ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

АДМИНИСТРИРОВАНИЕ WINDOWS-СЕРВЕРОВ

Методические рекомендации к лабораторным работам для студентов направления подготовки 09.03.04 «Программная инженерия» дневной формы обучения

Могилев 2018

УДК 004.4
ББК 32.973.202
А 31

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления» «13» марта 2018 г., протокол № 11

Составитель канд. техн. наук, доц. И. А. Евсеенко

Рецензент канд. техн. наук, доц. И. В. Лесковец

Методические рекомендации предназначены для студентов направления подготовки 09.03.04 «Программная инженерия» дневной формы обучения.

Учебно-методическое издание

АДМИНИСТРИРОВАНИЕ WINDOWS-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

Основной материал в приведенных методических рекомендациях посвящен серверным операционным системам семейства Windows.

Целью методических рекомендаций является предоставление возможности самостоятельного освоения студентами основных методов администрирования локальных сетей на основе серверных операционных систем Windows, получения практических навыков администрирования, а также изучения перспективных направлений в развитии современных серверных операционных систем Windows.

Целью дисциплины является изучение проблем администрирования Windows серверов, классификации операционных систем и применяемого программного обеспечения для администрирования серверов Windows, а также теоретических и практических основ администрирования локальных сетей на основе операционных систем Windows.

При освоении данной дисциплины студент приобретает практические навыки работы с серверными и клиентскими опрерационными системами Windows, создания клиент-серверных приложений, выбора и настройки операционных систем и программного обеспечения для них, диагностирования и устранения неполадок в локальных сетях, инсталлирования, тестирования и настройки конкретных конфигураций операционных систем, работы с различными операционными системами и их администрирования, а также настройки удаленного доступа к серверу и Интернет подключения.

1 Лабораторная работа № 1. Установка операционной системы Windows. Основные принципы функционирования ОС Windows

Цель работы: ознакомление с редакциями, набором сетевых служб, процессом установки и начальной настройки операционных систем семейства Windows Server.

Операционные системы семейства Windows Server являются универсальной платформой, на которой реализованы практически все сетевые службы – служба каталогов Active Directory, службы сетевой инфраструктуры (*DNS, DHCP, WINS, маршрутизация и удаленный доступ*), *службы файлов и печати, службы веб-публикаций и т. д.*

С дистрибутивного компакт-диска можно установить комплект ресурсов Windows Server Support Tools. Средства поддержки – это универсальный набор утилит для выполнения любых сервисных задач от диагностики системы до сетевого мониторинга.

Способы администрирования систем Windows Server.

Чаще всего применяются следующие.

Панель управления – набор средств для управления конфигурацией системы Windows Server. В классическом меню Пуск (Start) доступ к этим средствам открывает подменю Настройка (Settings), в упрощенном меню Пуск (Start) команда Панель управления (Control Panel) доступна сразу.

Графические средства администрирования – ключевые средства для управления компьютерами в сети и их ресурсами. Доступ к необходимому средству можно получить в подменю Администрирование (Administrative Tools).

Мастера администрирования – средства автоматизации ключевых административных задач. В отличие от Windows NT мастера не сосредоточены в центральном месте – доступ к ним происходит посредством выбора соответствующих параметров меню и других средствах администрирования.

Функции командной строки. Большинство административных действий можно выполнять из командной строки.

Серверы, обеспечивающие работу в сети TCP/IP, или серверы сетевой инфраструктуры. К ним относятся DHCP-, DNS- и WINS-серверы; обычно настройку работы в крупной сети начинают именно с них.

DHCP-серверы нужны, чтобы по запросу *DHCP-клиента* (компьютера, у которого в настройках протокола TCP/IP включен режим автоматического получения IP-адреса) выдать ему такие параметры, как уникальный IP-адрес и маску подсети. Кроме них, клиент может получать от DHCP-сервера ряд дополнительных параметров, важных для взаимодействия с другими сетями и удобной работы в сети: адрес основного шлюза, адреса DNS- и WINS-серверов, название домена, в который входит этот компьютер, и некоторые другие.

DNS-серверы выполняют очень важную функцию *преобразования* (*разрешения*) *имен узлов* (*host names*) *в соответствующие им IP-адреса*.

DNS (Domain Name System) расшифровывается как «система (служба) доменных имен». Служба DNS была реализована в Интернете в 1981 г., а с 2000 г. стала основной службой преобразования имен в сетях Microsoft.

Серверы файлов (файл-серверы) нужны для хранения больших объемов данных и предоставления к ним доступа пользователей. Один файловый сервер может поддерживать одновременную работу сотен и даже тысяч пользователей. Чтобы обеспечить сохранность информации, файл-серверы, как правило, оснащены отказоустойчивыми наборами (массивами) жестких дисков и системами резервного копирования на магнитную ленту или другой носитель.

Серверы приложений выполняют задачи обслуживания запросов пользователей на выборку или обработку какой-либо информации; их часто объединяют с **серверами баз данных**. Важно, что с серверами приложений и баз данных одновременно может работать большое число пользователей, причем выполнение клиентских запросов на специализированном многопроцессорном сервере производится намного быстрее, чем на компьютерах пользователей.

Серверы удаленного доступа и **серверы VPN** (Virtual Private Network – «виртуальная частная сеть») обеспечивают удаленное подключение к локальной сети по модему или через Интернет. Это дает пользователям возможность работать с ресурсами локальной сети предприятия, офиса или учебного заведения из дома или из любого места, где есть подключение к Интернету, например из Интернет-кафе.

Терминальные серверы предоставляют возможность работы с другими серверами через специальные программы – *терминальные клиенты*. С помощью этих программ администраторы, находясь вдалеке от локальной сети, оказываются как будто за консолью сервера и могут полностью управлять им, а пользователи могут удаленно работать с установленными на сервере приложениями.

Брандмауэры (межсетевые экраны) используются при подключении к Интернету для защиты внутренней сети от проникновения или атаки злоумышленников на корпоративные серверы. **Прокси-серверы (серверы-посредники)** выполняют функции контроля доступа пользователей в Интернете и кеширования часто запрашиваемых веб-страниц (что позволяет снизить расходы на пользование Интернетом). Поскольку оба этих сервера предназначены для установки на компьютер, связывающий локальную сеть с Интернетом, их часто объединяют в единую программно-аппаратную систему.

Серверы электронной почты (почтовые серверы, mail-серверы) обслуживают почтовые ящики пользователей в данной организации, обеспечивая подключения к ним *почтовых клиентов*, а также обрабатывают все входящие и исходящие сообщения. Их также можно использовать для ведения адресных книг, общих папок и систем электронного документооборота.

Веб- и FTP-серверы предоставляют для внешних (а часто – и для внутренних) пользователей доступ к веб- и FTP-ресурсам, размещенным в данной сети.

Контроллеры домена обеспечивают в сетях Microsoft работу служб *Активного каталога* (*Active Directory*) и поддерживают базу данных всех зарегистрированных в *домене* пользователей, компьютеров, групп и ресурсов.

Наличие такой базы данных позволяет администраторам централизованно управлять всеми сетевыми объектами и ресурсами. Пользователи же получают возможность входить в сеть с любого принадлежащего домену компьютера, а затем «прозрачно» (без ввода имени и пароля) подключаться к другим компьютерам и работать с их ресурсами.

Рабочая группа – это логическая группировка компьютеров, объединенных общим именем для облегчения навигации в пределах сети. Принципиально важно, что каждый компьютер в рабочей группе *равноправен* (т. е. сеть получается одноранговой) и *поддерживает собственную локальную базу данных учетных записей пользователей* (*Security Accounts Manager*, *SAM*).

Отсюда вытекает основная проблема, которая не позволяет использовать рабочие группы в крупных корпоративных сетях. Для обеспечения «прозрачного» взаимодействия в рабочей группе нужно *создавать одинаковые учетные записи с одинаковыми паролями на всех компьютерах*, где работают пользователи и расположены ресурсы.

Понятно, что управлять учетными записями и ресурсами в рабочей группе можно только при небольшом количестве компьютеров и пользователей. В крупных сетях следует применять домены.

Домен – это логическая группировка компьютеров, объединенных *общей базой данных пользователей и компьютеров, политикой безопасности и управления*.

Домены создаются на основе сетевых ОС Windows, а база данных поддерживается *контроллерами домена*. Важным в доменах является то, что все компьютеры здесь не сами осуществляют проверку пользователей при входе, а передоверяют эту процедуру контроллерам. Такая организация доступа позволяет легко осуществить однократную проверку пользователя при входе в сеть, а затем уже без проверки предоставлять ему доступ к ресурсам всех компьютеров домена.

Задание

1 Загрузите виртуальную машину, используя указанный преподавателем образ, и выполните настройку компонентов сетевых подключений для работы в сети.

2 Запустите установку операционной системы Windows.

3 Создайте разметку жесткого диска и выберите раздел для установки.

4 Введите ключ для установки системы.

5 Настройте время и дату.

6 Установите поддержку сети.

7 Настройте Главное меню, регистрацию компонентов и сохраните настройки.

8 Установите драйверы устройств. В процессе установки могут потребоваться драйверы для устройств, для которых в БД драйверов системы нет соответствующего драйвера. Если вы занимаетесь в группе под руководством преподавателя, то он предоставит все необходимые драйверы. Если вы занимаетесь индивидуально, то сами позаботьтесь о необходимых драйверах для ваших компьютеров.

Контрольные вопросы

1 Какие редакции систем входят в семейство Windows Server?

2 Функциональные возможности различных редакций системы Windows Server.

3 Какие сетевые службы функционируют в операционных системах семейства Windows Server?

2 Лабораторная работа № 2. Терминал и командная оболочка операционной системы Windows

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

Общие сведения о командной оболочке. 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. В командной оболочке программы выполняются и результат выполнения отображается на экране в виде, сходном с интерпретатором Command.com MS-DOS. Командная оболочка Windows 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.

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.

Пример процедурного файла. *В указанной директории удалить все файлы с указанными типам. Типы файлов задаются в командной строке.*

```
echo on
set deldir = %1
:one
Shift
if "%1"=="" goto two
del %deldir%\*.%1
goto one
:two
set deldir=
```

Задание

Разработать командный (процедурный) файл согласно указанного ниже варианта задания. Выполнить его отладку и тестирование. В формулировке задания указание – заданный, указанный и т. п. относится к имени объекта, которое задается как параметр командной строки.

1 В указанной директории удалить все файлы с указанными типам. Типы файлов задаются в командной строке.

2 Вывести на экран списки файлов из указанных директорий согласно указанному шаблону.

3 Удалить из указанных директорий файлы, имена которых заданы шаблоном.

4 Скопировать из заданных директорий все файлы в указанную директорию. Если указанной директории не существует – создать ее.

5 Создать в заданной директории поддиректории, имена которых совпадают с типами файлов, находящихся в заданной директории (для совместимости – считать, что расширения имен файлов – заданы).

6 Скопировать из заданной директории все файлы в поддиректории так, чтобы все файлы типа *.exe были скопированы в поддиректорию EXE, а остальные файлы – в поддиректорию XXX.

7 Из заданной директории переписать указанные шаблоном файлы в заданную директорию так, чтобы были переписаны только «новые» (не существующие в приемной директории) файлы.

8 Из заданной директории переписать указанные шаблоном файлы в заданную директорию так, чтобы были переписаны только «старые» (существующие в приемной директории) файлы.

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 Команды подготовки новых дискет к работе.

7 Процедурные файлы. Назначение и основные правила создания.

8 Основные команды для создания процедурных файлов.

9 Команда dir, more. Их параметры и варианты применения.

10 Атрибуты файлов. Их назначение и использование.

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

3 Лабораторная работа № 3. Изучение файловой системы и функций по обработке и управлению данными

Цель работы: изучение структуры файловой системы, изучение команд создания, удаления, модификации файлов и каталогов, функций манипулирования данными.

В операционной системе файлами считаются обычные файлы, каталоги, а также специальные файлы, соответствующие периферийным устройствам (каждое устройство представляется в виде файла). Доступ ко всем файлам однотипный, в том числе и к файлам периферийных устройств. Такой подход обеспечивает независимость программы пользователя от особенностей ввода/вывода на конкретное внешнее устройство.

Текущий каталог – это каталог, в котором в данный момент находится пользователь. При наличии прав доступа, пользователь может перейти после входа в систему в другой каталог. Текущий каталог обозначается точкой (.); родительский каталог, которому принадлежит текущий, обозначается двумя точками (..).

Полное имя файла может включать имена каталогов, включая корневой, разделенных косой чертой, например: /home/student/file.txt. Первая косая черта обозначает корневой каталог, и поиск файла будет начинаться с него, а затем в каталоге home и в его подкаталогах.

Задание

1 Ознакомтесь с файловой структурой ОС. Изучите команды работы с файлами.

2 Используя команды ОС, создайте два текстовых файла.

3 Полученные файлы объедините в один файл и его содержимое выведите на экран.

4 Создайте новую директорию и переместите в нее полученные файлы.

5 Выведите полную информацию обо всех файлах и проанализируйте уровни доступа.

6 Добавьте для всех трех файлов право выполнения членам группы и остальным пользователям.

7 Просмотрите атрибуты файлов.

8 Создайте еще один каталог.

9 Установите дополнительную связь объединенного файла с новым каталогом, но под другим именем.

10 Создайте символическую связь.

11 Сделайте текущим новый каталог и выведите на экран расширенный список информации о его файлах.

12 Произведите поиск заданной последовательности символов в файлах текущей директории и получите перечень соответствующих файлов.

13 Получите информацию об активных процессах и имена других пользователей.

Контрольные вопросы

1 Что считается файлами в ОС WINDOWS?

2 Объясните назначение связей с файлами и способы их создания.

3 Что определяет атрибуты файлов и каким образом их можно просмотреть и изменить?

4 Какие методы создания и удаления файлов (каталогов) Вы знаете?

5 В чем заключается поиск по шаблону?

6 Какой командой можно получить список работающих пользователей и сохранить его в файле?

4 Лабораторная работа № 4. Процессы в операционной системе Windows

Цель работы: практическое изучение возможностей утилит Windows на примере наиболее характерных операций.

Задачами работы являются:

– закрепление навыков использования аналога диспетчера задач;

– овладение методикой масштабирования и редактирования презентаций в демонстрации;

– приобретение навыков управления мониторингом системы, процессами, потоками.

Утилиты – обслуживающие программы, которые предоставляют пользователю сервисные функции. Многие из утилит обладают развитым диалоговым интерфейсом с пользователем и приближаются по уровню общения к оболочкам. Остальные же используются путем их запуска с определенными аргументами.

Программы-утилиты – это специальные компьютерные программы для выполнения особых функций, предназначенные для расширения возможностей операционных систем. Это могут быть безвозвратное удаление файлов, восстановление данных, оптимизация Windows, очистка реестра и другие важные задачи.

Обычно программы-утилиты имеют не очень большой размер и

значительно увеличивают диапазон взаимодействия с Windows. Утилиты могут быть встроенными в ОС и представлять собой часть оболочки или быть самостоятельными программами.

Существующие в настоящее время утилиты обеспечивают реализацию следующих функций:

а) обслуживание дисков, а именно:

– форматирование дисков в нескольких режимах;

– восстановление ошибочно удаленных файлов, а также в случае разрушения;

– дефрагментация файлов на диске, вследствие чего время доступа к файлам сокращается до 30 % и облегчается восстановление информации в случае разрушения;

– надежное затирание конфиденциальной информации;

б) утилиты печати;

в) шифрование информации;

г) защита от компьютерных вирусов;

д) архивация данных.

Для выполнения лабораторной работы воспользуемся пакетом программ Sysinternals Suite, который можно скачать по ссылке –http://technet.microsoft.com/ru-ru/sysinternals. Пакет содержит более 60 утилит для анализа и мониторинга системы, работы с файлами и дисками, сетевые и диагностические программы. Сервисные программы Sysinternals помогают как специалистам по информационным технологиям, так и разработчикам управлять, находить и устранять неисправности и выполнять диагностику приложений и операционной системы Windows.

Process Explorer – программа, предназначенная для мониторинга системных ресурсов. Обладает более богатым функционалом, чем стандартный Диспетчер Задач. Это системная утилита, позволяющая отслеживать процессы, происходящие на компьютере. С ее помощью всегда можно узнать, что происходит в операционной системе в данный момент. Среди основных отличий Process Explorer от Диспетчера Задач отмечают следующие:

– отображение запущенных процессов в виде древовидной структуры;

– наличие средств графического отображения данных (для повышения качества восприятия информации);

– применение разноцветной подсветки данных (для удобства восприятия);

– вывод информации об используемых системой динамически подключаемых библиотеках;

– возможность запуска программ с различными правами пользователя.

Интерфейс программы представляет собой два окна, в которых отображаются текущие процессы. В верхнем окне программы отображаются все процессы с указанием учетных записей, которым эти процессы соответствуют. В нижнем окне, в зависимости от режима работы, могут отображаться все запущенные в данный момент дескрипторы, а в режиме DLL – все динамические библиотеки, загруженные в память тем или иным процессом. Это

позволяет с точностью определить, какому процессу соответствует отображаемая информация. Это весьма полезно для пользователей, которые имеют навыки обращения с операционными системами по системной части. Кроме всего прочего, программа обладает качественным поиском. Всегда точно можно узнать, с помощью какого приложения был открыт тот или иной файл, какие библиотеки загружены тем или иным дескриптором. Возможности программы предусматривают изменение какого-либо процесса, или же, его полную остановку и выгрузку из памяти компьютера. Кроме всего прочего, с помощью Process Explorer предусмотрена возможность запускать любые приложения, выключать, перезагружать и даже заблокировать компьютер, сохранять в текстовом LOG-файле список всех процессов с подобным описанием процессов и размером занятой каждым из них памяти, находить используемые библиотеки, включать подсветку.

Задание

1 Запустите утилиту Process Explorer и выполните следующие задания:

а) запишите в отчет текущую общую загруженность процессора и объем занятой оперативной памяти (отдельно размер своп-файла и физической памяти);

б) откройте любой текстовый файл, сверните окно процесса winword с помощью программы process explorer;

в) назначьте для сочетания клавиш [ctrl + alt + delete] выполнение программы process explorer, а не диспетчера задач;

г) определите, сколько потоков у процессов explorer и procexp;

д) выясните, откуда запускается программа process explorer, какое процентное отношение ресурсов процессора и ОЗУ занимает данная программа;

е) расположите столбцы окна программы в следующем порядке: cpu, pid, working set, private bytes; сохраните такой вид программы;

ж) определите количество запущенных процессов;

з) отсортируйте по столбцу working set и определите, какой процесс занимет больше всего ОЗУ (имя процесса и id);

и) зарисуйте в отчет дерево процессов, подсчитайте количество родительских процессов и процессов-сирот;

к) определите, какой процесс больше всего загружает процессор в данный момент;

л) запишите, какие процессы стали «сиротами», после завершения процесса explorer.

2 Запустите утилиту Process Monitor и выполните следующие задания:

а) изучите панель инстурментов утилиты;

б) проанализиируйте результат выполнения запросов по всем процессам в таблице process monitor;

в) запустите диспетчер задач, найдите соответствующий процесс в окне данных process monitor, оставьте в списке процессов только его в режиме просмотра активности реестра и проанализируйте к какому ключу было последнее обращение и тип обращения, перейдите к данному

объекту в редакторе реестра;

г) настройте вид так, чтобы отображалась только активность файловой системы, выключите запись событий и проанализируйте: к какому файлу было последнее обращение, тип обращения, откройте в проводнике этот файл;

д) настройте фильтр, работающий по следующему правилу: «не отображать события, в поле имени компании которых присутствует значение microsoft corporation».

3 Запустите утилиту Desktops и выполните следующие задания:

а) создайте дополнительный рабочий стол и настройте горячие клавиши переключения между ними;

б) запустите одну и ту же программу (например, Paint) на разных рабочих столах и проанализируйте, каким образом отображаются процессы разных рабочих столов в диспетчере задач?

Контрольные вопросы

1 В чем отличие Process Explorer от Диспетчера задач?

2 Как установить двухоконный режим просмотра Process Explorer?

3 Как вызвать диалоговое окно свойств процесса, какую информацию можно получить во вкладке Performance?

4 Для чего нужен автоскроллинг?

5 Для каких операций выполняется наблюдение посредством утилиты Process Monitor?

6 Какую информацию содержит каждая строка в окне вывода данных программы Process Monitor?

7 Как сохранить события, отфильтрованные текущими фильтрами?

8 Какими тремя способами можно отобразить дерево процессов в Process Monitor?

9 Какими клавишами по умолчанию переключаются рабочие столы?

5 Лабораторная работа № 5. Организация ввода-вывода в ОС Windows

Цель работы: практическое ознакомление с управлением вводом/выводом в операционных системах Windows и кэширования операций ввода/вывода.

Необходимость обеспечить программам возможность осуществлять обмен данными с внешними устройствами и при этом не включать в каждую двоичную программу соответствующий двоичный код, осуществляющий собственно управление устройствами ввода/вывода, привела разработчиков к созданию системного программного обеспечения и, в частности, самих операционных систем.

Программирование задач управления вводом/выводом является наиболее сложным и трудоемким, требующим очень высокой квалификации. Поэтому

код, позволяющий осуществлять операции ввода/вывода, стали оформлять в виде системных библиотечных процедур; потом его стали включать не в системы программирования, а в операционную систему с тем, чтобы в каждую отдельно взятую программу его не вставлять, а только позволить обращаться к такому коду. Системы программирования стали генерировать обращения к этому системному коду ввода/вывода и осуществлять только подготовку к собственно операциям ввода/вывода, т. е. автоматизировать преобразование данных к соответствующему формату, понятному устройствам, избавляя прикладных программистов от этой сложной и трудоемкой работы. Другими словами, системы программирования вставляют в машинный код необходимые библиотечные подпрограммы ввода/вывода и обращения к тем системным программным модулям, которые, собственно, и управляют операциями обмена между оперативной памятью и внешними устройствами.

Таким образом, управление вводом/выводом – это одна из основных функций любой ОС. Одним из средств управления вводом/выводом, а также инструментом управления памятью является диспетчер задач Windows, он отображает приложения, процессы и службы, которые в текущий момент запущены на компьютере. С его помощью можно контролировать производительность компьютера или завершать работу приложений, которые не отвечают.

При наличии подключения к сети можно также просматривать состояние сети и параметры ее работы. Если к компьютеру подключились несколько пользователей, можно увидеть их имена, какие задачи они выполняют, а также отправить им сообщение.

Также управлять процессами можно и «вручную» при помощи командной строки.

Команды Windows для работы с процессами:

at – запуск программ в заданное время;

Schtasks – настраивает выполнение команд по расписанию;

Start – запускает определенную программу или команду в отдельном окне;

Taskkill – завершает процесс;

Tasklist – выводит информацию о работающих процессах, для получения более подробной информации можно использовать центр справки и поддержки или команду help (например: help at);

cmd.exe – запуск командной оболочки Windows.

Задание

1 Отследите выполнение процесса explorer.exe при помощи диспетчера задач и командной строки.

2 Продемонстрируйте преподавателю завершение и повторный запуск процесса explorer.exe из:

а) диспетчера задач;

б) командной строки.

3 Выполненные задания включите в отчет по выполнению лабораторной работы.

Контрольные вопросы

1 Дайте понятие процессу в операционной системе.
2 Дайте понятие службе в операционной системе.
3 Перечислите основные команды работы с процессами при помощи командной строки.

6 Лабораторная работа № 6. Создание и выполнение командных файлов в пользовательской среде ОС Windows

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

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. 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.

В следующем списке представлены различные типы переменных в порядке убывания приоритета.

Встроенные системные переменные.

Системные переменные куста HKEY_LOCAL_MACHINE.

Локальные переменные куста HKEY_CURRENT_USER.

Все переменные среды и пути указаны в файле Autoexec.bat.

Все переменные среды и пути указаны в сценарии входа в систему (если он имеется).

Переменные, используемые интерактивно в сценарии или пакетном файле.

В командной оболочке каждый экземпляр Cmd.exe наследует среду своего родительского приложения. Поэтому можно изменять переменные в новой среде Cmd.exe, что не повлияет на среду родительского приложения.

Подстановка значений в переменные среды. Чтобы иметь возможность подставлять значения в переменную среды из командной строки или из

сценариев, следует заключить имя соответствующей переменной в символы процентов (%имя_переменной%). Символы процентов указывают на то, что Cmd.exe должен обратиться к значениям переменных, а не делать посимвольное сравнение. После определения значения для имени переменной заключите имя переменной в символы процентов. Cmd.exe проводит поиск всех вхождений имени переменной и заменяет его на определенное значение переменной. Например, требуется создать сценарий, содержащий различные значения (например, имена пользователей), и требуется определить соответствующее значение переменной среды USERNAME для каждого пользователя. Для этого следует написать сценарий с использованием переменной USERNAME, заключенной в кавычки. При выполнении сценария Cmd.exe заменит вхождения %USERNAME% соответствующими значениями, что избавит от необходимости делать это вручную для каждого пользователя. Подстановка значений не является рекурсивной. Cmd.exe проверяет переменные один раз. Дополнительные сведения о подстановке значений в переменные см. в разделах For и Call.

Команды пакетной обработки. В таблице 1 приведены основные команды пакетной обработки.

Таблица 1 – Основные команды пакетной обработки

Наименование команды	Описание назначения команды
call	Вызов одного пакетного файла из другого, без завершения последнего
echo	Вывод или отмена вывода эха или вывод текущего состояния
for	Выполнение команды для группы файлов
goto	Переход на указанную метку
if	Выполнение команды при наличия некоторого условия
pause	Пауза в процессе выполнения файла
rem	Вывод коментария
shift	Увеличение количества формальных параметров

Все команды пакетной обработки являются резидентными (внутренними).

Задание

Согласно варианту в лабораторной работе 2 примените команды пакетной обработки.

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 Команды подготовки новых дискет к работе.
7 Процедурные файлы. Назначение и основные правила создания.
8 Основные команды для создания процедурных файлов.
9 Команда dir, more. Их параметры и варианты применения.
10 Атрибуты файлов. Их назначение и использование.
11 Принцип поиска внешней команды (программы) командной оболочкой. Команда set и path.

7 Лабораторная работа № 7. Удаленный доступ в Windows

Цель работы: научиться выполнять удаленное администрирование компьютеров.

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

Удаленный рабочий стол. 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 (как если бы вы работали за рабочим компьютером). 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), Windows, 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.

Virtual Network Computing (VNC). VNC система удалённого доступа к рабочему столу компьютера, использующая протокол RFB (Remote Frame Buffer). 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 платформонезависима: VNC-клиент, называемый VNC viewer, запущенный на одной операционной системе, может подключаться к VNC-серверу.

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 изменения 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.

Задание

Установите удаленно на виртуальный компьютер браузер Opera.

1 Подключитесь к виртуальному компьютеру VM-1 с помощью UltraVNC Viewer:

а) запустите с диска к лабораторным работам UltraVNC Viewer (vncviewer.exe);

б) введите в поле VNC Server – <имя удаленного компьютера> (VM-1);

в) установите режим подключения для просмотра (флажок View only);

г) активируйте подключение кнопкой Connect;

д) введите пароль для подключения – qwerasdf.

2 Переключитесь из режима просмотра в нормальный (контекстное меню окна UltraVNC Viewer/View only).

3 Установите браузер Opera:

а) перейдите в каталог с устанавливаемой программой;

б) запустите процесс установки (Opera_9 _Setup.exe);

в) укажите желаемый язык установки – Русский (ОК);

г) активизируйте процесс установки кнопкой Установить;

д) ознакомьтесь с лицензионным соглашением и согласитесь с ним кнопкой Принимаю;

е) укажите тип установки – Стандартная и щелкните Далее;

ж) установите приложение кнопкой Установить;

з) завершите процесс установки кнопкой Готово.

4 Создайте и сохраните в своей домашней папке снимок экрана виртуального компьютера с запущенным браузером Opera.

Контрольные вопросы

1 Назначение программного обеспечения VNC.

2 Опишите способы удаленного подключения к серверу.

3 Как определить, какой каталог является текущим на удаленном сервере?

4 Назначение протокола RDP.

8 Лабораторная работа № 8. Управление пользователями и обеспечение безопасности в ОС Windows

Цель работы: изучить модель безопасности операционной системы Windows, получить навыки практического использования ее средств обеспечения безопасности.

Классификация защиты семейства ОС Windows. Защита конфиденциальных данных от несанкционированного доступа является важнейшим фактором успешного функционирования любой многопользовательской системы. ОС Windows не является исключением и требования к защите объектов файловой системы, памяти, объектов ядра операционной системы внесли существенный вклад в процесс ее проектирования и реализации. Требования к операционной системе, защищенной по классу С2, включают:

– обязательную идентификацию и аутентификацию всех пользователей операционной системы. Под этим понимается способность операционной системы идентифицировать всех пользователей, которые получают санкционированный доступ к системе, и предоставление доступа к ресурсам только этим пользователям;

– разграничительный контроль доступа – предоставление пользователям

возможности защиты принадлежащих им данных;

– системный аудит – способность системы вести подробный аудит всех действий, выполняемых пользователями и самой операционной системой;

– защита объектов от повторного использования – способность системы предотвратить доступ пользователя к информации ресурсов, с которыми до этого работал другой пользователь.

Идентификация пользователей. Для защиты данных Windows использует следующие основные механизмы: аутентификация и авторизация пользователей, аудит событий в системе, шифрование данных, поддержка инфраструктуры открытых ключей, встроенные средства сетевой защиты. Эти механизмы поддерживаются такими подсистемами Windows как LSASS (Local Security Authority Subsystem Service, подсистема локальной аутентификации), SAM (Security Account Manager, диспетчер локальных записей безопасности), SRM (Security reference Monitor, монитор состояния защиты), Active Directory (служба каталогов), EFS (Encrypting File System, шифрующая файловая система) и др.

Защита объектов и аудит действий с ними в ОС Windows организованы на основе избирательного (дискреционного) доступа, когда права доступа (чтение, запись, удаление, изменение атрибутов) субъекта к объекту задается явно в специальной матрице доступа. Для укрупнения матрицы пользователи могут объединяться в группы. При попытке субъекта (одного из потоков процесса, запущенного от его имени) получить доступ к объекту указываются, какие операции пользователь собирается выполнять с объектом. Если подобный тип доступа разрешен, поток получает описатель (дескриптор) объекта и все потоки процесса могут выполнять операции с ним. Подобная схема доступа, очевидно, требует аутентификации каждого пользователя, получающего доступ к ресурсам и его надежную идентификацию в системе, а также механизмов описания прав пользователей и групп пользователей в системе, описания и проверки дискреционных прав доступа пользователей к объектам. Рассмотрим, как в ОС Windows организована аутентификация и авторизация пользователей. Все действующие в системе объекты (пользователи, группы, локальные компьютеры, домены) идентифицируются в Windows не по именам, уникальность которых не всегда удается достичь, а по идентификаторам защиты (Security Identifiers, SID). SID представляет собой числовое значение переменной длины:

S – неизменный идентификатор строкового представления SID;

R – уровень ревизии (версия). На сегодня I – (identifier-authority) идентификатор полномочий. Представляет собой 48-битную строку, идентифицирующую компьютер или сеть, который(ая) выдал SID объекту. Возможные значения:

– 0 (SECURITY_NULL_SID_AUTHORITY) – используются для сравнений, когда неизвестны полномочия идентификатора;

– 1 (SECURITY_WORLD_SID_AUTHORITY) – применяются для конструирования идентификаторов SID, которые представляют всех пользователей. Например, идентификатор SID для группы Everyone (Все пользователи) – это S-1-1-0;

– 2 (SECURITY_LOCAL_SID_AUTHORITY) – используются для построения идентификаторов SID, представляющих пользователей, которые входят

на локальный терминал;

– 5 (SECURITY_NT_AUTHORITY) – сама операционная система, т. е. данный идентификатор выпущен компьютером или доменом.

Sn – 32-битные коды (количеством 0 и более) субагентов, которым было передано право выдать SID. Значение первых подчиненных полномочий общеизвестно. Они могут иметь значение:

– 5 – идентификаторы SID присваиваются сеансам регистрации для выдачи прав любому приложению, запускаемому во время определенного сеанса регистрации. У таких идентификаторов SID первые подчиненные полномочия установлены как 5 и принимают форму S-1-5-5-x-y;

– 6 – когда процесс регистрируется как служба, он получает специальный идентификатор SID в свой маркер для обозначения данного действия. Этот S - R - I - S0 - S1 - … - Sn - RID 5 идентификатор SID имеет подчиненные полномочия 6 и всегда будет S-1-5-6;

– 21 (SECURITY_NT_NON_UNIQUE) – обозначают идентификатор SID пользователя и идентификатор SID компьютера, которые не являются уникальными в глобальном масштабе;

– 32 (SECURITY_BUILTIN_DOMAIN_RID) – обозначают встроенные идентификаторы SID. Например, известный идентификатор SID для встроенной группы администраторов S-1-5-32-544;

– 80 (SECURITY_SERVICE_ID_BASE_RID) – обозначают идентификатор SID, который принадлежит службе.

Остальные подчиненные полномочия идентификатора совместно обозначают домен или компьютер, который издал идентификатор SID.

RID – 32-битный относительный идентификатор. Он является идентификатором уникального объекта безопасности в области, для которой был определен SID. Например, 500 – обозначает встроенную учетную запись Administrator, 501 – обозначает встроенную учетную запись Guest, а 502 – RID для билета на получение билетов протокола Kerberos.

Аудит событий входа в систему. Аудит попыток пользователя войти в систему с другого компьютера или выйти из нее, при условии, что этот компьютер используется для проверки подлинности учетной записи.

Аудит управления учетными записями. Аудит событий, связанных с управлением учетными записями на компьютере: создание, изменение или удаление учетной записи пользователя или группы; переименование, отключение или включение учетной записи пользователя; задание или изменение пароля.

Аудит доступа к службе каталогов. Аудит событий доступа пользователя к объекту каталога Active Directory, для которого задана собственная системная таблица управления доступом (SACL).

Аудит входа в систему. Аудит попыток пользователя войти в систему с компьютера или выйти из нее.

Аудит доступа к объектам. Аудит событий доступа пользователя к объекту – например, к файлу, папке, разделу реестра, принтеру и т. п., – для которого задана собственная системная таблица управления доступом (SACL).

Аудит изменения политики. Аудит фактов изменения политик назначения прав пользователей, политик аудита или политик доверительных отношений.

Аудит использования привилегий. Аудит попыток пользователя воспользоваться предоставленным ему правом.

Аудит отслеживания процессов. Аудит таких событий, как активизация программы, завершение процесса, повторение дескрипторов и косвенный доступ к объекту.

Аудит системных событий. Аудит событий перезагрузки или отключения компьютера, а также событий, влияющих на системную безопасность или на журнал безопасности. Решения об аудите конкретного типа событий безопасности принимаются в соответствии с политикой аудита локальной системы.

События аудита записываются в журналы следующих типов.

1 **Журнал приложений.** В журнале приложений содержатся данные, относящиеся к работе приложений и программ.

2 **Журнал безопасности.** Журнал безопасности содержит записи о таких событиях, как **успешные** и безуспешные попытки доступа в систему, а также о событиях, относящихся к использованию ресурсов.

3 **Журнал системы.** В **журнале** системы содержатся события системных компонентов Windows. Например, в журнале системы регистрируются сбои при загрузке драйвера или других системных компонентов при запуске системы.

4 **Журнал службы каталогов.** В **журнале** службы каталогов содержатся события, заносимые службой каталогов Windows (на контроллере домена AD).

5 **Журнал службы репликации.** В журнале службы репликации файлов содержатся события, заносимые службой репликации файлов Windows (на контроллере домена AD).

Шифрующая файловая система. Начиная с версии Windows 2000, в операционных системах семейства Windows NT поддерживается шифрование данных на разделах файловой системы NTFS с использованием *шифрующей файловой системы* (**Encrypted File System, EFS**). Основное ее достоинство заключается в обеспечении конфиденциальности данных на дисках компьютера за счет использования надежных симметричных алгоритмов для шифрования данных в реальном режиме времени. Для шифрации данных EFS использует симметричный алгоритм шифрования (AES или DESX) со случайным ключом для каждого файла (**File Encryption Key, FEK**). По умолчанию данные шифруются в Windows 2000 и Windows XP по алгоритму DESX, а в Windows XP с Service Pack 1 (или выше) и Windows Server 2003 – по алгоритму AES. В версиях Windows, разрешенных к экспорту за пределы США, драйвер EFS реализует 56-битный ключ шифрования DESX, тогда как в версии, подлежащей использованию только в США, и в версиях с пакетом для 128-битного шифрования длина ключа DESX равна 128 битам. Алгоритм AES в Windows использует 256-битные ключи. При этом для обеспечения секретности самого ключа FEK шифруется асимметричным алгоритмом RSA открытым ключом пользователя, результат шифрации FEK – **Data Decryption Field, DDF** – добавляется в заголовок зашифрованного файла.

Задание

1 Ознакомьтесь с теоретическими основами защиты информации в ОС семейства Windows в настоящих указаниях и конспектах лекций.

2 Запустите в программе **Oracle VM Virtualbox** виртуальную клиентскую машину Windows. Войдите в систему под учетной записью администратора, пароль узнайте у преподавателя.

3 Создайте учетную запись нового пользователя **testUser** в оснастке **«Управление компьютером» (compmgmt.msc)**. При создании новой учетной записи запретите пользователю смену пароля и снимите ограничение на срок действия его пароля. Создайте новую группу **«testGroup»** и включите в нее нового пользователя. Удалите пользователя из других групп. Создайте на диске **C:** папку **forTesting**. Создайте или скопируйте в эту папку несколько текстовых файлов (*.txt).

4 С помощью команды **runas** запустите сеанс командной строки (**cmd.exe**) от имени вновь созданного пользователя. Командой **whoami** посмотрите SID пользователя и всех его групп, а также текущие привилегии пользователя. Строку запуска и результат работы этой и ***всех*** следующих консольных команд копируйте в файл протокола лабораторной работы.

5 Убедитесь в соответствии имени пользователя и полученного SID в реестре Windows. Найдите в реестре, какому пользователю в системе присвоен SID **S-1-5-21-1957994488-492894223-170857768-1004** (используйте ключ **HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\ProfileList**).

6 Командой **whoami** определите перечень текущих привилегий пользователя **testUser**. В сеансе командной строки пользователя попробуйте изменить системное время командой **time**. Чтобы предоставить пользователю подобную привилегию, запустите оснастку **«Локальные параметры безопасности» (secpol.msc).** Добавьте пользователя в список параметров политики **«Изменение системного времени»** раздела **Локальные политики -> Назначение прав пользователя**. После этого перезапустите сеанс командной строки от имени пользователя, убедитесь, что в списке привилегий добавилась **SeSystemtimePriviege**. Попробуйте изменить системное время командой **time**. Убедитесь, что привилегия **«Завершение работы системы» (SeShutdownPrivilege)** предоставлена пользователю testUser. После этого попробуйте завершить работу системы из сеанса командной строки пользователя командой **shutdown –s**. Добавьте ему привелегию **«Принудительное удаленное завершение» (SeRemoteShutdownPrivilege)**. Попробуйте завершить работу консольной командой еще раз (отменить команду завершения до ее непосредственного выполнения можно командой **shutdown -a**).

7 Ознакомьтесь со справкой по консольной команде **cacls.** Используя эту команду, просмотрите разрешения на папку **c:\forTesting**. Объясните все обозначения в описаниях прав пользователей и групп в выдаче команды.

Разрешите пользователю **testUse**r запись в папку **forTesting**, но запретите запись для группы **testGroup**. Попробуйте записать файлы или папки в **forTesting** от имени пользователя **testUser**. Объясните результат. Посмотрите

эффективные разрешения пользователя **testUser** к папке **forTesting** в окне свойств папки.

Используя стандартное окно свойств папки, задайте для пользователя **testUser** такие права доступа к папке, чтобы он мог записывать информацию в папку **forTesting**, но не мог просматривать ее содержимое. Проверьте, что папка **forTesting** является теперь для пользователя **testUser** «слепой», запустив, например, от его имени файловый менеджер и попробовав записать файлы в папку, просмотреть ее содержимое, удалить файл из папки.

Для вложенной папки **forTesting\Docs** отмените наследование ACL от родителя и разрешите пользователю просмотр, чтение и запись в папку. Проверьте, что для пользователя папка **forTesting\Docs** перестала быть «слепой» (например, сделайте ее текущей в сеансе работы файлового менеджера от имени пользователя и создайте в ней новый файл).

Снимите запрет на чтение папки **forTesting** для пользователя **testUser**. Используя команду **cacls**, запретите этому пользователю доступ к файлам с расширением txt в папке **forTesting**. Убедитесь в недоступности файлов для пользователя.

Командой **cacls** запретите пользователю все права на доступ к папке **forTesting** и разрешите полный доступ к вложенной папке **forTesting\Docs**. Убедитесь в доступности папки **forTesting\Docs** для пользователя. Удалите у пользователя **testUser** привилегию **SeChangeNotifyPrivilege**. Попробуйте получить доступ к папке **forTesting\Docs**. Объясните результат.

Запустите файловый менеджер от имени пользователя **testUser** и создайте в нем папку **newFolder** на диске C. Для папки **newFolder** очистите весь список ACL командой **cacls**. Попробуйте теперь получить доступ к папке от имени администратора и от имени пользователя. Кто и как теперь может вернуть доступ к папке? Верните полный доступ к папке для всех пользователей.

Создайте в разделе **HKLM\Software** реестра раздел **testKey**. Запретите пользователю **testUser** создание новых разделов в этом разделе реестра. Создайте для раздела **HKLM\Software\testKey** SACL, позволяющий протоколировать отказы при создании новых подразделов, а также успехи при перечислении подразделов и запросе значений (предварительно проверьте, что в локальной политике безопасности соответствующий тип аудита включен). Попробуйте от имени пользователя **testUser** запустить **regedit.exe** и создать раздел в **HKLM\Software**. Убедитесь, что записи аудита были размещены в журнале безопасности (**eventvwr.msc**).

8 Шифрование файлов и папок средствами EFS.

От имени пользователя **testUser** зашифруйте какой-нибудь файл на диске. Убедитесь, что после этого был создан сертификат пользователя, запустив оснастку **certmgr.msc** от имени пользователя (раздел **Личные**). Просмотрите основные параметры сертификата открытого ключа пользователя **testUser** (срок действия, используемые алгоритмы). Установите доверие к этому сертификату в вашей системе.

Создайте в папке **forTesting** новую папку **Encrypt**. В папке **Encrypt** создайте или скопируйте в нее текстовый файл. Зашифруйте папку **Encrypt** и

все ее содержимое из меню свойств папки от имени администратора. Попробуйте просмотреть или скопировать какой-нибудь файл этой папки от имени пользователя **testUser**. Объясните результат. Скопируйте зашифрованный файл в незашифрованную папку (например, **forTesting**). Убедитесь, что он остался зашифрованным. Добавьте пользователя **testUser** в список имеющих доступа к файлу пользователей в окне свойств шифрования файла. Повторите попытку получить доступ к файлу от имени пользователя **testUser**.

Создайте учетную запись нового пользователя **agentUser**, сделайте его членом группы Администраторы. Определите для пользователя **agentUser** роль агента восстановления EFS. Создайте в папке **forTesting** новый текстовый файл с произвольным содержимым. Зашифруйте этот файл от имени пользователя **testUser**. Убедитесь в окне подробностей шифрования файла, что пользователь **agentUser** является агентом восстановления для данного файла.

Контрольные вопросы

1 К какому классу безопасности относится ОС Windows по различным критериям оценки?

2 Каким образом пользователи идентифицируются в ОС Windows?

3 Что такое списки DACL и SACL?

4 Перечислите, каким образом можно запустить процесс от имени другого пользователя.

5 Как происходит проверка прав доступа пользователя к ресурсам в ОС Windows?

6 Что такое маркер безопасности, и какова его роль в модели безопасности Windows?

7 Как с использованием команды cacls добавить права на запись для всех файлов заданной папки?

8 Какие события подлежат аудиту в ОС Windows?

9 Каким образом шифруются файлы в файловой системе EFS? Что такое FEK? DDF? DDR?

10 Какие алгоритмы шифрования используются в EFS?

9 Лабораторная работа № 9. Администрирование DNS-сервера в ОС Windows

Цель работы: изучить понятие доменных имен, разобраться с назначением и принципом функционирования службы доменных имен (DNS), ознакомиться с базовыми понятиями протокола DNS, научиться устанавливать и настраивать простейший вариант DNS сервера.

Домен – область (ветвь) иерархического пространства доменных имён сети Интернет, которая обозначается уникальным доменным именем.

Доменное имя – символьное имя домена. Должно быть уникальным в рамках одного домена. Полное имя домена состоит из имён всех доменов, в которые он входит, разделённых точками. Например, полное имя www.tu-bryansk.ru. (с точкой в конце) обозначает домен третьего уровня www, который входит в домен второго уровня tu-bryansk, который входит в домен .ru, который входит в корневой домен. Доменное имя служит для адресации узлов сети Интернет и расположенных на них сетевых ресурсов (веб-сайтов, серверов электронной почты, сетевых сервисов) в удобной для человека форме.

Доменная зона – совокупность доменных имён определённого уровня, входящих в конкретный домен. Например, зона ctam.tu-bryansk.ru. означает все доменные имена третьего уровня в этом домене. Термин «доменная зона» в основном применяется в технической сфере, при настройке DNS-серверов (поддержание зоны, делегирование зоны, трансфер зоны).

Для обеспечения уникальности и защиты прав владельцев доменные имена 1-го и 2-го (в отдельных случаях и 3-го) уровней можно использовать только после их регистрации, которая производится уполномоченными на то регистраторами. Сведения о владельце (администраторе) того или иного регистрируемого домена общедоступны. Их можно узнать, воспользовавшись службой «whois» – например, http://www.ripn.net:8080/nic/whois.

Домены верхнего уровня общего назначения:

.aero – для субъектов авиатранспортной индустрии;

.biz – только коммерческие организации;

.cat – для использования каталанским языковым и культурным сообществом;

.com – коммерческие организации (без ограничений);

.coop – кооперативы;

.edu – высшие учебные заведения, признаваемые в качестве таковых Департаментом образования США;

.info – информационные ресурсы (без ограничений);

.jobs – кадровые агентства;

.mobi – для продавцов и поставщиков мобильного контента и услуг, связанных с мобильной связью;

.museum – музеи;

.name – физические лица;

.net – организации имеющие отношение к функционированию Интернета (без ограничений);

.org – некоммерческие организации (без ограничений);

.pro – сертифицированные профессионалы и смежные темы;

.travel – для субъектов туристического бизнеса.

Для удобство распределения и назначения доменных имен для каждой из стран были выделены собственные (в основном двухсимвольные) домены верхнего уровня. Правда, это вовсе не означает обязательную привязку серверов в данных доменных к их географическому расположению. Примеры доменов первого уровня для стран:

.au – Australia (Австралия);

.be – Belgium (Бельгия);
.ru – Russia (Россия);
.ua – Ukraine (Украина);
.uk – United Kingdom (Англия).

Первоначально преобразование между доменными и IP-адресами производилось с использованием специального текстового файла DHOSTS.TXT, который составлялся централизованно и обновлялся вручную на каждой из машин сети. С ростом сети возникла необходимость в эффективном, автоматизированном механизме, которым и стала DNS (Domain Name System) – система доменных имен.

Примечание – На самом деле на каждой сетевой машине имеется текстовый файл hosts (Windows – %windir%\system32\drivers\etc\hosts; *nix – /etc/hosts), в котором можно самостоятельно сопоставлять с некоторым IP-адресом доменные имена. Правда, эти действия валидны только для текущей машины.

Функции DNS. Существуют два принципиально разных способа идентификации хостов: с помощью имен и с помощью IP-адресов. Имя хоста удобно для людей в силу своей мнемоничности, а IP-адрес, являющийся компактной числовой величиной фиксированного размера, проще обрабатывать прикладными программами и маршрутизаторами. Для того чтобы установить связь между этими двумя идентификаторами, используется система доменных имен. DNS представляет собой, с одной стороны, базу данных, распределенную между иерархически структурированными серверами имен, и, с другой стороны, протокол прикладного уровня, организующий взаимодействие между хостами и серверами имен для выполнения операций преобразования.

DNS функционирует на принципе делегирования полномочий. Каждая машина либо знает ответ на вопрос, либо знает, кого спросить. При правильном функционировании система замкнута, т. е. если запрошенная информация имеется у кого-либо, то она будет найдена и сообщена клиенту, либо, если вопрос не имеет ответа, клиент получит сообщение о невозможности получения ответа на вопрос.

Обратный DNS-запрос (Reverce DNS). DNS используется в первую очередь для преобразования символьных имён в IP-адреса, но он также может выполнять обратный процесс. Для этого используются уже имеющиеся средства DNS. Дело в том, что с записью DNS могут быть сопоставлены различные данные, в том числе и какое-либо символьное имя. Существует специальный домен in-addr.arpa., записи в котором используются для преобразования IP-адресов в символьные имена. Например, для получения DNS-имени для адреса 192.168.128.5 можно запросить у DNS-сервера запись 5.128.168.192.in-addr.arpa, и тот вернёт соответствующее символьное имя. Обратный порядок записи частей IP-адреса объясняется тем, что в IP-адресах старшие биты расположены в начале, а в символьных DNS-именах старшие (находящиеся ближе к корню) части расположены в конце.

Одна из проблем состоит в том, что обратную зону можно выделить только на сеть класса A, B или C (на 16777216, 65536 или 256 адресов соответственно) и никак иначе (маски здесь не работают).

Задание

1 Выясните примерное географическое месторасположение корневых серверов имен (можно воспользоваться одним из сервисов whois).

2 Настройте DNS сервер на базе Windows Server 2016 (в качестве forward сервера рекомендуется использовать DNS-сервера 192.168.128.1 или 192.168.128.5). Настройка и управление DNS-сервером осуществляется через консоль управления dnsmgmt.

3 Проверьте работоспособность настроенного сервиса.

4 При помощи сниффера wireshark исследуйте механизм работы утилиты nslookup.

5 Сделайте выводы. Подготовьте отчет с результатами проделанной работы.

10 Лабораторная работа № 10. Маршрутизация в ОС Windows. Межсетевое экранирование в Windows

Цель работы: получить сведения о маршрутизации и научиться добавлять маршруты в таблицу маршрутизации.

В сетях, основанных на протоколе IP, концепция маршрутизации является одной из важных. Она создает или разбивает сеть. Неправильная конфигурация маршрутизации способна вывести из строя сеть.

Маршрутизация – технология определения пути доставки (маршрута) пакетов.

Каждая операционная система, поддерживающая стек TCP/IP, имеет маршрутизатор и таблицу маршрутизации.

Таблица маршрутизации используется только тогда, когда определяется, как доставлять пакеты.

Маршрутизация должна быть сконфигурирована корректно на обоих концах связи и на каждом участке между ними.

Для определения пути доставки пакета используется таблица маршрутизации. Пример таблицы маршрутизации можно получить командой route с параметром print (рисунок 1).

Активные маршруты:

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.4.1	192.168.4.7	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.4.0	255.255.255.0	192.168.4.7	192.168.4.7	1
192.168.4.7	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.4.255	255.255.255.255	192.168.4.7	192.168.4.7	1
224.0.0.0	224.0.0.0	192.168.4.7	192.168.4.7	1
255.255.255.255	255.255.255.255	192.168.4.7	192.168.4.7	1
Основной шлюз:	192.168.1.1			

Рисунок 1 – Пример таблицы маршрутизации

В общем случае для маршрутизации используется следующий алгоритм. Из пакета извлекается IP-адрес назначения пакета и производится попытка сопоставить его с адресом назначения (Сетевой адрес) каждого элемента таблицы маршрутизации, пока не найдется наилучшее совпадение. Если совпадений не найдено, то пакет удаляется и отправителю пакета может отправиться сообщение об ошибке. Сравнение производится с тремя порциями информации: Сетевой адрес (Network Destination), Маска сети (Netmask) и IP-адрес назначения пакета.

В основном, производится побитная операция AND между IP-адресом получателя и Маской сетии (Netmask): если полученное значение равно Сетевому адресу (Network Destination), то считается, что совпадение найдено.

Для работы с таблицей маршрутизации используется стандартная утилита ROUTE, которая выводит на экран и изменяет записи в локальной таблице IP-маршрутизации. Запущенная без параметров, команда route выводит справку. Назначение параметров команды ROUTE приведено в таблице 2.

Таблица 2 – Назначение параметров команды ROUTE

Параметр	Описание
add	Добавление маршрута
change	Изменение существующего маршрута
delete	Удаление маршрута или маршрутов
print	Печать маршрута или маршрутов

Пример добавления маршрута приведен на рисунке 2.

route ADD 172.25.0.0 MASK 255.255.0.0 192.168.1.8 METRIC

Рисунок 2 – Строка для добавление маршрута

Задание

Создайте новый маршрут для вашего компьютера и проследите его.

1 Запустите виртуальную машину и загрузите ОС Windows.

2 Откройте консоль (Пуск/Программы/Стандартные/Командная строка).

3 Определите IP-адрес вашего компьютера с помощью утилиты ipconfig.

4 Просмотрите таблицу маршрутизации на вашем компьютере: выведите справку по команде route (для этого необходимо ввести команду и нажать клавишу ENTER); выведите таблицу маршрутизации командой route с параметром PRINT: route PRINT; запомните маршрут по умолчанию (первая строка).

5 Проследите работу маршрутизатора с помощью утилиты TRACERT, отправив пакеты на узел www.opennet.ru. Введите: tracert www.opennet.ru.

6 Добавьте в таблицу маршрутизации компьютера строку для пересылки пакетов в сеть 172.21.0.0 (маска 255.255.0.0) через сетевой интерфейс компьютера. Введите:

route add 172.21.0.0 mask 255.255.0.0 192.168.1.4 METRIC 3.

7 Проверьте работу внесенных вами изменений с помощью утилиты TRACERT.

Контрольные вопросы

1 Назовите типы маршрутов.
2 Что такое таблица маршрутизации?
3 Что такое метрика в таблице маршрутизации?
4 Что такое шлюз?
5 Что такое адрес шлюза?

11 Лабораторная работа № 11. Обеспечение доступа в сеть Интернет

Цель работы: изучить различные варианты подключения к сети Интернет через локальную сеть, используя программные средства.

Постановка задачи. Имеется локальная сеть (Workstantion 1 – Workstantion 2), представленная на рисунке 3. На компьютере (шлюзе), через который планируется подключение локальной сети к Интернету, необходимо наличие двух сетевых адаптеров (подключений). Требуется обеспечить доступ к сети Интернет со всех рабочих станций.

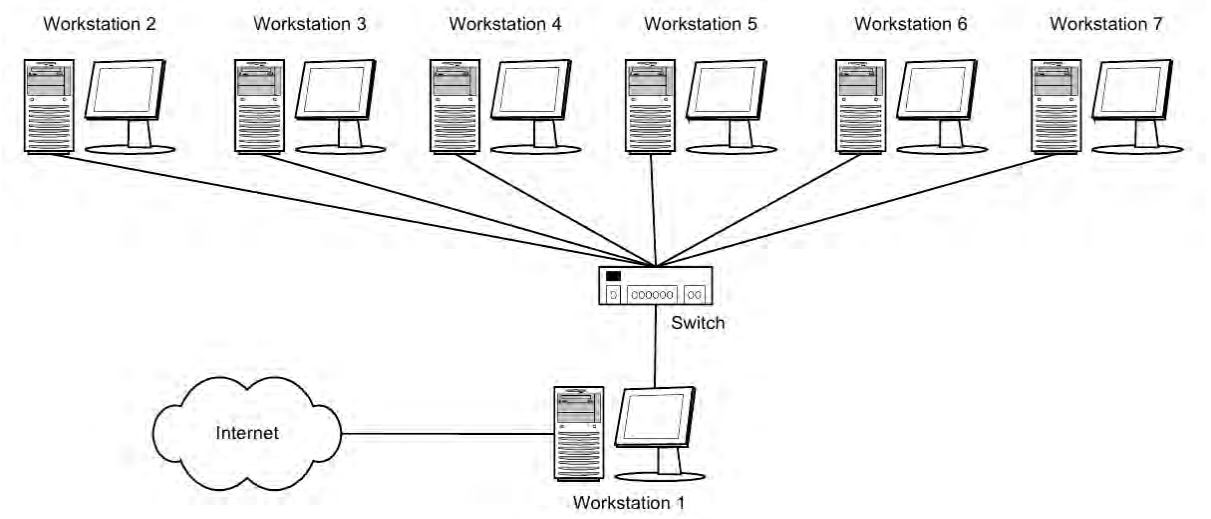

Рисунок 3 – Локальная сеть

Имеются три основных варианта подключения локальной сети к Интернету: «прямое» IP-подключение, подключение через NAT, подключение

через прокси-сервер. Рассмотрим преимущества, недостатки и область применения каждого метода, а также некоторые возникающие нюансы. 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. Наша задача – дать компьютерам локальной сети доступ к Интернету через подключенный к нему компьютер. Далее этот компьютер мы будем называть 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 Интернету. Для того, чтобы ваша локальная сеть была полноценно подключена к Интернету, должны соблюдаться как минимум три условия: каждая машина в локальной сети должна иметь «реальный», IP-адрес в сети Интернет; эти адреса должны быть не любыми, а выделенными вашим провайдером для вашей локальной сети (скорее всего, это будет подсеть класса С); 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 Ваша локальная сеть становится как бы частью Интернета. Собственно, это тот способ подключения, которым подключены к Интернету сами Интернет-провайдеры и хостинг-провайдеры. В 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 вы можете организовать в своей сети сервисы, доступные из Интернета – ведь при данном подключении не только Интернет полностью доступен из вашей сети, но и ваша сеть – из Интернета, т. к. является его частью. Однако такая «прозрачность» вашей 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:

– блокировку доступа извне к определенным TCP/IP-сервисам локальной сети;

– блокировку доступа к определенным компьютерам локальной сети, таким образом, можно запретить доступ извне ко всем машинам, кроме определенных серверов, предназначенных для доступа в Интернет;

– защиту от троянских программ на сетевом уровне.

Несмотря на универсальность такого метода подключения локальной сети к Интернету, этот метод имеет недостатки. Его 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, к тому же эту плату надо вносить периодически.

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

Подключение через NAT (IP-маскарадинг). 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 Интернета, нужно, чтобы: IP-пакеты, адресованные серверу в Интернете, смогли его достигнуть; ответные IP-пакеты, идущие от сервера Интернета 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» IP-адреса в Интернете! Как же они могут получать IP-пакеты из Интернета?

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 видим, работа NAT-сервера прозрачна для машин локальной сети (как и работа обычного IP-маршрутизатора). Единственным принципиальным ограничением этого метода подключения локальной сети к Интернету является невозможность установить входящее TCP-соединение из Интернета 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. Это 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, скажем, 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 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 конктретным 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 Интернету напрямую), это дает выигрыш в производительности, а при повременной оплате, соответственно, экономить время он-лайн. 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).

Задание

На виртуальных машинах осуществите все три способа подключения к Интернету («Прямое» IP-подключение к Интернету, подключение через NAT и подключение через прокси-сервер).

Контрольные вопросы

1 Назовите способы подключения к Интернету.
2 Опишите способ подключения к Интернету через NAT.
3 Опишите способ подключения к Интернету через прокси-сервер.
4 Опишите способ подключения к Интернету через Internet Connection Sharing.

Требования к отчетам и защите лабораторных работ

Лабораторная работа направляется на доработку, если количество ошибок и погрешностей позволяет отнести её к низкому уровню соответствия. Допустимые погрешности и ошибки при определении учебных достижений представлены в таблице 3.

Таблица 3 – Допустимые погрешности и ошибки при определении учебных достижений студентов

Шкала соответствия	Уровень соответствия	Балл	Количество ошибок, погрешности / несущественные / существенные
Соответствие	Высокий	5	3/2/0
	Средний	4	6/3/2
	Минимально необходимый	3	7/4/3
Несоответствие	Низкий	2	8/5/4

Погрешностями при определении учебных достижений считаются:

– неточные выражения в отчете по лабораторной работе;

– нерациональные, но правильные приемы, используемые для решения поставленных задач;

– незначительные погрешности при определении параметров.

К несущественным ошибкам относятся:

– неточности определения характеристик и параметров;

– нерациональный способ решения задачи или план ответа (нарушение логики изложения материала, подмена основных понятий второстепенными);

– несоблюдение требований ГОСТа и небрежное оформление отчета по лабораторной работе и графического материала.

К существенным ошибкам относятся:

– подмена понятий в изложении основных понятий;

– незнание фундаментальных понятий серверных операционных систем;

– неумение создавать учетные записи и диагностировать серверные операционные системы;

– неумение в ответе объяснить материал, делать выводы и обобщения, неумение письменно оформить материал;

– незнание клиентских операционных систем.

Оформление рисунков и схем в отчете по лабораторным работам должно соответствовать требованиям ГОСТ 2.105–95. Текстовая часть отчета выполняется либо чертежным шрифтом по ГОСТ 2304–81 с высотой букв не менее 5 мм либо машинным способом шрифтом Times с высотой букв 14 пунктов через одинарный интервал.

Формулы, иллюстрации и таблицы нумеруются в пределах отчета. Обозначения переменных и параметров, принятых в формулах, должны быть расшифрованы сразу после написания формулы. При этом указываются единицы

измерения переменных и параметров.

Рисунки, графики и таблицы сопровождаются наименованиями, отображающими их содержание (например, Рисунок 1 – Физическая топология компьютерной сети). Если на одном рисунке изображено несколько графиков различных процессов, то каждый график должен иметь отдельное обозначение, которое необходимо расшифровать в поясняющих данных к рисунку. Поясняющие данные помещаются под рисунком перед его наименованием.

Отчет или его часть могут быть представлены в электронном виде по согласованию с преподавателем.

Общие требования к содержанию отчета

1 Тема, цель работы.
2 Постановка задачи.
3 Вариант задания с исходными данными.
4 Выполненное задание согласно варианту: алгоритм решения поставленной задачи; результаты выполненного задания.
5 Результаты тестирования задания.
6 Выводы по теме лабораторной работы.

Список литературы

1 **Кенин, А.** Самоучитель системного администратора / А. Кенин. – Санкт-Петербург : БХВ-Петербург, 2012. – 512 с.

2 Microsoft Windows Server 2012. Полное руководство / Р. Моримото [и др.]. – Москва : Вильямс, 2013. – 1456 с.

3 **Поляк-Брагинский, А.** Администрирование сети на примерах / А. Поляк-Брагинский. – Санкт-Петербург : БХВ-Петербург, 2012. – 432 с.

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

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

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