ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

АДМИНИСТРИРОВАНИЕ СЕРВЕРОВ

Методические рекомендации к лабораторным работам для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника» дневной формы обучения

Могилев 2018

УДК 004.383.2
ББК 32.973.202-04
А 31

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления» «04» сентября 2018 г., протокол № 2

Составители: доц. А. И. Якимов;
ст. преподаватель В. Т. Садовский

Рецензент О. В. Леоненко

Методические рекомендации предназначены для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника» дневной формы обучения.

Учебно-методическое издание

АДМИНИСТРИРОВАНИЕ СЕРВЕРОВ

Ответственный за выпуск	А. И. Якимов
Технический редактор	С. Н. Красовская
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

Содержание

Введение

Целями преподавания дисциплины «Администрирование серверов» являются изучение основ теории и получение практических навыков сетевого администрирования информационной системы предприятия – управления сетевыми устройствами, сетевыми протоколами, сетевыми операционными системами, службами каталогов, сетевыми службами, управления файловыми ресурсами системы, правами доступа к ресурсам, устройствами печати, системами резервного копирования и восстановления информации, осуществления мониторинга сетевых устройств и служб, с применением сетевых операционных серверных систем Windows, Linux Server.

1 Лабораторная работа № 1. Настройка сетевой подсистемы Windows Server 2012

Цель работы: ознакомиться с основными аппаратными средствами и оборудованием локальной вычислительной сети.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования операций над подмножествами заданного универсума.
4 Выводы.

Основные теоретические положения

Рабочая станция под управлением пользовательской операционной системы, как правило, может поддерживать: выполнение нескольких процессов, создавать, хранить и обновлять список конфигурации компьютера, средства доступа в Internet, службу сообщений, службу локальной безопасности и защиты файлов, папок и других локальных ресурсов компьютера, надежность функционирования приложений в операционной системе (каждое приложение выполняется в отдельном адресном пространстве).

Серверная операционная система, например Windows Server, оптимизирована для работы в качестве сервера файлов, печати, а также для приложений с широким спектром применений: от администрирования нескольких рабочих групп до корпоративных сетей. Основными функциями операционной системы сервера являются: поддержка многопроцессорной обработки задач, администрирование сервера и сети, отслеживание входящего и исходящего трафика сервера, поддержка Web-сервера, интеграция с клиентами других фирм производителей, например Macintosh и др.

Чтобы получить доступ к ресурсам, пользователям необходимо прежде всего зарегистрироваться – идентифицировать себя в домене или компьютере. При этом необходимо ввести имя пользователя, пароль, а также название домена, в котором зарегистрирована учетная запись или название компьютера. Окно, в котором происходит регистрация пользователя, раскрывается при

загрузке операционной системы или при нажатии кнопок Ctrl-Alt-Delete.

В Windows Server глобальную запись можно создать средствами User Manager for Domain (Диспетчер пользователей доменов). Она размещается в основной базе данных каталогов на главном контроллере домена PDC (Primary domain controller). Копии базы данных хранятся на всех резервных контроллерах домена BDC (Backup domain controller), которые с интервалом в 5 мин обновляются с основного контроллера домена.

Локальная учетная запись содержит информацию о пользователе данного компьютера. С ее помощью пользователь может зарегистрироваться в системе и получить доступ к ресурсам компьютера. Чтобы иметь право обратиться к ресурсам другого компьютера, надо и на нем завести локальную учетную запись пользователя.

Программную настройку компьютера выполняет Пользователь, который обладает соответствующими правами на конфигурирование системы. Такими правами, как правило, обладает пользователь из группы «Администратор». Настроить сетевые установки можно путем нажатия правой кнопки мыши на значке «Мое сетевое окружение», которое, как правило, располагается на Рабочем столе операционной системы, и выбрать пункт меню «Свойства». При этом откроется окно «Сеть и удаленный доступ к сети».

Для того чтобы раскрыть окно «Подключения по локальной сети – свойства», в котором и настраиваются параметры подключения, необходимо правой кнопкой мыши нажать на значке «Подключение по локальной сети». В этом окне необходимо установить протокол передачи данных, службу доступа к информации по сети, а также указать, клиентом каких сетей является пользователь. Для выбора протокола передачи данных по сети необходимо в открывшемся окне нажать на кнопку «Установить», а затем в новом окно выбрать «Протокол», нажать «Добавить». Раскроется список доступных для установки протоколов. Выбирают, например, протокол передачи данных TCP/IP, для функционирования которого необходимо установить в свойствах данного протокола уникальный для каждого компьютера сети IP-адрес (например, 192.168.0.33) и маску подсети (например, 255.255.0.0).

Кроме того, чтобы получить возможность передавать данные по сети, а также иметь доступ к ресурсам другого компьютера, необходимо также установить, что пользователь является клиентом сети Microsoft, также службу доступа к файлам и принтерам сетей Microsoft. Для этого необходимо в окне «Подключения по локальной сети – свойства» выбрать «Установить», затем в открывшемся окне выбрать «Клиент», а затем из списка выбрать «Клиент для сетей Microsoft». Служба доступа к файлам и принтерам сетей Microsoft устанавливается аналогичным образом, только в окне «Выбор типа сетевого компонента» выбрать «Служба» и далее в открывшемся окне выбрать «Служба доступа к файлам и принтерам сетей Microsoft».

Контрольные вопросы

1 Для чего нужны сетевые операционные системы?
2 По каким основным признакам можно классифицировать ОС?
3 Для чего необходимы службы удаленного вызова процедур и сетевой динамический обмен данными?

2 Лабораторная работа № 2. Создание учетных записей пользователей и управление ими

Цель работы: изучить алгоритм администрирования пользователей и рабочих групп.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

В сетевой операционной системе Windows Server присутствует специальный инструмент, предназначенный для администрирования глобальных учетных записей пользователей и групп на основном контроллере домена, а также локальные учетные записи на любом компьютере домена – Active Directory Users and Computers.

Для того чтобы создать учетную запись нового пользователя в домене, необходимо в меню User выбрать «New User...». При этом появляются два последовательных окна (Password) и подтверждение пароля (Confirm Password) (рисунок 2.1).

Кроме того, в этом окне можно задать смену пароля при первой регистрации пользователя (User Must Change Password at Next Logon), запретить смену пользователем пароля (User Cannot Change Password), ограничение действия пароля (Password Never Expires), отключить учетную запись (Account Disabled).

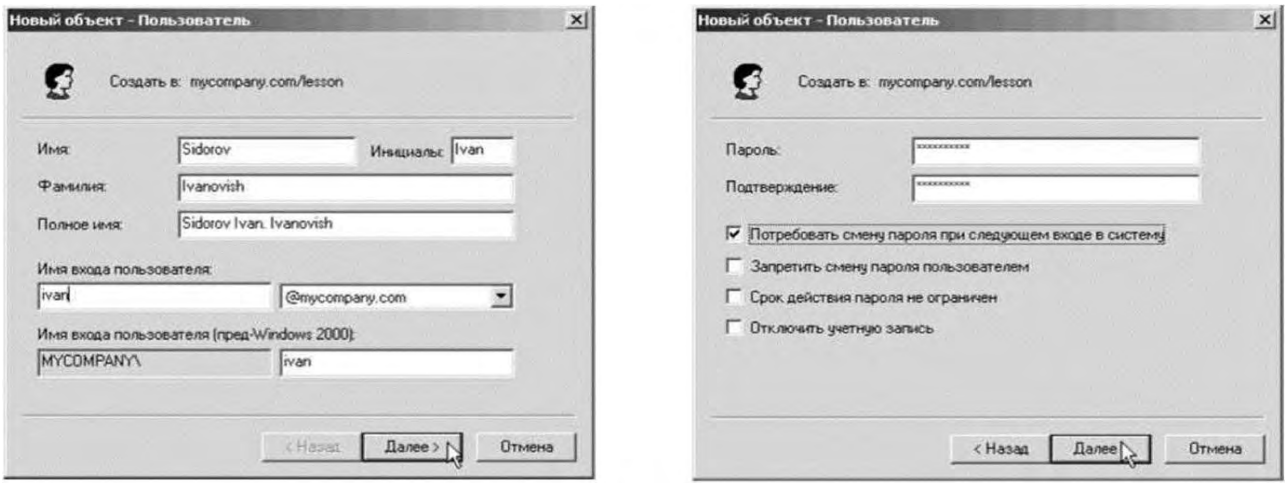

Рисунок 2.1 – Окно Новый объект: Пользователь

Контрольные вопросы

1 Опишите два основных подхода к построению ОС.

2 Каким образом обеспечивается взаимодействие подсистем с исполнительной системой?

3 В чем основное различие одноранговых и двухранговых классов сетей?

3 Лабораторная работа № 3. Настройка серверов DNS, DHCP и WINS. Маршрутизация

Цель работы: изучить базовые понятия, настройку параметров протокола TCP/IP; процесс установки службы DNS, создания зон прямого просмотра, настройку параметров регистрации узлов на сервере DNS; приобрести навыки применения диагностических утилит для поиска неисправностей и неверных конфигураций протокола TCP/IP и службы DNS.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

Требования к отчету

1 Цель работы.

2 Постановка задачи.

3 Результаты исследования.

4 Выводы.

Основные теоретические положения

В сетях TCP/IP принято различать адреса сетевых узлов трех уровней:

1) физический (или локальный) адрес узла (MAC-адрес сетевого адаптера или порта маршрутизатора); эти адреса назначаются производителями сетевого оборудования;

2) IP-адрес узла (например, 192.168.0.1), данные адреса назначаются сетевыми администраторами или Интернет-провайдерами;

3) символьное имя (например, www.microsoft.com); эти имена также назначаются сетевыми администраторами компаний или Интернет-провайдерами.

Компьютеры или другие сложные сетевые устройства, подсоединенные к нескольким физическим сетям, имеют несколько IP-адресов – по одному на каждый сетевой интерфейс. Схема адресации определяет три типа IP-адресов и позволяет проводить единичную, широковещательную и групповую адресацию. Таким образом, выделяют три типа IP-адрсов:

1) Unicast-адрес (единичная адресация конкретному узлу) – используется в коммуникациях «один-к-одному»;

2) Broadcast-адрес (широковещательный адрес, относящийся ко всем адресам подсети) – используется в коммуникациях «один-ко-всем». В этих адресах поле идентификатора устройства заполнено единицами. IP-адресация допускает широковещательную передачу, но не гарантирует ее – эта возможность зависит от конкретной физической сети. Например, в сетях Ethernet широковещательная передача выполняется с той же эффективностью, что и обычная передача данных, но есть сети, которые вообще не поддерживают такой тип передачи или поддерживают весьма ограничено;

3) Multicast-адрес (групповой адрес для многоадресной отправки пакетов) – используется в коммуникациях «один-ко-многим». Поддержка групповой адресации используется во многих приложениях, например, приложениях интерактивных конференций. Для групповой передачи рабочие станции и маршрутизаторы используют протокол IGMP, который предоставляет информацию о принадлежности устройств определенным группам.

Каждый сетевой интерфейс на каждом узле сети должен иметь уникальный unicast-адрес. IP-адрес имеет длину 4 байта (или 32 бита). Для удобства чтения адресов 32-битные числа разбивают на октеты по 8 бит, каждый октет переводят в десятичную систему счисления и при записи разделяют точками. Например, IP-адрес 11000000101010000000000000000001 записывается как 192.168.0.1.

IP-адрес состоит из двух частей – идентификатор сети (префикс сети, Network ID) и идентификатор узла (номер устройства, Host ID). Такая схема приводит к двухуровневой адресной иерархии.

Идентификатор сети идентифицирует все узлы, расположенные на одном физическом или логическом сегменте сети, ограниченном IP-маршрутизаторами. Все узлы, находящиеся в одном сегменте, должны иметь одинаковый идентификатор сети.

Идентификатор узла идентифицирует конкретный сетевой узел (сетевой

адаптер рабочей станции или сервера, порт маршрутизатора). Идентификатор узла должен быть уникален для каждого узла внутри IP-сети, имеющей один идентификатор сети.

Таким образом, в целом IP-адрес будет уникален для каждого сетевого интерфейса всей сети TCP/IP.

Соотношение между идентификатором сети и идентификатором узла в IP-адресе определяется с помощью маски подсети (Network mask), которая имеет длину также 4 байта и также записывается в десятичной форме по 4 октета, разделенных точками. Старшие биты маски подсети, состоящие из 1, определяют, какие разряды IP-адреса относятся к идентификатору сети. Младшие биты маски, состоящие из 0, определяют, какие разряды IP-адреса относятся к идентификатору узла.

IP-адрес и маска подсети – минимальный набор параметров для конфигурирования протокола TCP/IP на сетевом узле.

Правила назначения идентификаторов сети (Network ID):

– первый октет идентификатора сети не может быть равен 127 (адреса вида 127.x.y.z предназначены для отправки узлом пакетов самому себе и используются как правило для отладки сетевых приложений, такие адреса называются loopback-адресами, или адресами обратной связи);

– все разряды идентификатора сети не могут состоять из одних 1 (IP-адреса, все биты которых установлены в 1, используются при широковещательной передаче информации);

– все разряды идентификатора сети не могут состоять из одних 0 (в IP-адресах все биты, установленные в ноль, соответствуют либо данному устройству, либо данной сети);

– идентификатор каждой конкретной сети должен быть уникальным среди подсетей, объединенных в одну сеть с помощью маршрутизаторов.

Правила назначения идентификаторов узла (Host ID):

– все разряды идентификатора узла не могут состоять из одних 1 (идентификатор узла, состоящий из одних 1, используется для широковещательных адресов, или broadcast-адресов);

– все разряды идентификатора сети не могут состоять из одних 0 (если разряды идентификатора узла равны 0, то такой адрес обозначает всю подсеть, например, адрес 192.168.1.0 с маской подсети 255.255.255.0 обозначает всю подсеть с идентификатором сети «192.168.1»;

– идентификатор узла должен быть уникальным среди узлов одной подсети. Другим способом обозначения сети, более удобным и более кратким, является обозначение сети с сетевым префиксом. Такое обозначение имеет вид «/число бит маски подсети». Например, подсеть 192.168.1.0 с маской подсети 255.255.255.0 можно более кратко записать в виде 192.168.1.0/24, где число 24 длина маски подсети в битах.

Контрольные вопросы

1 Какие протоколы входят в стек протоколов TCP/IP?

2 Из каких частей состоит IP-адрес сетевого узла?

3 Каким образом влияет маска подсети на взаимодействие между узлами сети?

4 Назначение службы DNS.

5 Взаимосвязь между понятиями домена и зоны.

6 Назначение основной и дополнительной зоны.

7 Назначение зон прямого и обратного просмотра.

8 Как происходит процесс разрешения имен службой DNS с помощью итеративных и рекурсивных запросов?

4 Лабораторная работа № 4. Отслеживание проблем безопасности в установленном ПО

Цель работы: научиться устанавливать операционную систему Windows на компьютер.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

Требования к отчету

1 Цель работы.

2 Постановка задачи.

3 Результаты исследования.

4 Выводы.

Основные теоретические положения

Системная служба **Журнал событий** запускается по умолчанию при загрузке операционной системы и регистрирует события в трех журналах (в зависимости от роли сервера журналов может быть больше):

1) *Приложение* (содержит информацию об изменении конфигурации в системе);

2) *Система* (содержит данные о системных событиях);

3) *Безопасность* (содержит записи о событиях входа в систему и о доступе к ресурсам).

Для мониторинга и оптимизации работы компьютера в системах **Windows Server 2003** имеется несколько инструментов, позволяющих администратору следить за работой любых компонентов системы и конфигурировать ее оптимальным образом. Эти инструменты перечислены ниже.

***Task Manager* (*Диспетчер задач*)** служит для просмотра текущих данных о производительности системы. В этой утилите основными являются три индикатора: использование процессора, использование виртуальной памяти и запущенные процессы и программы.

Оснастка Event Viewer* (*Просмотр событий*)** позволяет просматривать журналы событий, генерируемых приложениями, службой безопасности и ***системой.

Performance* (*Производительность*)** – обновленная оснастка систем **Windows XP** и **Windows Server 2003**, аналог утилиты ***Performance Monitor в **Windows NT 4.0.** Оснастка ***Performance***включает в себя два компонента: *ActiveX-элемент System Monitor* и оснастку *Performance Logs and Alerts* (Оповещения и журналы безопасности). Графические средства *System Monitor* позволяют визуально отслеживать изменение производительности системы. С помощью *System Monitor* можно одновременно просматривать данные с нескольких компьютеров в виде динамических диаграмм, на которых отображается текущее состояние системы и показания счетчиков. Оснастка *Performance Logs and Alerts* позволяет создавать отчеты на основе текущих данных производительности или информации из журналов. При превышении счетчиками заданного значения или уменьшении ниже указанного уровня данная оснастка посредством службы сообщений (Messenger) посылает оповещения пользователю.

Диспетчер задач можно использовать для отслеживания ключевых индикаторов производительности компьютера, он позволяет определять статус запущенных программ и завершать приложения, которые перестали отвечать на запросы системы. С помощью диспетчера задач можно отслеживать активность запущенных процессов по 25 параметрам и просматривать графики использования процессора и памяти.

В **Windows Server 2003** диспетчер задач содержит пять вкладок/индикаторов. Ниже перечислены эти вкладки и указано их назначение.

Applications (*Приложения*) – показывает статус приложений, запущенных на компьютере. *Processes* (*Процессы*) – содержит информацию о процессах, запущенных на компьютере.

Performance (*Быстродействие*) – отображает динамическое состояние производительности компьютера, включая степень использования памяти и процессора.

Networking (*Сеть*) – показывает степень загрузки сети. Индикатор отображается только при наличии на компьютере сетевой карты.

Users (*Пользователи*) – содержит список зарегистрированных пользователей. Эти пользователи могут регистрироваться локально (из консоли) или являться клиентами служб *Terminal Services*, подключенных с использованием технологий *Terminal Server, Remote Access* или *Remote Assistant*.

В операционных системах **Windows** событием называется любое значительное «происшествие» в работе системы или приложения, о котором следует уведомить пользователей. В случае возникновения критических событий, таких как переполнение диска сервера или неполадки с электропитанием, на экран монитора будет выведено соответствующее сообщение. Остальные события, которые не требуют немедленных действий от пользователя, регистрируются в системных

журналах. Служба регистрации событий в системных журналах активизируется автоматически при каждом запуске системы **Windows Server 2003**.

Контрольные вопросы

1 Назовите порядок выполнения просмотра событий.
2 Назовите порядок выполнения мониторинга сетевых подключений.
3 Порядок отключения пользователя с отправкой ему уведомления.
4 Порядок просмотра сетевых подключений к компьютеру.

5 Лабораторная работа № 5. Создание и выполнение командных файлов в пользовательской среде ОС Windows

Цель работы: изучить основные команды командной оболочки операционной системы WINDOWS (в дальнейшем ОС), освоить программирование и создание простых пакетных (процедурных) файлов.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

Командная оболочка – это отдельный программный продукт, который обеспечивает прямую связь между пользователем и операционной системой. Текстовый пользовательский интерфейс командной строки предоставляет среду, в которой выполняются приложения и служебные программы с текстовым интерфейсом. Результат выполнения приложений в командной оболочке отображается на экране в виде, сходном с интерпретатором Command.com MS-DOS. Командная оболочка Windows XP использует интерпретатор команд Cmd.exe, который загружает приложения и направляет поток данных между приложениями, для перевода введенной команды

в понятный системе вид.

Имеется возможность использовать командную оболочку для создания и редактирования пакетных файлов (также называемых сценариями), что позволит автоматизировать выполнение обычных задач. Например, можно использовать сценарии для автоматизации управления учетными записями пользователей и ежедневной архивацией в нерабочие часы. Также можно использовать сервер сценариев Windows, CScript.exe для выполнения в командной оболочке сложных сценариев. Выполнение операций с помощью пакетных файлов является более эффективным, чем с помощью интерфейса пользователя. Пакетные файлы принимают все команды, доступные из командной строки.

Имеется возможность настроить окно командной строки для облегчения просмотра и для увеличения контроля за выполнением программ.

Синтаксическая структура выводится в том порядке, в котором следует вводить команду и следующие за ней параметры, если они есть. Следующий пример команды **xcopy** иллюстрирует разнообразие синтаксических форматов текста:

xcopy *источник* [*результат*] [**/w**] [**/p**] [**/c**] [**/v**] [**/q**] [**/f**] [**/l**] [**/g**] [**/d**[**:***мм-дд-гггг*]] [**/u**] [**/i**] [**/s** [**/e**]] [**/t**] [**/k**] [**/r**] [**/h**] [{**/a**|**/m**}] [**/n**] [**/o**] [**/x**] [**/exclude:***файл1*[+[*файл2*]][+[*файл3*]] [{**/y**|**/-y**}] [**/z**].

Можно выполнять несколько команд из одной командной строки или сценария с помощью символов условной обработки. При использовании нескольких команд, содержащих символы условной обработки, выполнение команд, стоящих справа от символа условной обработки, будет проводиться в зависимости от результатов выполнения команды, стоящей слева от символа. Например, требуется, чтобы команда выполнялась, только если предыдущая команда не была выполнена успешно. Или требуется, чтобы команда выполнялась, только если предыдущая команда была выполнена успешно.

Имеется возможность вкладывать командные оболочки в Cmd.exe, открывая новый экземпляр Cmd.exe из командной строки. По умолчанию каждый экземпляр Cmd.exe наследует среду своего родительского приложения Cmd.exe. Вложение экземпляров Cmd.exe позволяет вносить в локальную среду изменения, которые не повлияют на родительское приложение Cmd.exe. Это позволяет сохранять исходную среду Cmd.exe и возвращаться к ней после удаления вложенной командной оболочки. Изменения вложенной командной оболочки не сохраняются.

Чтобы создать вложенную командную оболочку, в командной строке вводится: **cmd.**

Появится сообщение следующего вида:

Microsoft (R) Windows XP (TM)
(C) Copyright 1985-2001 Microsoft Corp.

Чтобы закрыть все вложенные командные оболочки, вводится команда **exit**.

Можно локализовать изменения в экземпляре Cmd.exe (или в сценарии)

с помощью команд **setlocal** и **endlocal**. Команда **setlocal** создает локальную область, а **endlocal** ее удаляет. Любые изменения, сделанные внутри области, созданной командами **setlocal** и **endlocal**, не учитываются; таким образом, исходная среда остается без изменений. С помощью этих команд можно создать до 32 вложенных областей.

Среда командной оболочки Cmd.exe определяется переменными, задающими поведение командной оболочки и операционной системы. Имеется возможность определить поведение среды командной оболочки или среды всей операционной системы с помощью двух типов переменных среды: системных и локальных. Системные переменные среды определяют поведение глобальной среды операционной системы. Локальные переменные среды определяют поведение среды в данном экземпляре Cmd.exe.

Системные переменные среды заданы заранее в операционной системе и доступны для всех процессов Windows XP. Только пользователи с привилегиями администратора могут изменять эти переменные. Эти переменные наиболее часто используются в сценариях входа в систему.

Локальные переменные среды доступны, только когда пользователь, для которого они были созданы, вошел в систему. Локальные переменные из куста **HKEY_CURRENT_USER** подходят только для текущего пользователя, но определяют поведение глобальной среды операционной системы.

Контрольные вопросы

1 Понятие текущей директории и текущего диска. Команды изменения текущей директории и текущего диска.

2 Строка приглашения командного режима. Команда смены строки приглашения.

3 Команда создания и удаления директорий.

4 Команды переименования и копирования файлов.

5 Команды организации «среды выполнения» (set).

6 Процедурные файлы. Назначение и основные правила создания.

7 Основные команды для создания процедурных файлов.

8 Команды dir, more. Их параметры и варианты применения.

9 Атрибуты файлов. Их назначение и использование.

10 Принцип поиска внешней команды (программы) командной оболочкой. Команда set и path.

6 Лабораторная работа № 6. Удаленный доступ в Windows

Цель работы: научиться выполнять удаленное администрирование компьютеров.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

Для выполнения административных задач на компьютерах сети часто используется специальное программное обеспечение удаленного управления.

Подключение к удаленному рабочему столу – это технология, позволяющая пользователю, работающему за своим компьютером, связываться с удаленным компьютером, находящимся в другом месте. Например, можно подключиться к рабочему компьютеру из домашнего компьютера и получить доступ ко всем программам, файлам и сетевым ресурсам. Можно оставить программы выполняться на рабочем компьютере, а дома вывести на экран рабочий стол рабочего компьютера и продолжить работу с выполняющимися программами.

Удаленный рабочий стол работает по протоколу **RDP** (англ. *Remote Desktop Protocol*, протокол удалённого рабочего стола) – протокол прикладного уровня, использующийся для обеспечения удалённой работы пользователя с сервером, на котором запущен сервис терминальных подключений. Клиенты существуют практически для всех версий **Windows** (включая **Windows CE** и **Mobile**), **Linux, Free BSD, Mac OS X.** По умолчанию используется порт **TCP 3389**. Официальное название **Майкрософт** для клиентского программного обеспечения – **Remote Desktop Connection** или **Terminal Services Client (TSC)**, в частности, клиент в **Windows XP/2003/vista** называется ***mstsc.exe***.

***Virtual Network Computing* (*VNC*) –** система удалённого доступа к рабочему столу компьютера, использующая протокол **RFB** (*Remote FrameBuffer*). Управление осуществляется путём передачи нажатий клавиш на клавиатуре и движений мыши с одного компьютера на другой и ретрансляции содержимого

экрана через компьютерную сеть.

Система **VNC** платформа независима: **VNC**-клиент, называемый **VNC viewer**, запущенный на одной операционной системе, может подключаться к **VNC**-серверу, работающему на любой другой ОС. Существуют реализации клиентской и серверной части практически для всех операционных систем, в том числе и для Java. К одному **VNC**-серверу одновременно могут подключаться множественные клиенты. Наиболее популярные способы использования **VNC –** удалённая техническая поддержка и доступ к рабочему компьютеру из дома.

VNC была разработана компанией **AT&T**. Оригинальные исходные коды доступны на условиях лицензии **GNU General Public License**, как и многие варианты **VNC**, существующие на данный момент.

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.

VNC – очень простой протокол, основанный 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 **VNC** 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.

По умолчанию **VNC** использует диапазон 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**. Порты могут быть изменены. 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**.

Существует большое количество программ удаленного управления рабочим столом, основанных на **VNC**. Некоторые из них: **TightVNC**, **RealVNC**, **UltraVNC**, **TridiaVNC**, **Radmin.**

Контрольные вопросы

1 Что такое Подключение к удаленному рабочему столу?
2 По какому протоколу работает удаленный рабочий стол?
3 Что такое **Virtual Network Computing**?

7 Лабораторная работа № 7. Установка и настройка системы Linux/FreeBSD

Цель работы: освоить приемы установки и настройки операционной системы Linux/FreeBSD.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

В настоящее время все большее распространение приобретают операционные системы семейства Unix. В России Unix-подобные системы являются платформой для большинства брэндмауэров и Web-серверов.

Распространение Unix среди пользователей связано с бурным развитием сети Internet и проникновением ее во все сферы деятельности.

Наиболее распространенные версии Unix – FreeBSD, Linux, Sun Solaris, OpenBSD.

Для проведения исследований операционная система Unix (FreeBSD) устанавливается на виртуальную машину Virtual Box в следующей последовательности.

Запуск Oracle VM VirtualBox Менеджер – производится путем выбора соответствующего пункта в меню «Пуск» / «Программы» (рисунок 7.1).

После запуска Virtual Box необходимо создать новую виртуальную машину, на которую в последующем устанавливается операционная система FreeBSD. Создание новой машины производится путем нажатия на кноп-

ку «Создать…» в главном окне программы, которая запускает Мастер создания и добавления виртуальных машин в Virtual Box. Следует нажать на кнопку «Вперед».

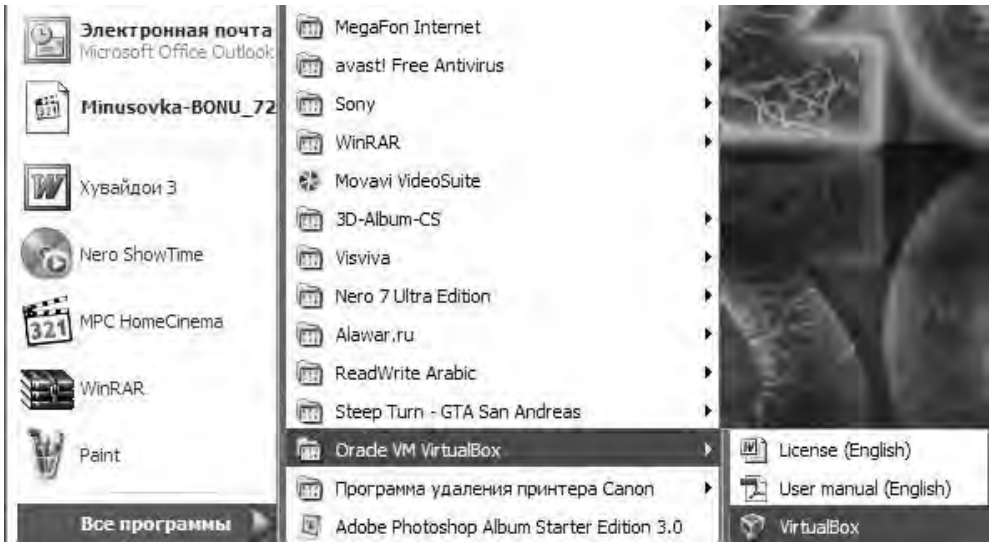

Рисунок 7.1 – Запуск Oracle VM VirtualBox Менеджер

На следующем шаге вводится название для виртуальной машины, и необходимо из списка выбрать тип устанавливаемой операционной системы (рисунок 7.2).

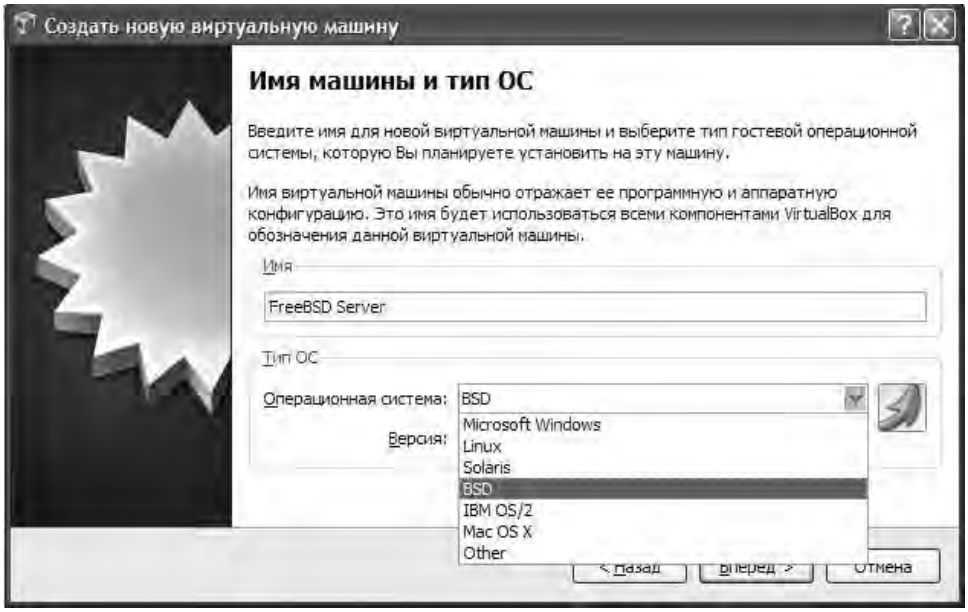

Рисунок 7.2 – Ввод названия для виртуальной машины

На следующем шаге предлагается указать размер оперативной памяти виртуальной машины либо путем принятия параметров по умолчанию, либо самостоятельно (рисунок 7.3), создается виртуальный жесткий диск для машины.

После нажатия на кнопку «Вперед» работа Мастера по созданию виртуальной машины завершается.

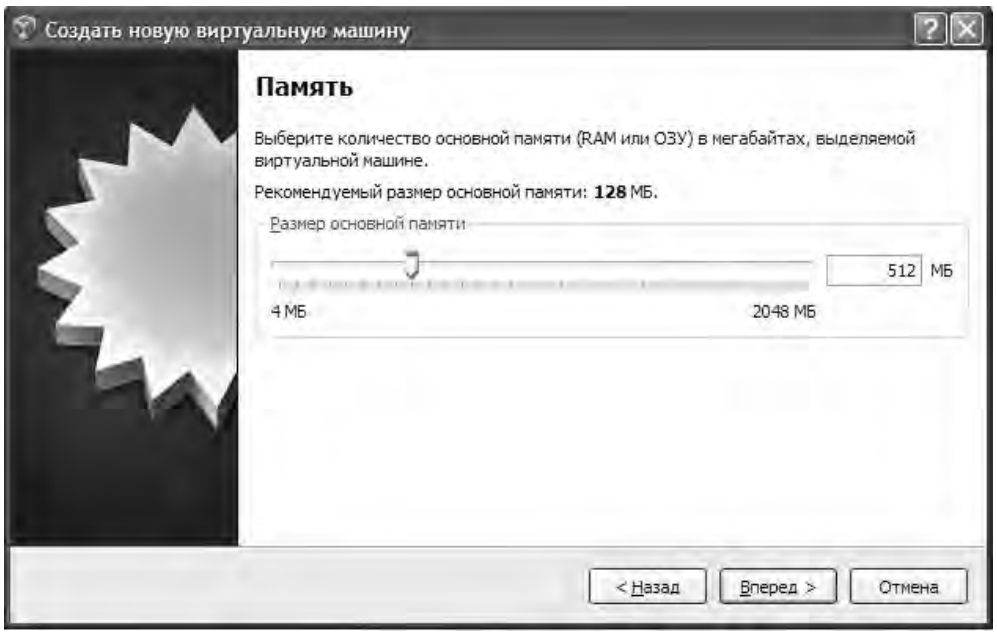

Рисунок 7.3 – Указание размера оперативной памяти

Запуск требуемой виртуальной машины из списка доступных в Virtual Box осуществляется путем ее выбора в списке и нажатия на кнопку «Старт».

Мастер первого запуска помогает осуществить выбор операционной системы на данную машину (рисунок 7.4).

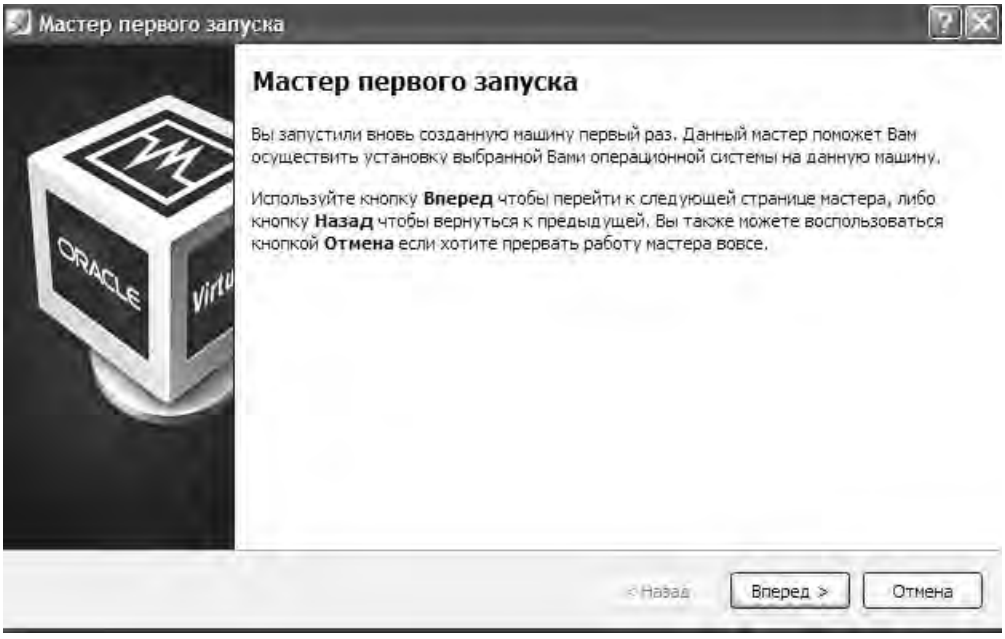

Рисунок 7.4 – Мастер первого запуска

После перехода к следующему шагу Мастер предлагает указать установочный носитель. Выбирается носитель, который содержит программу установки операционной системы. Этот носитель должен быть загрузочным, иначе программа установки не сможет начать работу.

После нажатия на кнопку «Вперед» работа Мастера завершается.

Работа с программой sysinstall (рисунок 7.5) осуществляется с помощью клавиатуры. Нажатие на кнопку производится с использованием клавиши «Enter».

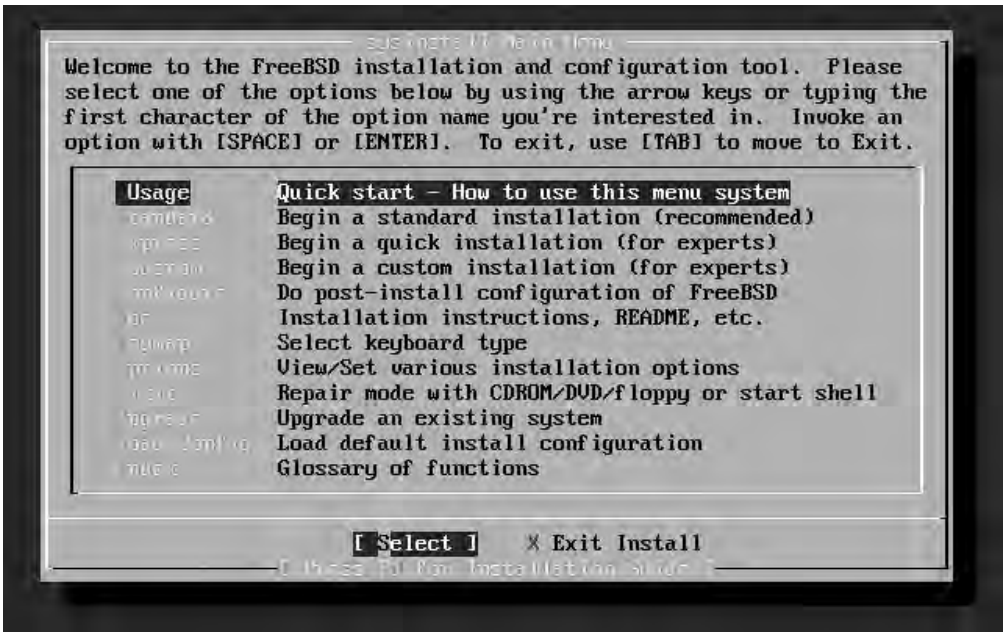

Рисунок 7.5 – Главное меню программы sysinstall

В главном меню программы можно выбрать различные варианты установки системы FreeBSD, модифицировать уже установленную ранее систему, а также получить справку по установке FreeBSD. В большинстве случаев подходит вариант «Standart».

Контрольные вопросы

1 Порядок установки виртуальной машины.
2 Порядок установки операционной системы FreeBSD.

8 Лабораторная работа № 8. Установка программ в Linux/FreeBSD

Цель работы: ознакомиться с программными продуктами для виртуализации, научиться устанавливать на виртуальную машину различные ОС и получить навыки их настройки.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

Виртуализация – это изоляция вычислительных процессов и ресурсов друг от друга. Это новый виртуальный взгляд на ресурсы составных частей, не ограниченных реализацией, физической конфигурацией или географическим положением. Обычно виртуализированные ресурсы включают в себя вычислительные мощности и хранилище данных. В широком смысле, понятие виртуализации представляет собой сокрытие настоящей реализации какого-либо процесса или объекта от истинного его представления для того, кто им пользуется. В компьютерных технологиях под термином **«виртуализация»** обычно понимается абстракция вычислительных ресурсов и предоставление пользователю системы, которая «инкапсулирует» (скрывает в себе) собственную реализацию. Проще говоря, пользователь работает с удобным для себя представлением объекта, и для него не имеет значения, как объект устроен в действительности.

Сам термин **«виртуализация»** в компьютерных технологиях появился в шестидесятых годах прошлого века вместе с термином **«виртуальная машина»**, означающим *продукт виртуализации программно-аппаратной платформы*.

Понятие виртуализации условно можно разделить на две фундаментально различающиеся категории:

1) виртуализация платформ. Продуктом этого вида виртуализации являются виртуальные машины – программные абстракции, запускаемые на платформе реальных аппаратно-программных систем;

2) виртуализация ресурсов. Данный вид виртуализации преследует своей целью комбинирование или упрощение представления аппаратных ресурсов для пользователя и получение неких пользовательских абстракций оборудования, пространств имен, сетей и т. п.

Под виртуализацией платформ понимают создание программных систем на основе существующих аппаратно-программных комплексов, зависящих или независящих от них. Система, предоставляющая аппаратные ресурсы и программное обеспечение, называется хостовой (host), а симулируемые ей системы – гостевыми (guest). Чтобы гостевые системы могли стабильно функционировать на платформе хостовой системы, необходимо, чтобы программное и аппаратное обеспечение хоста было достаточно надежным и предоставляло необходимый набор интерфейсов для доступа к его ресурсам.

Виртуальная машина (virtual machine) – программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы (target – целевая, или гостевая платформа) и исполняющая программы для target-платформы на host-платформе (host – хост-платформа, платформа-хозяин); или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы (песочница, sandbox).

Oracle VirtualBox – кроссплатформенный свободный (GNU GPL) программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, Mac OS X, Solaris/OpenSolaris, ReactOS, DOS и др. Поддерживаются как 32-битные, так и 64-битные версии ОС.

VMware Workstation – позволяет создавать и запускать одновременно несколько виртуальных машин (x86-архитектуры), в каждой из которых работает своя гостевая операционная система. Поддерживаются как 32-битные, так и 64-битные версии ОС.

VMware Player – бесплатный программный продукт, предназначенный для создания и запуска готовых виртуальных машин (созданных в VMware Workstation, либо VMware Server). Бесплатное решение с ограниченным, по сравнению с VMware Workstation, функционалом.

Microsoft Virtual PC – программный пакет виртуализации для операционной системы Windows.

Контрольные вопросы

1 Что такое операционная система? Назовите основные компоненты ОС.
2 Дайте определение понятию виртуализации.
3 Какие есть виды виртуализации? Охарактеризуйте каждый вид.
4 На какие виды подразделяется виртуализация платформ?
5 Что такое аппаратная виртуализация?
6 Что такое «виртуальная машина»? Назначение виртуальной машины.
7 Что такое хост-платформа?
8 Дайте определение гостевой ОС.
9 Дайте определение понятию песочницы («sandbox»).
10 Какие продукты для виртуализации Вы знаете?
11 Можно ли запустить несколько гостевых ОС на одном хосте?
12 Какие системы относятся к Unix, а какие системы относятся к Unix-подобным?
13 Что означает GNU GPL?
14 Какие системы относятся к Windows-подобным?
15 Расскажите про файловую структуру Unix-подобных систем.
16 Расскажите про файловую структуру Windows-подобных систем.
17 Кто является создателем ядра Linux?
18 Что такое ISO-образ?
19 Что такое виртуальный жесткий диск?

9 Лабораторная работа № 9. Настройка серверов

Цель работы: ознакомиться с порядком настройки серверов.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

Веб-сервер – это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы в виде HTML-страницы с изображением, текстами, медиа-потоком или другими данными. Веб-серверы – это основа Всемирной паутины WWW.

Веб-сервером называют как программное обеспечение, выполняющее функции **веб-сервера**, так и компьютер, на котором это программное обеспечение работает.

Клиенты получают доступ к **веб-серверу** по URL адресу нужной им **веб-страницы** или FTP ресурса.

FTP ресурс использует **FTP** (File Transfer Protocol – протокол передачи файлов) протокол, предназначенный для передачи файлов в компьютерных сетях. **FTP** позволяет подключаться к серверам **FTP**, просматривать содержимое каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами (рисунок 9.1).

Порядок установки WEB- и FTP-сервера:

1) скачивается на диск компьютера программное обеспечение WEB- и FTP- сервера, находящееся по адресу: ftp://10.242.48.45/software/Xitami/xitami.zip.

Этот файл представляет собой дистрибутив программного обеспечения WEB- и FTP-сервера, упакованный в виде архива ZIP;

2) дважды щелкнув в «Проводнике» по файлу xitami.zip открывается архив. Копируется содержимое каталога xitami из архива на диск D:\famili компьютера. Вместо famili создается папка со своей фамилией (латинскими буквами и без пробелов).

Рисунок 9.1 – Схема работы WEB- и FTP-сервера

Установка WEB- и FTP-сервера завершена. Теперь его необходимо настроить для работы.

Настройка программного обеспечения сервера заключается в редактировании параметров, находящихся в его конфигурационных файлах (файлах настройки). Параметров у сервера очень много, обычно устанавливаются базовые, которые непосредственно необходимы для корректной работы программного обеспечения.

Выполняется переход в папку, в которую установлен сервер (обычно это **D:\xitami**) и открывается в Блокноте файл **xitami.cfg**. Это главный файл конфигурации сервера.

Файл представляет собой набор различных параметров следующего вида: **Имя_параметра = значение.**

Находим параметр: **keep-alive-max=50.** Изменяем его значение на более высокое, например 100. Этот параметр задает максимальное количество одновременно подключаемых к серверу клиентов.

Находим параметр: **default1=index.htm.** Здесь указываются имена HTML-страниц, которые находятся и отображаются сервером по умолчанию, если клиент не указал имя страницы в адресной строке.

Измените это значение на следующее:

default1=index.htm
default2=index.html
default3=default.htm
default4=default.html
default5=index.php.

В данном случае указаны пять страниц: **index.htm, index.html, default.htm, default.html, index.php.**

Теперь при обращении к серверу по адресу http://адрес_сервера/ (например http://www.stu.ru, т. е. без указания имени страницы) сервер найдет и вернет пользователю страницу с одним из имеющихся имен списка настроек, если хотя бы одна из них есть на диске.

Находим параметр: **ipaddress = *.** Этот параметр задает IP-адрес сервера, вместо * указывается IP-адрес своего компьютера.

Для определения IP-адреса нажимаем «**Пуск / Выполнить**». В открывшимся окне вводим **cmd** и нажимаем **Enter**. Откроется командная строка. В командной строке вводится команда **ipconfig.** Эта системная команда позволяет просмотреть сетевые настройки компьютера, в том числе и IP-адрес нашего компьютера, который и вписывается в значение параметра **ipaddress =*** вместо *. Окно с командной строкой – **cmd** закрывается.

Находим строку **[Ftp]**, и сразу ниже после нее параметр: **enabled=0.** Устанавливаем значение единица вместо ноля. Этот параметр включает FTP-сервер (по умолчанию он отключен).

Находим параметр: **root=**. Этот параметр задает путь к корневой папке FTP-сервера. Здесь следует указать папку, в которой будут находиться все HTML-страницы. Это необходимо для того, чтобы иметь возможность удаленно, с другого компьютера, находящегося в сети, используя доступ к серверу через FTP-протокол закачивать на сервер HTML-страницы и другие файлы.

Если установлен сервер в папку **D:\xitami**, тогда путь к папке с HTML-файлам будет **root=D:\xitami\webpages.** Если установлен сервер в другую папку, вводится требуемый путь. В конце всегда должно быть webpages – это папка, в которой хранятся все HTML-страницы и файлы сервера.

Контрольные вопросы

1 Назначение параметра **keep-alive-max=50**.
2 Назначение параметра **default1=index.htm.**
3 Назначение параметра **ipaddress=*.**

Лабораторная работа № 10. Подключение Linux/FreeBSD к Windows-сети

Цель работы: ознакомиться с порядком подключения Linux/FreeBSD к Windows-сети.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

Большинство компьютеров в локальной сети работают под управлением ОС Windows. В таком случае к тем ресурсам этих компьютеров, которые «отданы» в общее пользование, проще всего подключаться, используя пакет Samba.

Samba – это набор приложений, позволяющих ОС Linux взаимодействовать с сетью, построенной на основе MS Windows, причем как в роли клиента сетей MS Windows, так и в роли сервера. Пакет Samba реализует протокол Server Message Block (SMB), который иногда называют также Session Message Block (SMB), протоколом NetBIOS или протоколом LanManager. Ниже будет рассмотрена работа клиентских программ этого пакета, а именно smbclient, smbmount и smbumount. Если этих программ нет на своем компьютере, то устанавливается пакет Samba (например, на дистрибутивном компакт-диске с Black Cat 6.02 имеется файл samba-client-2.0.5a-2bc.i386.rpm или, возможно, другая версия).

Программа **smbclient** предоставляет пользователю FTP-подобный интерфейс для переноса файлов с компьютеров, работающих под ОС Windows (или с компьютеров, на которых запущен сервер Samba). По сравнению с FTP **smbclient** имеет то преимущество, что не требует, чтобы на удаленном компьютере, работающем под Windows, была запущена специальная серверная программа, поскольку Windows поддерживает NetBIOS по умолчанию. Только должен быть открыт доступ к какому-либо каталогу из сети. Если же через Samba требуется получить доступ к UNIX-серверу, то на нем должна быть запущена серверная часть пакета Samba.

Предположим, что в сети имеется компьютер с именем PC1, работающий под ОС Windows, и на нем имеется каталог, открытый для доступа из сети, которому присвоено имя ресурса PUBLIC (в ОС Windows регистр символов не имеет значения).

Для начала подается команда

[root]# smbclient -L pc1,

чтобы увидеть доступные из сети ресурсы компьютера PC1. Если компьютер PC1 работает под управлением Windows NT, то надо сразу указать имя пользователя, который имеет права доступа к компьютеру:

[root]# smbclient -U user -L pc1,

в ответ на запрос программы ввести пароль этого пользователя.

В ответ на такой запрос получают примерно следующую информацию:

```
Domain=[WORKGR] OS=[Windows NT 4.0] Server=[NT LAN Manager 4.0]
Sharena
me          Type    Comment
---------   ----    -------
ADMI                Remote
N$          Disk    Admin
public      Disk
C$          Disk    Default share
IPC$        IPC     Remote IPC
G           Disk


Server Comment
--------- -------
PC2    Samba 1.9.15p8
PC5
PC25  Samba 1.9.15p8
```

Следует обратить внимание на то, что вслед за строкой Server Comment перечисляются другие SMB-сервера в сети с доступными ресурсами.

Чтобы получить доступ к ресурсу на удаленном компьютере, надо дать команду следующего вида:

[user]$ /usr/sbin/smbclient servicename -U user [password],

где servicename – это имя машины и ресурса, которые должны бы вообще-то иметь вид \\pc1\public, но из-за ограничений оболочки каждый слэш надо удваивать, поэтому команда принимает следующий вид:

[user]$ /usr/sbin/smbclient \\\\PC1\\public -U user mypasswd.

Указывать имя пользователя в опции необходимо только в том случае, если оно не совпадает с именем пользователя, от имени которого запущена программа **smbclient**. Естественно, что пароль необходим только в том случае, если доступ к ресурсу защищен паролем.

Если доступ к ресурсу дан, то будет получено приглашение программы:

```
Server time is Sat Mar 11 15:58:27 2017
Domain=[WORKGROUP] OS=[Windows NT 4.0] Server=[NT LAN Manager 4.0] smb: \>
```

В ответ на это приглашение можно вводить одну из следующих встроенных команд программы smbclient (этот перечень можно получить, введя команду h или ?

smb: \h

ls	dir	du	lcd	cd
pwd	get	mget	put	mput
rename	more	mask	del	open
rm	mkdir	md	rmdir	rd
prompt	recurse	translate	lowercase	print
printmod				
e	queue	cancel	quit	q
exit	newer	archive	tar	blocksize
tarmode	setmode	help	?	!

Команды эти во многом похожи на команды ftp-клиента и работа с программой **smbclient** не очень удобна. Но в пакет samba-client-2.0.5a-2bc.i386.rpm входят еще две программы, которые предоставляют некоторые дополнительные удобства. Эти программы называются **smbmount** и **smbumount**. С помощью команды **smbmount** можно смонтировать сетевой ресурс к локальной структуре каталогов. Формат команды:

[user]$ /usr/sbin/smbmount //PC1/public /mnt/pc1 -U 123 -W 456',

(в этом примере сетевой ресурс монтируется в локальный каталог /mnt/pc1, причем владельцем каталога объявляется пользователь 123 и группа 456). При необходимости нужно будет ввести пароль пользователя (тот же, по которому получен доступ к ресурсу в команде **smbclient**).

Команда **smbumount** позволяет обычным пользователям размонтировать файловую систему, смонтированную командой **smbmount** (пользователь **root** может воспользоваться обычной командой **umount**. Формат команды (используется название точки монтирования из того же примера):

[user]$ /usr/sbin/smbumount /mnt/pc1.

Если после монтирования сетевого ресурса запустить программу Midnight Commander, перейти в каталог /mnt/pc1, то можно увидеть файлы каталога public на компьютере PC1.

Контрольные вопросы

1 Назначение программы smbclient.
2 Назначение программы smbmount**.**
3 Назначение программы smbumount.

11 Лабораторная работа № 11. Обеспечение доступа в сеть Интернет

Цель работы: рассмотреть различные варианты подключения к сети Интернет локальной сети, используя различные программные средства.

Порядок выполнения работы

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
2 Получить задание у преподавателя, выполнить типовые задания.
3 Сделать выводы по результатам исследований.
4 Оформить отчет.

Требования к отчету

1 Цель работы.
2 Постановка задачи.
3 Результаты исследования.
4 Выводы.

Основные теоретические положения

Постановка задачи. Имеется локальная сеть (**Workstantion 1 – Workstantion 7**), представленная на рисунке 11.1. На компьютере (шлюзе), через который планируется подключение локальной сети к Интернет, необходимо наличие двух сетевых адаптеров (подключений).

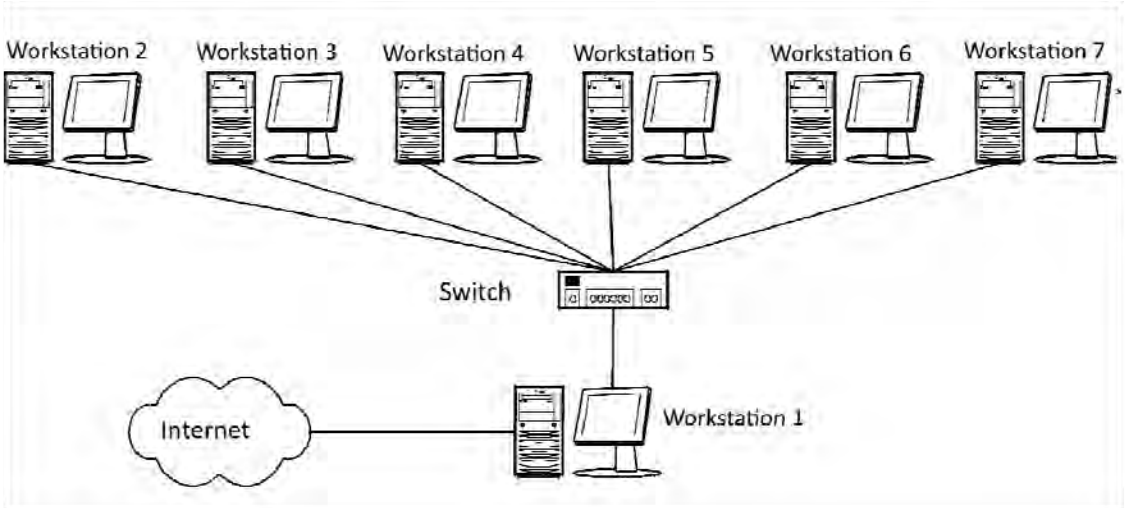

Рисунок 11.1 – Структура локальной сети

Необходимо обеспечить доступ к сети Интернет со всех рабочих станций. Имеются три основных варианта подключения локальной сети к Интернет:

1) «прямое» IP-подключение;
2) подключение через NAT;
3) подключение через прокси-сервер.

Выбор конкретного способа подключения зависит от потребностей пользователей, цели подключения и, в некоторой степени, финансовых возможностей.

Итак, компьютер ***Workstantion 1***. У него есть доступ как к Интернету, так и к локальной сети. Поставлена задача – дать компьютерам локальной сети доступ к Интернет через подключенный к нему компьютер. Этот компьютер называют шлюзом или маршрутизатором.

Рассмотрение способов мы начнем с наименее часто использующегося, наиболее дорогого, но также наиболее «правильного» и естественного способа, дающего наибольшие по сравнению с другими способами возможности. **«Прямое» IP-подключение к Internet**. Для того, чтобы локальная сеть была полноценно подключена к Интернету, должны соблюдаться, как минимум, три условия:

1) каждая машина в локальной сети должна иметь «реальный», интернетовский IP-адрес;
2) эти адреса должны быть не любыми, а выделенными провайдером для данной локальной сети;
3) на компьютере-шлюзе, подключенном к двум сетям – локальной сети и сети провайдера, должна быть организована IP-маршрутизация, т. е. передача пакетов из одной сети в другую.

В этом случае локальная сеть становится как бы частью Интернета. Собственно, это тот способ подключения, которым подключены к Интернету сами Интернет-провайдеры и хостинг-провайдеры.

Отличие от обычного подключения, рассчитанного на один компьютер, при таком подключении «под клиента» выделяется не один IP-адрес, а несколько, так называемая «IP-подсеть».

При таком способе подключения можно организовать в локальной сети сервисы, доступные из Интернета – ведь при данном подключении не только Интернет полностью доступен из локальной сети, но и локальная сеть – из Интернета, т. к. является его частью.

Однако такая «прозрачность» локальной сети резко снижает ее защищенность – ведь любые сервисы в локальной сети, даже предназначенные для «внутреннего» использования, станут доступными извне через Интернет. Чтобы это не имело места, доступ в локальную сеть извне несколько ограничивают. Обычно это делается установкой на шлюзе программы firewall. Это своеобразный фильтр пакетов, проходящих из одной сети в другую. Путем его настройки можно запретить вход-выход из локальной сети пакетов, соответствующих определенным критериям – типу IP-пакета, IP-адресу назначения, TCP/UDP-порту и т. п.

Firewall решает такие задачи, как блокировка доступа извне к определенным TCP/IP-сервисам локальной сети, блокировка доступа к определенным компьютерам локальной сети (таким образом можно запретить доступ извне ко всем машинам, кроме определенных серверов, предназначенных для доступа

из Интернета), защита от троянских программ на сетевом уровне.

Несмотря на универсальность такого метода подключения локальной сети к Интернету, этот метод имеет недостатки. Благодаря им, его реально и используют только лишь те организации, которым надо сделать свои сервера доступными из Интернета – в основном, те же интернет-провайдеры и хостинг-провайдеры, а также информационные службы. Самый главный недостаток заключается в дороговизне выделения IP-адресов и уж тем более IP-подсетей.

Поэтому на практике рассмотрим другие, описанные далее способы, не требующие больших затрат и, что самое главное, позволяющие подключить локальную сеть через обычное подключение с одним внешним IP-адресом.

Технология Network Address Translation (NAT) – «трансляция сетевых адресов» позволяет нескольким машинам локальной сети иметь доступ к Интернету через одно подключение и один реальный внешний IP-адрес.

Для того, чтобы компьютеры локальной сети могли устанавливать соединения с серверами сети Интернет, нужно, чтобы:

– IP-пакеты, адресованные серверу в Интернет, смогли его достигнуть;

– ответные IP-пакеты, идущие от сервера Интернет на машину в локальной сети, также смогли ее достигнуть.

С первым условием проблем не возникает, а как быть со вторым? Ведь компьютеры локальной сети не имеют своего «реального» интернетовского IP-адреса. Как же они могут получать IP-пакеты из Интернет?

А работает это следующим образом – на компьютере-шлюзе стоит программа NAT-сервера. Компьютер-шлюз прописан на машинах локальной сети как «основной шлюз», и на него поступают все пакеты, идущие в Интернет (не адресованные самой локальной сети). Перед передачей этих IP-пакетов в Интернет NAT-сервер заменяет в них IP-адрес отправителя на свой, одновременно запоминая у себя, с какой машины локальной сети пришел этот IP-пакет. Когда приходит ответный пакет (на адрес шлюза, конечно), NAT определяет, на какую машину локальной сети его надо направить. Затем в полученном пакете меняется адрес получателя на адрес нужной машины, и пакет доставляется этой машине через локальную сеть.

Как видно, работа NAT-сервера прозрачна для машин локальной сети (как и работа обычного IP-маршрутизатора). Единственным принципиальным ограничением этого метода подключения локальной сети к Интернет является невозможность установить входящее TCP-соединение из Интернет на машину локальной сети. Однако для «клиентских» сетей этот недостаток превращается в достоинство, резко увеличивающее (по сравнению с первым методом подключения) их защищенность и безопасность. Администраторы некоторых провайдеров даже употребляют слова NAT и Firewall как синонимы.

Подключение через прокси-сервер – самый простой тип подключения. При этом никакой маршрутизации IP-пакетов между локальной сетью и сетью Интернет не происходит. Машины локальной сети работают с Интернетом через программу-посредник, так называемый прокси-сервер, установленный на компьютере-шлюзе.

Основной особенностью этого метода является его «непрозрачность». Если

в случае NAT программа-клиент просто обращается к Интернет-серверу, не «задумываясь», в какой сети и через какую маршрутизацию она работает, то в случае работы через прокси-сервер программа должна явно обращаться к прокси-серверу. Мало того, клиентская программа должна уметь работать через прокси-сервер. Однако проблем с этим не возникает – все современные и не очень браузеры умеют работать через прокси-сервера.

Другой особенностью является то, что прокси-сервер работает на более высоком уровне, чем, скажем, NAT. Здесь уже обмен с Интернетом идет не на уровне маршрутизации пакетов, а на уровне работы по конкретным прикладным протоколам (HTTP, FTP, POP3 ...). Соответственно для каждого протокола, по которым должны «уметь» работать машины локальной сети, на шлюзе должен работать свой прокси-сервер.

Эта «протокольная зависимость» и есть основной недостаток этого метода подключения как самостоятельного.

Однако, с другой стороны, «маршрутизация» на таком высоком уровне может дать и немалые преимущества.

Почти каждый интернет-провайдер имеет один или несколько прокси-серверов, через которые рекомендует работать своим клиентам. Несмотря на то, что это совершенно необязательно (как правило, клиент провайдера может обращаться к Интернет напрямую), это дает выигрыш в производительности, а при повременной оплате, соответственно, экономить время онлайн. Это происходит потому, что прокси-сервера способны кэшировать (запоминать) запрашиваемые пользователем документы, и при следующих к ним обращениях выдавать копию из кэша, что быстрее, чем повторно запрашивать с Интернет-сервера. Кроме того, прокси-сервера могут быть настроены так, что будут блокировать загрузку баннеров наиболее распространенных баннерных служб, тем самым также (порой значительно) ускоряя загрузку веб-страниц.

При установке HTTP прокси-сервера в локальной сети и работе через него за счет кэширования экономится не только время, но и трафик, потому что кэширование происходит в самой локальной сети, до канала с провайдером, в котором считается трафик (при оплате за объем перекачанной информации).

Контрольные вопросы

1 Особенности подключения локальной сети к Интернету через «прямое» IP-подключение.

2 Особенности подключения локальной сети к Интернету через NAT.

3 Особенности подключения локальной сети к Интернету через прокси-сервер.

Список литературы

1 **Новиков, В. А**. Информационные системы и сети. С электронным приложением : учебное пособие / В. А. Новиков, А. В. Новиков, В. В. Матвеенко. – Минск : Изд-во Гревцова, 2014. – 448 с.

2 **Олифер, В. Г.** Компьютерные сети. Принципы, технологии, протоколы : учебник / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – Санкт- Петербург : Питер, 2013. – 944 с.

3 **Пескова, С. А.** Сети и телекоммуникации : учебное пособие / С. А. Пескова, А. В. Кузин, А. Н. Волков. – 2-е изд., стер. – Москва : Академия, 2007. – 352 с.

4 **Бройдо, В. Л.** Архитектура ЭВМ и систем : учебник для вузов/ В. Л. Бройдо, О. П. Ильина. – Санкт-Петербург : Питер, 2009. – 720 с.

5 **Бройдо, В. Л**. Вычислительные системы, сети и телекоммуникации : учебное пособие / В. Л. Бройдо, О. П. Ильина. – 4-е изд. – Санкт-Петербург : Питер, 2011. – 560 с.