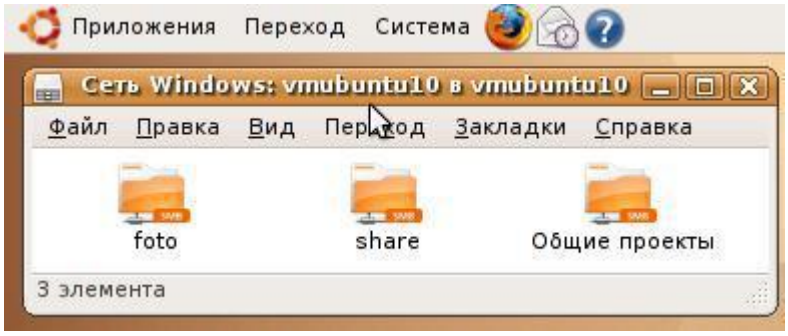

Рис. 10.24. Общесистемные ресурсы vmUbuntu10, доступные с vmUbuntu06.

10.3.3. Samba-сервер в одноранговой сети: персональные общие папки

Привлекательность данного примера в том, что он может быть реализован в небольшой офисной одноранговой сети из 15–20 компьютеров, где клиентами являются компьютеры с ОС Windows, а в качестве файлового сервера используется сервер Samba. Особенность реализованного подхода состоит в том, что:

- Для каждого пользователя (или группы пользователей с одинаковым паролем и логином) заводится личная папка на файл-сервере.
- Данная папка может подключаться как сетевой диск.
- Одна группа пользователей не может иметь доступ в папки других пользователей.
- Один из пользователей, например, руководитель подразделения, может просматривать папки всех пользователей.
- В сети нет сервера dhcp, и все адреса статические.

Пусть исходными данными для рассматриваемого примера будет небольшая ЛВС кафедры, реализованная на базе нашей виртуальной сети.

Таблица 10.1.

Перечень пользователей и компьютеров в составе ЛВС

Пользователь	Учетная запись	Пароль	Личная папка	Каталог на сервере	Рабочая станция
Зав. кафедрой	zam	zam-psw	zam	/kafedra	Main-PC
Лаборатория_2	lab2	lab2-psw	lab2	/kafedra/lab2	vm-WinXP
Лаборатория_3	lab3	lab3-psw	ab3	/kafedra/lab3	vmUbuntu06

В этом случае, одна из возможных последовательностей организации и настройки файлового сервера для небольшой ЛВС, структура которой приведена в табл. 10.1, может иметь вид, который представлен ниже.

1. Создание пользователей файл-сервера

Вначале создадим на файл-сервере группу, в которую потом будем включать всех сотрудников кафедры.

```
serp@vmUbuntu10:~$ sudo -i
[sudo] password for serp:
root@vmUbuntu10:~# groupadd kafedrastaff
```

Создадим новых unix-пользователей файл-сервера, входящих в только что созданную группу kafedrastaff. Для этого используем команду:

useradd -m -G <группа> -c <коммент> <ИмяУчетнЗаписиПользователя>

со следующим набором опций:

m — если домашнего каталога не существует, то он будет создан,

G — список дополнительных групп, в которых числится пользователь,

C — комментарий, строка для идентификации учетной записи.

```
root@vmUbuntu10:~# useradd -m -G kafedrastaff -c
"Зав.кафедрой" zam

root@vmUbuntu10:~# passwd zam
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Повторяем эту операцию для всех пользователей.

Для каждого нового unix-пользователей определяем его пароль, как пользователя Samba сервера.

```
root@vmUbuntu10:~# smbpasswd -a zam
New SMB password:
Retype new SMB password:
Added user zam.
```

Повторяем эту операцию для всех пользователей.

2. Создание общих ресурсов и назначение прав

Теперь создадим папки и раздадим права на доступ к ним следующим образом:

```
root@vmUbuntu10:~# mkdir -p /kafedra
root@vmUbuntu10:~# chown zam /kafedra

root@vmUbuntu10:~# for i in lab2 lab3
> do
> mkdir -p /kafedra/$i
```

```
> chown $i /kafedra/$i
> done

root@vmUbuntu10:~# chgrp -R kafedrastaff /kafedra
root@vmUbuntu10:~# chmod -R ug+rwx,o-r+x /kafedra
```

Проверим для каждой папки будущих общесистемных ресурсов, какие получились разрешения и кто их владельцы.

```
root@vmUbuntu10:~# ls -l / | grep kafedra
drwxrwx--x   4 zam  kafedrastaff  4096 2011-08-07 16:57
kafedra

root@vmUbuntu10:~# ls -l /kafedra
итого 8
drwxrwx--x 2 lab2 kafedrastaff 4096 2011-08-07 16:57 lab2
drwxrwx--x 2 lab3 kafedrastaff 4096 2011-08-07 16:57 lab3
```

3. Конфигурирование сервера Samba, запуск и проверка в работе

Файл /etc/samba/smb.conf заполняем примерно таким содержимым:

```
[global]
   workgroup = VIRTUAL-NET
   server string = ServerKafedra
   log file = /var/log/samba/log.%m
   max log size = 50

[files]
   comment = "наши данные"
   path = /kafedra/%U
   read only = No

[boss]
   comment = "для начальства"
   path = /kafedra
   valid users = zam
   read only = No
```

Запускаем сервер Samba. Чтобы все настройки файлового сервера Samba вступили в силу, необходимо перезапустить сервис Samba, используя для этого в Ubuntu 10.04 команду:

```
~# service smbd restart
root@vmUbuntu10:/home# service smbd restart
smbd start/running, process 1144
```

в более ранних версиях Ubuntu для этой цели использовалась команда:

```
~# /etc/init.d/samba restart
```

➢ Локальная проверка возможности подключения к серверу Samba.

Для проверки используем smbclient и тестируем возможность подключения к общему ресурсу пользователем zam:

```
root@vmUbuntu10:/etc/samba# smbclient -L
//192.168.1.10/boss -U zam
Enter zam's password:
Domain=[VIRTUAL-NET] OS=[Unix] Server=[Samba 3.4.7]
    Sharename       Type      Comment
    ---------       ----      -------
    files           Disk      наши данные
    boss            Disk      для начальства
    IPC$            IPC       IPC Service (ServerKafedra)

Domain=[VIRTUAL-NET] OS=[Unix] Server=[Samba 3.4.7]
    Server               Comment
    ---------            -------
    VMUBUNTU10           SambaServer

    Workgroup            Master
    ---------            -------
    HOME-NET             MAIN-PC
    VIRTUAL-NET          VMUBUNTU10
```

А теперь попробуем войти как пользователь zam, который имеет полный доступ, в общий ресурс и посмотреть, что у нас там есть:

```
root@vmUbuntu10:/etc/samba# smbclient  //192.168.1.10/boss
-U zam
Enter zam's password:
Domain=[VIRTUAL-NET] OS=[Unix] Server=[Samba 3.4.7]
smb: \> dir
  .           D        0  Sun Aug  7 16:57:52 2011
  ..          D        0  Sun Aug  7 16:55:24 2011
  lab2        D        0  Sun Aug  7 16:57:52 2011
  lab3        D        0  Sun Aug  7 16:57:52 2011

 61621 blocks of size 262144. 48652 blocks available
smb: \>
```

Так мы можем подключиться не только локально, но и из любой другой Linux-системы.

4. Подключение Ubuntu-клиентов

Итак, наш файловый сервер Samba реализован на виртуальной машине vmUbuntu10. Он полностью настроен и проверен утилитой smbclient в локальном режиме доступа.

Поставим теперь задачу проверки доступа к серверу Samba с другого Linux-клиента нашей виртуальной сети. Напомню, что в данном примере эта сеть моделирует ЛВС кафедры.

На основе табл. 10.1 таким клиентом может быть только компьютер лаборатории 3, так как именно на нем установлена Linux подобная Ubuntu 6.10. Другими словами, это виртуальная машина vmUbuntu06. Именно с нее мы и попробуем добраться до общедоступных сетевых ресурсов, находящихся на файловом сервере, организованном на vmUbuntu10.

С этой целью на компьютере vmUbuntu06, используя основное меню, последовательно выбираем «Переход» -> «Подключение к серверу». Появится диалоговое окно, в котором в качестве типа сервиса выбираем «Ресурс ОС Windows», а затем указываем адрес сервера Samba и пароль для входа. В данном примере пароль для входа – это lab3-psw. И получаем доступ к серверу (рис. 10.25).

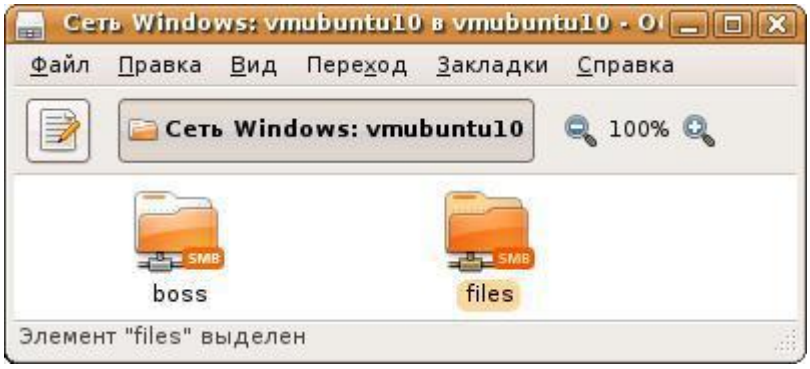

Рис. 10.25. Удаленный доступ к серверу кафедры с vmUbuntu06 лаборатории № 3.

При этом доступной нам является только папка files, заходя в которую и создавая там файлы и папки, мы фактически работаем в каталоге Samba-сервера /kafedra/lab3. Доступ к папке boss для нас закрыт. При попытке входа в нее от нас требуют дополнительную авторизацию с запросом логина и пароля пользователя этой папки.

5. Подключение Windows-клиентов

Создаем на Windows-машинах обычными средствами Windows пользователей и пароли такие же, как прописаны им сервере Samba. После этого в сетевом окружении мы видим компьютер ServerKafedra, а в нем две общие папки — boss и files. На компьютере лаборатории № 2 ситуация аналогичная компьютеру лаборатории № 3 (рис. 10.26).

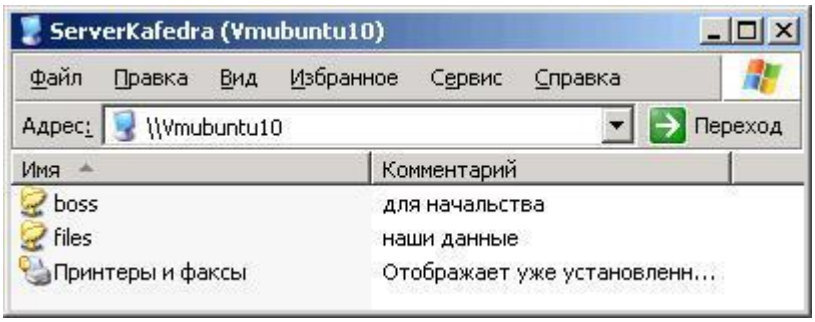

Рис. 10.26. Удаленный доступ к серверу кафедры с vm-WinXP лаборатории № 2.

Принципиальное различие заключается в том, что, работая в доступной им папке files, заходя в которую и создавая там файлы и папки, они фактически работают в каталоге Samba-сервера /kafedra/lab2.

Что касается компьютера заведующего кафедрой (рис. 10.27), то он не может войти в папку files по той причине, что она введет в папку /kafedra/имя_пользователя, а для зав. кафедрой мы такой личной папки не создавали. Зато из папки boss он может попасть в личные папки своих сотрудников.

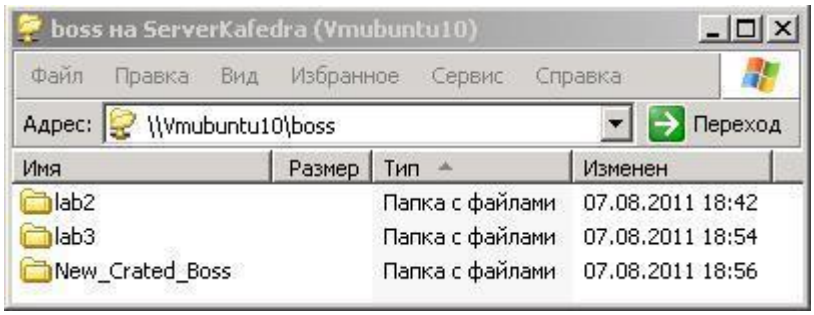

Рис. 10.27. Удаленный доступ к серверу кафедры с Main-PC администрации.

Сотрудники же лабораторий № 2 и № 3, попадают в папку files. Она является соответственно папками /kafedra/lab2 и /kafedra/lab3 на файл-сервере. В папку boss сотрудники лабораторий могут попасть, только если им станет известен пароль заведующего кафедрой.

10.3.4. Алиасы имен Samba пользователей

Попробуем еще раз взглянуть на соответствие имен Unix- и Samba-пользователей. Samba может использовать имена пользователей, которые уже зарегистрированы в Ubuntu-системе, то есть имеет доступ к этому узлу сети с теми или иными правами.

Так, например, Unix-пользователь с именем user, уже есть в Ubuntu-системе. Для того чтобы он имел доступ к файловому серверу Samba этой же системы, его надо внести в базу данных SMB и назначить пароль для доступа к общесистемным ресурсам. Кроме того, в конфигурации Samba надо для этого, теперь уже Samba-пользователя, определить список доступных для него ресурсов и права на них.

Один из возможных подходов был рассмотрен ранее, но возможен и другой подход на создание и добавление пользователей в Samba, при котором в файл /etc/samba/smb.conf добавляются две строки:

```
security = user
username map = /etc/samba/smbusers
```

Данные операции заставят Samba искать список пользователей в файле /etc/samba/smbusers. При таком подходе Samba пользователь будет создаваться в два этапа. Сначала, используя команду smbpasswd, создаем пароль для пользователя, который позже будет добавлен в файл списка пользователей Samba.

```
sudo smbpasswd -a <username>
```

А затем добавляем этого пользователя в файл пользователей Samba. С этой целью вызываем на редактирование файл /etc/samba/smbusers:

```
~# sudo gedit /etc/samba/smbusers
```

и добавляем в файл новую строку вида:

```
<unix user> = "<samba user1>" "<samba user2>"
```

В этой строке вместо user подставляем реальные имена Unix- и Samba-пользователей.

При таком подходе появляется возможность создать алиас для имени пользователя. Это может облегчить доступ с Windows-машины, на которой есть, например, пользователь с именем admin. Так, например, при инсталляции Samba файл /etc/samba/smbusers содержит два элемента:

```
# Unix_name = SMB_name1 SMB_name2 ...
root = administrator admin
nobody = guest pcguest smbguest
```

Первый элемент отображает два пользовательских имени из Windows (administrator и admin) на пользовательское имя root из Ubuntu. Второй элемент отображает три пользовательских имени из Windows, в том числе и guest, на пользовательское имя nobody из Ubuntu. Когда пользователь Windows пытается войти в сервер Samba с именем guest, Samba аутентифицирует пользователя Ubuntu с именем nobody.

В качестве примера и тренажа попробуйте использовать этот подход для рассмотренных ранее примеров по настройке файлового сервера Samba.

10.3.5. Полезные команды администрирования файл-сервера Samba

Мы уже несколько раз отмечали, что пакет Samba является достаточно мощным программным средством организации, конфигурирования и

поддержки файлового сервера Linux-систем. В его состав входит большое количество дополнительных команд и утилит по управлению сервером Samba.

Приведем, базируясь на их манах, совсем краткие описания и назначения основных утилит пакета Samba:

- **smbd —** предоставляет клиентам сервисы SMB (иначе LanManager)

  Синтаксис

  ```
  smbd [ -D ] [ -a ] [ -d уровень_отладки ]
       [ -l файл_протоколирования ]  [ -p номер_порта ]
       [ -O опции_socket'ов ] [ -s конфигурационный_файл ]
  ```

  Описание

  Это сервер, который может предоставлять сервисы SMB. Сервисами являются файловый сервер и сервер печати. Они предоставляются клиентам, которые используют SMB протокол. Этот протокол совместим с протоколом LanManager и с успехом может применяться клиентами LanManager, куда включаются MSCLIENT 3.0 для DOS, Windows for Workgroups, Windows 95, Windows NT, OS/2, DAVE для Macintosh, и smbfs для Linux.

- **smbstatus** — отчет о текущих соединениях Samba.

  Синтаксис

  ```
  smbstatus [-P] [-b] [-d <debug level>] [-v] [-L] [-B]
       [-p] [-S] [-s <configuration file>] [-u <username>]
  ```

  Описание

  Smbstatus это очень простая программа для вывода списка соединений Samba. Она может показать, какая машина и какой ресурс на Samba сервере в текущее время использует.

- **findsmb —** выводит список ресурсов сети, которые реагируют на запрос об SMB-имени.

  Синтаксис

  ```
  findsmb [subnet=broadcast=address]
  ```

  Описание

  Это perl скрипт, который отображает имеющиеся в сети общесетевые Windows-ресурсы. Для получения информации он использует обращение к nmblookup и smbclient.

- **smbcontrol** — утилита для отправки сообщений процессам smbd, nmbd и winbindd

  Синтаксис

  ```
  smbcontrol [-i] [-s]
  smbcontrol [destination] [message-type] [parameter]
  ```

  Описание

Эта утилита часть пакета samba, очень маленькая программа, отправляющая сообщения процессам smbd, nmbd или winbindd, запущенным в системе.

- **smbget —** подобно wget может получить файлы с windows-машин через smb-протокол

Синтаксис

```
smbget [-a,=--guest] [-r,=--resume] [-R,=--recursive]
       [-u,=--username=STRING] [-p,=--password=STRING]
       [-w,=--workgroup=STRING] [-n,=--nonprompt]
       [-d,=--debuglevel=INT] [-D,=--dots]
       [-P,=--keep-permissions] [-o,=--outputfile]
       [-f,=--rcfile] [-q,=--quiet] [-v,=--verbose]
       [-b,=--blocksize] [-?,=--help] [--usage]
       {smb://host/share/path/to/file} [smb://url2/] [...]
```

Описание

Это простая утилита с семантикой, подобной wget, которая позволяет получать файлы от SMB-сервера. Файлы должны быть описаны в smb-url стандарте. Например, smb://host/share/file.

- **smbspool** — утилита для отправки на печать файла на SMB принтер.

Синтаксис

```
smbspool {job} {user} {title} {copies} {options}
         [filename]
```

Описание

Это очень маленькая программа для печати файлов на SMB принтер. Аргументы командной строки позиционно-зависимы для совместимости с системой печати Common UNIX Printing System, но вы можете использовать smbspool с любыми системами печати или в сценариях, определяет назначение используя URI с методом «smb». Например, smb://server[:port]/printer или

smb://username:password@workgroup/server[:port]/printer.

- **smbtar** — shell скрипт для архивации общих ресурсов SMB/CIFS прямо на UNIX устройство хранения.

Синтаксис

```
smbtar [-r] [-i] [-a] [-v] {-s server} [-p password]
       [-x services] [-X] \ [-N filename] [-b blocksize]
       [-d directory] [-l loglevel] \
       [-u user] [-t tape] {filenames}
```

Описание

Это небольшой скрипт над smbclient(1), позволяющий сохранять общие ресурсы SMB прямо на ленту.

- **smbtree** — текстовый SMB обозреватель сети.

Синтаксис

```
smbtree [-b] [-D] [-S]
```

Описание

Это SMB обозреватель в текстовом режиме. Аналог «Обозревателя Сети» , существующего на компьютерах Windows. Отображает дерево со всеми доменами, серверы этих доменов и общие ресурсы на серверах.

Пример

Для случая рассмотренной ранее ЛВС с файловым сервером кафедры:

```
root@vmUbuntu10:~# smbtree
Enter root's password:
VIRTUAL-NET
   \\VMUBUNTU10             ServerKafedra
   \\VMUBUNTU10\IPC$        IPC Service (ServerKafedra)
   \\VMUBUNTU10\boss        для начальства
   \\VMUBUNTU10\files       наши данные
HOME-NET
session request to 192.168.1.2 failed (Called name not
present)
   \\MAIN-PC
   \\MAIN-PC\C$             Стандартный общий ресурс
   \\MAIN-PC\ADMIN$         Удаленный Admin
   \\MAIN-PC\vmPC_share
   \\MAIN-PC\D$             Стандартный общий ресурс
   \\MAIN-PC\IPC$           Удаленный IPC
root@vmUbuntu10:~#
```

- **testparm** — проверяет правильность оформления настроек в файле smb.conf.

Синтаксис

```
testparm [-s] [-h] [-v] [-V] [-L <server name>]
         [-t encoding] {config filename} \
         [hostname hostIP]
         [--parameter-name parametername]
         [--section-name sectionname]
```

Описание

Эта программа-тест для проверки правильности конфигурационного файла. Если программа сообщает об отсутствии ошибок, то можно использовать этот файл, не опасаясь ошибок при его загрузке. Заметьте, что testparm НЕ гарантирует того, что службы, определенные в конфигурационном файле, будут доступны и будут работать именно так,

как вы задумали. Если в командной строке указаны имя и IP хоста, программа-тест проверит службы по списку в smb.conf и сообщит, имеет ли указанный хост доступ к каждой из служб. Если testparm находит ошибку в файле smb.conf, то возвращает код «1» вызывающей программе, при отсутствии ошибок возвращен будет код «0». Это позволяет использовать testparm в скриптах.

С целью получения навыков по работе с этими утилитами, предлагаю выполнить самостоятельный тренинг на базе сформированной виртуальной или натуральной сети, с включенным в ее состав файловым сервером Samba на базе Ubuntu-машины.

10.3.6. Общие сведения об утилите pdbedit из пакета Samba

Отдельно хотелось бы остановиться на утилите pdbedit. Эта утилита — отличный инструмент для просмотра и управления учетными записями Samba. Описание ее назначения, синтаксиса и подробное рассмотрение всех доступных опций приведено в приложении 10.3 к данной главе.

В этом разделе рассмотрим только общие подходы к использованию данной утилиты. В частности, с ее помощью можно просмотреть список пользователей:

```
serp@vmUbuntu10:~$ sudo pdbedit -L
[sudo] password for serp:
samba-user:1001:
lab2:1003:Лаборатория-2
zam:1002:Зав.кафедрой
lab3:1004:Лаборатория-3
```

С ключом -v вы получите более полную информацию по всем пользователям. Так, например, можно получить исчерпывающую информацию по конкретному пользователю с именем zam:

```
serp@vmUbuntu10:~$ sudo pdbedit zam -v
Unix username:        zam
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-945640269-1393339082-1001
Primary Group SID:    S-1-5-21-945640269-1393339082-309
Full Name:            Зав.кафедрой
Home Directory:       \\vmubuntu10\zam
HomeDir Drive:
Logon Script:
Profile Path:         \\vmubuntu10\zam\profile
Domain:               VMUBUNTU10
```

```
Account desc:
Workstations:
Munged dial:
Logon time:            0
Logoff time:           never
Kickoff time:          never
Password last set:     Вск, 07 Авг 2011 16:45:54 MSD
Password can change:   Вск, 07 Авг 2011 16:45:54 MSD
Password must change:  never
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

В любой момент мы можем отключить неугодного вам пользователя, отключив его учетную запись. Например, для отключения от сервера пользователя с именем lab3, следует выполнить команду:

```
serp@vmUbuntu10:~$ sudo pdbedit -c "[D]" -u lab3
Unix username:         lab3
NT username:
Account Flags:         [DU          ]
Full Name:             Лаборатория-3
```

Обратите внимание, что запись не удалена, а отключена. При этом в строке Account Flags появилось значение D. При необходимости у нас всегда есть возможность обратно включить этого samba-пользователя на доступ к общесетевым ресурсам:

```
serp@vmUbuntu10:~$ sudo pdbedit -c "[E]" -u lab3
Unix username:         lab3
Account Flags:         [U           ]
Full Name:             Лаборатория-3
Home Directory:        \\vmubuntu10\lab3
```

Также вы можете, в любой момент создать нового пользователя Samba. Конечно, если этот пользователь уже зарегистрирован в самой системе, то есть является ее Unix-пользователем и уже имеет Unix-пароль. Пусть для примера таким пользователем является serp:

```
serp@vmUbuntu10:~$ sudo pdbedit -a  -u serp
new password:
retype new password:
Unix username:         serp
```

Если эта регистрация была выполнена случайно или по ошибке, то у вас есть возможность этого samba-пользователя вообще удалить из базы.

```
serp@vmUbuntu10:~$ sudo pdbedit -x  -u serp
serp:1000:serp
```

Это только начальные основы знакомства с утилитой pdbedit. Но рассматривая ее, хотелось бы всего пару слов сказать о команде smbstatus, которая может показать вам текущие подключения к общим ресурсам сервера.

```
serp@vmUbuntu10:~$ smbstatus
Samba version 3.4.7
PID     Username      Group         Machine
-------------------------------------------
Service      pid     machine       Connected at
-------------------------------------------
No locked files
```

Увидев этот лог, вы скажите: «Ну и что интересного в этих пустых строках?». Но вы не торопитесь с выводами, а лучше с основного компьютера, там где у нас работает зав. кафедрой, подключитесь к ServerKafedra и войдите в папку boss. А после этого на Ubuntu10, где у нас Samba-сервер, повторно выполните команду smbstatus.

```
serp@vmUbuntu10:~$ smbstatus
Samba version 3.4.7
PID  Username  Group  Machine
-------------------------------------------------------
2043     zam   zam    main-pc  (::ffff:192.168.1.2)

Service   pid    machine   Connected at
--------------   ----------------------------------
IPC$      2043   main-pc   Mon Aug  8 03:52:23 2011
boss      2043   main-pc   Mon Aug  8 03:52:28 2011

Locked files:
Pid       Uid    DenyMode    Access     R/W      Oplock
--------------------------------------------------------
2043      1002   DENY_NONE   0x100081   RDONLY    NONE

SharePath    Name    Time
----------------------------------------------
/kafedra     .       Mon Aug  8 03:52:29 2011
```

У вас полная информация, когда, кто, с какого узла и к какому ресурсу файлового сервера подключился. Мы думаем эта информация сетевому администратору никогда лишней не будет.

Сейчас мы видим полную картину, и при большом числе пользователей и ресурсов она будет очень объемной. Поэтому у нас есть возможность как-то отфильтровать ее. Для этого можно воспользоваться опциями команды smbstatus:

- ключ -b позволит увидеть краткую картину,
- ключ -S только список общих файлов,
- ключ -u <имя_пользователя> позволит просмотреть подключения конкретного пользователя.

Заканчивая краткое знакомство с Samba, отметим, что работа с ним, как и по протоколу SMB в Windows-сети, стала стандартом. Но сегодня уже нельзя привязываться только к этому протоколу. В офисе, где установлены стационарные компьютеры, и работа за его пределами не планируется, можно ограничиться SMB. Другое дело, когда пользователь работает за ноутбуком. В таком случае строить систему необходимо так, чтобы у пользователя были минимальные отличия в трудовом процессе при работе через Интернет. При этом остается единственная проблема — доступ к общедоступным ресурсам на сервере.

11. ОРГАНИЗАЦИЯ WEB- И FTP-СЕРВЕРОВ НА UBUNTU-МАШИНЕ

11.1. Типы серверов и технология клиент-сервер

Клиент-сервер — это сетевая архитектура, в которой устройства являются либо клиентами, либо серверами. Клиентом (front end) является запрашивающая машина, сервером (back end) — машина, которая отвечает на запрос. Термины клиент и сервер могут применяться как к физическим устройствам, так и к программному обеспечению.

• Применительно к программному обеспечению программа состоит из двух частей – клиента и сервера. Сервер работает в фоновом режиме и ждет запроса от клиента. Получив запрос, сервер выполняет специальные действия. Закончив обработку запроса, сервер «замолкает» и ждет нового запроса. Возможен вариант, когда одна и та же программа выступает и как клиент, и как сервер. В этом случае, две части программы — клиентская и серверная — существуют внутри одного исполняемого модуля.

• Применительно к физическим устройствам, сервер — это техническое решение, которое предоставляет множеству компьютеров доступ к файлам, данным, принтерам и факсам. Сервер оптимизирован для оказания услуг другим компьютерам, или «клиентам». Клиентами могут быть компьютеры, а также принтеры, факсы и другие устройства, подключенные к серверу. Вместе сервер и его клиенты образуют клиент-серверную сеть.

Существует большое разнообразие серверов, приспособленных для выполнения строго определенных функций: файловые серверы, серверы печати, почтовые серверы и FAX-серверы, серверы удаленного доступа и большая группа серверов приложений, среди которых можно выделить.

Ftp-серверы

Представляют собой своеобразные больше объемные библиотеки файлов, доступ к которым осуществляется по протоколу FTP (File Transfer Protocol — протокол передачи файлов). Многочисленные FTP-серверы Интернет предоставляют бесплатный анонимный доступ к гигабайтам самой разнообразной информации: текстовым документам, дистрибутивам программ, фотографиям и музыкальным файлам.

По FTP-протоколу можно закачивать свои домашние странички на бесплатных серверах, предоставляющих под них место. Это гораздо удобнее, нежели применять HTTP, когда на специальной страничке сервера вы указываете файлы, которые надо закачать.

При использовании FTP следует помнить некоторые особенности этого сервиса, прямо вытекающие из той операционной системы, где он возник, а именно UNIX. Любой FTP-сервер всегда требует авторизации пользователя, то есть ввод его имени и пароля. В зависимости от этого будет предоставлен доступ лишь к определенным каталогам и файлам с возможностью осуществлять только разрешенные действия над содержимым FTP-хранилища.

Web-серверы

Web-сервер — это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и компьютер, на котором это программное обеспечение работает, то есть по-другому — это программа, устанавливаемая на сервер и обслуживающая посетителей веб-сайта. Она принимает запросы от удаленных пользователей и выдает в ответ соответствующие «странички» сайта. Клиенты получают доступ к веб-серверу по URL адресу нужной им WEB-страницы или другого ресурса.

11.2. Организация Web-сервиса на Linux-компьютерах

В том случая, когда на компьютере с ОС Ubuntu надо организовать полноценный Web-сервис для работы интернет-сайтов, только одного Web-сервера бывает недостаточно. В этом случая надо использовать набор (комплекс) серверного программного обеспечения. Для Linux-подобных операционных систем, это так называемый LAMP-сервер.

LAMP — это акроним, обозначающий набор программного обеспечения, широко используемый в Интернете. LAMP назван по первым буквам входящих в его состав компонентов:

- **L**inux — операционная система GNU/Linux;
- **A**pache — веб-сервер Apache;
- **M**ySQL-server — система управления базой данных, которая обрабатывает запросы, поступающие от браузера;
- **P**HP — интерпретатор языка программирования PHP, используемый для работы PHP-сценариев сайта на стороне сервера.

Хоть изначально эти программные продукты не разрабатывались специально для работы друг с другом, такая связка стала весьма

популярной, в первую очередь из-за своей низкой стоимости. Все ее составляющие являются открытыми и могут быть бесплатно загружены из Интернет.

Следует отметить, что WEB-сервис на основе LAMP является кроссплатформенным. Он управляется через веб-сервер Apache, данные хранятся и обрабатываются посредством СУБД MySQL-server, а внутренним языком программирования сценариев является PHP (Hypertext Preprocessor). Если вам потребуется организация полноценного веб-сервиса на Ubuntu-машине, то для установки на ней LAMP можно использовать с правами root команду:

```
aptitude install apache2 php5 mysql-server phpmyadmin
```

Эта команда позволит установить необходимые пакеты для функционирования веб-сервера и администрирования сервера баз данных MySQL. В процессе установки этих пакетов запустится настройка MySQL-server. В открывшемся окне нужно будет ввести пароль для основной учетной записи root и его подтверждение. Они необходимы в последующем и для входа на СУБД. Также будет запрошен пароль для программы phpmyadmin, которая впоследствии будет доступна из браузера по адресу 127.0.0.1/phpmyadmin.

Однако на текущий момент организация полноценного веб-сервиса в наши задачи не входит. Наша цель пока проще — установка и настройка веб-сервера Apache, чтобы на его основе познакомиться с еще одной сетевой технологией взаимодействия Ubuntu- и Windows-машин.

11.3. Установка и настройка Web-сервера Apache

Веб-сервер Apache наиболее распространен и часто используется в Unix-подобных системах. Это свободно распространяемый веб-сервер. С апреля 1996 года — самый популярный веб-сервер в Интернете. Его основными достоинствами считаются надежность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает протокол IPv6. Недостатком наиболее часто называется отсутствие удобного стандартного интерфейса для администратора.

Основой функционирования веб-сервера на компьютере с ОС Ubuntu 10.04 является пакет: apache2 — это метапакет HTTP-сервера Apache (Deb-пакет: apache2_2.2.14-5ubuntu8.4_i386.deb).

К возможностям этого пакета обычно относят: HTTPS, виртуальный хостинг, CGI, SSI, IPv6, интеграцию с базами данных и возможности работы сценариев, фильтрацию запросов/ответов, много гибких схем аутентификации и прочее.

Под Linux существует несколько наборов мульти-процессных модулей (Multi-Processing Module — mpm), предоставляющих сервис HTTP-сервера Apache2:

- apache2-mpm-prefork — традиционная беспотоковая модель;
- apache2-mpm-worker — модель с высокоскоростными потоками;
- apache2-mpm-event — событийно управляемая модель. Этот mpm экспериментальный и менее тестируемый, чем другие;
- apache2-mpm-itk — многопользовательская модель.

По умолчанию из своих репозиториев Ubuntu 10.04 устанавливает пакет: **apache2-mpm-prefork**.

Deb-пакет: apache2-mpm-prefork_2.2.14-5ubuntu8.4_i386.deb

Этот пакет представляет традиционную беспотоковую модель HTTP-сервера Apache. Каждый многопроцессный модуль Apache предоставляет собой некоторую «разновидность» исполняемого файла веб-сервера, собранного с определенной моделью обработки.

Версия prefork-mpm является беспотоковой и работает посредством заранее запущенной копии сервера, которая обрабатывает запросы подобно Apache 1.3. Она не так быстра, как потоковая модель, но считается более стабильной и подходит для сайтов, которым нужна совместимость с потоко-небезопасными библиотеками. Лучше подходит для изоляции каждого запроса, так что проблемы с одним запросом не повлияют на остальные. Основными пакетами, которые относятся к apache2-mpm-prefork, являются:

apache2.2-bin – основные двоичные файлы HTTP сервера Apache.

Deb-пакет: apache2.2-bin_2.2.14-5ubuntu8.4_i386.deb.

Этот пакет содержит все двоичные файлы, но не содержит конфигурационных или вспомогательных скриптов. Чтобы получить автономный сервер, необходимо установить один из пакетов apache2-mpm-*. Другие пакеты, такие как gnome-user-share могут внести свою конфигурацию Apache. Пакеты, относящиеся к apache2.2-bin:

- libapr1 — переносимая библиотека Apache.
  Deb-пакет: libapr1_1.3.8-1ubuntu0.3_i386.deb.
- libaprutil1 — это переносимая библиотека времени выполнения, предоставляющая абстрактный интерфейс для разнообразных элементов, различных на каждой платформе.
  Deb-пакет: libaprutil1_1.3.9+dfsg-3ubuntu0.10.04.1_i386.deb.
- libaprutil1-dbd-sqlite3 — драйвер SQLite3.
  Deb-пакет:libaprutil1-dbd-sqlite3_1.3.9+dfsg-3ubuntu0.10.04.1_i386.deb.
- libaprutil1-ldap — драйвер LDAP.
  Deb-пакет: libaprutil1-ldap_1.3.9+dfsg-3ubuntu0.10.04.1_i386.deb.
- libssl0.9.8 — Динамическая библиотека для SSL.
  Deb-пакет: libssl0.9.8_0.9.8k-7ubuntu8.6_i386.deb.

apache2.2-common — общие файлы HTTP сервера Apache.

Deb-пакет: apache2.2-common_2.2.14-5ubuntu8.4_i386.deb.

Этот пакет содержит конфигурационные и вспомогательные скрипты, но не содержит сам сервер. Нужна установка одного из пакетов apache2-mpm-*.

К apache2.2-common относится пакет apache2-utils — утилиты для веб-серверов, который предоставляет программы, очень полезные для любого веб сервера. Среди них:

- ab — утилита тестирования производительности Apache;
- logresolve — соотнесение IP адресов и имен хостов в файлах журнала;
- htpasswd — управление основными файлами аутентификации;
- rotatelogs — периодическое прекращение записи в файл журнала и открытие нового;
- split-logfile — разделение файла журнала и много других;

Deb-пакет: apache2-utils_2.2.14-5ubuntu8.4_i386.deb

Для установки веб-сервера Apache можно воспользоваться Менеджером пакетов, Центром приложений или открыть терминал и ввести:

```
sudo apt-get install apache2
```

При отсутствии прямого доступа в Интернет, но при наличии на компьютере всех необходимых перечисленных выше 10 deb-архивов вам может помочь установка из deb-архива:

```
sudo dpkg -i <путь к deb-архивам Apache>/*.deb
```

Если установка пройдет успешно и не будет никаких предупреждений о нарушении зависимостей внутри пакета, то обратите внимание на правый верхний угол экрана Ubuntu. Пиктограмма режима работы стала красной, что свидетельствует о необходимости перезагрузки Ubuntu-машины (рис. 11.1).

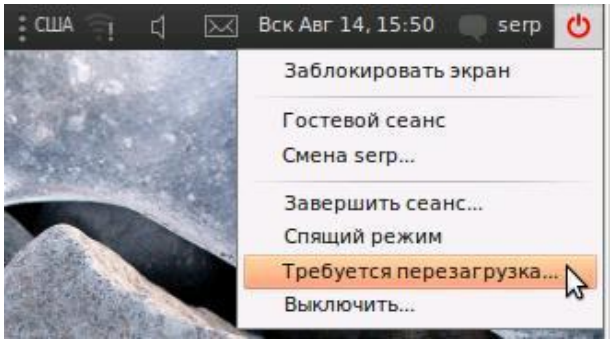

Рис. 11.1. Требование перезагрузки ОС после установки пакета Apache.

После установки пакетов и перезагрузке системы при вводе в адресную строку браузера Mozilla Firefox адреса http://localhost будет открываться страница вида, аналогичного рис. 11.2.

Если мы вспомним, что vmUbuntu10 имеет IP-адрес 192.168.1.10, то в строке браузера можно ввести URL вида http://192.168.1.10 и результат будет аналогичен тому, что приведен на рис. 11.2.

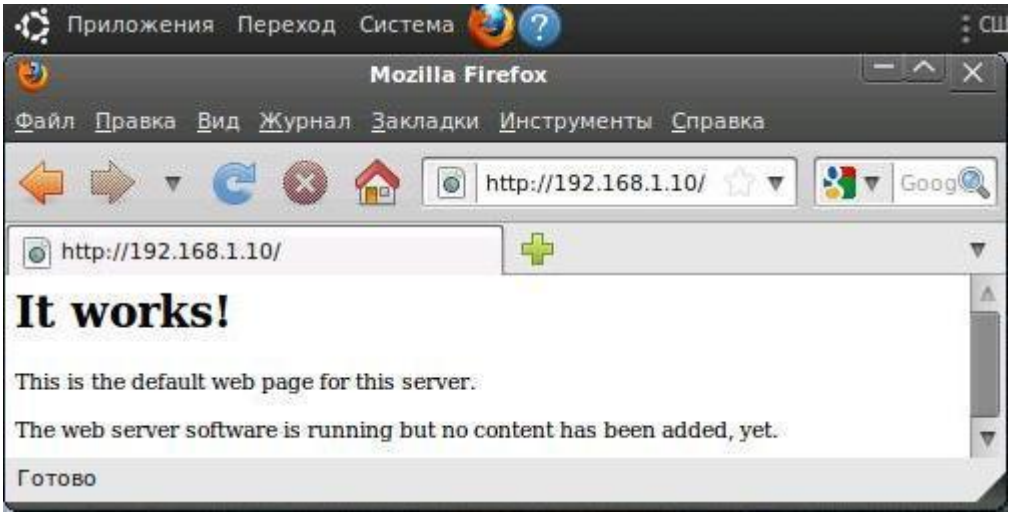

Рис. 11.2. Стартовая страница вновь установленного веб-сервера на vmUbuntu10.

При запуске Internet Explorer на основном компьютере нашей виртуальной сети и вводе того же URL должен быть получен аналогичный результат (рис. 11.3).

Рис. 11.3. Доступ к веб-серверу vmUbuntu10 с другого компьютера.

Замечание.
Если во втором случае у вас получился другой результат, то проверьте сетевые настройки виртуальных машин и выполните с основного компьютера ping на vmUbuntu10.

Из полученных результатов видно, что как веб-браузер Mozilla Firefox, который подключается локально, так и Internet Explorer, подключающийся к веб-серверу удаленно, ошибок о недоступности ресурса не выдают. Стало быть, наш веб-сервер работает и доступ к нему есть со всех узлов сети.

Но тут же встает вопрос: «Почему такой результат доступа и вид отображаемой страницы?». Ответ на этот вопрос можно получить, если

войти на vmUbuntu10 в автоматически созданный при установке Apache каталог /var/www (рис. 11.4).

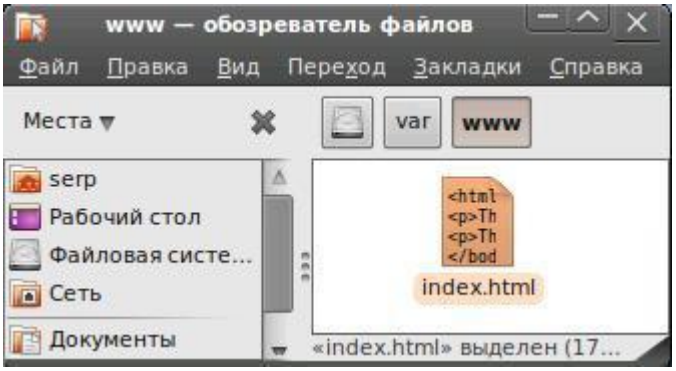

Рис. 11.4. Файл index.html в каталог /var/www vmUbuntu10.

Именно там хранится, созданный по умолчанию файл index.html, которой и является стартовой страницей веб-сервера, а каталог /var/www — это точка входа на наш новый веб-сервер. Чтобы убедиться в этом, можете поэкспериментировать с файлом index.html. Только помните, что права на него имеет суперпользователь root.

11.4. Установка Ftp-сервера proFTPd

В этом разделе мы познакомимся с установкой и настройкой Ftp-сервера ProFTPd. После его настройки можно будет получить доступ к директории /var/www для удаленного изменения содержимого главной страницы Web-сервера. А в конце раздела мы попробуем создать и сконфигурировать анонимный ftp-сервер.

В дистрибутиве Ubuntu доступно около 5 разновидностей программ, предоставляющих сервисы доступа к файлам по протоколу ftp. Наиболее известные из них pure-ftpd, wu-ftpd и proftpd. Полный список доступных Ftp-серверов для установки можно посмотреть, выполнив команду поиска по слову ftpd

```
aptitude search ftpd
```

или выполнив поиск по этому же слову в Менеджере пакетов или Центре приложений.

Мы будем использовать Ftp-сервер proFTPd, пришедший в мир Linux из Института Беркли из их набора программ с открытым кодом BSD. Для установки ftp-сервера необходимо, как минимум, два зависимых пакета:

- **proftpd-basic** — Ftp-сервер для Linux-подобных операционных систем.

Deb-пакет:proftpd-basic_1.3.2c-1ubuntu0.1_i386.deb

Сервер может быть настроен для работы нескольких виртуальных хостов, также поддерживает chroot. Может быть запущен в виде отдельного сервера (демона) или в составе суперсервера inetd. Также поддерживает IPv6 и использует лишь один конфигурационный файл /etc/proftpd/proftpd.conf.

➢ **openbsd-inetd** — метасервер OpenBSD, управляющий входящими соединениями

Deb-пакет:openbsd-inetd_0.20080125-4ubuntu2_i386.deb

Сервер inetd — это сетевая служба, которая управляет входящими сетевыми соединениями. При поступлении запроса на соединение она запускает программу его обработки. Службу можно настроить на обработку запросов с любого порта по протоколам tcp или udp.

11.4.1. Установка и настройка Ftp-сервера для доступа к файлам web-сайта

Для установки ftp-сервера proFTPd можно воспользоваться Менеджером пакетов, Центром приложений или в терминале ввести:

```
sudo apt-get install proftpd
```

При отсутствии прямого доступа в Интернет вам может помочь установка из вышеприведенных deb-архивов. Установку можно выполнить как непосредственно на Ubuntu-машине, так и удаленно с Ubuntu- или Windows-машины. В последнем случае можно использовать, например, vnc, rdp или ssh через PuTTY.

В качестве примера рассмотрим удаленную установку Ftp-сервера на vmUbuntu10 с виртуальной машины vmUbuntu06.

Выполним из vmUbuntu06 удаленный вход в консоль vmUbuntu10, используя протокол ssh. Для этого на vmUbuntu06 откроем Терминал, выбрав Приложения -> Стандартные. Затем выполним команду подключения к нашему серверу vmUbuntu10:

```
ssh 192.168.1.10 -l <имя пользователя>
```

После ввода пароля в терминале vmUbuntu06 появится приглашение командной оболочки удаленного сервера vmUbuntu10:

```
<имя пользователя>@vmUbuntu10:~$
```

В командной строке требуется ввести команду для инсталляции proFTPd. Один из возможных вариантов удаленной установки Ftp-сервера на vmUbuntu10 с машины vmUbuntu06 приведен на рис. 11.5.

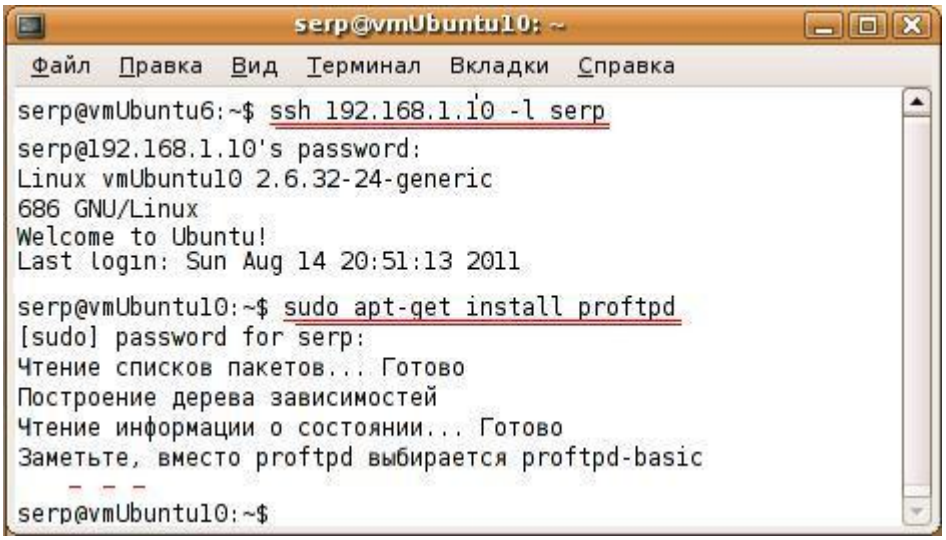

Рис. 11.5. Удаленная установка Ftp-сервера на vmUbuntu10 с vmUbuntu06.

В процессе установки система выдаст запрос: «Хотите ли вы запускать ProFTPd демон в режиме inetd или standalon? ». Выберите режим standalon. В этом случае демон будет запускаться при загрузке сервера и будет активен во все время работы сервера.

Если установка завершилась успешно и вы являетесь зарегистрированным пользователем на vmUbuntu10, и имеете на vmUbuntu10 свой домашний каталог, то для проверки работы установленного на vmUbuntu10 Ftp-сервера следует на vmUbuntu06 выполнить следующую последовательность действии:

- Выбрать Переход -> Подключение к серверу.
- В появившемся окне «Соединение с сервером» выбрать тип сервиса FTP (с авторизацией).
- В поле сервер ввести IP-адрес vmUbuntu10 — 192.168.1.10, заполнить поле «Имя пользователя» и нажать «Соединиться».
- На рабочем столе vmUbuntu06 появиться ярлык на доступ к Ftp-серверу (192.168.1.10), кликнув по которому можно получить доступ к vmUbuntu10 (рис. 11.6).

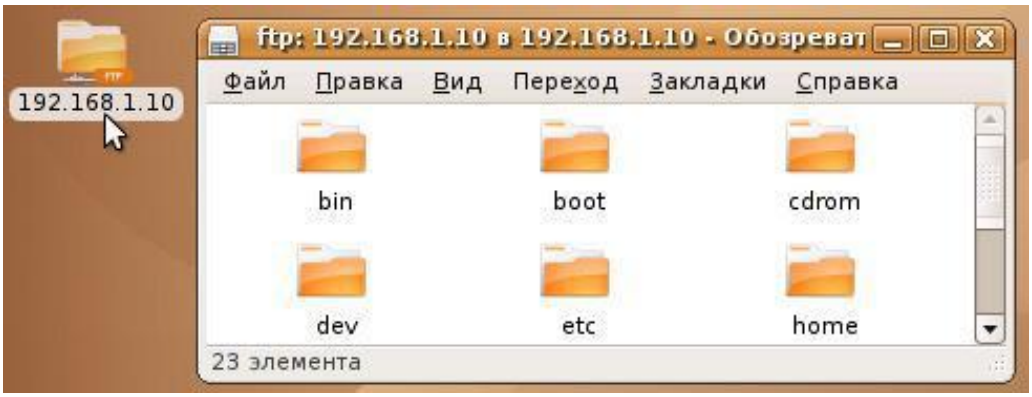

Рис. 11.6. Доступ к Ftp-серверу vmUbuntu10 с vmUbuntu06.

Проверить доступ к новому Ftp-серверу можно и с любого другого компьютера любым Ftp-клиентом. В том числе и из командной строки Windows. Для этого на основном Windows-компьютере нашей виртуальной сети следует:

- Выбрать Пуск -> Выполнить, в затем в окне Запуск программы в поле Открыть ввести cmd и нажать кнопку ОК.
- В окне командного режима Windows ввести команду ftp, и в ответ на появившееся приглашение ftp> ввести команду open 192.168.1.10.
- После ввода имени пользователя и пароля вам будут доступны любые команды протокола FTP по манипуляциям с ресурсами Ftp-сервера (рис. 11.7).

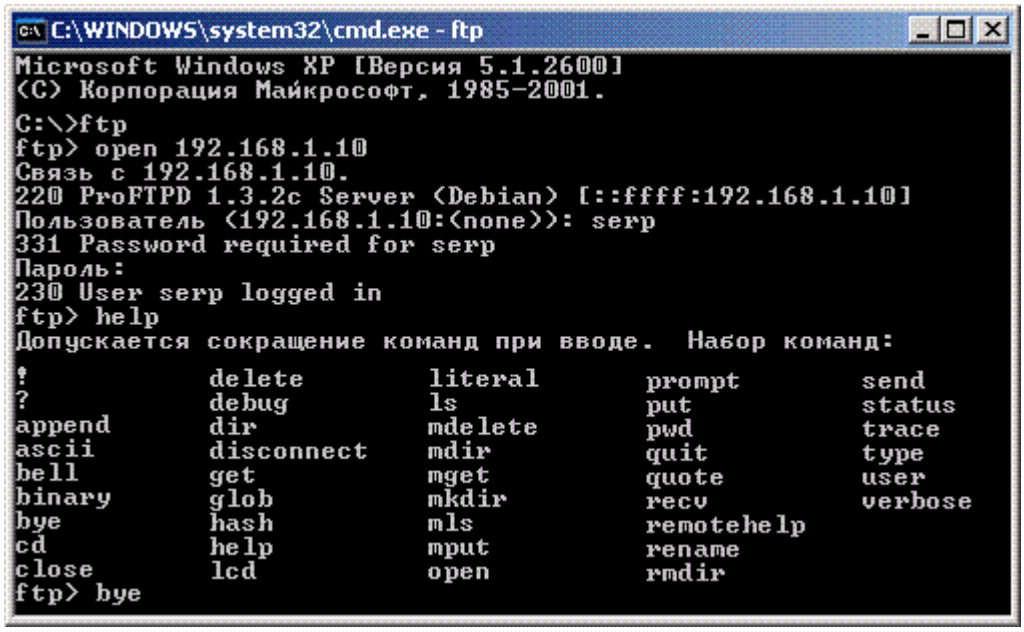

Рис. 11.7. Доступ к Ftp-серверу из командной строки Windows.

Кстати, напомню, что если вы справились с примерами предыдущей главы, то на vmUbuntu10 остались зарегистрированными пользователи подразделений кафедры. У вас есть возможность войти на Ftp-сервер с их логинами и паролями.

Внимание!!!

Обратите внимание, что при входе с любого компьютера любым Ftp-клиентом, вводя имя и пароль конкретного пользователя, вы получаете Ftp-доступ именно к рабочему каталогу этого пользователя на vmUbuntu10.

Но основная задача данного раздела — это настройка Ftp-сервера для доступа к файлам web-сайта. Например, кафедрального сайта, где должно быть предусмотрено удаленное изменение контента web-сайта рядом членов кафедры. Эта задача будет еще более актуальной, когда мы захотим разместить на нашем web-сервере несколько сайтов разных пользователей.

Аналогично серверу Samba вся информация о настройках Ftp-сервера proFTPd содержится в его конфигурационном файле /etc/proftpd/proftpd.conf. Исходный вид этого файла приведен в приложении 11.1. Именно этот файл надо редактировать для изменения настроек Ftp-сервера. Чтобы сохранить исходный конфигурационный файл выполните команду:

```
sudo cp /etc/proftpd/proftpd.conf /etc/proftpd/proftpd.conf.default
```

Обратите внимание в конфигурационном файле на строчки, которые определяют, что по умолчанию точкой входа Ftp-сервера (DefaultRoot) являются домашние каталоги пользователей (~):

```
DenyFilter                      \*.*/

# Use this to jail all users in their homes
# DefaultRoot                   ~
```

Чтобы получить доступ к файлам в папке /var/www, где хранится непосредственный контент нашего сайта в виде html-страниц и используемых в них изображений, следует изменить настройки в конфигурационном файле /etc/proftpd/proftpd.conf.

Откройте конфигурационный файл proftpd-сервера в текстовом редакторе, например nano, используя команду:

```
sudo nano /etc/proftpd/proftpd.conf
```

В файле большой набор параметров с комментариями к ним, но в текущий момент нас будут интересовать лишь два из них. Мы хотим изменить точку входу и немного ускорить работу Ftp-сервера.

Причем точку входа будем менять только для одного пользователя — предполагаемого администратора нашего сайта. Пусть таким пользователем будет пользователь serp, уже имеющий административные права на этом сервере.

```
DenyFilter                      \*.*/

# Use this to jail all users in their homes
# DefaultRoot                   ~
  DefaultRoot                   /var/www      serp
  UseReverseDNS                 off
```

Чтобы изменения в конфигурационном файле повлияли на работу Ftp-сервера, нужно перезапустить сервис proftpd:

```
sudo /etc/init.d/proftpd restart
```

Для проверки доступа по протоколу FTP с основного компьютера на Ftp-сервер, который сейчас должен нам предоставить доступ к каталогу, где хранится контент нашего Web-сайта следует:

- На основном Windows-компьютере открыть Мой компьютер.
- В адресной строке ввести ftp://192.168.1.10 адрес нашего Ftp-сервера в формате URL (рис. 11.8).

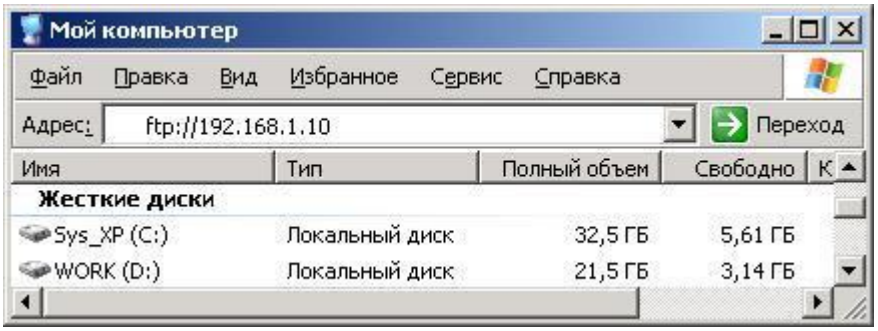

Рис. 11.8. Доступ к Ftp-серверу из Мой компьютер Windows.

- Появится окно Вход, где будет указано, что анонимный вход невозможен и требуется ввести логин и пароль пользователя.
- Если аутентификация пройдет успешно, то откроется папка с контентом нашего Web-сайта (рис. 11.9).

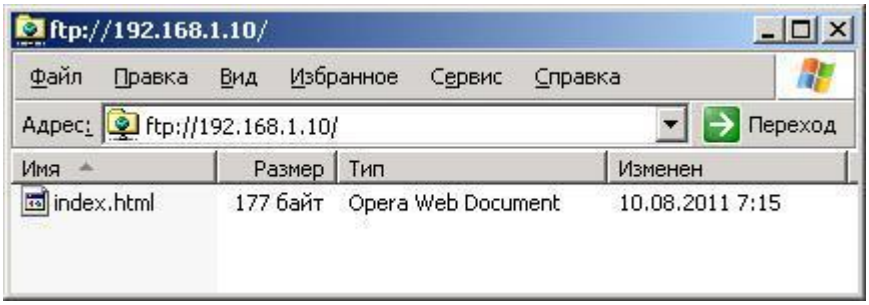

Рис. 11.9. Папка с контентом нашего Web-сайта.

Однако изменить файл index.html или добавить новый файл или каталог у вас не получится. И чтобы разобраться в этой проблеме нам придется вернуться к тому моменту, когда была закончена установка Web-сервера Apache.

11.4.2. Назначение прав на web-контент

Web-контент, который показывает Apache при доступе к нему через браузер любого компьютера (рис. 11.2 и 11.3), находится на vmUbuntu10 в директории /var/www. Войдем локально или удаленно на vmUbuntu10 с правами администратора этого компьютера, а далее:

- Перейдем в папку /var/www, где храниться файл index.html.

- Щелкнем правой кнопкой мыши по файлу index.html и в появившемся контекстном меню выберем Открыть с помощью -> Текстовый редактор.
- Просмотрим его содержимое — это именно тот файл, который отображался в браузерах компьютеров, когда мы входили по адресу http://192.168.1.10/.

Если теперь попробовать сделать изменения в этом файле, то получим «сбой операции сохранения». Это означает, что у пользователя, под которым мы входили по FTP, нет прав на изменение файла /var/www/index.html. В этом можно убедиться, выполнив в консоли vmUbuntu10 команду:

```
serp@vmUbuntu10:$ ls -l /var/www
итого 4
-rw-r--r-- 1 root root 177 2011-08-10 11:15 index.html
```

Пусть пользователь, например serp, является администратором vmUbuntu10. Давайте изменим «хозяина» директории /var/www и всех находящихся в ней файлов. Новым хозяином будет пользователь, под которым мы входим на ftp ресурс нашего сервера, этот пользователь — serp. Для этого можно использовать команду на изменение прав вида:

```
sudo chown <administrator> -R /var/www
```

Изменим права пользования и проверим сделанные изменения, выполнив следующую последовательность команд:

```
serp@vmUbuntu10:~$ sudo chown serp -R /var/www
serp@vmUbuntu10:~$ ls -l /var/www
итого 4
-rw-r--r-- 1 serp root 177 2011-08-10 11:15 index.html
```

Параметры доступа rw-r--r-- означают, что пользователь системы serp имеет право читать и записывать файл index.html, эти права распространяются и на весь каталог /var/www, «хозяином» которого стал serp. То есть он может выполнять функции администрирования сайта.

Проверить все это мы вам настоятельно рекомендуем, но уже самостоятельно. А мы переходим к рассмотрению еще одного аспекта конфигурирования ftp-сервера.

11.4.3. Настройка анонимного Ftp-сервера

Мы с вами обеспечили возможность изменять файлы и папки нашего Web-сервера. Теперь давайте создадим обычный Ftp-сервер, доступный для всех пользователей сети в режиме чтения.

Пользователи, желающие зайти на этот Ftp-сервер, смогут это сделать без ввода логина и пароля, но при этом они будут попадать в определенную папку на сервере и не смогут изменять данные в ней. Такой способ работы ftp сервера часто используется для создания общедоступных архивов фильмов, музыки, программ или обычных текстовых файлов.

Снова откроем конфигурационный файл proftpd и найдем в нем секцию Anonymous.

```
sudo nano /etc/proftpd/proftpd.conf
```

Эта секция представляет собой набор параметров, заключенных между тегами <Anonymous> и </Anonymous>. По умолчанию все строки этой секции закомментированы. Раскомментируем только часть параметров:

```
<Anonymous ~ftp>
   User                        ftp
   Group                       nogroup
#  Доступ для всех под именем "anonymous" и "ftp"
   UserAlias                   anonymous ftp
#  Косметика - все файлы принадлежат пользователю ftp
   DirFakeUser  on ftp
   DirFakeGroup on ftp
#
   RequireValidShell           off
#
#  Limit the maximum number of anonymous logins
   MaxClients                  10
#
#  Пусть 'welcome.msg' показывается при каждом входе и
#  и '.message' при каждой смене директории
   DisplayLogin               welcome.msg
   DisplayChdir               .message
 . . .
</Anonymous>
```

Сохраним изменения и перезапустим сервис proftpd для применения параметров.

```
sudo /etc/init.d/proftpd restart
```

Следует отметить, что все файлы, которые должны быть общедоступны через анонимный Ftp, надо располагать в директории /home/ftp. Адрес директории, куда попадает пользователь при анонимном входе, указан в теге <Anonymous ~ftp>. Параметр тега ~ftp соответствует домашней папке пользователя ftp, то есть папке /home/ftp.

Эта папка автоматически была создана в процессе установки Ftp-сервера и, аналогично папке /var/www. Ее «хозяином» является суперпользователь root. Создадим в директории /home/ftp, для примера, два файла file1.txt, file2.txt и одну папку NEW-MUSIC:

```
serp@vmUbuntu10:~$ sudo -i
[sudo] password for serp:

root@vmUbuntu10:~# cd /home/ftp

root@vmUbuntu10:/home/ftp# ls -l
итого 4
-rw-r--r-- 1 root root 170 2010-11-18 03:40 welcome.msg

root@vmUbuntu10:/home/ftp# echo FILE1 > file1.txt
root@vmUbuntu10:/home/ftp# echo FILE2 > file2.txt
root@vmUbuntu10:/home/ftp# mkdir NEW-MUSIC

root@vmUbuntu10:/home/ftp# ls
file1.txt  file2.txt  NEW-MUSIC  welcome.msg
```

Теперь проверим правильность работы анонимного доступа на Ftp-сервер. Для этого воспользуемся еще одним методом доступа к Ftp-серверам.

Запустим на основном Windows-компьютере Internet Explorer и в адресной строке введем URL-адрес нашего Ftp-сервера, то есть ftp://192.168.1.10. Если анонимный доступ на Ftp-сервер настроен, и настроен верно,то результат должен быть аналогичным рис. 11.10.

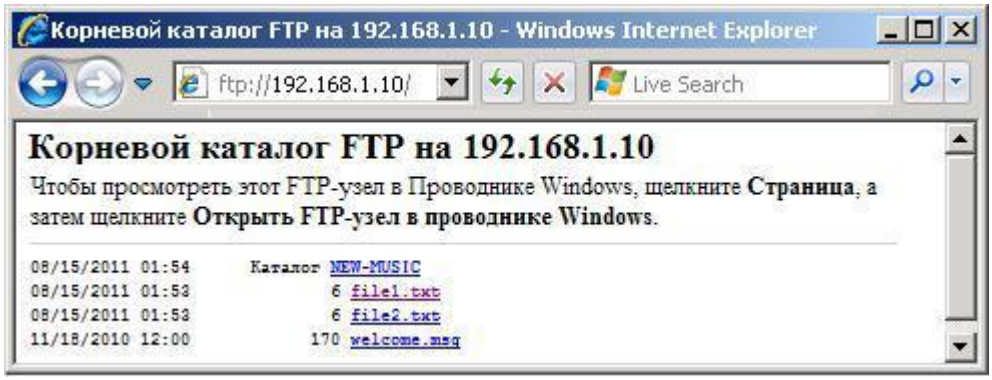

Рис. 11.10. Анонимный доступ на Ftp-сервер с Windows-компьютера через Internet Explorer.

Попробуйте просмотреть файлы и папки этого ресурса, скопировать их на свой компьютер, создать новый файл и новую папку, скопировать со своего компьютера файл в папку NEW-MUSIC.

11.4.4. Добавление в анонимный Ftp-сервер директории с возможностью публичной записи

В предыдущем разделе был настроен общедоступный анонимный Ftp-сервер. Но он позволял только удаленное чтение файлов. Попробуем расширить функциональность нашего Ftp-сервера. С этой целью давайте добавим в него еще одну директорию incoming и разрешим всем копировать в нее файлы и папки.

Решение поставленной задачи начнем с того, что снова откроем конфигурационный файл proftpd:

```
sudo nano /etc/proftpd/proftpd.conf
```

Найдем уже знакомую секцию <Anonymous> и внутри нее обратим внимание на подсекцию <Directory incoming> </Directory>. Чтобы получить желаемый результат потребуется раскомментировать строки именно этой секции.

```
<Directory incoming>
#   #   # Umask 022 хорошо подходит для предотвращения
#   #   # добавления прав записи
#   #   # всем вновь созданным файлам и папкам
Umask     022  022'
         <Limit READ WRITE>
             AllowAll
         </Limit>
         <Limit STOR>
             AllowAll
         </Limit>
</Directory>
```

Теперь следует создать в директории, доступной при анонимном входе, новую директорию, указанную в опции <Directory incoming>.

```
sudo mkdir /home/ftp/incoming
```

После чего требуется сменить права доступа всех пользователей на полный доступ к этой директории, то есть на rwxrwxrwx (777), что даст возможность процессу proftpd записывать в нее данные.

```
sudo chmod 777 /home/ftp/incoming
```

Перезапустим сервис proftpd для применения изменений.

```
sudo /etc/init.d/proftpd restart
```

Проверка правильности настройки Ftp-сервера, выполненная так, как указано в предыдущем параграфе, дает результат (рис. 11.11).

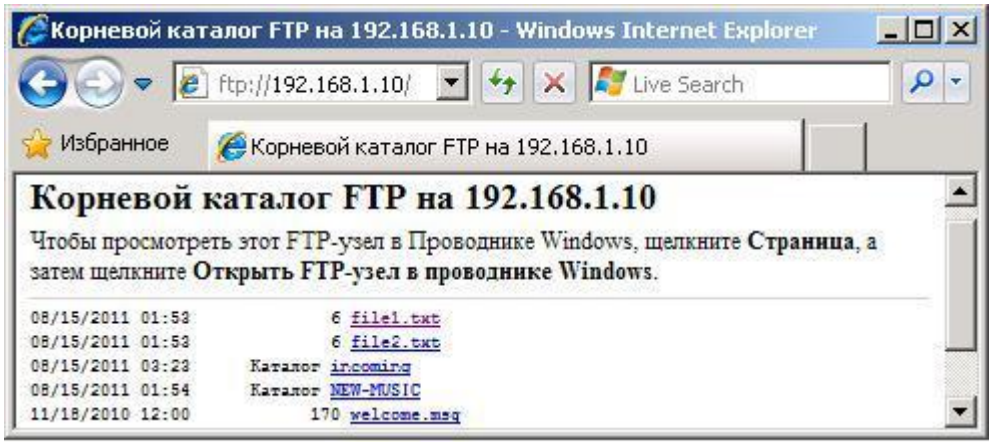

Рис. 11.11. Анонимный доступ на Ftp-сервер с Windows-компьютера через Internet Explorer.

На этот раз вы должны иметь возможность создавать файлы и папки в директории incoming и ошибку, при попытке создания файлов в других директориях.

11.4.5. Несколько общих замечаний об FTP доступе

Как мы уже выяснили, Ftp-сервер представляет собой своеобразную библиотеку файлов, и для перекачки файлов между серверами и компьютерами пользователей используется протокол FTP.

В Интернете тысячи Ftp-серверов, предоставляющих бесплатный анонимный доступ к гигабайтам самой разнообразной информации, но следует помнить, что любой Ftp-сервер всегда требует авторизации пользователя, то есть ввод его имени и пароля.

В зависимости от этого пользователю будет предоставлен доступ лишь к определенным каталогам и файлам вместе с возможностью осуществлять только разрешенные действия над содержимым Ftp-хранилища. Что же делать, если вы не являетесь зарегистрированным пользователем?

Практически каждый Ftp-сервер предоставляет так называемый анонимный вход, другое название этого сервиса — анонимный FTP. Для анонимного, или гостевого, входа на сервер необходимо:

- вместо имени пользователя указать ключевое слово anonymous,
- в качестве пароля набрать адрес своей электронной почты.

После чего вам будет предоставлен доступ к общим каталогам, к данным, которыми владелец сервера хочет поделиться. Обычно в таком режиме доступа к серверу пользователь может только просматривать каталоги и выкачивать файлы к себе на диск. Некоторые серверы создают специальные каталоги, куда желающие могут закачать свои собственные файлы. Это так называемые файлообменники.

Для работы с Ftp-сервером можно использовать обыкновенный веб-браузер. После набора в строке адреса URL нужного Ftp-сервера браузер

подключится к нему и выведет содержимое удаленного каталога. Для подключения к Ftp-серверу используют следующие формы записи URL:

При использовании Ftp-сервера, требующего авторизации:

```
ftp://логин:пароль@адрес_сервера:порт/путь_к_файлу
```

При использовании анонимного Ftp-сервера:

```
ftp://адрес_Ftp-сервера/путь_к_файлу
```

На первый взгляд, все замечательно, и подобное использование веб-браузера в качестве Ftp-клиента достаточно удобно. Однако необходимо отметить, что при этом способе работы отсутствует возможность докачки файла. Если связь с сервером внезапно оборвалась и вы не успели скачать файл целиком, что случается весьма часто при выкачивании больших файлов с очень удаленных серверов, то вам придется скачивать весь файл с самого начала.

Это одна из немалого количества достаточно веских причин, заставляющих использовать при работе с Ftp-сервером отдельных Ftp-клиентов. Такой клиент позволяет выгружать и посылать файлы на Ftp-сервер. В Интернете без труда можно найти большое количество FTP-клиентов для всевозможных операционных систем:

- WS_FTP — простейшая бесплатная программа для работы с Ftp-серверами.
- LeechFTP — бесплатный, но достаточно мощный Ftp-клиент с широкими возможностями и удобным графическим интерфейсом. Особенностью этой программы является возможность работы с несколькими FTP-серверами одновременно.
- CuteFTP — мощный Ftp-клиент с большим набором разнообразных функций, занимающий лидирующие позиции среди аналогичных программ. Необходимо отметить, что CuteFTP — коммерческий продукт, и большинство его возможностей будут отключены после месяца бесплатного использования.

Если LeechFTP осуществляет одновременно несколько соединений с разными Ftp-серверами, то CuteFTP поочередно взаимодействует с каждым сервером и последовательно выполняет задачи, которые записаны в созданную им «очередь».

CuteFTP поддерживает автоматическое продолжение закачивания или выкачивания файлов с сервера, сравнение удаленного каталога и каталога на локальном диске, поиск файлов в Сети.

Замечание.
Существует два режима FTP для перекачки файлов: ASCII и двоичный, его называют еще зеркальным.

Когда переписывается ASCII-файл между компьютерами различных типов с разными способами хранения информации в режиме ASCII, это преобразование выполняется автоматически, и поэтому на принимающей машине он записывается в виде понятного текстового файла. Двоичный же файл не обрабатывается и передается в неизменном виде.

Если вернуться к конфигурации Ftp-сервера, то следует сказать, что с использованием файла /etc/proftpd/proftpd.conf удобно выставлять глобальные настройки сервера и работать в удаленном текстовом терминале.

Если работать локально или через удаленный рабочий стол, то права доступа проще настраивать через графический интерфейс пакета gadmin-proftpd.

Следует отметить, что структура конфигурационных файлов пакета proftpd-basic и gadmin-proft частично отличаются. Поэтому имея готовый и настроенный Ftp-сервер надо понимать, что, установив и запустив gadmin-proftpd, вам может потребоваться заново конфигурировать ваш сервер.

Установить gadmin-proftpd можно стандартным образом, используя Менеджер пакетов, Центр приложений, или загрузить deb-архив с сайта, например http://debian.cs.binghamton.edu. После установки пакета в меню Приложения появится новая опция, аналогично рис. 11.12.

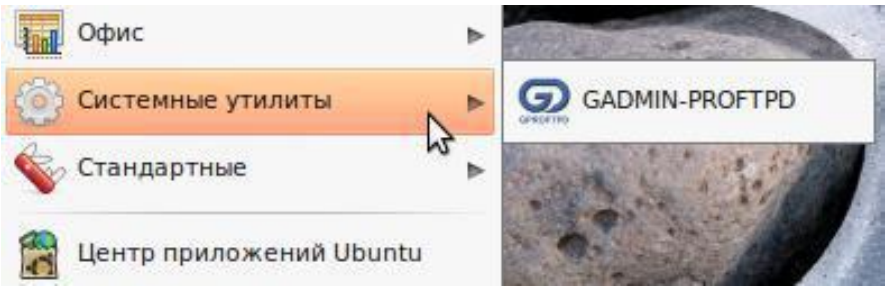

Рис. 11.12. Меню Приложение после установки пакета gadmin-proftpd.

После запуска пакета будет запрошен пароль пользователя и появится запрос о необходимости перезаписать конфигурацию файла proftpd.conf. Будьте внимательны и правильно принимайте решения при настроенном у вас Ftp-сервере. По окончании загрузки появится окно gadmin-proftpd (рис. 11.13).

Теперь у вас появляется возможность создавать пользователей, группы, устанавливать права и т. д. в удобном графическом интерфейсе. После того как вы сделали все настройки, gproftpd перепишет конфигурационный файл Ftp-сервера.

В любой момент конфигурационный файл proftpd.conf вам доступен во вкладке Configuration. В этой вкладке Вы можете посмотреть сделанные Вами настройки и, если необходимо, что-то исправить или добавить вручную.

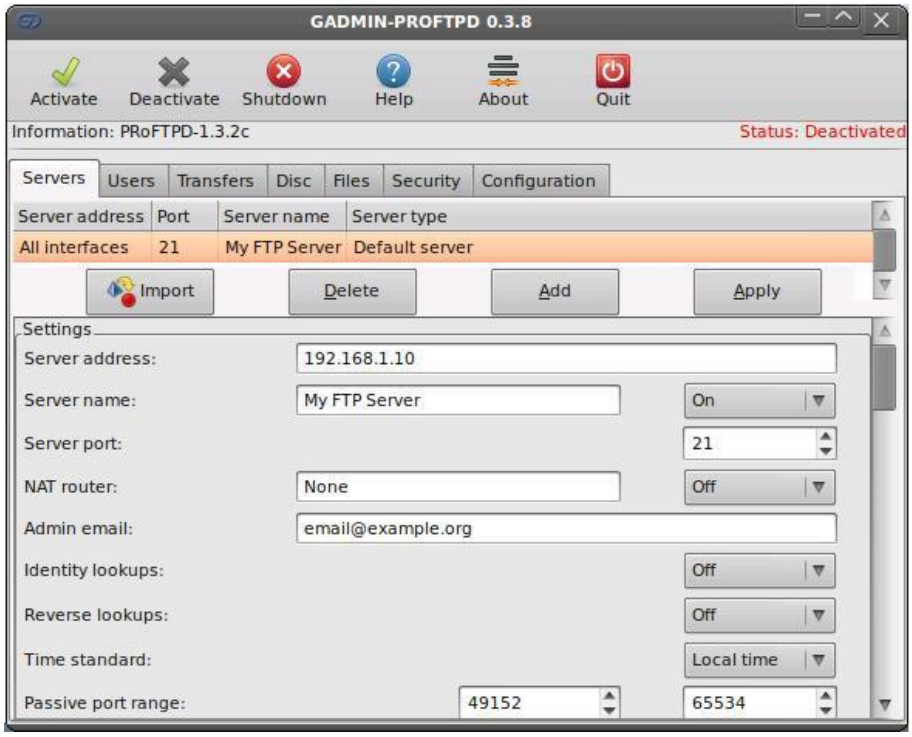

Рис. 11.13. Окно пакета gadmin-proftpd.

11.5. Настройка Web-сервера Apache2 и виртуальный хостинг

Этот раздел предназначен для тех, кого интересует процесс настройки Web-сервера Apache2. Если это не входит в ваши интересы, то вы можете спокойно пропустить этот раздел и вернуться к нему, когда возникнет такая необходимость.

Для тех, кто продолжит чтение, сразу же скажу, что, естественно, в этом коротком разделе будет дано только самое начальное представление о настройке такого сложного программного продукта, как Web-сервер Apache2. Для более глубоко знакомства могу рекомендовать официальный сайт документации по этому продукту http://httpd.apache.org/docs/2.2/.

Apache2 настраивается путем редактирования или добавления директив в обычных текстовых конфигурационных файлах. Эти директивы размещаются в следующих файлах и подкаталогах основного каталога /etc/apache2:

- apache2.conf — главный конфигурационный файл Apache2 который содержит глобальные для Apache2 параметры (см. Приложение 11.2 к этой главе);
- envvars — файл, где устанавливаются переменные окружения Apache2;
- httpd.conf — исторически так сложилось, что это главный конфигурационный файл Apache2. Назван по имени исполнимого файла httpd — «демон» HTTP. Этот файл может быть использован

для специфических, установленных пользователем настроек, которые глобально применяются к Apache2;
- ports.conf — содержит директивы, которые определяют, на каких TCP портах Apache2 принимает соединения;
- /conf.d — подкаталог, который содержит конфигурационные файлы, которые применяются к Apache2 глобально. Другие пакеты, использующие Apache2 для хранения содержимого, могут добавлять файлы или символические ссылки на этот каталог;
- /sites-available — подкаталог, который содержит конфигурационные файлы, описывающие виртуальные хосты;
- /sites-enabled — подкаталог, который содержит символические ссылки на виртуальные хосты, описанные в /etc/apache2/sites-availab, чтобы сделать их активными при рестарте Apache2.

Таким образом, для описания всех доступных сайтов используется папка sites-available. В ней расположены файлы с описанием виртуальных хостов — VirtualHosts, опубликованные же сайты находятся в папке sites-enabled в виде ссылок на файлы доступных сайтов из папки sites-available.

Аналогично в папках mods-available и mods-enabled настраивается доступность модулей, используемых сервером. То есть, чтобы добавить виртуальный хост в apache2, необходимо создать файл нового виртуального хоста в sites-available, а чтобы включить виртуальный хост, необходимо, чтобы в директории sites-enabled была ссылка на файл, описывающий виртуальный хост.

Это сделано для того, чтобы разделить виртуальные домены на уровне хостинга. Например, хостёр чтобы временно удалить какой-то домен, удаляет ссылку из папки sites-enabled и перезапускает веб-сервер, и так же быстро он может включить домен снова, без правки общей конфигурации.

11.5.1. Основные настройки Apache2

По умолчанию Apache 2 имеет конфигурацию, совместимую с виртуальными хостами. В его настройках указан через директиву VirtualHost единственный виртуальный хост. Он может быть оставлен, как есть, если у вас всего один сайт, либо использован как шаблон для других виртуальных хостов, если сайтов у вас несколько.

Если оставить его настройку, как есть, виртуальный хост по умолчанию будет обслуживать ваш основной сайт, а также использовать этот сайт, если URL, по которому пользователи попали на ваш сервер, не обрабатывается ни одним из остальных виртуальных хостов. То есть активным будет основной сайт, если имя хоста не найдено ни в одной директиве ServerName.

Если надо изменить виртуальный хост, описанный по умолчанию, то следует отредактировать файл /etc/apache2/sites-available/default.

Замечание.
Директивы, установленные для виртуального хоста, применяются только для того хоста, для которого они установлены.

Если директива установлена в основной конфигурации сервера и не установлена для конкретного виртуального хоста, то будет использовано значение по умолчанию. Например, вы можете указать адрес электронной почты веб-мастера в основном конфигурационном файле сервера и не указывать его для каждого виртуального хоста.

Если вы хотите настроить новый виртуальный хост или сайт, скопируйте файл /etc/apache2/sites-available/default в папку с выбранным вами для нового сайта именем. Для примера:

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/mysite
```

После этого, чтобы настроить новый сайт, следует отредактировать новый конфигурационный файл, используя директивы, часть из которых описанна ниже.

Директива ServerAdmin

Определяет почтовый адрес администратора сервера, который будет отображаться у пользователя. Значение по умолчанию — webmaster@localhost. Данная переменная должна быть изменена на доступный для вас почтовый адрес, если вы — администратор сервера. Если на вашем сайте возникнут проблемы, Apache2 отобразит ошибку, в которой также будет отображен указанный почтовый адрес с целью сообщения проблемы. Эту директиву можно найти в файле конфигурации сайтов, в каталоге /etc/apache2/sites-available.

Директива Listen

Определяет порт и, при указании, IP адрес, на котором должен работать Apache2. Если IP адрес не указан, Apache2 работает на всех IP адресах, которые доступны компьютеру, на котором он запущен. Значение директивы по умолчанию — порт 80.

Вы можете изменить значение на 127.0.0.1:80, чтобы Apache2 работал только на локальном интерфейсе и не был доступен извне. Также можно указать, например, значение 81 для изменения порта сервера или оставить все, как есть, для работы по умолчанию. Данная директива может быть найдена и изменена в ее собственном файле /etc/apache2/ports.conf.

Директива ServerName

Является необязательной и определяет, на какие FQDN должен отвечать ваш сайт. По умолчанию виртуальный хост не имеет установленной директивы ServerName, поэтому он будет отвечать на все

запросы, которые не совпадают с директивой ServerName на другом виртуальном хосте.

Если вы только что приобрели доменное имя ubunturocks.com и хотите поместить его на своем Ubuntu-сервере, то значение директивы ServerName в конфигурационном файле вашего виртуального хоста должно быть ubunturocks.com. Добавьте эту директиву к новому файлу виртуального хоста, который вы создали ранее (/etc/apache2/sites-available/mynewsite).

Возможно, вы захотите, чтобы ваш сайт отвечал на www.ubunturocks.com, поскольку многие пользователи сочтут подходящим использовать префикс www. Для этого используйте директиву ServerAlias. В директиве ServerAlias вы также можете использовать метасимволы.

Например, следующая конфигурация заставит ваш сайт отвечать на любой запрос домена, оканчивающийся на .ubunturocks.com.

ServerAlias *.ubunturocks.com

Директива DocumentRoot

Она определяет, где Apache2 должен искать файлы, которые составляют сайт. Значение по умолчанию /var/www. Если сайт не сконфигурирован, но вы раскомментировали директиву RedirectMatch в файле /etc/apache2/apache2.conf, то запросы будут перенаправляться в /var/www/apache2-default, где по умолчанию Apache2 будет искать сайт. Измените это значение на виртуальный хост-файл вашего сайта и не забудьте создать этот каталог, если необходимо!

Внимание!!!
Директория /etc/apache2/sites-available не анализируется Apache2. Символические ссылки в /etc/apache2/sites-enabled указывают на «доступные» сайты.

Для включения нового VirtualHost следует использовать утилиту a2ensite, а затем перезапустить Apache2:

```
sudo a2ensite mysite
sudo /etc/init.d/apache2 restart
```

Не забудьте заменить mysite более подходящим именем для VirtualHost. Один из способов — это назвать файл так же, как в директиве ServerName для VirtualHost.

Аналогично следует использовать утилиту a2dissite для выключения сайтов. Это может быть полезным для устранения неполадок в конфигурации для нескольких VirtualHost.

```
sudo a2dissite mynewsite
sudo /etc/init.d/apache2 restart
```

11.5.2. Настройки параметров Apache2 по умолчанию

Знать значения параметров Apache2 по умолчанию необходимо в том случае, если вы, например, добавляете виртуальный хост, настраиваете нужные директивы, а некоторые не указываете. В этом случае будут использоваться значения по умолчанию.

Директива DirectoryIndex

Указывает на страницу (файл) по умолчанию, которую отдает пользователю сервер при запросе индекса каталога, указывая слеш (/) в конце имени каталога. Например, когда пользователь запрашивает страницу http://www.example.com/this_directory/, то он получит страницу из DirectoryIndex, если она существует, или Permission Denied, если ее нет.

Сервер попытается найти один из файлов, перечисленных в директиве DirectoryIndex, и вернет первый найденный. Если он не находит ни одного из этих файлов, и если Options Indexes установлен для этого каталога, то сервер генерирует и возвращает список в HTML-формате в подкаталоги и файлы в каталоге.

Значения по умолчанию определяются в файле /etc/apache2/mods-available/dir.conf и имеют значения «index.html index.cgi index.pl index.php index.xhtml index.htm». Таким образом, если Apache2 находит в запрашиваемом каталоге файл, соответствующий любому из этих имен, то он и будет первым отображаться.

Директива ErrorDocument

Позволяет указать для Apache2 файл, который будет использоваться при возникновении ошибок. Например, если пользователь запрашивает ресурс, который не существует, то возникает ошибка 404, и в соответствии с конфигурацией Apache2 по умолчанию будет отображаться содержимое файла /usr/share/apache2/error/HTTP_NOT_FOUND.html.var.

Этот файл находится не в DocumentRoot сервера, но есть директива Alias в /etc/apache2/apache2.conf, которая перенаправляет запросы к каталогу /errror в каталог /usr/share/apache2/error/. Для просмотра списка всех директив по умолчанию используйте следующую команду:

```
grep ErrorDocument /etc/apache2/apache2.conf
```

Директива CustomLog

По умолчанию сервер ведет журнал обращений в файле /var/log/apache2/access.log. Вы можете изменить этот журнал индивидуально для каждого сайта в вашем виртуальном хостинге, используя директиву CustomLog или принять по умолчанию, как указано в /etc/apache2/apache2.conf.

Директива ErrorLog.

Позволяет указать файл, в котором ведется журнал ошибок. По умолчанию — это /var/log/apache2/error.log. Ошибки протоколируются отдельно от журнала обращений. Содержимое этого файла оказывает помощь в устранении проблем, возникающих с сервером Apache2.

Вы также можете указать параметр LogLevel, его значение по умолчанию — «предупредить», а также параметр LogFormat, значение которого по умолчанию определены в /etc/apache2/apache2.conf.

11.5.3. Настройки httpd

В этом разделе кратко познакомимся с некоторыми основными настройками демона httpd:

Директива LockFile

Устанавливает путь к log-файлу сервера, который используется, если сервер собран с параметрами USE_FCNTL_SERIALIZED_ACCEPT или USE_FLOCK_SERIALIZED_ACCEPT. Он должен располагаться на локальном диске. Значение директивы должно быть оставлено по умолчанию за исключением случая, когда каталог логов находится в разделе NFS. Доступ к файлу должен быть только у суперпользователя.

Директива PidFile

Устанавливает имя файла, в который сервер записывает свой номер процесса (pid). Файл должен читаться только суперпользователем (root). В большинстве случаев следует оставить значение по умолчанию.

Директива User

Определяет параметр userid, по которому сервер отвечает на запросы. Она определяет возможности в доступе для сервера. Любые файлы, недоступные для этого пользователя не будут доступны и для посетителей сайтов. Значение по умолчанию www-data.

Замечание.
Без полного понимания того, что вы делаете, не устанавливайте директиву User в значение root. Использование root как пользователя создаст очень серьезные дыры в безопасности вашего веб-сервера.

Директива Group

По значению похожа на директиву User. Она устанавливает группу, от которой работает веб-сервер. Значение по умолчанию — www-data.

11.5.4. Apache2 — модульный сервер

Apache2 Modules — это значит, что в ядро сервера включены только базовые функции. Расширенные возможности доступны в виде модулей, которые могут быть загружены в Apache2. По умолчанию базовый набор модулей включается в сервер во время компиляции. Если сервер скомпилирован с возможностью использования динамически загруженных модулей, модули могут быть скомпилированы отдельно и добавлены в любое время с помощью директивы LoadModule. Иначе Apache2 должен быть перекомпилирован для добавления и/или удаления модулей.

Ubuntu компилирует Apache2 с возможностью динамической загрузки модулей. Конфигурационные директивы могут быть включены для присутствия конкретного модуля при условии заключения их в блок <IfModule>. Вы можете установить дополнительные модули для Apache2 и использовать их на вашем веб-сервере.

Например, можно использовать следующую команду из командной строки терминала, чтобы установить модуль MySQL Authentication:

```
sudo apt-get install libapache2-mod-auth-mysql
```

Для просмотра доступных дополнительных модулей можно обратиться в директорию /etc/apache2/mods-available. Чтобы включить модуль следует использовать утилиту a2enmod:

```
sudo a2enmod auth_mysql
sudo /etc/init.d/apache2 restart
```

Аналогично используется утилита a2dismod, чтобы отключить модуль:

```
sudo a2dismod auth_mysql
sudo /etc/init.d/apache2 restart
```

11.5.5. Конфигурация HTTPS

Модуль mod_ssl добавляет серверу Apache2 важную особенность — возможность защищенных коммуникаций. Соответственно, когда ваш браузер соединяется с использованием SSL, в адресной строке браузера перед URL используется префикс https://.

Модуль mod_ssl доступен в составе пакета apache2-common. Выполните нижеследующую команду в терминале для включения модуля mod_ssl:

```
sudo a2enmod ssl
```

По умолчанию настройка HTTPS описывается в конфигурационном файле / etc/apache2/sites-available/default-ssl. Для того чтобы Apache2 мог поддерживать HTTPS также требуются ключевой файл и файл сертификатов. Конфигурации по умолчанию будут использовать HTTPS-

сертификат и ключ, сгенерированные пакетом SSL-сертификатов. Они хороши для тестирования, но автоматически сгенерированный сертификат и ключ должны быть заменены на уникальные для конкретного сайта или сервера. О генерации ключей и получении сертификатов обратитесь к специализированной документации.

Чтобы сконфигурировать Apache2 на режим работы по протоколу HTTPS, следует использовать утилиту a2ensite:

```
sudo a2ensite default-ssl
```

Каталоги /etc/ssl/certs и /etc/ssl/private — это места по умолчанию. Если вы установили сертификат и ключ в другие каталоги, убедитесь, что SSLCertificateFile и SSLCertificateKeyFile тоже изменены.

Для того чтобы активировать настройку Apache2 для работы по HTTPS, следует выполнить рестарт сервиса:

```
sudo /etc/init.d/apache2 restart
```

После выполнения этих настроек и наличия на сервере файлов уникальных ключей и сертификатов пользователи могут получить доступ к страницам через безопасное соединение, набрав в адресной строке браузера https://your_hostname/url/.

11.6. Простейший пример виртуального хостинга

Виртуальный хостинг предполагает возможность организации на одном Web-сервере нескольких автономных сайтов, к каждому из которых можно обращаться по собственному зарегистрированному адресу.

Стандартный подход к организации виртуального хостинга включает в себя следующий набор действий:

- создание контента каждого сайта в отдельном каталоге,
- формирование файла описания сайта,
- подключение сайта в состав Web-сервера.

При этом подходе предполагается, что у каждого хоста будет собственное доменное имя. В нашей виртуальной сети отсутствует DNS сервер. Но при этом в файлах hosts на всех виртуальных машинах мы уже ранее прописали соответствия символических имен хостов нашей виртуальной сети их IP-адресам.

Однако для того, чтобы при выполнении этого маленького примера не мучиться с файлами hosts на всех машинах, мы пойдем несколько нестандартным путем. Стандартно доступ по протоколу HTTP к сайтам с различными доменными именами осуществляется по 80 порту.

Мы поступим наоборот: оставим постоянное имя сервера, а для каждого сайта будем использовать различные порты. Это упростит наш пример и

позволит просто изложить основные моменты технологии виртуального хостинга.

Предположим, что, кроме основного, уже настроенного на vmUbuntu10 ранее Web-сайта (рис. 11.3), мы хотим разместить на этом Web-сервере еще два web-сайта. В этом случае одна из возможных последовательностей может включать в себя следующие этапы:

➢ Пусть администратором нашего Web-сервера будет пользователь serp, который для организации виртуального хостинга создаст в своей рабочей директории каталог www с двумя подкаталогами test1 и test2 для контента каждого из двух сайтов.

```
serp@vmUbuntu10:~$ mkdir www
serp@vmUbuntu10:~$ mkdir www/test1
serp@vmUbuntu10:~$ mkdir www/test2
```

➢ Сформируем контент каждого из сайтов, который в нашем примере будет состоять всего из одного файла index.html. Каждый из этих файлов будет содержать всего по одной строке TEST 1 и TEST 2 соответственно.

```
serp@vmUbuntu10:~$ echo "<html><body><h1> TEST 1 </h1>
</body></html>" > ./www/test1/index.html

serp@vmUbuntu10:~$ echo "<html><body><h1> TEST 2 </h1>
</body></html>" > ./www/test2/index.html
```

➢ Сформируем файл конфигурации для первого виртуального хоста. Вместо шаблона стандартного файла описаний (см. Приложение 11.3) сформируем в /etc/apache2/sites-available/ простейший файл, в котором укажем, что контент первого сайта находится в директории /home/serp/www/test1/, а входным портом этого сайта будет порт 81 вместо стандартного 80.

```
serp@vmUbuntu10:~$ cat /etc/apache2/sites-available/test1
<VirtualHost *:81>
   DocumentRoot /home/serp/www/test1/
</VirtualHost>
```

➢ Сформируем файл конфигурации для второго виртуального хоста. Все аналогично вышесказанному, только его входной порт будет 82, а стартовый каталог /home/serp/www/test2/.

```
serp@vmUbuntu10:~$ cat /etc/apache2/sites-available/test2
<VirtualHost *:82>
   DocumentRoot /home/serp/www/test2/
</VirtualHost>
```

➢ Для формируемых сайтов решили использовать порты, которые не являются стандартными порты HTTP. Это вызывает необходимость описания этих портов в файле конфигураций /etc/apache2/ports.conf.

С этой целью в него необходимо добавить две новые строки Listen 81 и Listen 82.

```
serp@vmUbuntu10:~$ cat /etc/apache2/ports.conf
 . . .
NameVirtualHost *:80
Listen 80
Listen 81
Listen 82
 . . .
```

➢ Имя нашего Web-сервера опишем с помощью директивы ServerName в конфигурационном файле /etc/apache2/httpd.conf.

```
serp@vmUbuntu10:~$ cat /etc/apache2/httpd.conf
    ServerName vmUbuntu10
```

➢ Следующие команды будут выполняться от имени суперпользователя.

```
serp@vmUbuntu10:~$ sudo -s
root@vmUbuntu10:~#
```

➢ Убедимся в том, что конфигурационные файлы виртуальных хостов существуют в каталоге /etc/apache2/sites-available.

```
root@vmUbuntu10:~# ls   -l /etc/apache2/sites-available
-rw-r--r-- 1 root root  948 2010-11-19 00:16 default
-rw-r--r-- 1 root root 7467 2010-11-19 00:16 default-ssl
-rw-r--r-- 1 root root   70 2011-08-16 18:47 test1
-rw-r--r-- 1 root root   71 2011-08-16 18:48 test2
```

➢ Для включения новых виртуальных хостов в состав Web-сервера Apache используем его утилиту a2ensite.

```
root@vmUbuntu10:~# sudo a2ensite test1
Enabling site test1.
Run '/etc/init.d/apache2 reload' to activate new
configuration!

root@vmUbuntu10:~# sudo a2ensite test2
Enabling site test2.
Run '/etc/init.d/apache2 reload' to activate new
configuration!
```

➢ Убедимся, что подкаталог /etc/apache2/sites-availab содержит символические ссылки на вновь сформированные виртуальные хосты вместе со ссылкой на ранее организованный сайт по умолчанию.

```
root@vmUbuntu10:~# ls -l  /etc/apache2/sites-enabled/
lrwxrwxrwx 1 root root 26 2011-08-10 11:14 000-default ->
../sites-available/default
lrwxrwxrwx 1 root root 24 2011-08-16 19:07 test1       ->
../sites-available/test1
lrwxrwxrwx 1 root root 24 2011-08-16 19:07 test2       ->
../sites-available/test2
```

➢ Чтобы сделать активными новые виртуальные хосты перезапустим Apache2.

```
root@vmUbuntu10:~# sudo /etc/init.d/apache2 restart
 * Restarting web server apache2        ... waiting
 [ OK ]
```

➢ Выйдем из режима суперпользователя.

```
root@vmUbuntu10:~# exit
exit
serp@vmUbuntu10:~$
```

На этом настройку виртуального хостинга в рассматриваемом нами небольшом примере можно считать законченной и следует перейти к его тестированию. Таким образом, на текущей момент в нашей виртуальной сети на базе одной основной и нескольких виртуальных машин:

- на vmUbuntu10 установлен и настроен Web-сервер;
- на Web-сервере организован и настроен виртуальный хостинг трех различных сайтов;
- для каждого сайта сформирован простейший контент, состоящий из одной html-страницы для каждого из сайтов.

Если все настройки выполнены верно, то, запустив Internet Explorer на любой Windows-машине нашей виртуальной сети и вводя последовательно в строке адреса http://vmUbuntu10:81 и http://vmUbuntu10:82 получим результат, аналогичный приведенному на рис. 11.14.

Рис. 11.14. Доступ к виртуальным хостам Web-сервера Apache с Windows-узла сети.

Из приведенного рисунка видно, что мы имеем доступ к двум различным ресурсам, которые расположены на одном и том же Web-сервере. Причем доступ к каждому из них осуществляется по одному и тому же IP-адресу, но разным TCP-портам.

Следует отметить, что сохранился доступ и к основному хосту сервера. В этом можно убедиться, если в строке адреса Internet Explorer ввести:

- http://vmUbuntu10:80,
- либо просто http://vmUbuntu10,
- или http://192.168.1.10.

Результат должен быть один и тот же и соответствовать рис. 11.3. И на этом мы закончим очень краткое знакомство с сетевой технологией на базе web-серверов и свободным доступом к ним. То есть с технологией которая составляет основу сегодняшнего World Wide Web.

12. ПОДКЛЮЧЕНИЕ К UBUNTU И WINDOWS МОБИЛЬНЫХ УСТРОЙСТВ

12.1. Доступ с мобильных устройств к удаленным рабочим столам

Развитие беспроводных коммуникаций, с одной стороны, и широкое распространение мобильных устройств, с другой — привели к изменению подходов к работе людей ряда профессий. Врач в больнице, брокер на бирже, главный строитель на стапеле и многие другие не могут сегодня обойтись без удаленного мобильного доступа как к своим, так и к корпоративным информационным ресурсам.

Возможны различные варианты доступа. Исторически первыми были корпоративные базы данных, когда имелось специальное клиентское программное обеспечение для доступа к ним. Постепенно, с развитием технологий Интернет появлялся Web-доступ к корпоративным данным. Следующий этап — это облачные хранилища данных. Но все эти этапы характеризуются тем, что пользователь может воспользоваться и иметь доступ только к тем данным, которые являются корпоративно ценными.

Представьте себе инженера-конструктора, который пришел на техническое совещание к руководству, и возник какой-либо сложный вопрос. Чертежи, исходные данные и программы расчетов находятся на его компьютере двумя этажами ниже. Что ему делать? Срываться с места и бежать за документацией? И это при условии, что в комнате совещаний есть беспроводная точка доступа в корпоративную сеть и Интернет.

Да, какие-то готовые решения хранятся в его разделе на сервере организации, но странно хранить там свои программы и текущие расчеты. Тем более сомнительным представляется хранение этих достаточно конфиденциальных данных в общедоступных облачных хранилищах. Единственный вариант, который может помочь этому специалисту, — непосредственный доступ к его собственному компьютеру, хотя бы и удаленный. Но у него с собой ничего нет, кроме мобильного устройства.

Такая же проблема может возникнуть и у малыша, который захочет смотреть мультики, но они на папином десктопном компьютере. А у папы срочная работа, и отойти от компьютера он не может. Но у мамы есть смартфон с достаточно большим дисплеем, который, подключив к папиному компьютеру, можно отдать ребенку. И все довольны при

условии, что папа пробросил мамину любимую музыку со своего компьютера на свой маленький телефон и отдал его маме.

Вы скажете: «А в чем проблема? Мы уже прочитали главу 8 и знаем про удаленные рабочие столы». Вы правы, но там была виртуальная сеть с Windows и Ubuntu, а тут «живые» гаджеты: iPhone или iPad с iOS, либо Samsung Galaxy S или Samsung Galaxy Tab с операционной системой Android. Как их превратить в узел нашей гетерогенной сети для доступа к компьютерам на базе операционных систем Windows или Ubuntu.

12.2. Использование 2X Client для доступа к удаленным рабочим столам

Существует множество программных продуктов, осуществляющих доступ к удаленным рабочим столам по протоколам RDP или VNC. На момент подготовки рукописи к изданию для мобильных устройств наиболее популярным и, что немаловажно, бесплатным продуктом является 2X Client.

Установив и применяя приложение 2X Client, вы всегда будете на связи с домашним или офисным компьютером. Оно позволяет подключаться к удаленному рабочему столу, используя протокол RDP. Версии этой программы существуют для многих мобильных устройств на основе iOS, Android и др. К основным возможностям этой программы можно отнести:

- Доступ к удаленным рабочим столам и файлам.

Из среды 2X Client вы можете просто подключиться к любому доступному компьютеру в сети, используя подключение по RDP. Настройки подключений сохраняются для последующих сеансов.

Возможен безопасный доступ к домашнему или рабочему ПК, размещенным на сервере, виртуальным рабочим столам и файлам из любой точки мира. Например, используя Facebook.

- Работа с удаленными приложениями.

Бесперебойная работа на вашем ПК или ноутбуке с размещенными на сервере удаленными приложениями, такими как Microsoft Office.

- Высокий уровень безопасности.

Прозрачное подключение к опубликованным Windows приложениям или рабочим столам прямо из вашего Facebook с помощью защищенного RDP SSL подключения.

- Интеграция с 2X ApplicationServer XG.

Доступ к виртуальному рабочему столу и приложениям, расположенным на таких гипервизорах, как Microsoft Hyper-V, Citrix Xen, VMware vSphere, и других.

Существует несколько способов загрузки 2X Client на ваше устройство. Это можно сделать, посетив сайт адресу www.2x.com, либо интернет-

магазин iTunes (IPad и iPhone) или Google Play (Samsung Galaxy S и Samsung Galaxy Tab).

Дальнейшее изложение процесса загрузки, запуска, настройки и подключения рассмотрим на примере Apple iPad с операционной системой iOS. Для других мобильных устройств и смартфонов, типа iPhone и Samsung Galaxy эти процессы практически аналогичны.

После загрузки 2X Client с какого-либо источника на Интернет-планшет iPad, на рабочем столе последнего появится новая пиктограмма с надписью 2X (рис. 12.1).

Рис. 12.1. Рабочий стол после загрузки 2X Client.

Для запуска приложения достаточно нажать на рабочем столе пиктограмму 2X. Откроется окно Connection, где будут указаны все настроенные ранее подключения. Если на данный момент настроенных подключений нет, то вид окна будет такой, как приведен на рис. 12.2.

Рис. 12.2. Рабочий стол после загрузки 2X Client.

Сообщение, которое появляется в этом окне, предлагает нажать на пиктограмму с символом '+' для создания нового 2X подключения или нового RDP подключения.

Первый вариант предполагает организацию подключения к X-серверу, второй к RDP-серверу. Напомню, что RDP-сервер является составной частью Windows, а на Ubuntu мы его устанавливали отдельно. Именно RDP-сервер поддерживает работу Windows в режиме ее удаленного рабочего стола. Следует отметить, что с помощью 2X Client для iOS есть возможность организовать подключения к нескольким рабочим столам терминальных серверов.

Создание нового подключения по протоколу RDP

Для создания нового подключения, нажимаем пиктограмму с символом '+' в предыдущем окне. Появляется окно «New Connection» (рис. 12.3), в котором появляется возможность выбора одного из двух типов организации подключения.

Рис. 12.3. Вид окна New Connection.

При желании организовать подключение к компьютеру на базе Windows, выбираем подключение по протоколу RDP. Открывается окно New RDP Connection (рис. 12.4).

Рис. 12.4. Вид окна New RDP Connection.

Это окно служит для задания основных параметров настройки подключения к конкретному рабочему столу определенного компьютера, на котором разрешен доступ к его рабочему столу. К основным параметрам относятся

- Alias — название, которое будет определять создаваемое подключение, и отображаться при всех входах на соответствующий удаленный рабочий стол.
- *Address — имя или IP-адрес сервера, к которому формируется подключение.
- *Port — номер порта, который для всех подключений RDP по умолчанию устанавливается 3389.
- *User Name/Password — имя и пароль пользователя, имеющего право на подключения к удаленному рабочему столу.
- * — данный символ перед параметром, означает, что задания этого параметра обязательно для формирования подключения.
- Connect to Console (Подключиться к консоли) – позволяет выбрать консольное подключение к рабочему столу RDP (используется только для сервера Windows 2003).

Дополнительные настройки

Фрейм Additional Setting окна новых RDP подключений дает возможность выполнить настройку параметров передачи и отрисовку экрана удаленного рабочего стола на клиентском устройстве. Коротко рассмотрим наиболее полезные настройки в каждой из предлагаемых групп.

➢ Display.

При выборе этой опции появится новое окно в котором можно установить нужные пользователю параметры дисплея (рис. 12.5).

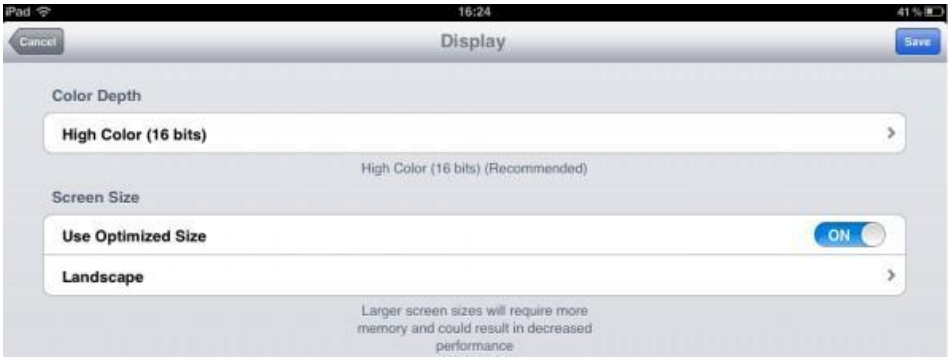

Рис. 12.5. Окно настройки параметров экрана.

В этом окне есть возможность установить цветовое разрешение (Color Depth), выбрав один из трех предлагаемых вариантов: 256 цветов, High Color (16-бит) или True Color (24-бит). По умолчанию используется 16-битовая передача цвета. Наряду с этим можно включить или отключить

опцию, которая заставляет компьютер, самому выбирать наиболее подходящее разрешение экрана. И, наконец, опция Landscape, которая позволяет осуществить выбор между пейзажной или портретной ориентациями экрана.

- Local Resources.

  Название опции — локальные ресурсы. Звучит солидно, но она обеспечивает только одну функцию: включение или отключение передачи и воспроизведения звука.

- Experience Settings.

  Опция Настройки производительности выглядит значительно солиднее, и при ее выборе открывается новое окно (рис. 12.6).

Рис. 12.6. Окно настройки параметров экрана.

Появляется возможность, в зависимости от мощности вашего мобильного устройства и скорости канала связи с удаленным рабочим столом, включить или отключить некоторые режимы доступа к удаленному рабочему столу и процессу работы в нем.

- Фон рабочего стола — при отключении обои удаленного рабочего стола не будут отображаться у клиента.
- Сглаживание шрифтов — включение этого параметра обеспечивает сглаживание всех шрифтов для удобства чтения.
- Анимация окон и меню — когда отключен этот параметр, меню появится мгновенно.
- Расположение рабочего стола.
- Показ содержимого окна во время перетаскивания.
- Темы.
- Кэширование изображений.
- Разрешить сжатие.

Рекомендуется включать сжатие и кэширование растровых изображений, для обеспечения более эффективной связи. При работе в

ЛВС на скоростях 100 мегабит и выше доступно ON всех параметров. В иных случаях — все зависит от мощности канала связи.

12.3. Пример подключения мобильных устройств к Ubuntu и Windows машинам

Если вы помните, в нашей виртуальной сети используются виртуальные машины как на базе Windows, так и на базе Ubuntu Linux. Можно ли удаленно управлять этими машинами с современных мобильных устройств? Несомненно, и точно так же, как это используется между платформами Microsoft. Единственное требование — это наличие беспроводного канала связи между этими устройствами. И тогда вы можете на удаленном рабочем столе сколько угодно тыкать пальцами и запускать приложения точно так же, как если бы вы сидели прямо перед компьютером, но двигали мышь. В рассматриваемом примере, для реализации задуманного, требуется, чтобы:

- основной компьютер был подключен к точке доступа, которая поддерживает публичную Wi-Fi сеть,
- были в наличии мобильные устройства, способные подключиться к этой беспроводной сети.

При этом желательно, чтобы мобильные устройства были разного класса и базировались на разных платформах. В качестве таких устройств рассмотрим (рис. 12.7) наиболее широко распространенные на сегодняшний день типы, а именно:

- Планшетный компьютер iPad с на платформе iOS.
- Смартфон Samsung Galaxy S II на платформе Android.

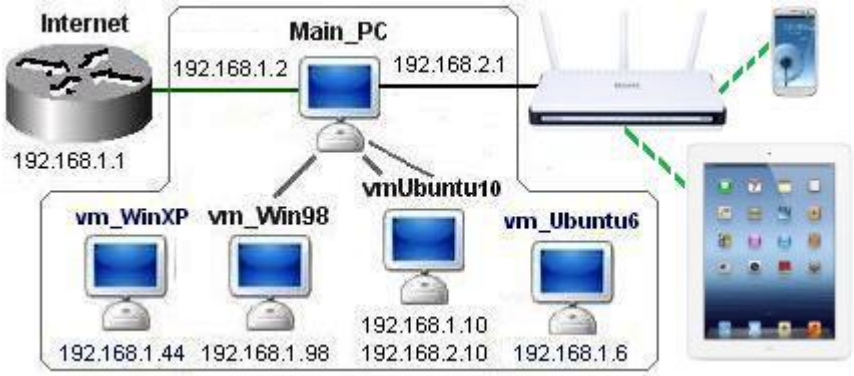

Рис. 12.7. Структура тестовой системы.

Выбор этих устройств не случаен, так как настройка подключений и работа с удаленным рабочим столом в iPad, полностью аналогична тем же действиям в медиапроигрывателях iPod и смартфонах iPhone. Это же можно сказать и об однотипности действий, выполняемых со смартфонами Samsung Galaxy S и планшетными компьютерами Samsung Galaxy Tab.

Естественно, это высказывание справедливо в том случае, когда на всех этих устройствах используется один и тот же программный продукт, например 2X Client, разработанный компанией 2X Software.

До начала работы с мобильными устройствами, следует проверить, существует ли реально доступ с основного компьютера к удаленному рабочему столу, например, vm-WinXP (192.168.1.44), и вы помните логин и пароль пользователя, которому этот доступ разрешен. Это нужно для того, чтобы не было проблем в дальнейшей работе с мобильными устройствами.

Для примера рассмотрим процесс подключения с мобильных устройств на одну из виртуальных машин нашей тестовой сети. Пусть это будет, уже упомянутая выше vm-WinXP (192.168.1.44). Для этого надо:

- Загрузить и установить на устройстве приложение 2X client.
- Запустить приложение на выполнение.
- Создать новое подключение, выбрав тип Подключение RDP.
- Настроить это подключение на связь с vm-WinXP (192.168.1.44).

  Для этого необходимо указать, как минимум, адрес хоста рабочего стола, к которому должно выполняться подключение, а также логин и пароль пользователя, у которого есть права на работу с этим рабочим столом. Для случая использования Apple iPad это может быть аналогично фото, приведенному на рис. 12.8.

Рис. 12.8. Фрагмент экрана iPad — новое подключения RDP.

- Просмотреть настройки подключения, которые установлены по умолчанию. В нашем примере их можно не менять.
- Сохранить описание вновь созданного подключения по RDP.

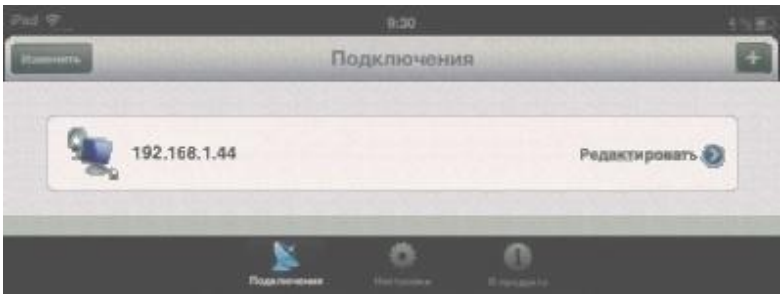

Рис. 12.9. Список сформированных подключений RDP.

После сохранения нового подключения откроется окно, в котором будет представлен список (рис. 12.9) всех сформированных подключений к доступным рабочим столам. Видно только одно подключение, которое было только что сформировано. При необходимости оно может быть отредактировано. Для этого используют кнопку Редактирование.

В нижней части экрана три пиктограммы: Подключение, Настройки и О продукте. Уже из названий ясно назначение каждой из них. При этом вполне естественным является желание попробовать: «А что же будет, если …?». И если выбор остановился на опции Подключение, то Apple iPad, используя приложение 2X Client, установит связь с vm-WinXP.

При этом на экране Apple iPad отобразится удаленный рабочий стол именно того пользователя, логин и пароль которого указан в настройках подключения. Один из возможных вариантов выполнения описанных выше действий приведен на фото, представленном на рис. 12.10.

Рис. 12.10. Планшет Apple iPad, подключенный к виртуальной vm-WinXP базового компьютера.

На фото планшет расположен рядом с базовым компьютером, но результат не изменится при его перемещении в области действия Wi-Fi сети, которая даже для бытовых точек доступа может составлять 30–60 метров.

Следует отметить, что если в предыдущих главах мы в основном работали с виртуальными устройствами, то в этом разделе мы оперируем с реальными мобильными устройствами. Ранее было заявлено желание познакомиться с классом этих устройств на базе различных платформ.

Технология работы с iPad на платформе iOS, полностью аналогична работе медиапроигрывателей iPod и смартфонов iPhone. Перейдем теперь к другому типу устройств – смартфону Samsung Galaxy S II на платформе Android.

Выполним всю вышеописанную последовательность действий, используя для этого также приложение 2X Client компанией 2X Software,

но только в версии под Android. Различие будет заключаться только в несколько видоизмененном графическом интерфейсе, но суть, параметры настройки, основные режимы работы программы остаются неизменными. Поэтому повторять ту же технологию, но применительно к Samsung Galaxy S II, является нецелесообразным. Достаточно только сказать, что если на смартфоне будет выбран режим подключения, то его экран примет вид, аналогичный фото, приведенному на рис. 12.11.

Рис. 12.11. Планшет Apple iPad, подключенный к виртуальной vm-WinXP базового компьютера.

Как видно из рис. 12.10 и 12.11, мы получили доступ к vm-WinXP, точнее, к рабочему столу одного из пользователей этой машины. Причем существует возможность выполнять на этой машине все, доступные конкретному пользователю действия, включая запуск любых приложений. Например, у нас появилась возможность как на смартфоне с Android, так и на планшете iPad запустить пасьянс Windows и спокойно переставлять карты на своем мобильном устройстве, хотя все операции выполняются на удаленной машине.

Однако если такой подход доступен дома или в небольшой офисной ЛВС, то корпоративный подход к применению данной сетевой технологии имеет свои сложности и особенности.

12.4. Безопасность мобильного доступа

С развитием беспроводных сетей у любого современного человека появилась возможность пользоваться публичными и корпоративными сервисами в любом месте. Однако мобилизация, кроме преимуществ, таит в себе и опасности. Утеря устройства, ошибки при задании параметров, использование публичных сетей при работе с конфиденциальной информацией — все это может привести к утечкам ценных данных.

Все больше организаций предоставляют своим сотрудникам доступ к корпоративным ресурсам, что позволяет им выполнять свою работу как в

офисе, так и удаленно, используя планшеты и смартфоны. И если пять лет назад большинство телефонов, применяемых для работы, управлялись операционной системой Blackberry, то теперь к ней добавились Apple и Android. При этом использование удаленного доступа помимо удобств содержит в себе и потенциальные угрозы политике корпоративной информационной безопасности. Для минимизации этих рисков возникает необходимость наладить контроль над мобильными устройствами.

Наиболее популярный сервис для корпоративных пользователей мобильных устройств — это почта. И уже здесь надо думать о настройке политик безопасности клиентов. Так, в сервере Microsoft Exchange наряду с управлением паролями используются такие политики безопасности, как: требование ручной синхронизации в роуминге; разрешение на использование камеры; разрешение на использование доступа в Web; удаленная полная очистка данных с устройства.

Более продвинутым является доступ для мобильных устройств к приложениям документооборота, ERP, CRM и другим. Здесь требования безопасности послужили основной появления систем класса Mobile Device Management (MDM), предназначенных для управления политиками безопасности мобильных устройств и распространения настроек для доступа к различным сервисам (VPN, почта, Wi-Fi и т. д.).

Набор политик для каждой из мобильных платформ (Apple iOS, Android, Windows Phone, Symbian и т. д.) включает совпадающие настройки, такие как парольная политика, и индивидуальные, которые сильно зависят от архитектуры. Например, функционал, связанный с iCloud и iTunes, уникален для платформы iOS, а возможность отключения фотокамеры есть и в iOS, и в Android. Для определения политик, поддерживаемых разными платформами, необходимо установить агенты системы MDM.

Для поддержки мобильных устройств надо иметь платформу, поддерживающую управление устройствами сотрудников и параметрами их безопасности. В решениях от компании Good Technology ключевой компонент — центр управления сетью, который проверяет подлинность соединения устройства с серверами компании, управляет маршрутизацией и гарантирует подключение к серверам организации только конкретных устройств. Вся передаваемая информация шифруется. Поддерживаются популярные операционные системы: iOS, Android и Windows Phone.

Существуют и другие подходы к обеспечению безопасности мобильных подключений. Выше рассматривался бесплатный программный продукт 2X Client компании 2X Software. Но эта же компания разрабатывает и достаточно надежные серверные платформы мобильного доступа различных структурных организаций. Она создает решения виртуализации рабочих столов и доставки приложений, удаленного и корпоративного доступа.

Сегодня облачные вычисления уже не столько обеспечивают конкурентное преимущество, сколько являются насущной необходимостью. 2X Software предлагает широкий выбор решений, которые значительно упрощают и удешевляют переход к облачным вычислениям. Семейство продуктов компании включает:

- 2X ApplicationServer XG. Совмещает платформенно независимые виртуальные рабочие столы, доставку приложений и централизованное управление тонкими клиентами в одном продукте.
- 2XOS. Для трансформации настольных ПК в тонкие клиенты
- 2X Client RDP/Remote Desktop. Для удаленного доступа к виртуальным рабочим столам и приложениям Windows, используя платформы Android, iOS, BlackBerry и другие.

2X Software предлагает централизованное управление тонкими клиентами для виртуальных рабочих столов и доставки приложений, используя модуль 2X ClientManager для 2X ApplicationServer XG. В арсенале фирмы и продукт Facebook Client App для безопасного доступа к приложениям и рабочим столам Windows, превращающий Facebook в мощный бизнес-инструмент.

13. ВЗАИМОДЕЙСТВИЕ UBUNTU И WINDOWS ЧЕРЕЗ ОБЛАКА

13.1. Общие сведения об облачных вычислениях

Под облачными вычислениями (cloud computing) понимают такую технологию обработки и хранения данных, при которых вычислительные мощности и инфраструктура хранения данных предоставляются пользователю как Интернет-сервис, доступный с любого устройства, имеющего выход в Интернет, будь то домашний компьютер, ноутбук или смартфон, или даже игровая приставка. Данные и приложения хранятся постоянно на серверах, образующих облако, и кэшируются на вычислительных устройствах пользователя, с которых он осуществляет доступ к облачному хранилищу.

Для объединения серверов в облака используется специализированное программное обеспечение, которое обеспечивает мониторинг состояния серверного оборудования, балансировку нагрузки между серверами, обеспечение ресурсов для решения задачи. На устройствах пользователя для доступа к облакам могут использоваться как специализированное клиентское программное обеспечение, так и Web-браузеры. Развитие облачных технологий идет по нескольким основным направлениям:

- Infrastructure as a service (IaaS) — инфраструктура как сервис.
- Platform as a service (PaaS) — платформа как сервис.
- Software as a service (SaaS) — программное обеспечение как сервис.
- Storage as a service (STaaS) — хранилище как услуга и ряду других.

Концепция IaaS подразумевает предоставление пользователю вычислительных, сетевых ресурсов и ресурсов хранения данных в виде единой инфраструктуры. Таким образом, пользователь получает в распоряжение требуемые ему вычислительные мощности без капитальных и временных затрат на приобретение, установку и обслуживание оборудования и программного обеспечения, без необходимости содержать штат системных администраторов и т. п. При этом пользователь оплачивает использование инфраструктуры повременно, за то время, которое он фактически пользовался предоставляемыми ему мощностями.

Остальные направления развития облачных технологий можно рассматривать как составные части концепции IaaS.

Направление PaaS подразумевает предоставление пользователю некоторой компьютерной платформы, которая может включать в себя в общем виде операционную систему, среду разработки ПО, базы данных, веб-сервер.

Если модели IaaS и PaaS подходят больше для корпоративных пользователей, то концепции SaaS и STaaS представляют интерес и для конечных пользователей.

Концепция SaaS подразумевает предоставление пользователю программного обеспечения на основе повременной оплаты. Согласно данной концепции пользователь не покупает ПО, а как бы берет его в аренду, причем использует только те функции, которые ему нужны, и соответственно платит только за них. Этот подход удобен, когда пользователю изредка (например, раз в год) требуется какая-либо программа на короткий промежуток времени для решения определенной задачи или круга задач. Это может быть составление годового отчета, создание видеоролика и т. п. В этом случае стоимость покупки программного обеспечения может оказаться неоправданно высокой, а использование программного обеспечения как облачного сервиса позволяет избежать больших затрат. Кроме того, существуют сервисы SaaS, предоставляющие определенное ПО бесплатно.

Наверное, наиболее распространенной моделью облачных вычислений является STaaS — хранилище как сервис. Концепция STaaS подразумевает предоставление пользователю в облаке дискового пространства для хранения различных пользовательских данных: документов, фото- и видеоматериалов, списка контактов, электронной почты и т. п. Использование облачных хранилищ позволяет осуществлять доступ к данным с любых устройств, имеющих доступ в Интернет, что повышает мобильность, осуществляет обмен данными между устройствами, а при использовании специализированного ПО осуществляет синхронизацию данных между различными вычислительными устройствами пользователя, будь то офисный компьютер, домашний ноутбук или смартфон.

13.2.Облачные вычисления в Ubuntu

Впервые поддержка облачных решений в Ubuntu была реализована с запуском сервиса Ubuntu One, начиная с версии ОС Ubuntu 9.10. Согласно информации из Википедии: «Ubuntu One — онлайн-хранилище, которое разрабатывается компанией Canonical, для пользователей Ubuntu, предназначенное для обмена файлами и синхронизации между компьютерами и мобильными устройствами, а также сервис потоковой музыки, с помощью которого можно прослушивать загруженную музыку на мобильных устройствах с Android, iPhone или iPad».

Таким образом, в Ubuntu реализована концепция STaaS. Для работы с Ubuntu One существуют специализированные клиенты, доступные для ОС Ubuntu, ОС Windows, Mac OS, Android, iOS, также доступ к хранилищу может быть осуществлен через Web-интерфейс (https://one.ubuntu.com/).

Для того чтобы воспользоваться сервисом Ubuntu One, необходимо пройти процедуру регистрации. Регистрация может быть произведена на сайте https://login.ubuntu.com или через программу клиент, установленную на компьютере. При регистрации нового аккаунта пользователю бесплатно выделяется 5 Gb пространства в online-хранилище для пользовательских данных. При необходимости пользователь может увеличить этот объем за определенную плату.

При входе в online-хранилище через Интернет пользователь может загружать файлы в хранилище и скачивать их оттуда, создавать и редактировать фотоальбомы, создавать и редактировать список контактов, прослушивать музыку. На рис. 13.1 приведен скриншот Web-интерфейса хранилища Ubuntu One.

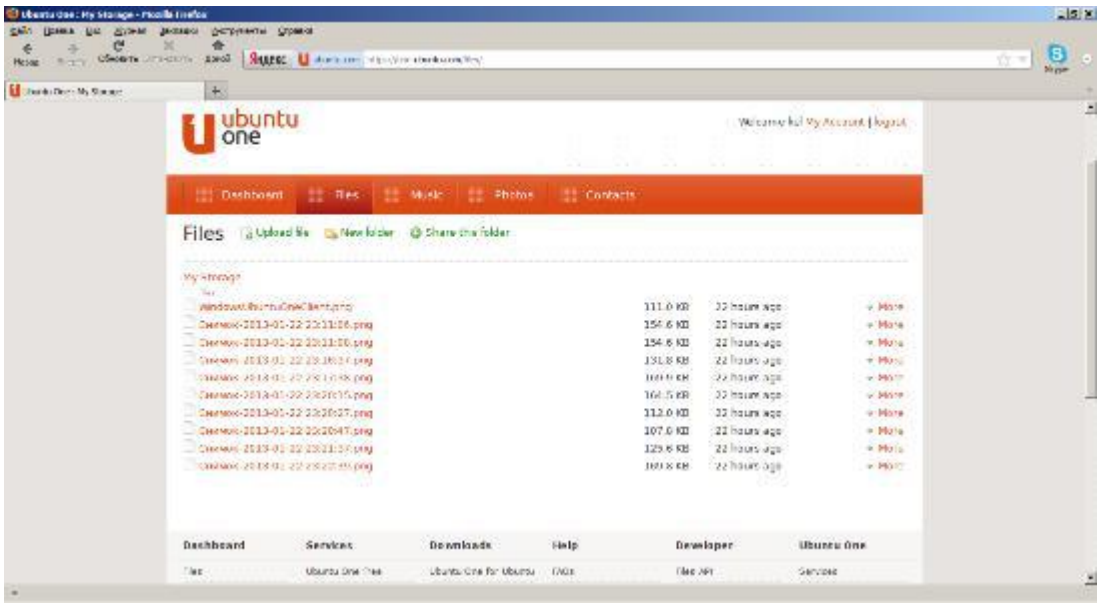

Рис.13.1. Доступ к online-хранилищу Ubuntu One через Web-интерфейс.

Доступ к хранилищу может быть осуществлен с любого устройства, имеющего выход в Интернет и Web-браузер. Однако у использования Web-интерфейса есть существенный недостаток: допустим, пользователю необходимо иметь возможность редактировать файл, находящийся в хранилище, с домашнего и рабочего компьютера. При работе через Web-интерфейс каждый раз, когда пользователю понадобится отредактировать файл, он должен его скачать на компьютер, отредактировать и затем вновь загрузить его в хранилище. Такой подход, кроме того что он явно неудобен, еще и небезопасен: фактически имеется несколько копий одного файла на разных устройствах, а задача синхронизации файлов возлагается на пользователя.

Если после редактирования файла пользователь однажды забудет загрузить его обратно в хранилище, синхронизация файлов нарушится, и часть данных может быть утеряна.

Гораздо более удобный способ работы с данными из хранилища представляет использование специализированных клиентов. Ниже представлено стартовое окно клиента Ubuntu One в ОС Ubuntu (рис. 13.2).

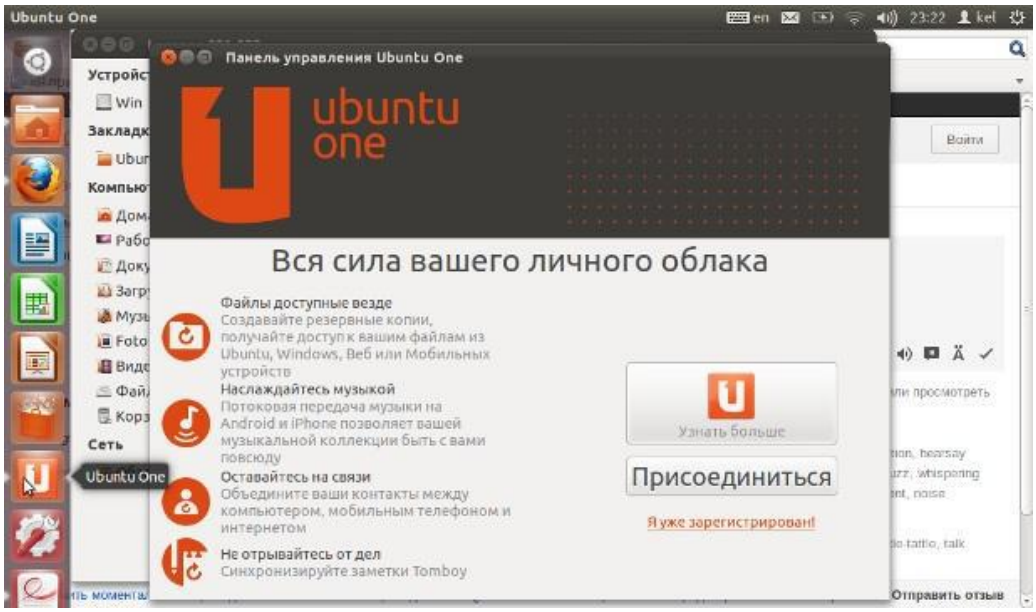

Рис.13.2. Панель управления Ubuntu One в ОС Ubuntu.

При установке на компьютере клиента Ubuntu One в домашней папке пользователя появляется новая папка с именем Ubuntu One. Особенность этой папки в том, что ее содержимое автоматически синхронизируется с online-хранилищем:

- все файлы и папки, которые пользователь добавляет в каталог Ubuntu One, автоматически добавляются в хранилище;
- в процессе редактирования файлов – при сохранении измененный документ автоматически синхронизируется с online-хранилищем.

При этом пользователь с документами из online-хранилища работает точно так, как и с обычными локальными файлами, хранящимися на данном компьютере. Синхронизация файлов между online-хранилищем и компьютером происходит прозрачно для пользователя.

Но если полезная для пользователя информация отделена от компьютера, на котором она создавалась или редактировалась, то, стало быть, становится неважно, с какого из компьютеров осуществляется доступ к этой информации, важно — каким способом.

Выше был рассмотрен подход, когда доступ к облачному хранилищу выполнялся с помощью клиента Ubuntu One. А если такого же клиента пользователь установит не только на рабочем, но и на домашнем компьютере, то он сможет работать с файлами на любом из компьютеров как с локальными файлами, не заботясь об их синхронизации.

Причем эти компьютеры могут работать под управлением совершенно разных операционных систем. И сами компьютеры могут быть совершенно разных типов: десктопы, ноутбуки или планшеты. Важно, чтобы для этих платформ существовал клиент доступа к облачному хранилищу, то есть клиент Ubuntu One. На рис. 13.3 и 13.4 приведены скриншоты Windows и Ubuntu машин, на которых установлены клиенты Ubuntu One, подключенные к одному хранилищу.

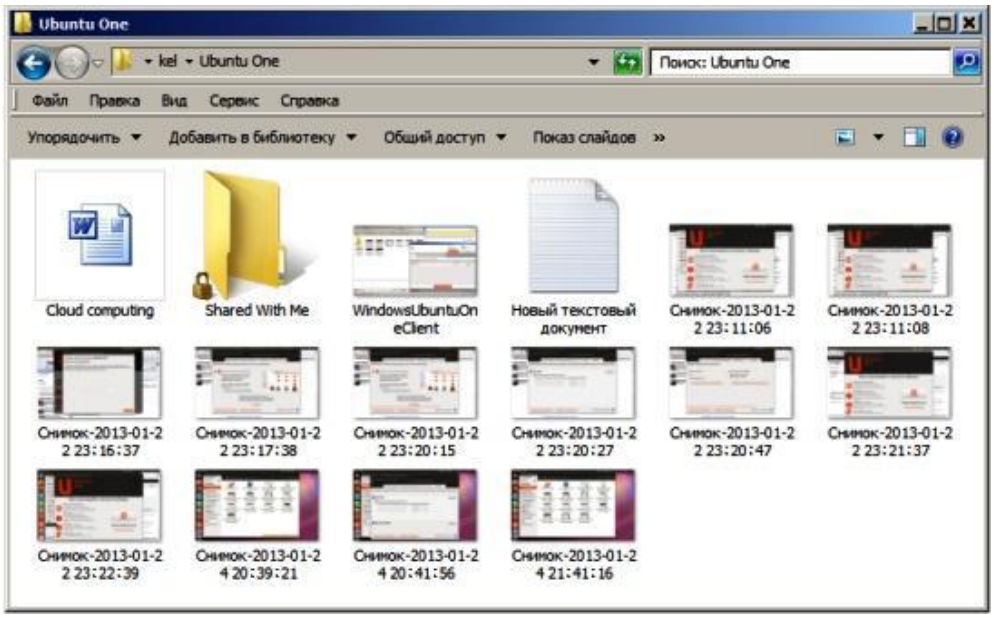

Рис.13.3. Содержимое каталога Ubuntu One на компьютере под управлением ОС Windows XP.

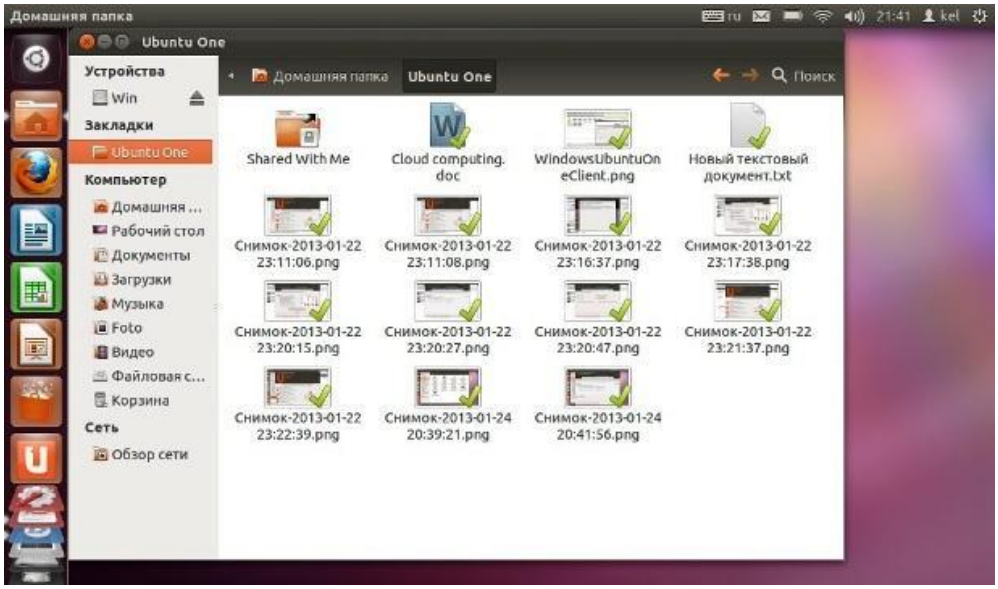

Рис.13.4. Содержимое каталога Ubuntu One на компьютере под управлением ОС Ubuntu.

При необходимости, пользователь может добавить собственные папки, содержимое которых необходимо синхронизировать. Например,

пользователю дополнительно к каталогу Ubuntu One необходимо добавить в online-хранилище содержимое папки Downloads, находящейся на компьютере под управлением ОС Windows. Для этого в панели управления Ubuntu One во вкладке Folder необходимо нажать кнопку «Add folder from this computer» и в открывшемся диалоге выбрать требуемую папку, как показано на рис.13.5.

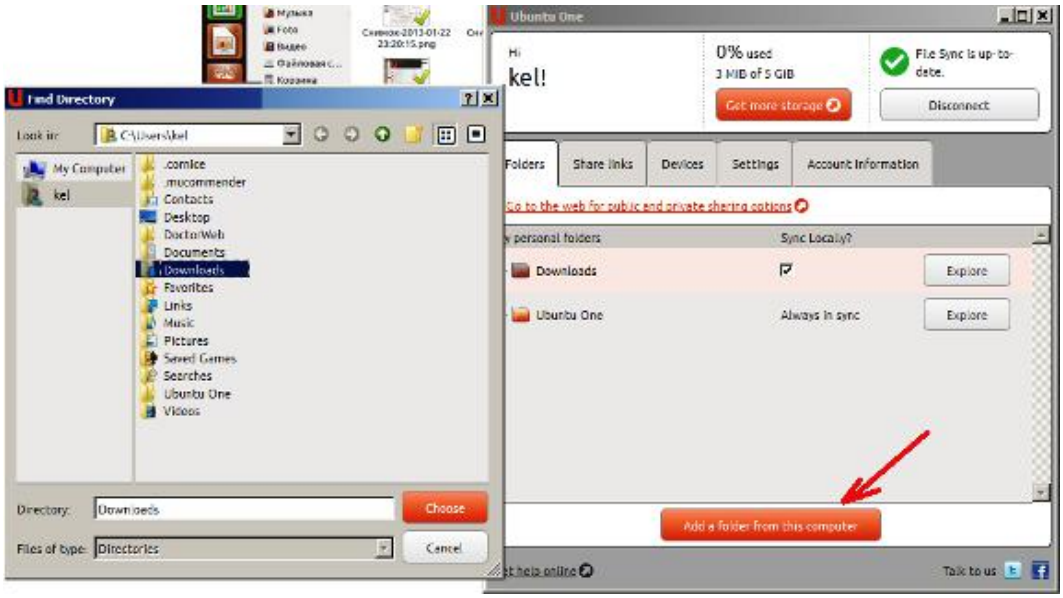

Рис.13.5. Добавление каталога в Ubuntu One.

После этого каталог будет автоматически добавлен в online-хранилище, и его содержимое будет автоматически синхронизироваться с online-хранилищем (рис. 13.6).

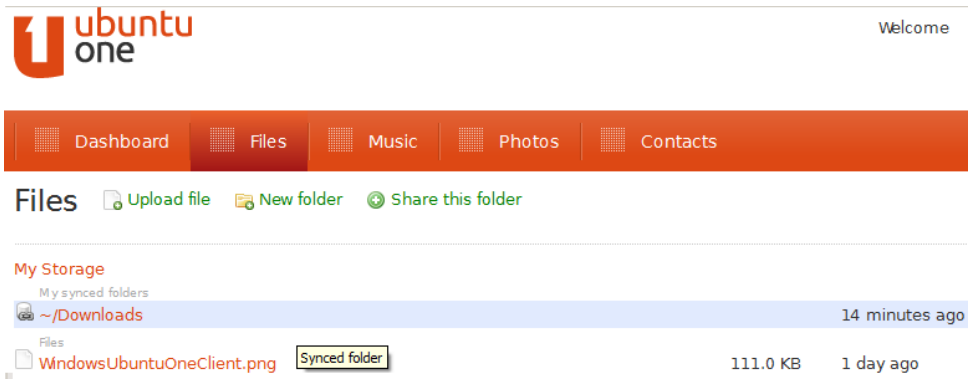

Рис.13.6. Доступ ко вновь добавленному каталогу через Web-интерейс.

Кроме того, на всех компьютерах, на которых установлены клиенты Ubuntu One, подключенные к данному хранилищу, также появится папка Downloads. Эта папку можно увидеть и из среды ОС Ubuntu (рис. 13.7).

Таким образом, облачные технологии в Ubuntu предоставляют пользователям удобные возможности по хранению и синхронизации различных пользовательских данных, работать с которыми можно так же, как если бы эти данные хранились на локальном компьютере.

«Мы надеемся дать всем пользователям, вне зависимости от аппаратной платформы, доступ к одному из лучших персональных облачных сервисов», — утверждает команда Ubuntu One. «Мы хотим, чтобы сервисом Ubuntu One насладилось максимальное количество людей. Вполне вероятно, что часть этих пользователей заинтересуется и преимуществами использования Ubuntu».

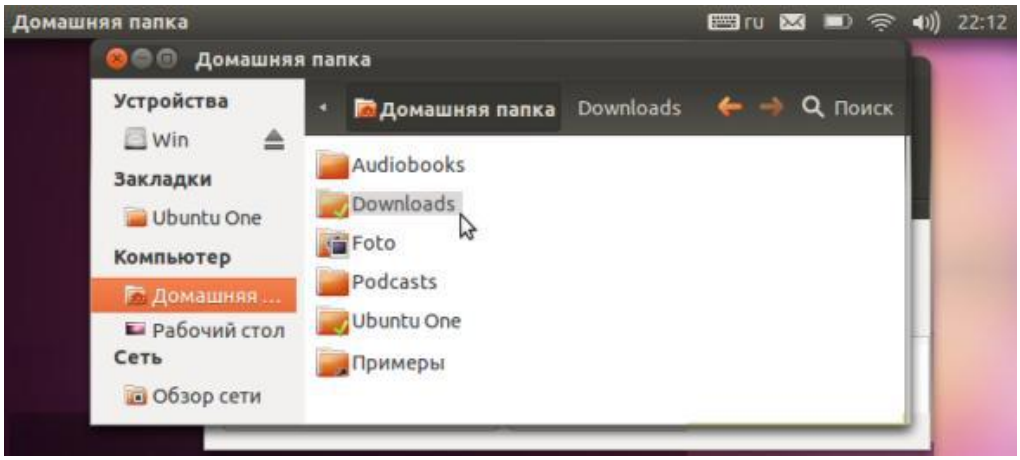

Рис.13.7. Доступ ко вновь добавленному каталогу в ОС Ubuntu.

Однако применение облачных технологий имеет и свои недостатки. Так, для полноценного использования облачных сервисов требуется постоянный высокоскоростной доступ в Интернет. Кроме того, нет ясности с безопасностью данных — насколько хорошо они защищены в хранилище, существует ли вероятность, что данными может воспользоваться владелец вычислительно центра, где размещены облачные серверы?

13.3. Ubuntu One и мобильные устройства

После выпуска клиента Ubuntu One под Android было не очень ясно, а зачем же он нужен. Ведь на телефоне редко хранятся важные файлы, которые постоянно нужно сохранять в отдельном месте, контакты легко синхронизируются с аккаунтом Google. Так зачем Ubuntu One?

Возьмем простой пример — фотографии. Любой пользователь делает фото своим мобильником. Где хранятся фото? А что делать, если вдруг телефон потеряется, его украдут, либо просто выйдет из строя карта памяти на телефоне? Фотографии потеряны. Но если пользоваться клиентом Ubuntu One, то бекап фотографий будет делаться автоматически.

Второй пример. Программы, которыми вы пользуетесь, при их потере очень просто восстановить с помощью маркета либо специальных утилит. А что делать с данными программ, которые сохраняются отдельно? Можно с легкостью настроить синхронизацию данных всех программ,

установленных на смартфоне. Так что, используя клиент Ubuntu One, можно настроить синхронизацию и не бояться потерять свои данные!

Рис.13.8. Клиент Ubuntu One под Android

На сегодняшний день, наряду с клиентом Ubuntu One под Windows разработаны и доступны для загрузки клиенты Ubuntu One для телефонов с Android (рис. 13.8), для устройств от Apple (рис. 13.9), для Nokia N9.

Клиент для Ubuntu One под устройства от Apple с iOS получил название Ubuntu One Files и доступен для всех устройств с этой операционной системой: для iPhone, iPod Touch и iPad. Приложение достаточно функциональное и удобное.

Рис.13.9. Клиент Ubuntu One под iOS.

Ubuntu One Files может автоматически загружать ваши медиафайлы на облачный сервис Ubuntu One и другие подключенные к сервису устройства. Для работы нужно только установить приложение, войти в него (или зарегистрироваться, если вы не сделали этого ранее) и начать пользование.

ПРИЛОЖЕНИЕ

ПРИЛОЖЕНИЕ К РАЗДЕЛУ 2

Приложение 2.1. Использование режима Windows XP и Windows Virtual PC для Windows 7

Если ваш компьютер с Windows 7 не поддерживает старые приложения, то в этом случае вам может помочь использование Windows XP Mode. Этот режим позволяет легко установить и использовать приложения для Windows XP на компьютере с Windows 7.

Использование Windows XP Mode предполагает наличие на вашем компьютере Windows Virtual PC, которая позволяет запускать одновременно более одной операционной системы на одном компьютере!

Обратите внимание, что Windows XP Mode и Windows Virtual PC доступны только для Windows 7 Professional, Enterprise and Ultimate. Компьютер должен поддерживать технологии «Виртуализация».

Если ваш компьютер не поддерживает эту технологию, то не волнуйтесь, Microsoft выпустила обновление KB977206 для Windows 7, которое дает возможность запускать Windows XP и Virtual PC. Для реализации этих целей Вы должны загрузить следующие компоненты:

- Windows XP Mode,
- Windows Virtual PC,
- и обновление KB977206.

Это обновление устраняет необходимые условия для работы в режиме Windows Virtual PC и Windows XP. К необходимым условиям относится наличие процессора с поддержкой аппаратной виртуализации, которая также включается в BIOS. После установки этого обновления может потребоваться перезагрузить компьютер.

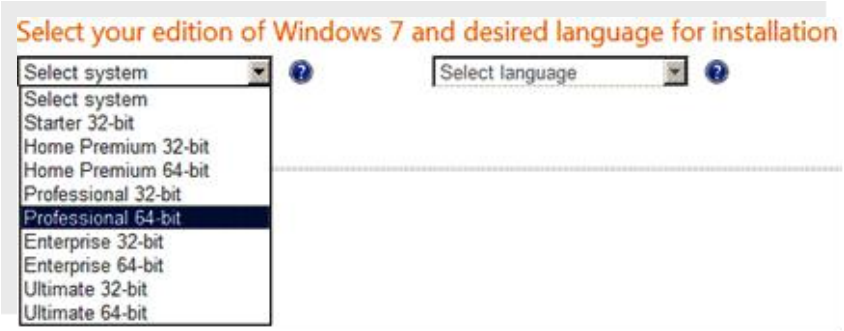

Рис. 2.1.1. Выбор версии операционной системы.

Все эти компоненты доступны для бесплатной загрузки с официального сайта Microsoft («http://www.microsoft.com/windows/virtual-pc/download.aspx»). При его использовании, вы прежде всего должны указать версию вашей операционной системы (рис. 2.1.1).

Затем необходимо указать язык инсталляции пакета. Вид экрана буден аналогичен скриншоту на рис. 2.1.2.

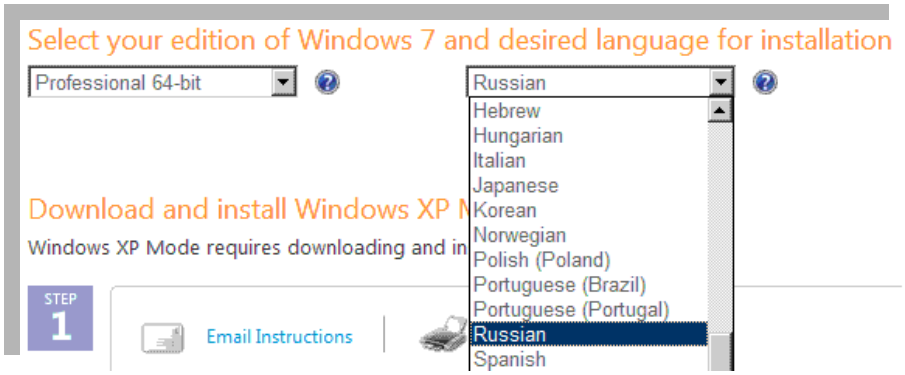

Рис. 2.1.2. Выбор языка загрузки пакетов.

После этого вам будут представлены пошаговые инструкции со ссылками для загрузки (рис. 2.1.3).

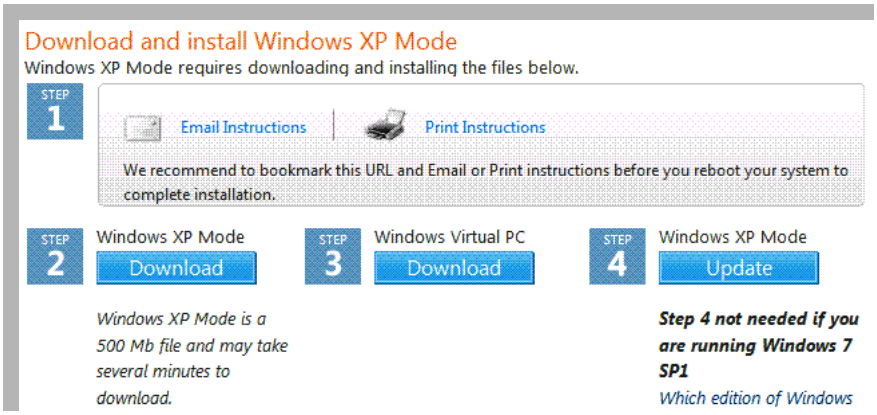

Рис. 2.1.3. Пошаговая инструкция инсталляции пакетов.

Сначала надо загрузить и установить Windows XP Mode. Это все достаточно просто. Затем требуется скачать и установить Windows Virtual PC. И, наконец, если ваш компьютер не поддерживает технологию «Виртуализация», то надо загрузить и установить обновление KB977206 для Windows XP Mode.

Если вам удобнее использовать автономные инсталляторы указанных выше программных продуктов, то ниже для вашего удобства приведены прямые ссылки для загрузки этих программных продуктов:

Windows XP Mode (прямые ссылки для загрузки):

```
English –
http://download.microsoft.com/download/7/2/C/72C7BAB7-
2F32-4530-878A-292C20E1845A/WindowsXPMode_en-us.exe
```

```
Russian  -
http://download.microsoft.com/download/7/5/B/75B73FA4-
3ACB-4131-8A32-B3E51CC3FEF5/WindowsXPMode_ru-ru.exe
```

Windows Virtual PC (прямые ссылки для загрузки):

```
Для 32-битных {x 86} -
http://download.microsoft.com/download/0/5/5/0554AE99-
785F-45CB-B1F2-0E3ED1E6117D/Windows6.1-KB958559-x86.msu"
Для 64-битных {x 64}
http://download.microsoft.com/download/0/5/5/0554AE99-
785F-45CB-B1F2-0E3ED1E6117D/Windows6.1-KB958559-x64.msu"
```

KB977206 Windows Update (прямые ссылки для загрузки):

```
Для 32-битных {x 86} -
href="http://download.microsoft.com/download/E/7/4/E742FBD
2-AE2E-4920-AED1-ABE3F8173585/Windows6.1-KB977206-x86.msu
Для 64-битных {x 64} -
http://download.microsoft.com/download/0/A/3/0A326AC6-
2F94-423F-B760-C61CB8439182/Windows6.1-KB977206-x64.msu
```

Более подробную инструкцию можно получить на сайте технической поддержки Microsoft в документе «Руководство по установке и настройке».

Приложение 2.2. Знакомство с ОС Android и ее виртуализация на Windows Virtual PC

1. Общие сведения об Android

Говоря о гетерогенных сетях, взаимодействиях и доступе между узлами этих сетей, нельзя обойти стороной такую популярную сейчас группу устройств, как мобильные устройства. Довольно большая часть этих устройств работает под управлением под управлением ОС Android.

Android — это портативная сетевая операционная система для коммуникаторов, планшетных компьютеров, электронных книжек, цифровых проигрывателей, наручных часов, нетбуков и смартфонов. Она построена на открытом ядре Linux. Кроме того, она использует пользовательскую виртуальную машину, которая была предназначена для оптимизации памяти и аппаратных ресурсов в мобильной среде.

Речь идет о так называемой Dalvik — виртуальной машине (DalvikVM), которая основана на регистрах. Она разработана и написана Dan Bornstein и некоторыми другими инженерами Google, чтобы быть важной частью платформы Android. В словах «на основе регистров» мы находим первое

отличие от нормальных виртуальных машин Java (JVM), которые основаны на стеке.

Android имеет открытый исходный код, и в него могут быть включены новые современные технологии по мере их появления. Эта платформа будет продолжать развиваться, пока разработчики производят новые приложения.

ОС Android явилась новой вехой в истории развития мобильных платформ, становясь одной из самых популярных ОС для этого класса устройств. В этих условиях вполне естественно желание получить общее представление об этой ОС и познакомиться с ее возможностями.

При отсутствии под рукой какой-либо мобильной платформы у вас есть возможность попробовать Google Android OS на своем ноутбуке или на обычном десктопном Windows-компьютере.

Если вас действительно интересует Android, то я не могу не привести цитату из статьи «Технологический бублик с мармеладом», журнал Upgrate № 2 (609) за 2013 год: «Для начинающих «андрофилов» – учиться, учиться и еще раз учиться (управлять «зеленым роботом», а не расстреливать зеленых свиней!). Осваивать меню, попытать себя в программировании, выучить назубок настройки. И что особенно ценно, придется навести порядок в именах хранилищ (или запомнить их) неважно, встроены они внутрь или подключены извне».

2. Виртуализация Android на Windows Virtual PC

В этом разделе в очень сжатой форме постараемся познакомить читателей с тем, как можно запустить Android на обычном компьютере с помощью Windows Virtual PC. Прежде всего, для этого надо:

- Загрузить и установить на своем компьютере Windows Virtual PC (http://www.microsoft.com/windows/virtual-pc/download.aspx).
- Загрузить образ одной из версий Google Android OS (http://www.android-x86.org/download).

Считая, что с Windows Virtual PC все ясно, переходим на сайт, например http://www.android-x86.org (рис. 2.2.1), находим интересующую вас версию Android и загружаем ее образ к себе на компьютер.

Для примера, которой рассмотрен ниже, выбрана версия Android 2.2 froyo-eeepc. Выбор этот версии связан еще и с тем, что на официальном сайте Android любой желающий может скачать исходный код этой версии ОС для мобильных устройств, а также инструментарий разработчика.

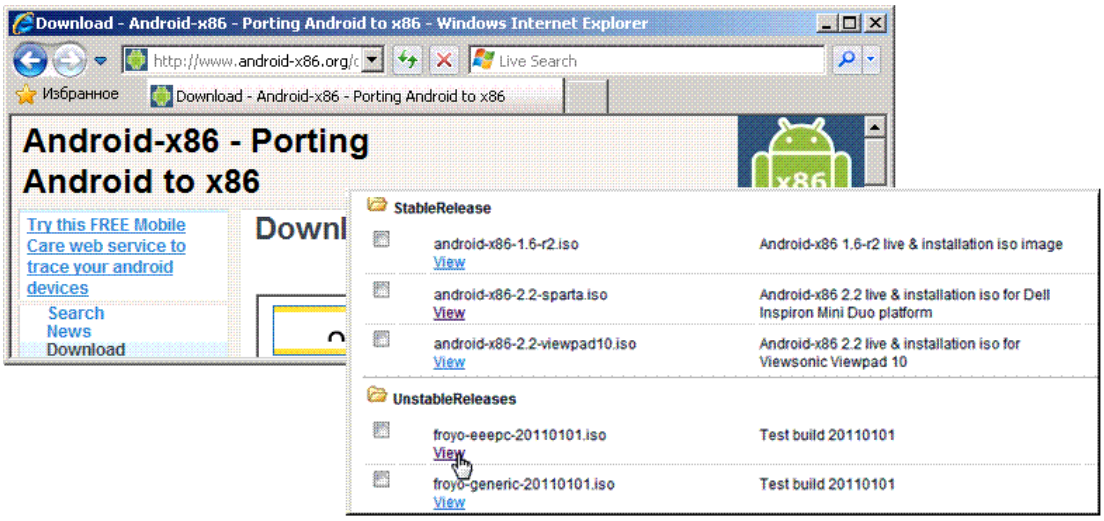

Рис. 2.2.1. Загрузка образа OS Android.

Чтобы начать процесс загрузки образа ОС, следует, выбрав нужную версию, нажать на ссылку View, расположенную ниже названия версии (рис. 2.2.1), и сохранить файл *.iso на вашем компьютере. После того как загрузка завершится, можно переходить к созданию и настройке виртуальной Android-машины.

3. Создание новой виртуальной машины в Windows Virtual PC

Запустим Windows Virtual PC. Для этого выполним Пуск -> Все программы, открываем папку Windows Virtual PC и в ней выбираем Windows Virtual PC. После того как программа Windows Virtual PC будет запущена, следует перейти к созданию новой виртуальной машины. Для выбора этого режима достаточно в меню Windows Virtual PC нажать кнопку Create virtual машина (рис. 2.2.2).

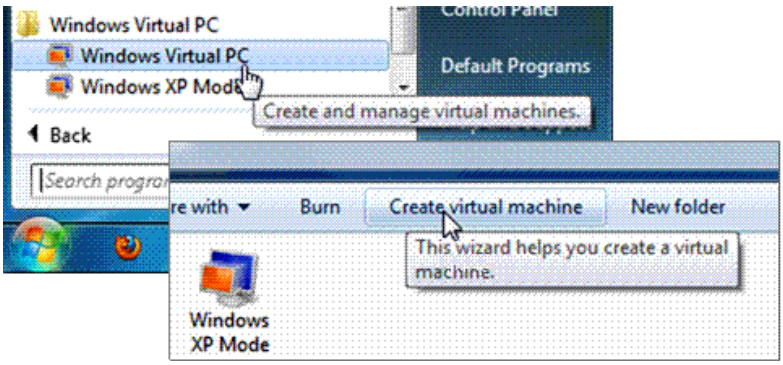

Рис. 2.2.2. Создание новой машины Windows Virtual PC.

Появится окно, в котором надо указать имя и месторасположение файлов новой виртуальной машины. Задаем ее имя, например Android, или что-либо аналогичное этому. Чтобы изменить указанное по умолчанию, ее местоположение на диске вашего компьютера, следует кликнуть по кнопке

Browse и выбрать нужную папку для хранения файлов Android OS, точнее виртуальной машины Android. По окончании следует нажать кнопку Next.

В следующем окне указывается размер ОЗУ, который будет использовать вновь создаваемая виртуальная машина. Рекомендуется использовать минимум 300 Мб. Указываем, требуется или нет для вновь создаваемой виртуальной машины использовать сетевое подключение, а затем для продолжения настройки нажимаем кнопку Next.

Появится новый экран в котором надо ввести, или оставить по умолчанию, запрашиваемые программой настройки параметры (рис. 2.2.3). Далее создаем виртуальный жесткий диск для виртуальной Android-машины. Для этого надо нажать кнопку Create.

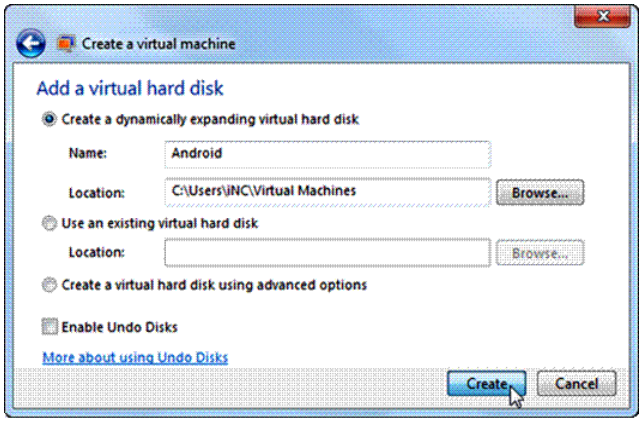

Рис. 2.2.3. Окно добавления виртуального диска.

4. Настройка вновь созданной виртуальной машины

Виртуальная машина создана, и можно переходить к её настройке. Вновь запускаем Windows Virtual PC. В её основном окне отобразится пиктограмма с именем Android. Это ссылка на только что созданную виртуальную машину. Для её настройки, правой кнопкой мыши вызываем выпадающее меню, в котором выбираем опцию Setting (рис. 2.2.4).

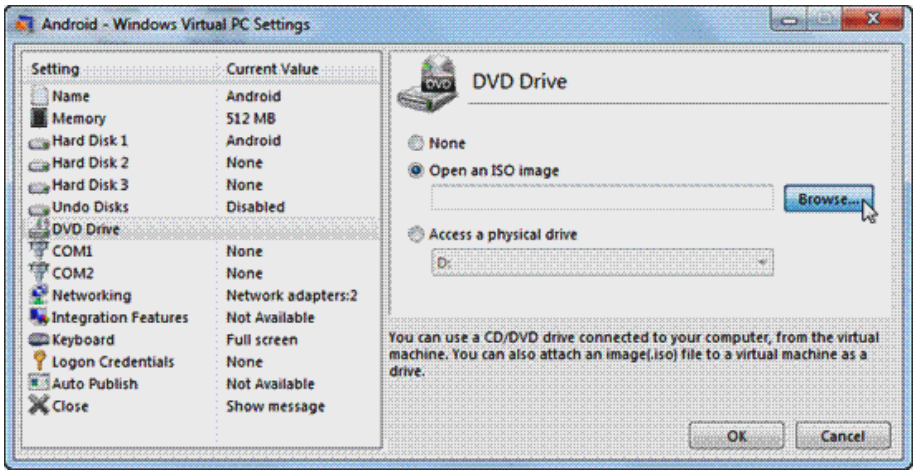

Рис. 2.2.4. Окно настройки виртуальной машины в среде Windows Virtual PC

На этом этапе следует в левой вкладке выбрать опцию DVD Drive, а в правой установить режим Open an ISO image, затем кликнуть по кнопке Browse и подключить iso файл Android, который был только что загружен и сохранен на вашем компьютере. Для этого нажмите Open или просто дважды щелкните на имени файла, после чего подтвердите ваш выбор.

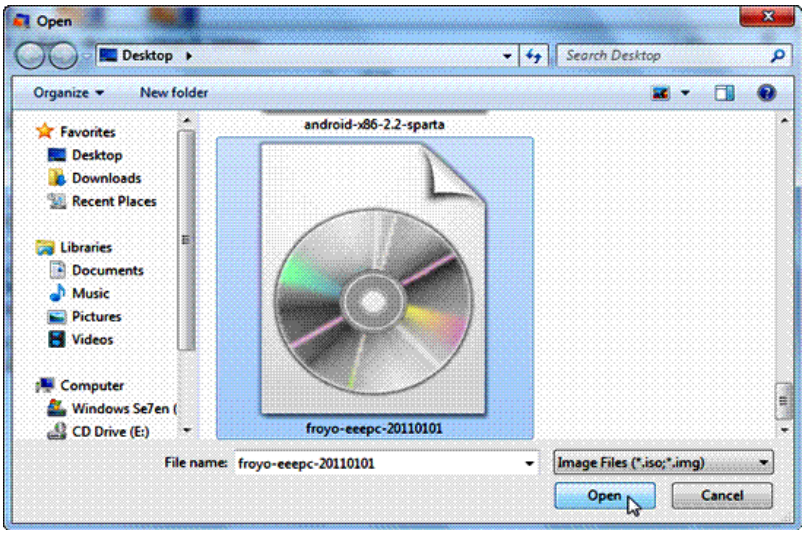

Рис. 2.2.5. Выбор файла с образом операционной системы Android.

Наступил момент, когда у нас появился новый, пусть и виртуальный компьютер, со своим процессором, памятью, дисками и DVD-приводом, в который вставлен установочный диск операционной системы Android. Можно переходить к установке ОС на этот компьютер.

5. Установка Android на виртуальной машине

Прежде всего надо включить компьютер, то есть запустить в работу созданную нами виртуальную машину. Для этого достаточно найти пиктограмму виртуальной машины с именем Android, правой кнопкой мыши вызвать выпадающее меню, в котором выбрать режим Open (рис. 2.2.6). Возможен и более короткий путь — просто дважды щелкнуть мышкой по пиктограмме этой машины.

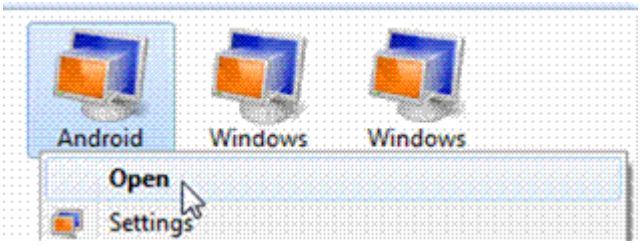

Рис. 2.2.6. Запуск виртуальной машины с именем Android.

Так как в приводе виртуальной машины установлен загрузочный диск, а точнее его iso-образ, то одновременно с запуском в работу виртуальной машины начинается и загрузка на нее с этого iso-диска операционной

системы Android версии Android 2.2 froyo-eeehc. Именно той, образ которой был загружен на наш компьютер на первом этапе.

Через несколько мгновений окно нашей виртуальной машины приобретает вид стартовой страницы загрузчика операционной системы Android (рис. 2.2.7).

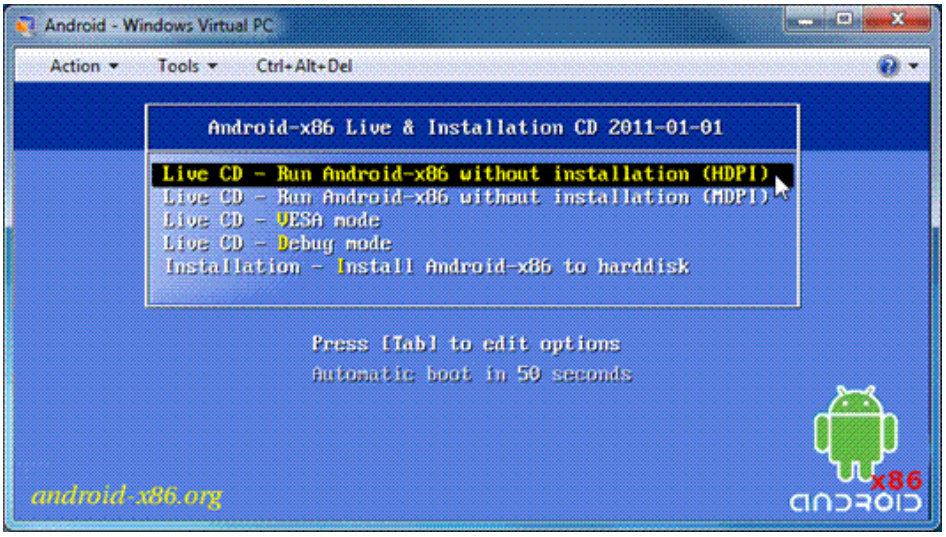

Рис. 2.2.6. Окно виртуальной машины со стартовой страницей загрузчика ОС Android.

На стартовой странице выберите «Run Android-x86 without installation». Обратите внимание, что существуют два варианта для запуска Android без установки.

- HDPI предназначен для отображения в режиме высокой четкости;
- MDPI — для отображения в режима средней четкости.

Здесь H (High) — высокая четкость, M (Medium) — средняя четкость, а DPI (Dots Per Inch - точек на дюйм) — мера плотности пикселей экрана.

Кроме того, предлагается вариант с видеодрайвером VESA — «Boot with vesa video mode». Этот режим пригодится в том случае, если система не распознает видеоплату.

Рис. 2.2.7. Логотип Android при загрузке операционной системы.

Можно выбрать любой вариант запуска Android без установки. При загрузке появиться яркий логотип Android (рис.2.2.7). Выглядит стильно, не правда ли?

После небольшого промежутка времени, связанного с загрузкой ОС, на экране появиться стартовая страница Android, аналогичная той, что представлена на рис. 2.2.8.

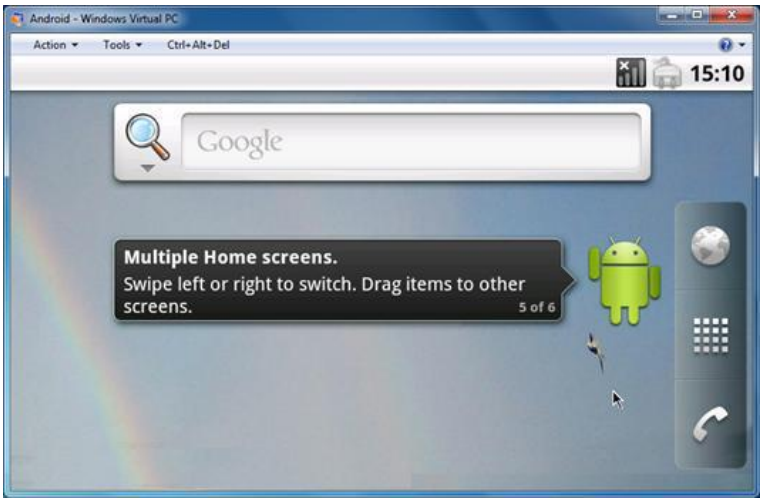

Рис. 2.2.8.Стартовая страница Android на виртуальной машине.

Активизируем вход в систему, и перед нами главное меню Android, вид которого будет аналогичен приведенному на рис. 2.2.9.

Рис. 2.2.9.Главное меню Android.

Итак, Android запущен на виртуальной машине без каких-либо проблем! Все просто и легко. Однако если у вас появилось желание запустить что-либо, то следует принять во внимание тот факт, что некоторые приложения могут работать некорректно, так как мы

используем компьютер, а не мобильное устройство (телефон, смартфон или планшетник).

Сетевые подключения ОС Android

Имея виртуальную машину с Android, у вас появляется возможность познакомиться с основными приложениями для работы в этой среде. Однако суть данной книги — сетевое взаимодействие узлов гетерогенных сетей. В этом плане виртуальная Android-машина рассматривается, лишь как один из Linux-узлов этой сети. Но говоря о сетевом информационном обмене, мы должны быть прежде всего уверенными, что есть физическое подключение узла к сети. Если это не так, то надо уметь настроить это подключение. Об этом подробно было рассмотрено выше.

В этом небольшом разделе обратим внимание лишь на то, где, но не как, в Android выполняются эти настройки. В Android всегда доступен вызов всплывающего системного меню, аналогичного рис. 2.2.10.

Рис. 2.2.10. Всплывающее меню рабочего стола Android.

Щелкнув мышкой по кнопке Setting, попадаем на экран настроек системы (рис. 2.2.11).

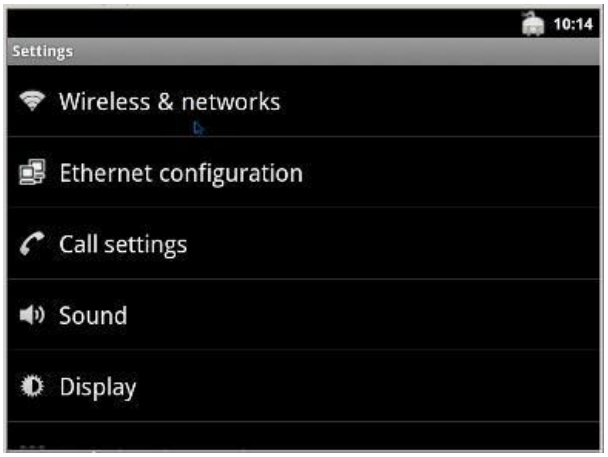

Рис. 2.2.11. Всплывающее меню рабочего стола Android.

Из всех возможных настроек, которые возможны в Android, для нас интересны только две первые строки: это настройка подключения к беспроводным сетям и конфигурирования Ethernet подключения. При выборе первой строки вы попадаете на экран (рис. 2.2.12), где возможны настройки беспроводных соединений на базе различных протоколов, просмотр доступных вам беспроводных сетей и т. д.

Рис. 2.2.12. Настройка беспроводных подключений.

В том случае, если будет выбран режим Ethernet configuration, то будет доступен экран, аналогичный тому, что приведен на рис. 2.2.13.

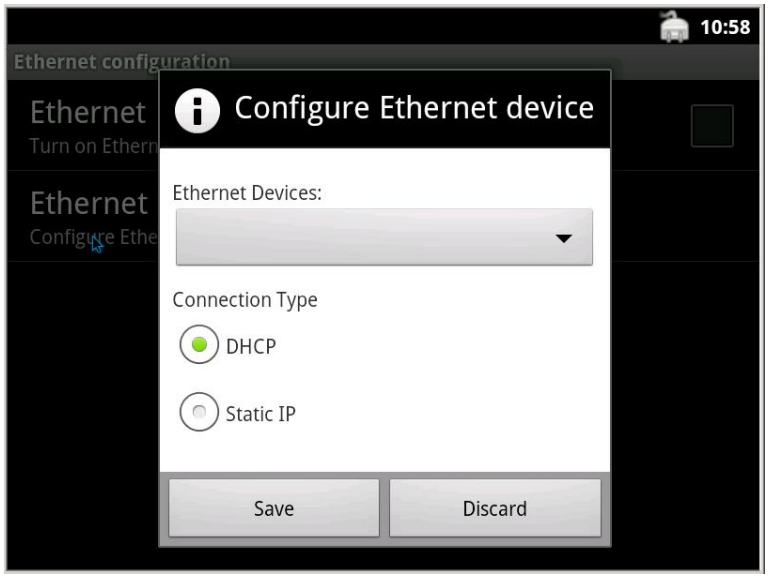

Рис. 2.2.13. Экран конфигурирования Ethernet подключения.

На этом этапе доступна полноценная настройка интерфейса Ethernet, с указанием всех необходимых DHCP и DNS сервером, шлюзов, IP-адреса и маски сети вашего устройства. Об этом более подробно смотри в главе 6. Но там все рассмотрено на примере ОС Ubuntu. Вы можете возразить: «А

зачем мне ваша Ubuntu, если я читаю про Android?». Все дело в том, что в основе обеих операционных систем лежит ядро Linux. Поэтому, несмотря на существенные различия в графическом интерфейсе, большинство действий в них выполняется практически одинаково.

Рис. 2.2.14. Текстовая Linux-консоль Android.

Можно на вашем красивом нетбуке или планшетнике нажать пару клавиш, и исчезнет вся его красота, а черный экран будет аналогичен первой строке скриншота, приведенного на рис. 2.2.14. Это текстовая консоль, где можно вводить любые команды Linux и заставлять ваше мобильное устройство выполнять нужные вам действия.

Но это на любителя или профессионала! Более подробно о командах Linux см. в главах 3 и 4. В качестве примера на рис. 2.2.14 приведен результат просмотра всех папок, присутствующих в вашей виртуальной машине. Для этого использована команда ls. Там же приведен пример использования еще одной команды – echo и результат ее выполнения. Об Android см. также в самом конце, в главе 12.

ПРИЛОЖЕНИЕ К РАЗДЕЛУ 3

Приложение 3.1. Руководство по Терминалу среды GNOME

1. Введение

Терминал среды GNOME – это программа эмуляции терминала, которую можно использовать для выполнения следующих задач:

• Доступ к интерпретатору командной строки UNIX из среды GNOME. Интерпретатор командной строки исполняет команды, вводимые в командной строке. При запуске Терминала запускается интерпретатор, указанный по умолчанию в вашей учетной записи. Вы можете перейти к использованию другого интерпретатора в любое время.

• Запуск любых приложений, спроектированных для запуска на терминалах VT102, VT220 и xterm. Терминал среды GNOME эмулирует программу xterm. В свою очередь программа xterm эмулирует терминал DEC VT102 и также поддерживает управляющие последовательности DEC VT220. Управляющие последовательности (ESC-последовательности) — это серия символов, обычно используемая для задания неотображаемых символов и символов, имеющих специальное значение).

2. Приступая к работе

2.1. Запуск Терминала среды GNOME

Запустить Терминал среды GNOME можно следующими способами:

• Из основного меню: Приложения -> Стандартные -> Терминал.

• Из командной строки: gnome-terminal. Можно использовать параметры командной строки, чтобы изменить способ запуска Терминала. Чтобы просмотреть параметры командной строки, надо выполнить команду: gnome-terminal --help

2.2. Ваш первый запуск Терминала среды GNOME

Когда вы запускаете Терминал среды GNOME в первый раз, приложение открывает терминальное окно с группой установок по умолчанию.

Группа установок по умолчанию называется Профилем по умолчанию. Название профиля появляется в строке заголовка окна Терминала среды GNOME.

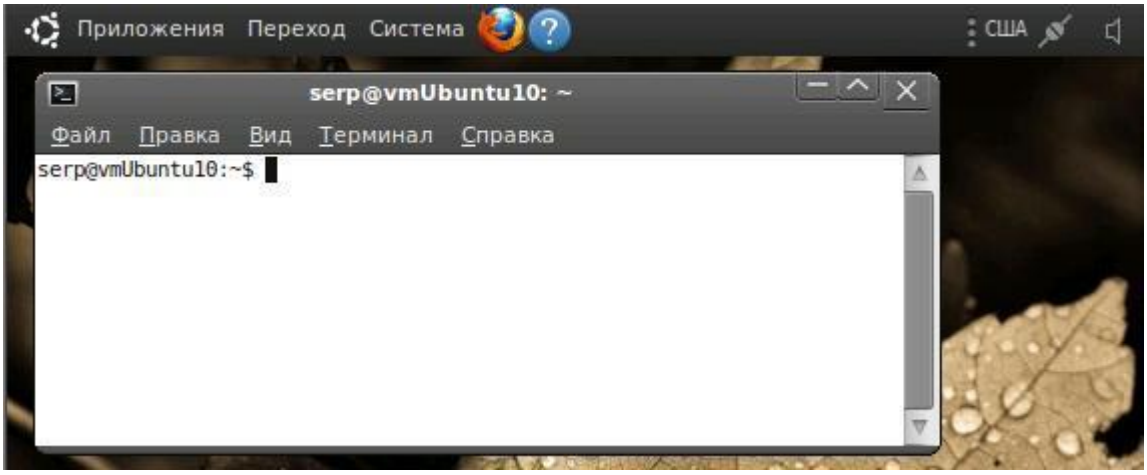

Рис. 3.1.1. Пример окна по умолчанию Терминала среды GNOME Ubuntu 10.04.

Окно терминала отображает приглашение на ввод команды, в котором можно набирать команды UNIX. Приглашение на ввод команды может быть в виде %, #, >, $ или любых других специальных символов. Около приглашения на ввод расположен курсор. Когда вы набираете команду UNIX и нажимаете Enter, компьютер выполняет команду.

По умолчанию Терминал среды GNOME использует интерпретатор командной строки пользователя, запустившего приложение. Терминал среды GNOME также устанавливает следующие переменные окружения:

- TERM — устанавливается по умолчанию к xterm.
- COLORTERM — устанавливается по умолчанию к gnome-terminal.
- WINDOWID — устанавливается к оконному идентификатору X11 по умолчанию.

2.3. Профили Терминала

Назначение профилей в том, чтобы в Терминале изменить такие характеристики, как шрифт, цвета и эффекты, вид полос прокрутки, заголовка окна, совместимости. Также можно определить команду, которая запускается автоматически при запуске Терминала.

Определяют профили в окне «Изменить профиль», которое доступно из меню «Правка». Можно определить столько профилей, сколько нужно. При запуске терминала можно выбрать профиль, который вы хотите использовать. Доступно изменить профиль «Терминала» во время его использования. Чтобы определить первоначальный профиль «Терминала» при запуске из командной строки, используют команду:

```
gnome-terminal --window-with-profile=название профиля
```

Название текущего профиля появится в строке заголовка Терминала среды GNOME, если вы не определили другое название заголовка в окне «Изменить профиль». В разделе 3.2 данного Приложения приведена информация о том, как определить и использовать новый профиль терминала.

2.4. Работа с несколькими терминалами

Особенностью Терминала среды GNOME являются вкладки, которые дают возможность вам открывать несколько терминалов в одном окне. Каждый терминал открывается в отдельной вкладке. Клик мышью на соответствующей вкладке отображает окно терминала. Каждая вкладка терминала в общем окне — это отдельный процесс, так что вы можете использовать каждый терминал для различных задач.

Вы можете применять различные профили к каждой вкладке терминала в общем окне. Заголовок окна терминала отображает или имя текущего профиля, или название, присвоенное текущему профилю. На рис. 3.1.2 показано окно Терминала среды GNOME с несколькими вкладками, в каждой из которых может быть свой профиль. Название профиля активной вкладки появляется в строке заголовка.

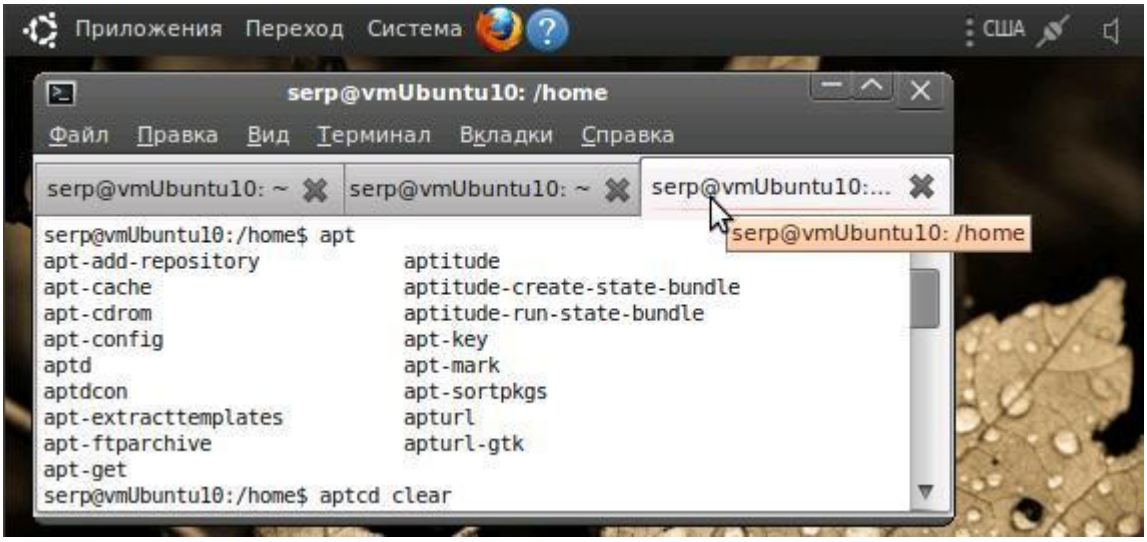

Рис. 3 1.2. Пример окна терминала с вкладками.

3. Применение

3.1. Открытие и закрытие терминалов

Чтобы открыть новое окно терминала, достаточно выбрать Файл -> Открыть терминал. При этом новый терминал наследует установки программы и интерпретатор командной строки по умолчанию из родительского терминала.

Чтобы закрыть окно терминала, следует выбрать Файл -> Закрыть окно. Это действие закроет терминал и все подпроцессы, которые были запущены из терминала. Если вы закроете последнее окно терминала, Терминал среды GNOME завершает работу.

Для добавления нового терминал-вкладки используйте Файл -> Открыть вкладку. Для отображения нужного терминала следует нажать мышкой на заголовке соответствующей вкладки или выбрать название вкладки из меню Вкладки. Другие способы: это Вкладки -> Следующая вкладка или Вкладки -> Предыдущая вкладка.

Чтобы закрыть терминал-вкладку, отобразите вкладку терминала, которую хотите закрыть и выберите Файл -> Закрыть вкладку.

3.2. Управление профилями

Чтобы добавить новый профиль выберите Файл -> Новый профиль и откроется диалоговое окно «Новый профиль», в котором следует:

- набрать имя нового профиля в текстовом поле «Имя профиля».
- используя выпадающий список, указать профиль, на котором вы хотите основать новый профиль.

После этого нажмите кнопку «Создать», а затем закройте появившееся окно «Изменение профиля». Терминал среды GNOME добавит новый профиль в подменю Терминал -> Использовать профиль.

Чтобы изменить профиль терминала-вкладки щелкните на вкладке терминала, профиль которого надо изменить, а затем выберите Терминал -> Изменить профиль -> [имя профиля].

Чтобы редактировать профиль следует активировать окно «Изменение профиля», доступ к которому можно получить несколькими способами:

- Путем выбора Правка -> Текущий профиль.
- Щелкнув правой кнопкой в окне терминала и, выбрав из контекстного меню «Изменить текущий профиль».
- И, наконец, выбрать Правка -> Профили, указать профиль, который надо изменить и затем щелкнуть Правка.

Информацию о параметрах, которые можно устанавливать в профиле, приведены в разделе 4 данного Приложения.

Чтобы удалить профиль выберите Правка -> Профили, в открывшемся списке укажите имя профиля и щелкните по кнопке «Удалить». Появится окно «Удаление профиля», где надо подтвердить удаление.

3.3. Изменение окна терминала

Чтобы скрыть строку меню надо выбрать Вид -> Показать строку меню. Для восстановления скрытой строки меню следует щёлкнуть правой кнопкой на заголовке окна терминала и из контекстного меню выбрать «Показать меню».

Чтобы отобразить окно Терминала в полноэкранном режиме, используйте Вид -> Развернуть на полный экран. Окно не содержит рамки окна или заголовка. Чтобы выйти из этого режима, выберите Вид -> Развернуть на полный экран снова. Изменить внешний вид окна терминала можно, выбирая профиль с теми или иными параметрами.

3.4. Работа с содержимым окон терминала

Чтобы просмотреть предыдущие команды и информацию о результатах их запуска выполните одно из следующих действий:

- Используйте прокрутку, которая обычно отображается в правой части терминального окна.
- Нажмите кнопки Shift+PageUp, Shift+PageDown, Shift+Home или Shift+End для перехода на страницу вперед или назад, в начало или конец окна прокрутки. Количество строк, которые можно прокручивать в окне терминала определяется установкой параметра Scrollback setting в секции Scrolling tabbed окна редактора профиля.

Чтобы выделить и копировать текст можно поступить одним из следующих способов:

- Чтобы выделить посимвольно, щёлкните на первом символе и тяните мышь до последнего символа, который надо выделить.
- Чтобы выделить по словам, дважды щелкните на первом слове и тяните мышь до последнего слова, которое следует выделить.
- Чтобы выделить по строкам, трижды щелкните на первой строке, которую надо выделить, и тяните мышь до последней строки, которую вы хотите выделить.

Эти действия выделяют весь текст от первого до последнего знака. Терминал копирует выделенный текст в буфер обмена когда вы отпускаете кнопку мыши. Для копирования явным образом выделенного текста выберите Правка -> Копировать.

Чтобы вставить текст в терминал, который предварительно был скопирован в буфер обмена можно выполнить одно из следующих действий:

- Для вставки текста, который копировали только выделением, надо щелкнуть средней кнопкой мыши в приглашении на ввод.
- Чтобы вставить копированный явным образом текст, выберите Правка -> Вставить.

Чтобы получить доступ к ссылке (URL), которая будет отображаться в терминале, выполните следующие шаги:

- Двигайте мышь над URL до тех пор, пока она не будет подчеркнута.
- Затем щелкните правой кнопкой на URL, чтобы открыть контекстное меню. Выберите опцию «Открыть ссылку …», на которое указывает ссылка, и отобразить файл, размещенный по URL.

3.5. Просмотр установленных комбинаций клавиш

Чтобы просмотреть установки комбинаций клавиш, которые определены для Терминала, выберите Правка -> Комбинации клавиш. В диалоговом окне Комбинации клавиш отображаются такие опции, как «Отключить все клавиши для доступа в меню (такие, как Alt+Ф для меню «Файл»)», «Отключить клавишу для доступа в меню (F10 по умолчанию)» и «Комбинации клавиш». Вам доступна перенастройка ряда параметров.

3.6. Размер текста

Вы можете изменить размер текста в окне Терминала. Чтобы увеличить размер текста, выберите Вид -> Увеличить. Чтобы уменьшить размер текста, выберите Вид -> Уменьшить. Чтобы отобразить текст в обычном размере, выберите Вид -> Обычный размер.

3.7. Смена заголовка терминала

Чтобы изменить заголовок отображаемого в данный момент терминала, надо выбрать Терминал -> Установить заголовок, ввести новый заголовок в текстовом поле Заголовок. Изменения в Терминале среды GNOME вступят в силу немедленно.

3.8. Смена кодировки символов

Чтобы изменить кодировку символов, следует выбрать Терминал -> Установить кодировку символов, а затем подходящую кодировку.

Чтобы изменить список кодировок, надо выбрать Терминал -> Установить кодировку символов -> Добавить или удалить. Чтобы добавить кодировку, выберите нужную из списка возможных кодировок и нажмите клавишу стрелка вправо. Чтобы удалить, нажмите клавишу стрелка влево.

3.9. Восстановление вашего терминала

При возникновении проблем с терминалами можно сбросить терминал в первоначальное состояние, выберите Терминал -> Сброс. Чтобы сбросить терминал и очистить экран, используйте Терминал -> Сброс и очистка.

4. Параметры

Чтобы настроить Терминал, выберите Правка -> Текущий профиль. Чтобы настроить другой профиль, выберите Правка -> Профили, затем профиль, который хотите редактировать и щёлкните Правка.

4.1. Общие

Имя профиля. — Это текстовое поле используется для установки имени текущего профиля.

Шрифт. — Эта кнопка вызывает окно, в котором можно осуществить выбор типа и размера шрифта, отображаемого в Терминале. Кнопка доступна в том случае, если не выбран параметр. Использовать системный терминальный шрифт.

Разрешать полужирный текст. — Выбор этого параметра используется для возможности использования в отображении Терминалом полужирного текста.

Показывать в новых терминалах строку меню. — Выберите этот параметр, чтобы в новых окнах терминала отображалась строка меню.

Подавать гудок. — Выберите этот параметр, чтобы был доступен звуковой сигнал терминала.

Выбирающие слово символы. — Используйте этот параметр, чтобы определить символы или группы символов, которые рассматриваются Терминалом как слова, когда вы выбираете текст по словам.

4.2. Заголовок и команда

Исходный заголовок. — Используйте это текстовое поле для определения исходного заголовка терминалов, которые используют профиль. Новые терминалы, которые были запущены из текущего терминала, получат новый заголовок.

Команды терминала устанавливают собственные заголовки окон. — Используйте этот выпадающий список, чтобы установить, как поступать с динамически устанавливаемыми заголовками, которые устанавливаются командами, запущенными в терминале.

Запускать команду как команду входа в сеанс. — Выберите этот параметр, чтобы заставить команду выполнять действия, нужные для входа в сеанс. Если команда не является интерпретатором командной строки, то параметр не имеет эффекта.

Запускать другую программу вместо Bash-интерпретатора. — Выберите этот параметр, чтобы запускать в терминале команду, отличную от обычного интерпретатора командной строки. Определите пользовательскую команду в поле Другая команда.

При выходе из команды. — Используйте этот выпадающий список, чтобы определить действия при выходе.

4.3. Текст и фон

Используйте выпадающий список «Встроенные схемы» чтобы определить цвета текста и фона. Терминал поддерживает следующие комбинации цвета текста и фона:

- Черный на светло-желтом
- Черный на белом
- Серый на черном

- Зеленый на черном
- Белый на черном
- Другая — позволяет выбрать цвета, которых нет в цветовой схеме.

Фактическое отображение цветов текста и фона может различаться в зависимости от выбранной цветовой схемы (палитры). Выпадающий список Встроенные схемы доступен, если только не выбран параметр Использовать цвета из системной темы.

- Для выбора цвета текста — нажать кнопку «Цвет текста». Откроется окно, в котором, используя цветной круг или треугольник, выбрать желаемый цвет для текста Терминала. Затем нажать OK. Кнопка Цвет текста доступна, только если не выбран параметр Использовать цвета из системной темы.
- Для установки цвета фона Терминала - нажать кнопку «Цвет фона». Отобразится окно, в котором следует выбрать желаемый цвет и нажать кнопку OK. Кнопка Цвет фона доступен, если только не выбран параметр Использовать цвета из системной темы.

4.4. Палитра

Терминал для прорисовки текста использует сразу только 16 цветов. Эти 16 цветов определяет Цветовая палитра. Приложения, которые запускаются в терминале, используют индексный номер, определяющий цвет из этой палитры.

Следует использовать выпадающий список Встроенные схемы, чтобы выбрать предустановленные цветовые схемы. Нижняя цветовая палитра обновляется совместно с содержанием окна терминала, чтобы отобразить схему.

Используйте Палитру чтобы установить 16 цветов, используемых по умолчанию, для собственной цветовой палитры. Чтобы установить цвет, щелкните на цвете для появления диалога Изменить цвет палитры. Используйте цветной круг или треугольник для установки цвета.

4.5. Эффекты

Данный режим позволяет выбрать фон для окна терминала. Доступны несколько вариантов.

Фоновое изображение. — Выберите этот параметр, чтобы использовать файл рисунка в качестве фона для терминала. Используйте комбинированное окно диалога Фоновое изображение для определения места нахождения и имени файла рисунка. Иначе, щелкните Обзор для поиска и выбора файла рисунка.

Выберите параметр «Прокручивать изображение фона» для того, чтобы фоновое изображение прокручивалось вместе с текстом, когда вы прокручиваете его в терминале. Если вы не выберете этот пункт,

изображение фона останется зафиксированным на фоне терминала и прокручивается только текст. Этот параметр доступен, если только будет выбран вариант Фоновое изображение.

Прозрачный фон. — Выберите этот пункт, чтобы использовать прозрачный фон для терминала.

Затемнять прозрачность или изображение фона. — Используйте этот ползунок, чтобы затемнить или замутнить фон терминала. Этот параметр доступен, если только будет выбран вариант Фоновое изображение или Прозрачный фон.

4.6. Прокрутка

Полоса прокрутки. — Используйте этот выпадающий список для установки положения полосы прокрутки в окне терминала.

Обратная прокрутка ... линий. — Используйте этот прокручиваемый список для установки количества линий, которые вы сможете прокрутить назад, используя прокрутку. Например, если вы установите 100, вы сможете прокрутить назад последние 100 линий, отображенных в терминале.

Прокручивать при выводе. — Выберите этот параметр для возможности прокручивать при выводе терминалом информации, когда терминал продолжает выводить дальнейшую информацию выполняющейся команды.

Прокручивать при нажатии клавиши. — Выберите этот параметр для возможности при нажатии любой клавиши на клавиатуре прокручивать текст в окне терминала вниз, до приглашения на выполнение команды. Это действие применимо, если вы прокрутили вверх текст в окне терминала и хотите вернуться к запросу на выполнение команды.

ПРИЛОЖЕНИЕ К РАЗДЕЛУ 6

Приложение 6.1. Общая информация о сетевых настройках системы

Несмотря на то что во многих современных дистрибутивах есть красивые графические утилиты по настройке сетевых конфигураций, в ряде случаев проще настроить все руками. Это очень просто — вписать несколько строчек в нужные конфигурационные файлы или в терминале набрать несколько команд.

Цель данного приложения — дать самое начальное представление о возможности сетевых настроек в местном или удаленном терминальном доступе на Linux-компьютерах. Рассматриваются простейшие и наиболее часто используемые команды, используемые при сетевом администрировании.

Основу данного приложения составляют материалы статьи «Настройка сетевой карты, краткое пособие для начинающих», представленной на сайте http://www.altlinux.org/.

Узнаем имя компьютера

Чтобы узнать имя компьютера, в консоли которого вы сейчас работаете, следует использовать команду hostname:

```
serp@VMubuntu:~$ hostname
VMubuntu
```

До первой перезагрузки изменить имя компьютера можно командой:

```
hostname <новое_имя>.
```

Чтобы изменить имя компьютера окончательно, откройте файл по адресу

```
/etc/hostname
```

Найдите там строку с именем компьютера, поменяйте имя на новое и сохраните изменения в файле и перезагрузите компьютер.

Какие сетевые карты есть в системе

Чтобы выяснить, есть ли вообще сетевые карты в компьютере, введем команду:

```
lspci -v
```

Будет выдан список многих устройств, где сетевой карте вашей виртуальной машины соответствует, примерно такой раздел:

```
serp@VMubuntu:~$ lspci -v
 . . .
00:0a.0 Ethernet controller: Digital Equipment Corporation
DECchip 21140 [FasterNet] (rev 20)
  Subsystem: Unknown device 0a00:2114
  Flags: bus master, medium devsel, latency 64, IRQ 11
  I/O ports at ec00 [size=128]
  Memory at febff000 (32-bit, non-prefetchable) [size=4K]
  Expansion ROM at febe0000 [disabled] [size=64K
```

Если вы хотите увидеть только сетевую карту, следует к команде применить фильтр grep, то есть

```
lspci | grep Eth
```

Тогда результат тестирования сетевой карты будет:

```
serp@VMubuntu:~$ lspci -v | grep Eth
00:0a.0 Ethernet controller: Digital Equipment Corporation
DECchip 21140 [FasterNet] (rev 20)
```

Получение информации о сетевом адресе компьютера

Для того чтобы узнать сетевой адрес компьютера, можно воспользоваться командой

```
ip addr show
```

или ее более кратким вариантом ip a. В ответ вы должны получить примерно следующее:

```
serp@VMubuntu:~$ ip addr show
1: lo:  mtu 16436 qdisc noqueue
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
  inet6 ::1/128 scope host
     valid_lft forever preferred_lft forever
2: eth0:  mtu 1500 qdisc pfifo_fast qlen 1000
  link/ether 00:03:ff:04:40:14 brd ff:ff:ff:ff:ff:ff
  inet 192.168.1.10/24 brd 192.168.1.255 scope global eth0
  inet6 fe80::203:ffff:fe04:4014/64 scope link
     valid_lft forever preferred_lft forever
3: sit0:  mtu 1480 qdisc noop
  link/sit 0.0.0.0 brd 0.0.0.0
```

В протоколе вывода: eth0 — имя сетевого интерфейса, link/ether 00:03:ff:04:40:14 — его MAC-адрес, а inet 192.168.1.10/24 — IP-адрес сетевого интерфейса с маской в 24 бита.

Следует обратить внимание: если сетевой кабель не будет физически подключен к разъему сетевой карты, то при выводе по этой команде появится слово NO-CARRIER.

Существует еще одна полезная команда ip route show или ее краткий вариант ip r, которая покажет шлюз:

```
serp@VMubuntu:~$ ip route show
192.168.1.0/24 dev eth0  proto kernel  scope link  src
192.168.1.10
default via 192.168.1.1 dev eth0
```

Характеристики работы сетевой карты

Команда ethtool <имя_интерфейса> покажет вам некоторые характеристики, с которыми работает ваша сетевая карта. В нашем варианте с виртуальной машиной она не поддерживается. А в других случаях протокол вывода имел бы примерно такой вид:

```
# ethtool eth0
Settings for eth0:
    Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                            100baseT/Half 100baseT/Full
                            1000baseT/Full
    Supports auto-negotiation: Yes
    Advertised link modes:  10baseT/Half 10baseT/Full
                            100baseT/Half 100baseT/Full
                            1000baseT/Full
    Advertised auto-negotiation: Yes
    Speed: 1000Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: on
    Supports Wake-on: umbg
    Wake-on: d
    Link detected: yes
```

Посредством команды ethtool вы также можете изменять эти характеристики.

Контроль наличия сетевого соединения

Для контроля наличия соединения используется команда ping <адрес_узла>.

Протокол работы этой команды может иметь вид, аналогичный, приведенному ниже:

```
serp@VMubuntu:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=128 time=3.34 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=128 time=8.58 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=128 time=0.34 ms

--- 192.168.1.2 ping statistics ---
18 packets transmitted, 18 received, 0% packet loss, time
17004ms
rtt min/avg/max/mdev = 0.337/0.975/8.586/1.968 ms
```

Просмотр текущих сетевых настроек командой ifconfig

В результате выполнения этой команды в консоли vmUbuntu10 на экране появится информация, аналогичная нижеприведенной:

```
serp@vmUbuntu10:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:03:ff:07:40:14
          inet addr:192.168.1.10  Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::203:ffff:fe07:4014/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:156 errors:3 dropped:0 overruns:0 frame:0
          TX packets:245 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17231 (17.2 KB)  TX bytes:24544 (24.5 KB)
          Interrupt:11 Base address:0xe880

eth1      Link encap:Ethernet  HWaddr 00:03:ff:08:40:14
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::203:ffff:fe08:4014/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21826 errors:3 dropped:0 overruns:0 frame:0
          TX packets:1314 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3940928 (3.9 MB)  TX bytes:141408 (141.4 KB)
          Interrupt:11 Base address:0xec00
```

```
lo        Link encap:Локальная петля (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:168 errors:0 dropped:0 overruns:0 frame:0
          TX packets:168 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Из этого листинга видно, что в виртуальной машине vmUbuntu10 настроены три сетевых интерфейса:

• Интерфейс lo. Это интерфейс обратной петли, который позволяет компьютеру обращаться к самому себе. Интерфейс имеет IP-адрес 127.0.0.1 и необходим для нормальной работы системы:

• Интерфейсы eth0 и eth1. Это две сетевые карты Ethernet, к которым можно подключить сетевой кабель. Интерфейс eth0 имеет ip-адрес 192.168.1.10/24, а eth1 — 192.168.1.10/24, то есть маски подсетей 255.255.255.0.

В Ubuntu, как и вообще в Linux, имена проводных сетевых устройств принимают вид ethN, где N — число, означающее номер устройства связи в системе. Нумерация устройств начинается с нуля.

Если в компьютере несколько сетевых карт, то они получат имена eth0, eth1 и т. д. Если в сетевую карту ethN вставлен сетевой провод, идущий в модем, роутер или свитч, то для нее будет написано RUNNING. Это видно в нашем примере как для eth0, так и для eth1. Если сетевой кабель не подключен, то интерфейс будет неактивен, и для него не отображаются IP-адрес, широковещательный адрес и маска подсети.

Команда ifconfig также выдает информацию о количестве отправленных и полученных пакетов (параметры RX и TX) и сведения об аппаратных адресах сетевых карт (MAC — адресах). Это 48-разрядный серийный номер сетевого адаптера, присваиваемый производителем. Так как lo создан программно, у него не может быть аппаратного адреса.

Настройка сетевого интерфейса посредством ifconfig

Команда ifconfig позволяет сконфигурировать сетевой интерфейс, имеет очень широкие возможности, требует прав суперпользователя. Для подробной информации следует использовать man ifconfig. Простая настройка сетевого интерфейса возможна командой:

```
sudo ifconfig <ethN> <параметры>
```

где в качестве параметров, например, могут выступать: up – поднять интерфейс, down — остановит интерфейс, netmask – определить маску подсети и т. д.

Тогда, чтобы изменить IP-адрес, можно использовать следующий формат команды:

```
sudo ifconfig eth0 192.168.1.100 ,
```

а для изменения маски подсети этого интерфейса использовать:

```
sudo ifconfig eth0 netmask 255.255.248.0
```

Но возможен и другой подход, когда эти два действия будут выполнены одновременно, и измененный интерфейс будет активирован:

```
sudo ifconfig eth0 192.168.1.100 netmask 255.255.248.0 up
```

Иногда администраторы сетей делают привязку к MAC-адресу сетевой карты. В случае смены сетевой карты или всего системного блока в такой сети ничего работать не будет.

В этом случае можно изменить MAC-адрес на тот, что был у предыдущей сетевой карты, вручную. Для этого можно использовать, например, команду:

```
sudo ifconfig eth0 hw ether 00:e0:4c:d0:99:28
```

Все действия, которые были описаны выше, можно выполнить одной командой. Например, так:

```
sudo ifconfig eth0 down  &&  ifconfig eth0 192.168.1.100
netmask 255.255.248.0 hw ether 00:e0:4c:d0:99:28 up
```

где && — означает, успешное выполнение предыдущей команды. То есть если ifconfig eth0 down будет выполнено успешно, то на управление передается следующая команда ifconfig eth0 192.168.1.100 netmask 255.255.248.0 hw ether 00:e0:4c:d0:99:28 up.

Следует отметить, что для того чтобы заглушить сетевой интерфейс, можно использовать и другую команду:

```
ifdown eth0
```

Внимание!!!

Все эти настройки действительны лишь до перезагрузки системы. В других случаях изменения должны прописываться в сетевых конфигурационных файлах.

Настройка сетевого соединения в Ubuntu Linux

После, приведенного выше небольшого обзора перейдем к описанию настройки простых сетей в Linux. Для этого нужны права администратора и текстовый редактор. Так же следует представлять, какой IP-адрес надо присвоить сетевой карте, адрес шлюза сети, адреса вспомогательного и основного серверов DNS и прочие параметры.

Настройка сетевого соединения со статическим IP

Почти вся информация о настройках сети и методах ее активации, хранится в файле /etc/network/interfaces. Для настройки статического сетевого соединения следует отредактировать этот файл:

```
sudo nano /etc/network/interfaces
```

Если вы подключаетесь к уже настроенной сети — хорошо бы посмотреть, какие настройки там уже существуют. Если вы подключаетесь к провайдеру, то настройки вам дадут. При построении небольшой ЛВС на базе статических адресов нужно указать: IP-адрес (address), маску подсети (netmask) и адрес шлюза (gateway).

Для одного из компьютеров нашей виртуальной сети файл /etc/network/interfaces может иметь вид:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.10
netmask 255.255.255.0
gateway 192.168.1.1
```

где auto lo- говорит о том, что интерфейс lo поднимается автоматически при загрузке системы, auto eth0 — сетевая карта eth0 поднимается автоматически во время загрузки системы, iface eth0 inet static — указывает, что интерфейс (iface) сетевой карты (eth0) находится в диапазоне адресов ipv4 (inet) со статическим ip (static).

В этом примере сетевой карте eth0 назначен IP-адрес 192.168.1.10. Этот параметр, как и маску, и шлюз, можно изменить при иной конфигурации сети. После окончания редактирования следует сохранить и закрыть файл.

Настройка сетевого соединения с динамическим IP

В этом случае все может быть еще проще, и конфигурационный файл /etc/network/interfaces может иметь вид:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
```

Если по каким-либо причинам надо, чтобы при загрузке сетевой карты eth0 был другой MAC-адрес, то для этого в файл нужно дописать строчку:

```
hwaddress ether 00:e0:4c:d0:99:28
```

После окончания редактирования следует сохранить и закрыть файл. Аналогично, адреса серверов DNS надо описать в файле /etc/resolv.conf.

Настройка адресов DNS-серверов

Если нет адресов DNS-серверов, то можно к ресурсам Интернета подключаться только по IP-адресам, что несколько неудобно. Для того чтобы в Linux прописать адреса DNS-серверов, нужно отредактировать файл /etc/resolv.conf:

```
sudo nano /etc/resolv.conf
```

Если такого файла нет, то надо создать его и прописать адреса DNS-серверов следующим образом:

```
nameserver 192.168.1.1
nameserver 192.168.2.1
```

Пишем столько строк, сколько нужно указать DNS-серверов. Слово nameserver добавлять обязательно.

Активизация новых сетевых настроек

Чтобы изменения настройки сетевых интерфейсов вступили в силу, следует перезагрузить компьютер. Если этот подход вас не устраивает, то без перезагрузки следует дать команду:

```
sudo /etc/init.d/networking restart
```

Настройка закончена!
Изменения вступают в силу для всех сетевых интерфейсов, и можно продолжать работу в сети с новыми настройками.

Несколько IP адресов на одной сетевой карте

На одной сетевой карте может быть несколько IP-адресов. Это бывает необходимо, если в коммутатор идут два провода от разных сетей и один от — компьютера. В этом случае можно настроить на компьютере адреса обеих сетей без использования дополнительной сетевой карты.

В этом случае в Linux используется механизм сетевых алиасов (нескольких IP-адресов на одном интерфейсе). При его использовании:

- обязательно присваивается основной адрес интерфейсу ethX,
- алиасы (добавочные IP-адреса) присваиваются как

```
ethX:Y, где Y — номер алиаса.
```

Например, присвоить еще два IP-адреса интерфейсу eth0 можно, используя команду ifconfig совместно с алиасами:

```
# ifconfig eth0:1 inet 192.168.2.10 netmask 255.255.255.0
# ifconfig eth0:2 inet 192.168.3.10 netmask 255.255.255.0
```

Интересно то, что основной интерфейс можно настроить через dhcp (автоматически) dhclient eth0, а алиас eth0:1 в статику. Все это хорошо, но после перезагрузки все настройки слетают. Чтобы исключить этот

недостаток алиасы интерфейсов должны быть указаны в том же файле, где и все остальные описания интерфейсов. А именно в файле /etc/network/interfaces, который, например, может иметь вид:

```
# интерфейс обратной петли
auto lo
iface lo inet loopback
# основной сетевой интерфейс
auto eth0
iface eth0 inet static
        address 192.168.1.10
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
        dns-nameservers 127.0.0.1
auto eth0:1
iface eth0:1 inet static
        address 192.168.2.10
        netmask 255.255.255.0
        network 192.168.2.0
        broadcast 192.168.2.255
auto eth0:2
iface eth0:2 inet static
        address 192.168.3.10
        netmask 255.255.255.0
        network 192.168.3.0
        broadcast 192.168.3.255
```

Тут видно, что мы указали один основной адрес и 3 алиаса. Отметим, что к алиасам нельзя применять опции gateway или dns-nameservers.

Коротко о беспроводных сетях

Чтобы изменить тип интерфейса, используются команды:

```
wlanconfig ath0 destroy
wlanconfig ath0 create wlandev wifi0
wlanmode adhoc
```

Для того чтобы посмотреть список сетей, набираем:

```
iwlist ath0 scan
  или
wlanconfig ath0 list ap
```

Если есть открытая сеть и нужно к ней подключиться, набираем от рута:

```
iwconfig ath0 essid SomeESSID
```

Для шифрования соединения с использованием WEP используем:

```
iwconfig ath0 key ...
```

ПРИЛОЖЕНИЕ К РАЗДЕЛУ 7

Приложение 7.1. Пример файла конфигурации /etc/ssh/ssh/sshd_config

```
# Package generated configuration file
# See the sshd(8) manpage for details
# Задает порт, на котором будет работать SSH-сервер. Если директива
# не указана (закомментирована), то по умолчанию используется порт 22
#Port 22

# Локальный адрес, который должен прослушиваться SSH-сервером
#ListenAddress 0,0.0.0

# Директива Protocol позволяет выбрать версию протокола,
# рекомендуется использовать вторую версию
# Protocol 2,1
Protocol 2

# Ключевые файлы для второй версии протокола SSH
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# Директива AddressFamily задает семейство интерфейсов, которые должен
# прослушивать SSH-сервер
#AddressFamily any

# Время жизни ключа протокола первой версии. Время можно задавать в
# секундах или в часах (постфикс h, например, lh — это 1 час или
# 3600 секунд). По истечении указанного времени ключевой файл будет
# сгенерирован заново
#KeyRegenerationlnterval lh

# Разрядность ключа сервера в битах (только для 1- ой версии
# протокола SSH)
#ServerKeyBits 768

# Директивы управления протоколированием (можно не изменять)
```

```
#SyslogFacility AUTH
#LogLevel INFO

# Директивы аутентификации
# Время, предоставляемое клиенту для аутентификации. Задается в
# секундах или минутах. Если за это время клиент не аутентифицировал
# себя, соединение будет прекращено
#LoginGraceTime 2m

# Директива разрешает (yes) удаленный доступ пользователя root
PermitRootLogin yes

# Максимальное количество попыток аутентификации
#MaxAuthTries б

# Использование RSA (yes)
#RSAAuthentication yes
# Аутентификация с открытым ключом (при значении yes)
#PubkeyAuthentication yes
#AuthorizedKeysFile .ssh/authorized_keys

# Использование rhosts-аутентификации с поддержкой RSA,
# rhosts-аутентификацию использовать не рекомендуется, поэтому по
# умолчанию для этой директивы указано значение no. Если вы все-таки
# установите значение yes для этой директивы, то не забудьте указать
# в файле /etc/ssh/ssh-known_hosts IP-адреса компьютеров, которым
# разрешен доступ к SSH-серверу. Только для первой версии протокола
#RhostsRSAAuthentication no

# Если вы используете вторую версию протокола и хотите разрешить
# rhosts-аутентификацию, то вам нужно включить директиву
# HostbasedAuthentiaation,
# а разрешенные узлы указываются в файле ~/.ssh/known_hosts
# HostbasedAuthentication no

# Если вы не доверяете пользовательским файлам -/.ssh/known_hosts,
# установите значение yes для директивы IgnoreUserKnownHosts. Тогда
# будет использован только файл /etc/ssh/ssh_known_hosts
#IgnoreUserKnownHosts no

# Игнорировать файлы -/.rhosts и -/.shosts (рекомендуется установить yes)
#IgnoreRhosts yes
```

```
# Следующие директивы не рекомендуется изменять из соображений
# безопасности — они включают аутентификацию по паролю
# (а не IP-адресу компьютера, указанному в файле etc/ssh/ssh_known_hosta)
# и запрещают использование пустых паролей
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Параметры протокола аутентификации Kerberos.
# Рекомендуется использовать RSA-аутентификацию
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# Параметры GSSAPI
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Разрешить TCP-форвардинг
#AllowTcpForwarding yes

# Использовать порты шлюза
#GatewayPorts no

# Использовать Xll-форвардинг (для запуска Xll-приложений)
#XllForwarding yes

# Выводить сообщение дня (содержится в файле /etc/motd)
#PrintMotd yes

# Выводить время последней регистрации пользователя
#IPrintLastLog yes

# Не обрывать TCP-соединения после выполнения команды по SSH
#TCPKeepAlive yes

# Отключение (no) этой опции позволяет немного ускорить работу SSH,
# поскольку DNS не будет использоваться для разрешения доменных имен
#UseDNS yes

# Остальные параметры рекомендуется оставить как есть
#UseLogin по
#UsePrivilegeSeparation yes
```

```
#PermitUserEnvironment no
#Compression delayed
#ClientAlivelnterval 0
#ClientAliveCountMax 3
#PidFile /var/run/sshd.pid
#MaxStartups 10
#Banner /etc/issue.net
Subsystem sftp /usr/lib/openssh/sftp-server

# Использовать для аутентификации модули PAM
#UsePAM yes
```

Приложение 7.2. Краткая справка о протоколе SSH

> Основу раздела составляют материалы портала Википедия: http://ru.wikipedia.org/wiki/SSH.

SSH (*англ.* Secure Shell — «безопасная оболочка») — сетевой протокол сеансового уровня, позволяющий производить удаленное управление операционной системой и туннелирование TCP-соединений, например для передачи файлов. Схож по функциональности с протоколами Telnet и rlogin, но в отличие от них шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

SSH позволяет безопасно передавать в незащищенной среде практически любой другой сетевой протокол. Таким образом, можно не только удаленно работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео, например с веб-камеры. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно для удаленного запуска клиентов X Window System.

Большинство хостинг-провайдеров за определенную плату предоставляют клиентам доступ к их домашнему каталогу по SSH. Это может быть удобно как для работы в командной строке, так и для удаленного запуска программ, в том числе и графических приложений.

Стандарты

Первая версия протокола, SSH-1, была разработана в 1995 году исследователем Тату Улёненом из Технологического университета Хельсинки, Финляндия. SSH-1 был написан для обеспечения большей конфиденциальности, чем протоколы rlogin, telnet и rsh. В 1996 году была разработана более безопасная версия протокола, SSH-2, несовместимая с SSH-1. Сейчас под термином SSH обычно подразумевается именно SSH-2.

В 2006 году протокол был утвержден рабочей группой IETF в качестве стандарта Интернет. Однако в некоторых странах (Франция, Россия, Ирак и Пакистан) до сих пор требуется специальное разрешение в соответствующих структурах для использования определенных методов шифрования, включая SSH.

Известны две реализации SSH: собственническая коммерческая и бесплатная свободная. Свободная реализация называется OpenSSH. К 2006 году 80 % компьютеров сети Интернет использовало именно OpenSSH. Собственническая реализация разрабатывается организацией SSH Inc., она бесплатна для некоммерческого использования. Эти реализации содержат практически одинаковый набор команд.

Протокол SSH-2, в отличие от протокола telnet, устойчив к атакам прослушивания трафика («снифинг»), но неустойчив к атакам «человек посередине». Протокол SSH-2 также устойчив к атакам путем присоединения посредине — невозможно включиться в уже установленную сессию или перехватить ее.

Для предотвращения атак «человек посередине» при подключении к хосту, ключ которого еще не известен клиенту, клиентское ПО показывает пользователю «слепок ключа» (*англ.* key fingerprint). Рекомендуется тщательно проверять показываемый клиентским ПО "слепок ключа" со слепком ключа сервера, желательно полученным по надежным каналам связи или лично.

Поддержка SSH реализована во всех UNIX подобных системах, и на большинстве из них в числе стандартных утилит присутствуют клиент и сервер ssh. Существует множество реализаций SSH-клиентов и для не-UNIX ОС. Большую популярность протокол получил после широкого развития анализаторов трафика и способов нарушения работы локальных сетей, как альтернативное небезопасному протоколу Telnet решение для управления важными узлами.

Для работы по SSH нужен SSH-сервер и SSH-клиент. Сервер прослушивает соединения от клиентских машин и при установлении связи производит аутентификацию, после чего начинает обслуживание клиента. Клиент используется для входа на удаленную машину и выполнения команд. Для соединения сервер и клиент должны создать пары ключей — открытых и закрытых — и обменяться открытыми ключами. Обычно используется также и пароль.

Программные реализации

Программные реализации SSH-серверов в различных ОС:

- *BSD: OpenSSH.
- Linux: dropbear, lsh-server, openssh-server, ssh.
- Windows: freeSSHd, copssh, WinSSHD, KpyM Telnet/SSH Server, MobaSSH, OpenSSH через Cygwin.

Программные реализации SSH-клиентов и оболочек в различных ОС :

- GNU/Linux, *BSD: kdessh, lsh-client, openssh-client, PuTTY, ssh.
- MS Windows и Windows NT: PuTTY, SecureCRT, ShellGuard, Axessh, ZOC, SSHWindows, ProSSHD, Xshell.
- MS Windows Mobile: PocketPuTTY, mToken, sshCE, PocketTTY, OpenSSH, PocketConsole.
- Mac OS: NiftyTelnet SSH.
- Symbian OS: PuTTY.
- Java: MindTerm, AppGate Security Server.
- iPhone: i-SSH, ssh (в комплекте с Terminal).
- Android: connectBot.
- Blackberry: BBSSH.

Советы по безопасности использования SSH

1. Запрещение удаленного root-доступа.
2. Запрещение подключения с пустым паролем или отключение входа по паролю.
3. Выбор нестандартного порта для SSH-сервера.
4. Использование длинных SSH2 RSA-ключей (2048 бит и более). Системы шифрования на основе RSA считаются надёжными, если длина ключа не менее 1024 бит.
5. Ограничение списка IP-адресов, с которых разрешен доступ (например, настройкой фаервола).
6. Запрещение доступа с некоторых потенциально опасных адресов.
7. Отказ от использования распространенных или широко известных системных логинов для доступа по SSH.
8. Регулярный просмотр сообщений об ошибках аутентификации.
9. Установка систем обнаружения вторжений (IDS — Intrusion Detection System).
10. Использование ловушек, подделывающих SSH-сервис (honeypots).

SSH-туннелирование

SSH-туннель — это туннель, создаваемый посредством SSH-соединения и используемый для шифрования туннелированных данных. Используется для того, чтобы обезопасить передачу данных в Интернете. Особенность состоит в том, что незашифрованный трафик какого-либо протокола шифруется на одном конце SSH-соединения и расшифровывается на другом.

Практическая реализация выполняет несколькими способами:

- Созданием Socks-прокси для приложений, которые не умеют работать через SSH-туннель, но могут работать через Socks-прокси
- Использованием приложений, умеющих работать через SSH-туннель.

• Созданием VPN-туннеля, подходит практически для любых приложений.

• Если приложение работает с одним определенным сервером, можно настроить SSH-клиент таким образом, чтобы он пропускал через SSH-туннель TCP-соединения, приходящие на определенный TCP-порт машины, на которой запущен SSH-клиент. Например, клиенты ICQ подключаются по умолчанию на порт 5190. Тогда, чтобы настроить подключение к серверу ICQ через SSH-туннель, SSH-клиент настраивается на перенаправление подключений с любого порта локальной машины (например, тот же порт 5190) на удаленный сервер (например, login.icq.com и порт 5190). В данном случае клиент ICQ настраивается на подключение к серверу localhost(если ssh-клиент запущен на той же машине что и icq-клиент) и порт 5190.

Техническая информация о протоколе

SSH — это протокол сеансового уровня. SSH-сервер обычно прослушивает соединения на TCP-порту 22. Спецификация протокола SSH-2 содержится в RFC 4251. Для аутентификации сервера в SSH используется протокол аутентификации сторон на основе алгоритмов электронно-цифровой подписи RSA или DSA. Для аутентификации клиента также может использоваться ЭЦП RSA или DSA, но допускается также аутентификация при помощи пароля (режим обратной совместимости с Telnet) и даже ip-адреса хоста (режим обратной совместимости с rlogin).

Аутентификация по паролю наиболее распространена; она безопасна, так как пароль передается по зашифрованному виртуальному каналу. Аутентификация по ip-адресу небезопасна, эту возможность чаще всего отключают. Для создания общего секрета (сеансового ключа) используется алгоритм Диффи – Хеллмана (DH). Для шифрования передаваемых данных используется симметричное шифрование, алгоритмы AES, Blowfish или 3DES. Целостность переданных данных проверяется с помощью CRC32 в SSH1 или HMAC-SHA1/HMAC-MD5 в SSH2.

Для сжатия шифруемых данных может использоваться алгоритм LempelZiv (LZ77), который обеспечивает такой же уровень сжатия, что и архиватор ZIP. Сжатие SSH включается лишь по запросу клиента, и на практике используется редко.

ПРИЛОЖЕНИЕ К РАЗДЕЛУ 8

Приложение 8.1. Краткая справка о настройке удаленного рабочего стола в Windows XP.

В данном разделе очень сжато, базируясь на выдержках из Help-системы Windows XP, приводятся основные действия, которые необходимо выполнить по организации взаимодействия компьютеров с целью возможности использования и подключения к их удаленным рабочим столам.

- Чтобы настроить компьютер на использование дистанционного управления рабочим столом:
    1. Войдите в систему с учетной записью администратора или как участник группы «Администраторы».
    2. Выберите Пуск -> Панель управления -> Система.
    3. На вкладке «Удаленные сеансы» установите флажок «Разрешить удаленное подключение к этому компьютеру» и нажмите кнопку ОК.
    4. Убедитесь в наличии необходимых разрешений на удаленное подключение к данному компьютеру и нажмите кнопку ОК. Необходимо быть администратором компьютера или членом группы «Пользователи удаленного рабочего стола» на этом компьютере.
- Чтобы разрешить подключение других пользователей к компьютеру на вкладке «Удаленные сеансы»:
    1. Нажмите кнопку «Выбрать удаленных пользователей».
    2. В диалоговом окне «Пользователи удаленного рабочего стола» нажмите кнопку «Добавить».
    3. В диалоговом окне «Выбор: пользователи» нажмите кнопку «Размещение», чтобы указать область поиска.
    4. Нажмите кнопку «Типы объектов», чтобы указать типы объектов, поиск которых требуется выполнить.
    5. В поле «Введите имена выбираемых объектов (примеры):» введите имена объектов, которые требуется найти.
    6. Нажмите кнопку Проверить имена.

7. Найдя имя, нажмите кнопку ОК. Теперь имя появится в списке пользователей в диалоговом окне «Пользователи удаленного рабочего стола».

- Чтобы определить сетевое имя компьютера:
  1. Выберите Пуск -> Панель управления -> Система.
  2. На вкладке «Имя компьютера» найдите запись «Полное имя компьютера». Имя вашего компьютера в этой текстовой строке слева от первой точки.
  3. При необходимости смены имени компьютера или рабочей группы (домена) нажмите кнопку «Изменить».
- Для подключения к удаленному рабочему столу:
  1. Подключитесь к сети организации при помощи локального сетевого подключения (LAN), модема или подключения к виртуальной частной сети (VPN).
  2. Выберите Пуск -> Программы -> Стандартные -> Связь -> Подключение к удаленному рабочему столу (или Пуск -> Выполнить -> в поле «Открыть» ввести mstsc.exe и нажать кнопку «ОК»).
  3. В окне «Подключение к удаленному рабочему столу» введите имя удаленного компьютера, к которому требуется подключиться, и нажмите кнопку «Подключить».

Приложение 8.2. Удаленная установка xrdp на Ubuntu с использованием SSH и PuTTY

Ниже приведен протокол терминального доступа с использованием протокола SSH и утилиты PuTTY с Windows-машины на виртуальную vmUbuntu10 для удаленной инсталляции на ней RDP-сервера.

```
login as: serp
serp@192.168.1.10's password:
Linux vmUbuntu10 2.6.32-24-generic #39-Ubuntu SMP Wed Jul
28 06:07:29 UTC 2010 i686 GNU/Linux
Ubuntu 10.04.1 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

Your CPU appears to be lacking expected security
protections.
Please check your BIOS settings, or for more information,
run:
  /usr/bin/check-bios-nx --verbose
```

```
Last login: Sat Jul  9 23:24:41 2011 from 192.168.1.2
serp@vmUbuntu10:~$ sudo apt-get install xrdp
[sudo] password for serp:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
НОВЫЕ пакеты, которые будут установлены:
  xrdp
обновлено 0, установлено 1 новых пакетов, для удаления
отмечено 0 пакетов, и 296 пакетов не обновлено.
Необходимо скачать 0B/209kB архивов.
После данной операции, объём занятого дискового
пространства возрастёт на 733kB.
Выбор ранее не выбранного пакета xrdp.
(Чтение базы данных ... на данный момент установлено
125730 файлов и каталогов.)
Распаковывается пакет xrdp (из файла .../xrdp_0.4.1~dfsg-
2_i386.deb)...
Обрабатываются триггеры для man-db ...
Обрабатываются триггеры для ureadahead ...
Настраивается пакет xrdp (0.4.1~dfsg-2) ...
Starting xrdp: xrdp sesman.

serp@vmUbuntu10:~$
```

Как видно из приведенного листинга, удаленная установка RDP-сервера на Ubuntu-машину завершилась успешно. Теперь появляется реальная возможность стандартными средствами Windows подключаться к рабочему столу vmUbuntu10.

Для этого в Windows можно использовать, например, Пуск -> Программы -> Стандартные ->Связь -> Подключение к удаленному рабочему столу.

Приложение 8.3. Краткая справка о протоколе VNC

> Основу раздела составляют материалы портала Википедия: http://ru.wikipedia.org/wiki/Virtual_Network_Computing.

Virtual Network Computing (VNC) — система удаленного доступа к рабочему столу компьютера, использующая протокол RFB (*англ.* Remote Frame Buffer, удаленный кадровый буфер). Управление осуществляется путем передачи нажатий клавиш на клавиатуре и движений мыши с одного компьютера на другой и ретрансляции содержимого экрана через компьютерную сеть.

Система VNC платформонезависима: VNC-клиент, запущенный на одной ОС, может подключаться к VNC-серверу, работающему на другой ОС. Существуют реализации клиентской и серверной части практически для всех ОС. К одному VNC-серверу одновременно могут подключаться множественные клиенты.

История

VNC была создана в Olivetti & Oracle Research Lab, которая в то время принадлежала Olivetti и Oracle Corporation. В 1999 году лаборатория была приобретена компанией AT&T, которая закрыла отдел разработок в 2002 году. Оригинальные исходные коды доступны на условиях лицензии GPL. Разработчики, работавшие над VNC в AT&T Research Lab: Tristan Richardson, Квентин Стаффорд-Фрейзер (англ.), James Weatherall, Ken Wood, Энди Хоппер (англ.), Charlie McLachlan, Paul Webster.

Устройство

VNC состоит из двух частей: клиента и сервера. Сервер — программа, предоставляющая доступ к экрану компьютера, на котором она запущена. Клиент (или viewer) — программа, получающая изображение экрана с сервера и взаимодействующая с ним по протоколу RFB.

Протокол RFB

RFB («remote frame buffer») простой клиент-серверный сетевой протокол прикладного уровня для удаленного доступа к графическому рабочему столу компьютера. Так как он работает на уровне кадрового буфера, то его можно применять для таких пользовательских графических оконных систем как X11, Windows, Macintosh. В начале своего развития RFB — был относительно простым протоколом, основанным на графических примитивах: «Положить прямоугольник пиксельных данных на заданную координатами позицию». Сервер посылает небольшие прямоугольники клиенту. Такая схема в своей примитивной форме потребляет большую часть пропускной возможности канала.

Для снижения нагрузки на канал используются различные методы. Существуют различные кодировки — методы определения наиболее эффективного способа передачи этих прямоугольников. Протокол RFB позволяет клиенту и серверу «договориться» о том, какая кодировка будет использована. Самый простой метод кодирования, поддерживаемый всеми клиентами и серверами — «raw encoding», при котором пиксели передаются в порядке слева направо, сверху вниз, и после передачи первоначального состояния экрана передаются только изменившиеся пиксели.

Этот метод работает очень хорошо при незначительных изменениях изображения на экране (движения указателя мыши по рабочему столу, набор текста под курсором), но загрузка канала становится очень высокой при одновременном изменении большого количества пикселей, например при просмотре видео в полноэкранном режиме. За время своего развития протокол оброс различными дополнительными функциями и опциями такими как передача файлов, сжатие, безопасность.

По умолчанию RFB использует диапазон TCP-портов с 5900 до 5906. Каждый порт представляет собой соответствующий экран X-сервера (порты с 5900 по 5906 ассоциированы с экранами с :0 по :6). Java-клиенты, доступные во многих реализациях, использующих встроенный Web-сервер для этой цели, например, в RealVNC, связаны с экранами таким же образом, но на диапазоне портов с 5800 до 5806. Многие компьютеры под управлением ОС Windows могут использовать лишь один порт из-за отсутствия многопользовательских свойств, присущих UNIX-системам. Для Windows-систем экран по умолчанию — :0, что соответствует порту 5900.

Также существует возможность обратного подключения от сервера к клиенту. В этом случае клиент переводится в слушающий (Listening) режим, и соединение инициируется сервером на 5500 TCP-порт клиента. Порты могут быть изменены.

Безопасность

Изначально VNC не использует шифрование трафика, однако в процедуре аутентификации пароль не передается в открытом виде, а используется алгоритм «вызов — ответ» с DES-шифрованием (эффективная длина ключа составляет 56-бит). Во многих реализациях существует ограничение в 8 символов на длину пароля, и если его длина превосходит 8 символов, то пароль урезается, а лишние символы игнорируются.

При необходимости надежного шифрования всей VNC-сессии, она может быть установлена через SSL, SSH или VPN-туннель, а также поверх IPSec. Технология IPSec поддерживается подавляющим большинством современных ОС и используется как при соединении через Интернет, так и в локальных сетях. SSH-клиенты позволяют создавать SSH-туннели как со всех основных платформ (UNIX, Windows, Macintosh и др.), так и для менее популярных.

UltraVNC позволяет использовать специальный плагин, распространяемый с открытым исходным кодом, который шифрует всю сессию VNC, используя алгоритмы AES или RC4, включая аутентификацию и передачу данных. Также существуют варианты аутентификации на основе NTLM и учетных записей пользователей в

Active Directory. UltraVNC позволяет передавать файлы между сервером и клиентом, в любых направлениях.

RealVNC в коммерческой версии продукта использует алгоритм AES для шифрования соединения и алгоритм RSA для аутентификации.

Приложение 8.4. Настройка брандмауэра при установке удаленных соединений

Перед тем как перейти к установке удаленных соединений или к их приему, необходимо настроить программное обеспечение брандмауэра. На компьютерах, к которым вы хотите подключиться удаленно, необходимо разрешить трафик VNC или RDP через брандмауэр.

В Windows при старте сервера вы должны получить запрос на Блокирование или Разрешение доступа к сети приложению сервера удаленного рабочего стола. Если нажать кнопку «Разрешить», все должно заработать. Если вы не получили запрос, то можете зайти в свойства брандмауэра Windows и добавить разрешение для этого приложения вручную, используя номера портов, указанных ниже.

В Linux вам, скорее всего, необходимо будет вручную добавить правила для входящих соединений в брандмауэре, на компьютере, принимающем запросы на подключение. Если необходимо, вы можете вызвать из меню браузер и поискать в Google информацию о том, как настроить брандмауэр. Ваш дистрибутив Linux может включать GUI (графический интерфейс пользователя) для вашего брандмауэра или вы можете использовать командную строку для его настройки.

Таким же образом добавьте исключение или правило для того, чтобы разрешить трафик на соответствующих портах, перечисленных ниже.

- RDP использует TCP порт 3389.
- VNC использует порты, начиная с 5900, поэтому, когда создают правила или исключения для брандмауэра, то лучшим вариантом будет определение области портов 5900–5905.

После этого у вас появится возможность удаленно подключаться к компьютерам в вашей локальной сети. Для удаленного соединения через Интернет вы должны также настроить ваш маршрутизатор.

Приложение 8.5. Краткая справка о Xming и X Window System

> Основу раздела составляют материалы портала Википедия: http://ru.wikipedia.org/wiki/Xming.

Xming — порт сервера X Window System для операционной системы Microsoft Windows (XP/2003/Vista). Сервер Xming основан на сервере Xorg (X11R6.9). Несмотря на сходство кода, Xming отличается от X-сервера Cygwin. Главным отличием является то, что для работы Xming не

требуется библиотека Cygwin, что позволяет использовать его под Windows без установки добавочных библиотек. Это в свою очередь позволяет устанавливать Xming на переносных устройствах, таких, как USB flash drive.

Xming можно использовать вместе с приложениями, работающими по SSH (такими, как PuTTY) для обеспечения шифрованной передачи сессии X11 с Unix. В этом случае Xming может использоваться для безопасной работы с графическими приложениями удаленного компьютера.

X Window System — оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя. Используется в UNIX-подобных ОС.

X Window System обеспечивает базовые функции графической среды: отрисовку и перемещение окон на экране, взаимодействие с устройствами ввода, такими как, например, мышь и клавиатура. X Window System не определяет деталей интерфейса пользователя — этим занимаются менеджеры окон, которых разработано множество. По этой причине внешний вид программ в среде X Window System может очень сильно различаться в зависимости от возможностей и настроек конкретного оконного менеджера.

В X Window System предусмотрена сетевая прозрачность: графические приложения могут выполняться на другой машине в сети, а их интерфейс при этом будет передаваться по сети и отображаться на локальной машине пользователя (в случае, если это разрешено в настройках). В контексте X Window System, термины «Клиент» и «Сервер» имеют непривычное для многих пользователей значение. «Сервер» означает локальный дисплей пользователя (дисплейный сервер), а «Клиент» — программу, которая этот дисплей использует (она может выполняться на удалённом компьютере).

Система X Window System была разработана в Массачусетском технологическом институте в 1984 году. Нынешняя версия протокола — X11 — появилась в сентябре 1987 года. Проект X возглавляет фонд X.Org Foundation. Референсная (или образцовая) реализация системы свободно доступна на условиях лицензии MIT и подобных ей лицензий. X Window System часто называют X11 или просто X (в разговорной речи — «иксы»).

В главе 8 и приложениях к ней использованы ресурсы Интернет, оригиналы которых можно найти по этим ссылкам:

http://ru.wikipedia.org/wiki/Virtual_Network_Computing,
http://www.linuxspace.org/archives/3444,
http://www.linuxplanet.com/linuxplanet/tutorials/6641/1/,
http://debback.blogspot.com,
http://ru.wikipedia.org/wiki/X_Window_System.
http://ru.wikipedia.org/wiki/Xming

ПРИЛОЖЕНИЕ К РАЗДЕЛУ 9

Приложение 9.1. Краткая справка о файл /etc/fstab

> По материалам Nana Langstedt (http://www.tuxfiles.org/linuxhelp/fstab.html) в переводе Алексея Дмитриева (http://rus-linux.net/lib.php?name=/MyLDP/file-sys/fstab.html)

Назначение файла /etc/fstab

Этот файл содержит информацию обо всех разделах жесткого диска и других носителях информации в компьютере. В нем прописано, куда и как разделы винчестера и другие носители должны быть примонтированы.

В каждой системе файл /etc/fstab выглядит по-своему, так как разделы, устройства, и их свойства, различаются в разных системах. Но скелет структуры файла всегда одинаков и может иметь вид, например:

```
/dev/hda2  /              ext2  defaults             1 1
/dev/hdb1  /home          ext2  defaults             1 2
/dev/cdrom /media/cdrom   auto  ro,noauto,user,exec  0 0
/dev/fd0   /media/floppy  auto  rw,noauto,user,sync  0 0
proc       /proc          proc  defaults             0 0
/dev/hda1  swap           swap  pri=42               0 0
```

Каждая строка содержит информацию об одном разделе или устройстве. Первый столбец содержит имя устройства, второй — точку его монтирования, третий — тип файловой системы, четвертый — опции монтирования, пятый (число) — опции дампа, шестой (число) опции проверки файловой системы. Рассмотрим подробно эту информацию.

Устройство и точка монтирования

Это первый и второй столбцы файла, которые содержат то, что вы пишете в командной строке при вызове команды mount. То есть это имя устройства (раздела) и точка его монтирования. Точка монтирования, указанная в /etc/fstab, является точкой монтирования по умолчанию. Эта та директория, куда будет примонтировано устройство, если не была указана другая при вводе команды mount.

Разные дистрибутивы Linux создают специальные директории для точек монтирования в каталоге /mnt или /media. Что это означает на практике? Если ввести команду:

```
$ mount /dev/fd0/ ,
```

то дискета будет смонтирована в /media/floppy, потому что эта точка монтирования указана в /etc/fstab и поэтому используется по умолчанию. Если строчки /dev/fd0 в файле /etc/fstab не окажется, то команда mount не будет знать, куда следует монтировать дискету.

Точки монтирования по умолчанию легко изменить, если они вас почему-либо не устраивают. Для этого надо заменить директории в файле /etc/fstab на любые другие, реально существующие директории. Если подходящих не существует, то их надо просто создать.

Некоторые разделы и устройства монтируются автоматически, в процессе загрузки системы. Рассмотрим первые две строки:

```
/dev/hda2   /       ext2   defaults    1 1
/dev/hdb1   /home   ext2   defaults    1 2
```

Они означают, что /dev/hda2 будет примонтирован в директорию /, а /dev/hdb1 — в директорию /home. Это произойдет автоматически, когда система загружается. Если этого не произойдет, то система не сможет работать, так как все программы находятся именно в директории /, и, если она не смонтирована, то и доступа к программам нет! Именно из файла /etc/fstab система узнает, куда примонтировать /dev/hda2, а куда /dev/hdb1.

Тип файловой системы

В третьем столбце файла указывается тип файловой системы раздела или устройства. Поддерживается множество различных файловых систем, но наиболее употребительные:

- ext2 и ext3.

  Раньше стандартом была система ext2. Затем ext3 и ext4, но в наши дни почти все дистрибутивы используют по умолчанию Ext4 или ReiserFS.

- reiser.

  Возможно, что ваши Linux разделы отформатированы в ReiserFS. Подобно Ext3, ReiserFS тоже журналируемая файловая система, но она является гораздо более «продвинутой»

- swap.

  Своп значит подкачка. Файловая система типа "swap" используется в разделах подкачки.

- vfat и ntfs.

  Windows разделы используют либо Vfat, либо NTFS. В 9х сериях (95, 98, ME) применялась Vfat, более известная как FAT32, в последующих сериях используется NTFS. В XP можно применять и Vfat тоже. Для возможности записи в Windows-разделы из Linux, то надо их отформатировать соответствующим образом.

- auto.

  Опция «auto» означает, что тип файловой системы определяется автоматически. Если взглянуть на пример файла /etc/fstab, приведенный выше, то видно, что и floppy и CD-ROM имеют вместо типа файловой системы опцию «auto». Дело в том, что в этих устройствах могут применяться различные типы файловых систем.

Опции монтирования

В четвертом столбце перечисляются все опции, с которыми устройство или раздел будут смонтированы. Рассмотрим наиболее распространенные опции:

- auto (по умолчанию) и noauto.

  Устройство с опцией auto монтируется автоматически во время запуска компьютера. Если не нужна автоматическое монтирование, то в файле /etc/fstab прописывают опцию noauto. С опцией noauto, устройство или раздел могут быть смонтированы только явно.
- exec (по умолчанию) и noexec.

  Чтобы запускать двоичные программы, находящиеся в данном разделе, применяют опцию exec, а если нет — noexec. Последнее может быть полезно, если на разделе содержатся программы, которые не могут работать в вашей системе, например Windows-приложения, либо программы, нежелательные к запуску по той или иной причине.
- Ro и rw

  Монтирует файловую систему в режиме «только чтение» или в режиме «чтение и запись», соответственно.
- sync and async (по умолчанию).

  Определяют, как выполняется ввод/вывод в данную файловую систему: синхронно или асинхронно. В примере у дискеты - опция sync. То есть, когда вы копируете файл на дискету, то запись физически происходит в тот самый момент, когда дана команда копировать. При использовании опции async, в случае с дискетой это означает, что физически запись может произойти много позже команды.
- defaults.

  По умолчанию включены: rw, suid, dev, exec, auto, nouser и async.

Опции dump и fsck

Дамп — это опция резервного копирования, а fsck — опция проверки файловой системы. Они соответствуют пятому и шестому столбам файла.

Пятый столбец файла /etc/fstab — это опция дампа, выраженная числом. От его значения зависит, будет ли создаваться резервная копия файловой

системы. Если ноль — программа dump проигнорирует такую файловую систему. В примере, большинство строк в пятом столбце — 0.

В шестой колонке опция программы fsck (filesystem check-проверка файловой системы). Программа fsck использует значение чисел в этом столбце, чтобы определить, в каком порядке проверять файловые системы. Если там ноль, то файловая система вообще не будет проверяться.

Примеры записей в файл /etc/fstab

Для примера мы разберем два случая, которые чаще прочих расстраивают новых пользователей Linux: дискета и CD-ROM (хотя дискеты в последнее время практически не употребляются).

```
/dev/fd0  /media/floppy  auto  rw,noauto,user,sync  0  0
```

Эта строка означает, что дискета монтируется по умолчанию в директорию /media/floppy и что тип системы определяется автоматически. Это полезно, так как тип файловой системы на дискетах может быть различным. Особое внимание обратите на опции rw и user: они обязательно должны быть прописаны, если вы хотите монтировать дискету и записывать на нее, будучи рядовым пользователем.

Еще обратите внимание на опцию sync. С таким же успехом может быть и async, по причинам, которые мы уже обсудили.

```
/dev/cdrom /media/cdrom auto ro,noauto,user,exec 0 0
```

Снова отметим опцию user, позволяющую рядовому пользователю монтировать компакт-диски. Опция ro установлена потому, что нет смысла монтировать CD-ROM в режиме «чтение — запись», ведь на него все равно ничего не запишешь. А вот опция exec очень кстати, если надо запустить какую-либо программу с компакт-диска.

Обратите также внимание на применение опции noauto как с дискетой, так и с CD-ROM, это означает, что они не будут автоматически смонтированы при запуске системы. Это очень разумно для съемных носителей, которых при запуске может просто не быть в дисководах, ведь нет смысла пытаться монтировать то, чего нет.

В главе 9 и приложениях к ней использованы ресурсы Интернет, оригиналы которых можно найти по этим ссылкам:

http://ru.wikipedia.org/wiki/SMB
http://ru.wikipedia.org/wiki/GVFS
http://ru.wikipedia.org/wiki/Filesystem_in_Userspace
http://ru.wikipedia.org/wiki/GIO
http://ru.wikipedia.org/wiki/D-Bus
http://help.ubuntu.ru/
http://www.unix.com/man-page/All/1/gvfs-copy/
http://www.tuxfiles.org/linuxhelp/fstab.html"

ПРИЛОЖЕНИЯ К РАЗДЕЛУ 10

Приложение 10.1. Исходный конфигурационный файл Samba сервера /etc/samba/smb.conf

```
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
#  - When such options are commented with ";", the proposed setting
#    differs from the default Samba behaviour
#  - When commented with "#", the proposed setting is the default
#    behaviour of Samba but the option is considered important
#    enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.
# A well-established practice is to name the original file
# "smb.conf.master" and create the "real" config file with
# testparm -s smb.conf.master >smb.conf
# This minimizes the size of the really used smb.conf file
# which, according to the Samba Team, impacts performance
# However, use this with caution if your smb.conf file contains nested
# "include" statements. See Debian bug #483187 for a case
# where using a master file is not a good idea.
#

#=================== Global Settings =====================
```

```
[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba will part of
   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
   server string = %h server (Samba, Ubuntu)

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
#   wins support = no

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
;   wins server = w.x.y.z

# This will prevent nmbd to search for NetBIOS names through DNS.
   dns proxy = no

# What naming service and in what order should we use to resolve host names
# to IP addresses
;   name resolve order = lmhosts host wins bcast

#### Networking ####

# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
;   interfaces = 127.0.0.0/8 eth0

# Only bind to the named interfaces and/or networks; you must use the
# 'interfaces' option above to use this.
# It is recommended that you enable this feature if your Samba machine is
# not protected by a firewall or is a firewall itself.  However, this
# option cannot handle dynamic or non-broadcast interfaces correctly.
;   bind interfaces only = yes
```

```
#### Debugging/Accounting ####

# This tells Samba to use a separate log file for each machine
# that connects
   log file = /var/log/samba/log.%m

# Cap the size of the individual log files (in KiB).
   max log size = 1000

# If you want Samba to only log through syslog then set the following
# parameter to 'yes'.
#   syslog only = no

# We want Samba to log a minimum amount of information to syslog. Everything
# should go to /var/log/samba/log.{smbd,nmbd} instead. If you want to log
# through syslog you should set the following parameter to something higher.
   syslog = 0

# Do something sensible when Samba crashes: mail the admin a backtrace
   panic action = /usr/share/samba/panic-action %d

####### Authentication #######

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/ServerType.html
# in the samba-doc package for details.
#   security = user

# You may wish to use password encryption.  See the section on
# 'encrypt passwords' in the smb.conf(5) manpage before enabling.
   encrypt passwords = true

# If you are using encrypted passwords, Samba will need to know what
# password database type you are using.
   passdb backend = tdbsam

   obey pam restrictions = yes

# This boolean parameter controls whether Samba attempts to sync the Unix
# password with the SMB password when the encrypted SMB password in the
# passdb is changed.
```

```
   unix password sync = yes

# For Unix password sync to work on a Debian GNU/Linux system, the following
# parameters must be set (thanks to Ian Kahan < for
# sending the correct chat script for the passwd program in Debian Sarge).
   passwd program = /usr/bin/passwd %u
   passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .

# This boolean controls whether PAM will be used for password changes
# when requested by an SMB client instead of the program listed in
# 'passwd program'. The default is 'no'.
   pam password change = yes

# This option controls how unsuccessful authentication attempts are mapped
# to anonymous connections
   map to guest = bad user

########## Domains ###########

# Is this machine able to authenticate users. Both PDC and BDC
# must have this setting enabled. If you are the BDC you must
# change the 'domain master' setting to no
#
;   domain logons = yes
#
# The following setting only takes effect if 'domain logons' is set
# It specifies the location of the user's profile directory
# from the client point of view)
# The following required a [profiles] share to be setup on the
# samba server (see below)
;   logon path = \\%N\profiles\%U
# Another common choice is storing the profile in the user's home directory
# (this is Samba's default)
#   logon path = \\%N\%U\profile

# The following setting only takes effect if 'domain logons' is set
# It specifies the location of a user's home directory (from the client
# point of view)
;   logon drive = H:
#   logon home = \\%N\%U
```

```
# The following setting only takes effect if 'domain logons' is set
# It specifies the script to run during logon. The script must be stored
# in the [netlogon] share
# NOTE: Must be store in 'DOS' file format convention
;   logon script = logon.cmd

# This allows Unix users to be created on the domain controller via the SAMR
# RPC pipe.  The example command creates a user account with a disabled Unix
# password; please adapt to your needs
; add user script = /usr/sbin/adduser --quiet --disabled-password --gecos "" %u

# This allows machine accounts to be created on the domain controller via the
# SAMR RPC pipe.
# The following assumes a "machines" group exists on the system
; add machine script  = /usr/sbin/useradd -g machines -c "%u machine account" -
d /var/lib/samba -s /bin/false %u

# This allows Unix groups to be created on the domain controller via the SAMR
# RPC pipe.
; add group script = /usr/sbin/addgroup --force-badname %g

########### Printing ###########

# If you want to automatically load your printer list rather
# than setting them up individually then you'll need this
#   load printers = yes

# lpr(ng) printing. You may wish to override the location of the
# printcap file
;   printing = bsd
;   printcap name = /etc/printcap

# CUPS printing.  See also the cupsaddsmb(8) manpage in the
# cupsys-client package.
;   printing = cups
;   printcap name = cups

############# Misc #############

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
;   include = /home/samba/etc/smb.conf.%m
```

```
# Most people will find that this option gives better performance.
# See smb.conf(5) and /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/speed.html
# for details
# You may want to add the following on a Linux system:
#        SO_RCVBUF=8192 SO_SNDBUF=8192
#   socket options = TCP_NODELAY

# The following parameter is useful only if you have the linpopup package
# installed. The samba maintainer and the linpopup maintainer are
# working to ease installation and configuration of linpopup and samba.
;   message command = /bin/sh -c '/usr/bin/linpopup "%f" "%m" %s; rm %s' &

# Domain Master specifies Samba to be the Domain Master Browser. If this
# machine will be configured as a BDC (a secondary logon server), you
# must set this to 'no'; otherwise, the default behavior is recommended.
#   domain master = auto

# Some defaults for winbind (make sure you're not using the ranges
# for something else.)
;   idmap uid = 10000-20000
;   idmap gid = 10000-20000
;   template shell = /bin/bash

# The following was the default behaviour in sarge,
# but samba upstream reverted the default because it might induce
# performance issues in large organizations.
# See Debian bug #368251 for some of the consequences of *not*
# having this setting and smb.conf(5) for details.
;   winbind enum groups = yes
;   winbind enum users = yes

# Setup usershare options to enable non-root users to share folders
# with the net usershare command.

# Maximum number of usershare. 0 (default) means that usershare is disabled.
;   usershare max shares = 100

# Allow users who've been granted usershare privileges to create
# public shares, not just authenticated ones
   usershare allow guests = yes
```

```
#===================== Share Definitions =====================

# Un-comment the following (and tweak the other settings below to suit)
# to enable the default home directory shares.  This will share each
# user's home directory as \\server\username
;[homes]
;   comment = Home Directories
;   browseable = no

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
;   read only = yes

# File creation mask is set to 0700 for security reasons. If you want to
# create files with group=rw permissions, set next parameter to 0775.
;   create mask = 0700

# Directory creation mask is set to 0700 for security reasons. If you want to
# create dirs. with group=rw permissions, set next parameter to 0775.
;   directory mask = 0700

# By default, \\server\username shares can be connected to by anyone
# with access to the samba server.  Un-comment the following parameter
# to make sure that only "username" can connect to \\server\username
# This might need tweaking when using external authentication schemes
;   valid users = %S

# Un-comment the following and create the netlogon directory for Domain
Logons
# (you need to configure Samba to act as a domain controller too.)
;[netlogon]
;   comment = Network Logon Service
;   path = /home/samba/netlogon
;   guest ok = yes
;   read only = yes
;   share modes = no

# Un-comment the following and create the profiles directory to store
# users profiles (see the "logon path" option above)
# (you need to configure Samba to act as a domain controller too.)
# The path below should be writable by all users so that their
# profile directory may be created the first time they log on
;[profiles]
```

```
;   comment = Users profiles
;   path = /home/samba/profiles
;   guest ok = no
;   browseable = no
;   create mask = 0600
;   directory mask = 0700

[printers]
   comment = All Printers
   browseable = no
   path = /var/spool/samba
   printable = yes
   guest ok = no
   read only = yes
   create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
   comment = Printer Drivers
   path = /var/lib/samba/printers
   browseable = yes
   read only = yes
   guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin

# A sample share for sharing your CD-ROM with others.
;[cdrom]
;   comment = Samba server's CD-ROM
;   read only = yes
;   locking = no
;   path = /cdrom
;   guest ok = yes

# The next two parameters show how to auto-mount a CD-ROM when the
#       cdrom share is accesed. For this to work /etc/fstab must contain
#       an entry like this:
#
```

```
#       /dev/scd0   /cdrom  iso9660 defaults,noauto,ro,user   0 0
#
# The CD-ROM gets unmounted automatically after the connection to the
#
# If you don't want to use auto-mounting/unmounting make sure the CD
#       is mounted on /cdrom
#
;   preexec = /bin/mount /cdrom
;   postexec = /bin/umount /cdrom
```

Приложение 10.2. Переменные, зарезервированные для работы с файловым сервером Samba

Ниже приводится список переменных, которые используются при конфигурирование сервера Samba и в макроподстановках файла /etc/samba/smb.conf. Дается краткое описания каждой из переменных:

```
Переменные клиента
  %a - Архитектура клиента. Например: WinNT, Win95, Samba
  %I - IP адрес клиента
  %m - NETBIOS имя клиента
  %M - DNS имя клиента
Переменные пользователя
  %g - первичная группа
  %H - домашняя директория
  %u - имя пользователя
Переменные расшаренных ресурсов
  %P - Корневая папка ресурса
  %S - Имя ресурса
Переменные сервера
  %h - DNS имя
  %L - NETBIOS имя
  %v - версия Samba
Другое
  %T - Текущая дата и время
```

Приложение 10.3. Утилита для администрирования базы данных пользователей Samba

Это приложение представляет собой сокращенный перевод страницы man pdbedit(8) из пакета Samba, написанной Simo Sorce и Jelmer Vernooij и

переведенной на русский язык Айзятуллен Рамилем (Mail-to:argo-uln@yandex.ru).

Имя

pdbedit — утилита для администрирования базы данных SAM (База данных Пользователей Samba)

Синтаксис

pdbedit [-L] [-v] [-w] [-u username] [-f fullname] [-h homedir] [-D drive] \
[-S script] [-p profile] [-a] [-t, --password-from-stdin] \
[-m] [-r] [-x] [-i passdb-backend] [-e passdb-backend] \
[-b passdb-backend] [-g] [-d debuglevel] [-s configfile] \
[-P account-policy] [-C value] [-c account-control] [-y]

Описание

Эта утилита является частью пакета samba(7).

Программа pdbedit используется для управления пользовательскими учетными записями, сохраненными в базе данных sam, и может быть запущена только с правами пользователя root.

Утилита pdbedit использует модульный интерфейс passdb и не зависит от выбранной базы данных пользователей (базы smbpasswd, ldap, nis+ и tdb и другие могут использоваться с этой утилитой).

Существует пять способов использования утилиты pdbedit:

1. Добавление учетной записи пользователя.
2. Удаление учетной записи пользователя.
3. Модификация учетной записи пользователя.
4. Просмотр учетных записей пользователей.
5. Импорт учетных записей пользователей.

Параметры

- **-L**

Этот параметр выводит список всех учетных записей пользователей существующих в базе данных пользователей. Выводится список пар user/uid, разделенных символом ':'.

- **-v**

Параметр включает расширенный формат вывода. Вынуждает pdbedit выводить список пользователей в расширенном формате. Пример: pdbedit -L -v.

- **-w**

Параметр включает формат вывода совместимый с форматом smbpasswd. Вынуждает pdbedit выводить список пользователей из базы в формате совместимом с форматом файла smbpasswd (более подробно смотрите smbpasswd(5)). Пример: pdbedit -L -w.

- **-u username**

Параметр передает имя пользователя в запрос (просмотр, добавление, удаление). Опция обязательна для параметров добавление, удаление, изменение и опционально для просмотра.

- **-f fullname**

Опция используется при добавлении или модификации пользовательской учетной записи. Она определяет полное имя пользователя. Пример: -f "Simo Sorce".

- **-h homedir**

Опция используется при добавлении или модификации пользовательской учетной записи. Определяет путь к домашней директории пользователя. Пример: -h \\\\BERSERKER\\sorce.

- **-D drive**

Опция используется при добавлении или модификации пользовательской учетной записи. Определяет букву логического диска для домашней директории. Пример: -D "H:".

- **-S script**

Опция используется при добавлении или модификации пользовательской учетной записи. Определяет путь к пользовательскому logon-скрипту. Пример: -S \\\\BERSERKER\\netlogon\\sorce.bat.

- **-p profile**

Опция используется при добавлении или модификации пользовательской учетной записи. Определяет путь к профилю пользователя. Пример: -p \\\\BERSERKER\\netlogon.

- **-G SID|rid**

Опция используется при добавлении или модификации пользовательской учетной записи. Определяет для пользователя новый групповой SID (Security Identifier) или rid. Пример: -G S-1-5-21-2447931902-1787058256-3961074038-1201.

- **-U SID|rid**

Опция используется при добавлении или модификации пользовательской учетной записи. Определяет для пользователя новый SID (Security Identifier) или rid. Пример: -U S-1-5-21-2447931902-1787058256-3961074038-5004.

- **-c account-control**

Опция используется при добавлении или модификации пользовательской учетной записи. Она определяет настройки для пользователя. Ниже дан список флагов атрибутов пользователя.

```
N: Пароль не требуется
D: Учетная запись отключена
H: Домашняя директория
T: Временная копия другой учетной записи
```

```
U: Постоянная пользовательская учетная запись
M: Учетная запись входа MNS
W: Доверенная учетная запись рабочей станции
S: Серверная доверенная учетная запись
L: Автоматическая блокировка
X: Пароль не имеет срока давности
I: Доменная доверенная учетная запись
Пример: -c "[X ]"
```

- **-a**

Используется для добавления пользователя в базу данных. Эта опция требует ввода имени пользователя в параметре -u. Также при добавлении пользователя будет запрошен пароль для создаваемого пользователя. Пример: pdbedit -a -u sorce.

Замечание. pdbedit не вызывает скрипт синхронизации unix пароля, даже если параметр unix password sync установлен. Он обновит только базу пользователей Samba.

Если вы хотите добавить пользователя и незамедлительно синхронизировать пароль, используйте опцию -a команды smbpasswd.

- **-t | --password-from-stdin**

Этот параметр вынуждает pdbedit читать пароль со стандартного ввода, предпочтительно с /dev/tty **(**дружелюбно для программы passwd(1)**)**. Пароль будет передан дважды и прерывается переходом на новую строку.

- **-r**

Используется для изменения существующего пользователя в базе. Параметр включает имя пользователя, определенное в ключе -u. Другие опции могут быть определены для изменения свойств определенного пользователя. Этот флаг сохранен для обратной совместимости, его использование больше не обязательно.

- **-m**

Этот параметр может использоваться только совместно с параметром -a. Он вынудит pdbedit добавить доверенную учетную запись компьютера вместо учетной записи пользователя (-u username будет представлять имя машины). Пример: pdbedit -a -m -u w2k-wks.

- **-x**

Параметр вынудит pdbedit удалить учетную запись из базы. Он требует указания имени пользователя в ключе -u. Пример: pdbedit -x -u bob.

- **-i passdb-backend**

Использовать альтернативное хранилище паролей для выбора пользователей определенных в smb.conf. Может использоваться для импорта данных в локальную пользовательскую базу. Эта опция упощает процесс миграции из одного хранилища паролей в другое. Пример: pdbedit -i smbpasswd:/etc/smbpasswd.old.

- **-e passdb-backend**

Используется для экспорта пользователей в альтернативное хранилище пользовательских данных. Упрощает процесс миграции из одного хранилища данных в другое через резервную копию. Пример: pdbedit -e smbpasswd:/root/samba-users.backup.

- **-g**

Если использовать ключ -g, тогда ключи -i in-backend -e out-backend будут применены к отображению группы вместо пользовательской базы. Упрощает процесс миграции из одного хранилища данных в другое через резервную копию.

- **-b passdb-backend**

Использовать отличный от принятого по умолчанию passdb backend. Пример: pdbedit -b xml:/root/pdb-backup.xml –l.

- **-P account-policy**

Выводит политики учетной записи. Существующие политики:

- minimum password age — минимальное время жизни пароля;
- reset count minutes — сброс счетчика минут;
- disconnect time — время до разъединения;
- user must logon to change password — требовать смену пароля при следующем входе в систему;
- password history — история пароля;
- lockout duration — время блокировки при неудачном входе;
- min password length — мин длина пароля;
- maximum password age — максимальное время жизни пароля;
- bad lockout attempt — блокировка при неудачной попытке.
- Пример: pdbedit -P "bad lockout attempt"

- **-C account-policy-value**

Устанавливает для политики пользователя специфичное значение. Этот параметр используется совместно с ключом -P. Пример: pdbedit -P "bad lockout attempt" -C 3.

- **-y**

Если определен ключ -y, тогда ключи -i in-backend -e out-backend будут применены к политикам учетной записи альтернативной пользовательской базы. Этот параметр позволяет политикам учетной записи мигрировать из умолчального tdb хранилища в другие хранилища, например на сервер LDAP. Пример: pdbedit -y -i tdbsam: -e ldapsam:ldap://my.ldap.host

- **-h | --help**

Выводит строку со всеми возможными ключами.

- **-d | --debuglevel=level**

level целое число от 0 до 10. Значение по умолчанию не определено, то есть 0. Чем выше значение параметра, тем более подробный log-файл мы получим. При уровне 0, в log запишутся только критические ошибки и

серьезные предупреждения. Уровень 1 это разумный уровень для ежедневной работы — генерируется немного информации об осуществляемых операциях.

Уровни выше 1 будут создавать значительный объем информации в log-файл и должны использоваться при возникновении проблем. Уровни выше 3 используются разработчиками и создают огромный объем информации в лог файл, большинство из которой будет вам не понятно.

Заметьте что этот параметр переопределяет значение параметра log level в файле smb.conf.

- **-V**

  Выводит версию программы.

- **-s <configuration file>**

  Задает файл конфигурации для сервера. Этот файл содержит специфичную для сервера конфигурационную информацию, например какой использовать printcap-файл, а так же описание всех сервисов, предоставляемых сервером. Для получения более подробной информации см. файл smb.conf. Имя файла конфигурации по умолчанию определяется во время компиляции.

- **-l | --log-basename=logdirectory**

  Путь к директории для файлов log/debug. Могут использоваться расширения, представляющие собой < .имя_программы > (например: log.smbclient, log.smbd, и т.д…). Log-файл не может быть удален клиентом.

Замечание

Это команда может быть запущена только с правами root. Этот man написан для версии 3 пакета Samba.

Автор

Изначально Samba и сопутствующие утилиты были разработаны Эндрю Тридгеллом (Andrew Tridgell). Сейчас Samba разрабатывается Samba Team в качестве Open Source проекта – напоминает то, как разрабатывается ядро Linux.

ПРИЛОЖЕНИЯ К РАЗДЕЛУ 11

Приложение 11.1. Исходный конфигурационный файл Ftp-сервера /etc/proftpd/proftpd.conf

```
#
# /etc/proftpd/proftpd.conf -- This is a basic ProFTPD configuration file.
# To really apply changes reload proftpd after modifications.
#

# Includes DSO modules
Include /etc/proftpd/modules.conf

# Set off to disable IPv6 support which is annoying on IPv4 only boxes.
UseIPv6                 on
# If set on you can experience a longer connection delay in many cases.
IdentLookups            off

ServerName              "Debian"
ServerType              standalone
DeferWelcome            off

MultilineRFC2228        on
DefaultServer           on
ShowSymlinks            on

TimeoutNoTransfer       600
TimeoutStalled          600
TimeoutIdle             1200

DisplayLogin            welcome.msg
DisplayChdir            .message true
ListOptions             "-l"

DenyFilter               \*.*/
```

```
# Use this to jail all users in their homes
# DefaultRoot                ~

# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
# RequireValidShell          off

# Port 21 is the standard FTP port.
Port                    21

# In some cases you have to specify passive ports range to by-pass
# firewall limitations. Ephemeral ports can be used for that, but
# feel free to use a more narrow range.
# PassivePorts               49152 65534

# If your host was NATted, this option is useful in order to
# allow passive tranfers to work. You have to use your public
# address and opening the passive ports used on your firewall as well.
# MasqueradeAddress          1.2.3.4

# This is useful for masquerading address with dynamic IPs:
# refresh any configured MasqueradeAddress directives every 8 hours

# DynMasqRefresh 28800


# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances        30

# Set the user and group that the server normally runs at.
User                    proftpd
Group                   nogroup

# Umask 022 is a good standard umask to prevent new files and dirs
# (second parm) from being group and world writable.
Umask                   022  022
# Normally, we want files to be overwriteable.
AllowOverwrite      on
```

```
# Uncomment this if you are using NIS or LDAP via NSS to retrieve passwords:
# PersistentPasswd            off

# This is required to use both PAM-based authentication and local passwords
# AuthOrder                   mod_auth_pam.c* mod_auth_unix.c

# Be warned: use of this directive impacts CPU average load!
# Uncomment this if you like to see progress and transfer rate with ftpwho
# in downloads. That is not needed for uploads rates.
#
# UseSendFile                 off

TransferLog /var/log/proftpd/xferlog
SystemLog   /var/log/proftpd/proftpd.log

<IfModule mod_quotatab.c>
QuotaEngine off
</IfModule>

<IfModule mod_ratio.c>
Ratios off
</IfModule>


# Delay engine reduces impact of the so-called Timing Attack described in
# http://security.lss.hr/index.php?page=details&ID=LSS-2004-10-02
# It is on by default.
<IfModule mod_delay.c>
DelayEngine on
</IfModule>

<IfModule mod_ctrls.c>
ControlsEngine          off
ControlsMaxClients      2
ControlsLog             /var/log/proftpd/controls.log
ControlsInterval        5
ControlsSocket          /var/run/proftpd/proftpd.sock
</IfModule>

<IfModule mod_ctrls_admin.c>
AdminControlsEngine off
</IfModule>
```

```
#
# Alternative authentication frameworks
#
#Include /etc/proftpd/ldap.conf
#Include /etc/proftpd/sql.conf

#
# This is used for FTPS connections
#
#Include /etc/proftpd/tls.conf

# A basic anonymous configuration, no upload directories.

# <Anonymous ~ftp>
#   User                            ftp
#   Group                           nogroup
#   # We want clients to be able to login with "anonymous" as well as "ftp"
#   UserAlias                   anonymous ftp
#   # Cosmetic changes, all files belongs to ftp user
#   DirFakeUser on ftp
#   DirFakeGroup on ftp
#
#   RequireValidShell           off
#
#   # Limit the maximum number of anonymous logins
#   MaxClients                  10
#
#   # We want 'welcome.msg' displayed at login, and '.message' displayed
#   # in each newly chdired directory.
#   DisplayLogin                    welcome.msg
#   DisplayChdir                .message
#
#   # Limit WRITE everywhere in the anonymous chroot
#   <Directory *>
#     <Limit WRITE>
#       DenyAll
#     </Limit>
#   </Directory>
#
#   # Uncomment this if you're brave.
#   # <Directory incoming>
#   #   # Umask 022 is a good standard umask to prevent new files and dirs
```

```
#   #   # (second parm) from being group and world writable.
#   #   Umask                    022  022
#   #           <Limit READ WRITE>
#   #           DenyAll
#   #           </Limit>
#   #
#   #           AllowAll
#   #           </Limit>
#   # </Directory>
#
# </Anonymous>
```

Приложение 11.2. Главный конфигурационный файл Apache2

```
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.2/ for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# The configuration directives are grouped into three basic sections:
#  1. Directives that control the operation of the Apache server process as a
#     whole (the 'global environment').
#  2. Directives that define the parameters of the 'main' or 'default' server,
#     which responds to requests that aren't handled by a virtual host.
#     These directives also provide default values for the settings
#     of all virtual hosts.
#  3. Settings for virtual hosts, which allow Web requests to be sent to
#     different IP addresses or hostnames and have them handled by the
#     same Apache server process.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "/var/log/apache2/foo.log"
# with ServerRoot set to "" will be interpreted by the
# server as "//var/log/apache2/foo.log".
```

```
#

### Section 1: Global Environment
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.

# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the LockFile documentation
# (available  at <URL:
# http://httpd.apache.org/docs-2.1/mod/mpm_common.html#lockfile>);
#  you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/apache2"

# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
#<IfModule !mpm_winnt.c>
#<IfModule !mpm_netware.c>
LockFile /var/lock/apache2/accept.lock
#</IfModule>
#</IfModule>

# PidFile: The file in which the server should record its process
# identification number when it starts.
# This needs to be set in /etc/apache2/envvars

PidFile ${APACHE_PID_FILE}

# Timeout: The number of seconds before receives and sends time out.

Timeout 300

# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.

KeepAlive On
```

```
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.

MaxKeepAliveRequests 100

# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.

KeepAliveTimeout 15

## Server-Pool Size Regulation (MPM specific)

# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_worker_module>
    StartServers              2
    MinSpareThreads          25
    MaxSpareThreads          75
    ThreadLimit              64
    ThreadsPerChild          25
    MaxClients              150
    MaxRequestsPerChild       0
```

```
</IfModule>

# event MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_event_module>
    StartServers              2
    MaxClients              150
    MinSpareThreads          25
    MaxSpareThreads          75
    ThreadLimit              64
    ThreadsPerChild          25
    MaxRequestsPerChild       0
</IfModule>

# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives.  See also the AllowOverride
# directive.

AccessFileName .htaccess

# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy all
</Files>

# DefaultType is the default MIME type the server will use for a document
# if it cannot otherwise determine one, such as from filename extensions.
# If your server contains mostly text or HTML documents, "text/plain" is
# a good value.  If most of your content is binary, such as applications
# or images, you may want to use "application/octet-stream" instead to
```

```
# keep browsers from trying to display binary files as though they are
# text.

DefaultType text/plain

# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.

HostnameLookups Off

# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.

ErrorLog /var/log/apache2/error.log

# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.

LogLevel warn

# Include module configuration:
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf

# Include all the user configurations:
Include /etc/apache2/httpd.conf

# Include ports listing
Include /etc/apache2/ports.conf

# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
# If you are behind a reverse proxy, you might want
# to change %h into %{X-Forwarded-For}i
```

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Define an access log for VirtualHosts that don't define their own logfile
CustomLog /var/log/apache2/other_vhosts_access.log vhost_combined

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
Include /etc/apache2/conf.d/

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/
```

Приложение 11.3. Файл шаблона для виртуальных хостов

```
<VirtualHost *:80>
	ServerAdmin webmaster@localhost

	DocumentRoot /var/www
	<Directory />
		Options FollowSymLinks
		AllowOverride None
	</Directory>
	<Directory /var/www/>
		Options Indexes FollowSymLinks MultiViews
		AllowOverride None
		Order allow,deny
		allow from all
	</Directory>

	ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
	<Directory "/usr/lib/cgi-bin">
		AllowOverride None
		Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
		Order allow,deny
```

```
            Allow from all
        </Directory>

        ErrorLog /var/log/apache2/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog /var/log/apache2/access.log combined

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>

</VirtualHost>
```

ЛИТЕРАТУРА И ИНТЕРНЕТ-РЕСУРСЫ

Литература

1. *Колесниченко Д. Н.* Ubuntu Linux. Краткое руководство пользователя. — СПб.: БХВ-Петербург, 2007. — 304 с.: ил. + CD-ROM.
2. *Колесниченко Д. Н.* Ubuntu Linux. Краткое руководство пользователя. — СПб.: БХВ-Петербург, 2007. — 304 с.: ил. + CD-ROM.
3. *Колесниченко Д. Н.* Ubuntu 10. Краткое руководство пользователя. — СПб.: БХВ-Петербург, 2010. — 342 с.: ил. + CD-ROM.
4. *Колесниченко Д. Н.* Ubuntu 10. Библия пользователя. — М.: Изд-во Диалектика, Вильямс, 2010. – 592 с. + DVD-ROM.
5. *Комягин В. Б.* Ubuntu Linux 10.04. Русская версия. Серия: Официальный дистрибутив + учебный курс. — М.: Изд-во Триумф,, 2011. —208 с.: ил. + CD-ROM.
6. *Негус К., Каэн Ф.* Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд . — СПб.: Питер, 2011. — 354 с.
7. *Бусаргин М.* Linux Ubuntu. Секреты и настройки. — Интернет-издание, 2010. Формат PDF.
8. Ubuntu Linux: официальный учебный курс / Бенджамин Мако Хилл [и др.]. — М.: Изд-во Триумф, 2008. — 384 с.
9. *Робин Никсон.* Ubuntu для всех. — СПб.: БХВ-Петербург, Русская Редакция 2011. — 464 с.: ил. + DVD-ROM.
10. *Эви Немет, Гарт Снайдер, Трент Хейн, Бэн Уэйли.* Unix и Linux. Руководство системного администратора — М.: Изд-во Вильямс, 2012. — 1312 с.
11. *Резников Ф. А.* Ubuntu Server 2012-2015 + настольные ПК с Ubuntu в офисе. — М.: Изд-во Триумф, 2012. — 256 с. + DVD-ROM.
12. *Комягин В. Б.* Ubuntu Linux 11.04. Русская версия. — М.: Изд-во Триумф, 2012. — 208 с. + DVD-ROM.
13. 20 лучших бесплатных книг о Linux (http://rus-linux.net/nlib.php?name=/MyLDP/freesoft/FreeBooks/free20books.html).

Российские UBUNTU - порталы

- Ubuntu по-русски — http://ubuntu.ru/

- Всё об Ubuntu Linux — http://ubuntovod.ru/
- Убунтология (http://ubuntologia.ru)
- Ubuntu новости — http://ubuntunews.ru/
- Всё об Ubuntu — http://myubuntu.ru/
- Всё про Убунту — http://www.pro-ubuntu.ru/
- Сайт для пользователей Ubuntu Linux — http://ubuntulinux.ru/
- Новости Ubuntu Linux — http://ubuntu-news.ru/
- Установка игр в Ubuntu — http://ubuntu-wine.ru/

Интернет-ресурсы

Для начинающих пользователей-чайников (обучение)

- Убунтология (http://ubuntologia.ru/start-learning).
- Ubuntu для начинающих (http://startubuntu.ru).
- Ubuntu Master (http://ubuntumaster.ru).
- Ubuntu — Викиучебник (http://ru.wikibooks.org/wiki/Ubuntu).
- Руководство по Ubuntu для новичков (http://help.ubuntu.ru/manual/).
- Ubuntu для новичков (http://ubuntu-for-novices.blogspot.com /2007/10/2.html).
- Видеоуроки для Linux Ubuntu 10.04 (http://linux.panzins.ru/).
- www Видео: Install Ubuntu Desktop in Virtual PC (Part 1) и (Part 2) http://www.youtube.com/watch?v=PSxN0cg76Cc&feature=related.
- Видео: Linux Ubuntu and Virtual Windows 7 (http://www.youtube.com/watch?v=rwYoXy-Bxf8&feature=related").
- Видео: Начинаем Ubuntu в Windows (http://www.youtube.com/watch?v=duQ3mgP_Ssk).
- Видео: Введение в Ubuntu (http://www.youtube.com/watch?v=9K3Em2kFZqI&feature=related).
- Видео: Общий доступ папок Ubuntu в Windows (http://www.youtube.com/watch?v=UppITKeeoME).
- Ubuntu Quantal == http://ubuntuguide.org/wiki.

Для нечайников (документация и форумы)

- Официальная документация на сайте разработчика (https://help.ubuntu.com/).
- Русскоязычная документация — Wiki раздел (http://help.ubuntu.ru/wiki/).
- Русскоязычная документация про Ubuntu = http://help.ubuntu.ru/

- Практические руководства по Ubuntu Linux (http://ubuntueasy.com/about).
- Пингвинус -> Интернет и сети (http://pingvinus.ru/notes/network).
- Практика с Операционной Системой Ubuntu (http://ubuntual.com/).
- Ubuntu — Пакеты программ в «lucid», Подсекция net (http://packages.ubuntu.com/ru/lucid/i386/net/).
- Ubuntu — Список секций в «lucid» http://packages.ubuntu.com/ru/lucid/i386/).
- Англоязычный форум по Ubuntu (http://ubuntuforums.org/).
- Перезапуск, остановка и запуск X-сервера (http://help.ubuntu.ru/wiki/).
- Строка приглашения в Bash (http://ubuntologia.ru/bash-prompt-tuning).
- Создание учётной записи пользователя без пароля (http://help.ubuntu.ru/wiki/).
- Форум русскоязычного сообщества Ubuntu — http://forum.ubuntu.ru/
- Форум программистов и сисадминов — http://www.cyberforum.ru/ubuntu-linux/
- Тесты и экзамены Ubuntu — http://www.eureca.ru/edu/study/index.php?goto=list&vendor=ubuntu&type=exam.
- Ubuntu Certification — http://www.ubuntu.com/news/ubuntucert.
- Ubuntu Уровень 1 экзамен uCertify 117-199 — http://ru.downv.com/download-uCertify-117-199-Ubuntu-%D3%F0%EE%E2%E5%ED%FC-1-%FD%EA%E7%E0%EC%E5%ED-10484656.htm.