

4757

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***ИНФОРМАЦИОННЫЕ СЕТИ И
ТЕЛЕКОММУНИКАЦИИ***

Методические указания к лабораторным работам

Рязань 2014

УДК 681. 326 (075)

Информационные сети и телекоммуникации: методические указания к лабораторным работам/ Рязан. гос. радиотехн. ун-т.; сост.: Е.М.Кузьмина, Т.А. Куличенко, В.А. Лашин. Рязань, 2014. 32 с.

Содержат описание трех лабораторных работ по дисциплине «Информационные сети телекоммуникации» для изучения отдельных принципиальных вопросов построения сетей и некоторых приложений.

Предназначены для студентов направления 220700 «Автоматизация технологических процессов и производств».

Ил. 22. Библиогр.: 4 назв.

Адрес, IP-адрес, маска, сеть, топология, технология, HTML, Web-страница, гиперссылка, тег, шаблон

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра АИТП РГРТУ (зав. кафедрой д-р техн. наук, проф. А.К. Мусолин)

ОБМЕН ДАННЫМИ В ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ (ЛВС). НАСТРОЙКИ

Цель работы: знакомство с основными приемами работы в одноранговой локальной вычислительной сети (ЛВС) и видами применяемых адресов.

Топологии и технологии

Любое сетевое взаимодействие в ЛВС основано на соединении компьютеров с помощью кабельной системы (КС). Варианты исполнения КС могут быть различными даже при одинаковой технологии функционирования сети. ЛВС типа Ethernet стандартов 10 Base работают по шинной технологии с использованием случайного метода доступа CSMA/CD к общей среде передачи. Этот принцип работы может быть реализован применением различных топологий (рис. 1, 2). На рис. 1 показана традиционная шинная топология, в которой сетевые карты (сетевые адаптеры) всех узлов с помощью приемопередатчиков (трансиверов) подключены к общей для всех узлов кабельной системе [1]. Таким образом, сеть конфигурируется на двух разновидностях коаксиального кабеля: «толстом» и «тонком» Ethernet (стандарты 10Base-5 и 10Base-2).

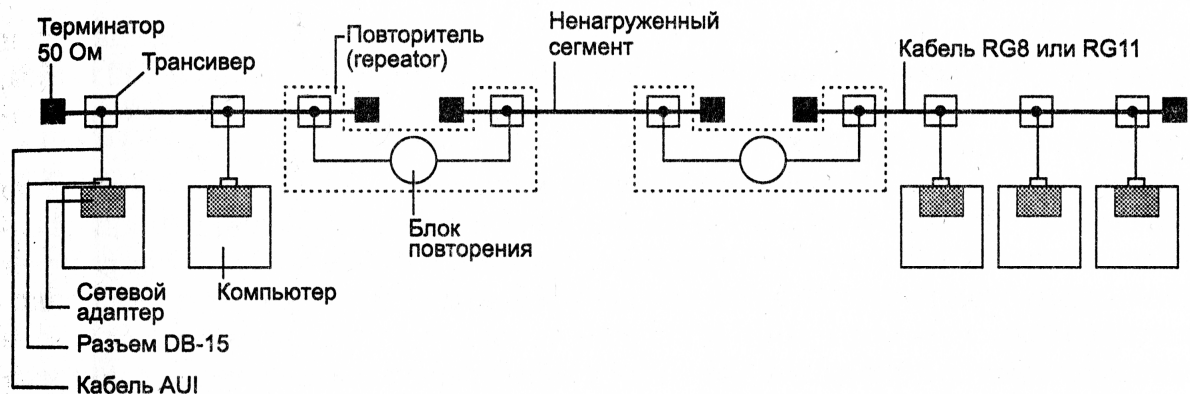


Рис. 1

В стандартах 10Base-T и 10Base-F каждый из узлов (рис. 2) подключается к одному из портов многопортового повторителя – **концентратора Hub** (хаб), работа которого построена по принципу: принятый на одном из портов сигнал хаб транслирует на выходы всех портов концентратора, за исключением того порта, на котором был принят этот сигнал. Соединение выполняется витыми парами (10Base-T) или волоконно-оптическим кабелем (10Base-F). По характеру подключения эта разновидность ЛВС физи-

чески выполняется как «звезда», а по алгоритму работы хаба – сохраняет все признаки шинной топологии.

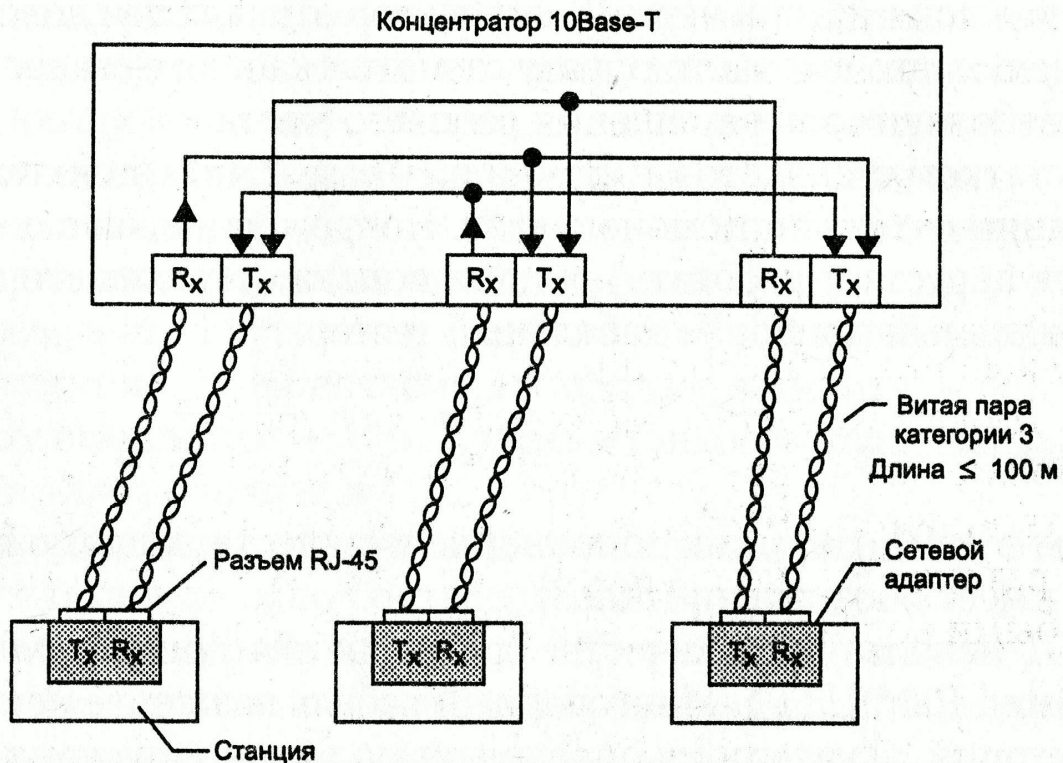


Рис. 2

Подобным же образом сеть Token Ring, работающая по технологии маркерного кольца, физически конфигурируется как «звезда». Принцип концентрации некоторой части межузловых соединений во внутренней структуре хаба может быть развит с целью достижения более высокого быстродействия за счет параллельной обработки внутрисетевого трафика подключением сегментов узлов к портам *коммутатора Switch* (свитч), но пока это находит применение при логической структуризации сети.

Обмен данными по сети Ethernet происходит при соблюдении следующего формата кадра:

Преамбула	Адрес получателя	Адрес отправителя	Длина поля данных	Поле данных	CRC
8 байт	6 байт	6 байт	2 байта	46-1800 байт	4 байта

Преамбула – это своего рода синхросигнал, в течение 7 байт которого передается последовательность чередующихся 1 и 0, которая заканчивается (в восьмом байте) начальным ограничителем кадра 10101011. После преамбулы приемник готов анализировать Адрес получателя сообщений.

Адреса получателя и отправителя – это уникальные адреса для каждой из сетевых карт, задаваемые фирмой-изготовителем. Это так называемые **физические** адреса. Расположение этих адресов в начале кадра убеждает в том, что знать их просто необходимо независимо от места нахождения получателя и отправителя в общей распределенной межсетевой структуре.

CRC – поле защиты информации циклическим кодом.

Вообще говоря, физических адресов могло бы быть достаточно для обмена внутри небольшой изолированной сети, но совершенно недостаточно для организации сеанса между узлами, находящимися в разных подсетях. Это происходит по той простой причине, что невозможно отслеживать миллиарды адресов адаптеров, состав которых динамично меняется из-за пополнения сети новыми узлами или исключения какой-либо части из них, замены сетевого оборудования и т.д.

Поэтому адресация узлов независимо от их расположения выполняется по типовой процедуре, когда каждому узлу в дополнение к физическому адресу присваивается еще **сетевой** адрес, однозначно определяющий и сеть, в которой каждый узел находится, и адрес самого узла в этой сети [2, 3]. Поскольку определением маршрута доставки сообщения в стеке протоколов TCP/IP занимается протокол сетевого уровня IP, этот адрес часто называют IP-адресом и помещается он в заголовке этого протокола (рис. 3).

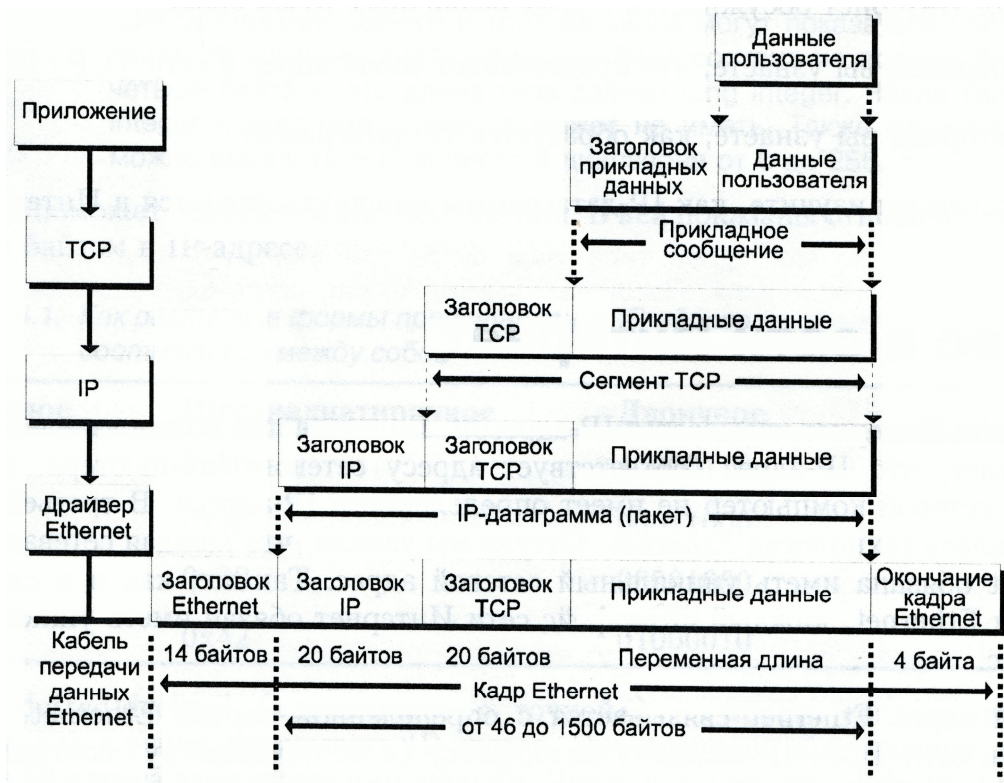


Рис. 3

Таким образом, в соответствии с физическим адресом каждый из адаптеров решает, принимать или не принимать ему сигнал, действующий на его входе, а по IP-адресу просто находится местоположение требуемого узла сети.

Так как IP-адрес получателя сообщений изначально известен (или может быть определен через службу доменных имен **DNS**), а физический адрес необходимо определить, в сетевом программном обеспечении предусмотрена стандартная процедура широковещательного ARP-запроса, смысл которого означает: «Узел с таким-то IP-адресом! Сообщите свой физический адрес». Несмотря на наличие других средств, таких, как хранение некоторых совокупностей адресов в кэш-памяти с последующим их извлечением при необходимости, определение физического адреса (так называемое разрешение аппаратного адреса) через широковещательный запрос является универсальным.

Структура и классы IP-адресов

Прежде всего, IP-адрес – это адрес не компьютера, а его сетевой карты. Поэтому, если компьютер имеет несколько сетевых карт, то он имеет столько же IP-адресов (рис. 4).

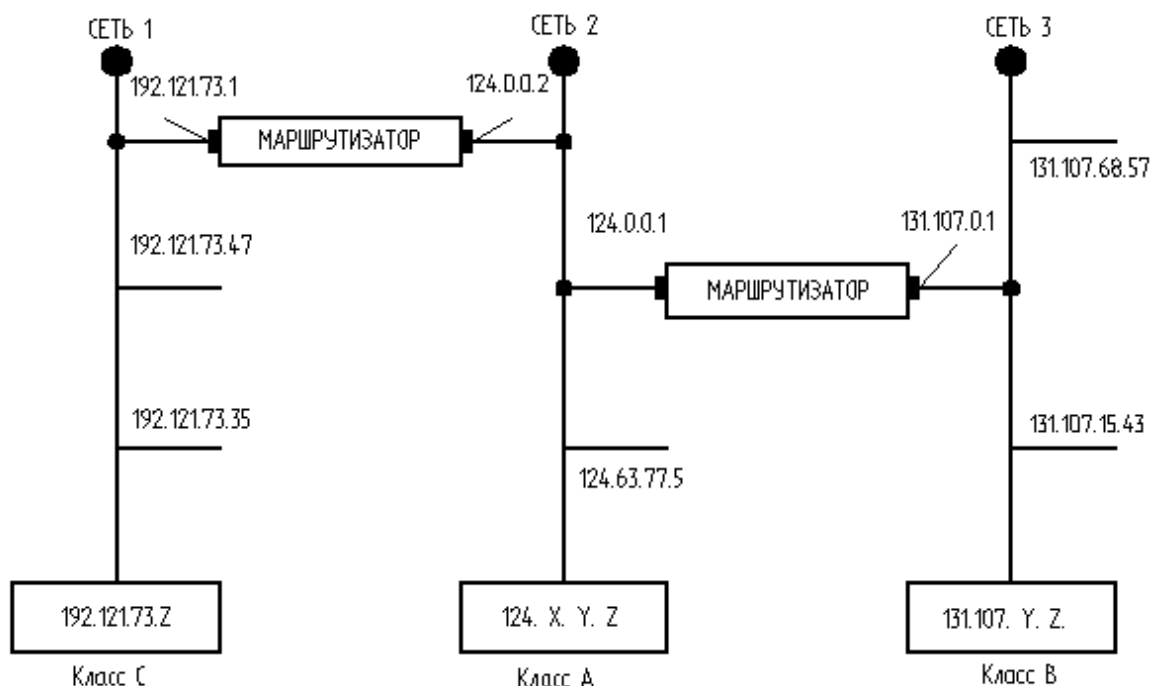


Рис. 4

IP-адрес пока состоит из 4 байт, и вот эту 32-разрядную двоичную комбинацию можно записать по-разному, например:

- в двоичном виде: 10000110 00011000 00001000 01000010;

- в десятичном: 2249721922;
- в шестнадцатеричном: 0x86180842;
- в десятичном с точкой: 134.24.8.66.

Из-за большего удобства восприятия принято IP-адрес записывать в виде: десятичное с точкой.

По своей структуре он состоит из двух частей: идентификатора (номера) сети и идентификатора узла, занимающего правую (младшую) часть адреса. Чтобы иметь возможность рационально распределять адресное пространство среди имеющихся сетей различного размера, применена система классификации адресов. Как видно из таблицы, для нумерации относительно небольшого числа очень крупных сетей ($N_{\max} = 127$), каждая из которых содержит до $M_{\max} = 16\,777\,216$ узлов, предусмотрен класс А. Нулевое значение старшего бита идентификатора сети определяет принадлежность к классу А.

Класс	Старшие биты	IP – адрес	Идентификатор сети	Идентификатор узла
A	0-----	w.x.y.z	w	x.y.z
B	10-----	w.x.y.z	w.x	y.z
C	110-----	w.x.y.z	w.x.y	z
D	1110----	Для широковещания		

Подобно этому класс В содержит до $N_{\max} = 16\,384$ сетей с числом узлов в каждой до $M_{\max} = 65\,536$, а класс С включает в себя $N_{\max} = 2\,097\,152$ сетей с $M_{\max} < 256$ узлов.

Определяя десятичное значение старшего байта идентификатора сети, можно установить по IP-адресу принадлежность к определенному классу.

Класс сети	Диапазон значений старшего байта
A	От 0 до 126
B	От 128 до 191
C	От 192 до 224
D	От 225 до 240

Здесь учтены дополнительные соглашения об использовании IP-адресов:

- если идентификатор сети состоит из одних нулей, это означает, что узлы назначения и отправления находятся в одной сети;

- если все разряды IP-адреса равны 1, то пакет рассылается всем узлам, находящимся в той же сети, что и отправитель, – это ограниченное широковещательное сообщение;
- если в поле номера узла стоят одни 1, то пакет предназначен всем узлам той сети, которая обозначена в поле идентификатора сети, – это широковещательное сообщение;
- если первый октет адреса равен 127, то адрес используется только для диагностических целей.

Из этих соглашений следует, что в реальной работе сети может встретиться применение адресов, полностью или по частям состоящих из одних нулей или одних единиц, но это связано с заранее оговоренными применениями, что не допускает использования их для адресации конкретных компьютеров.

В полной мере использовать информацию, содержащуюся в IP-адресе, можно только при наличии тщательно обработанного механизма для разделения областей идентификатора узла в общем битовом потоке IP-адреса. В современных сетях таким механизмом является использование **маски сети**.

Маска сети

Маска – это 32-битовая двоичная последовательность, в которой позиции, соответствующие идентификатору сети, заполнены единицами, а область, характеризующая идентификатор узла, - нулями.

Простым наложением маски на IP-адрес четко и достоверно определяется граница раздела между идентификаторами.

Существуют два способа назначения маски.

1. В тех случаях, когда сеть не делится на подсети, маска назначается по умолчанию в соответствии с классом сети.

Класс адресов	Биты, используемые для маски сети	Десятичная с точкой запись маски
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

Процедура разделения общего IP-адреса на идентификаторы сети и узла поясняется на примере десятичного с точкой варианта записи адреса для сети класса B следующим образом:

IP-адрес	131 . 107 . 16 . 200
Маска сети	255 . 255 . 0 . 0
Идентификатор сети	131 . 107
Идентификатор узла	16 . 200

Рассмотрим другой пример. Узел-отправитель, находящийся в сети класса В, пытается организовать соединение с узлом-получателем, расположение которого в межсетевом объединении может быть произвольным. Как должна поступить сеть с передаваемым пакетом? Для этого надо выделить идентификаторы сетей с помощью маски.

IP-адрес отправителя	10011001 10101010 00100101 10100011
Маска сети отправителя	11111111 11111111 00000000 00000000
Результат	10011001 10101010

IP-адрес получателя	11011001 10101010 10101100 11101001
Маска сети отправителя	11111111 11111111 00000000 00000000
Результат	11011001 10101010

Как видно, выделенные идентификаторы сетей различны. Это говорит о том, что отправитель и получатель расположены в разных сетях. Следовательно, сообщение должно быть направлено на адрес шлюза по умолчанию для выхода за пределы сети, которой принадлежит отправитель, и поиска сети, в которой расположен получатель. IP-адрес шлюза обязательно указывается при начальных установках сети.

2. Назначение маски в сетях, делящихся на подсети.

При объединении сетей с малым количеством рабочих станций назначение каждой из них номера по классу сети может быть нерационально с точки зрения использования адресов. В этих случаях более целесообразным может оказаться разделение одной сети соответствующего класса на ряд отдельных подсетей с независимой адресацией каждой из них. При этом часть поля идентификатора узлов используется для нумерации подсетей, а оставшаяся часть – нумерует узлы в составе каждой из них.

Назначение маски для такого применения производится в следующей последовательности:

- 1) пусть требуется организовать 6 подсетей. Для представления числа 6 в двоичной форме требуется 3 разряда;
- 2) заполняются эти три разряда единицами и присоединяются к маске разделяемой сети со стороны младшего разряда.

Для разделяемой на 6 подсетей сети класса В это примет вид:

IP-адреса подсетей	10110011	00100100	001	Идентификаторы узлов в каждой из 6 подсетей	
	10110011	00100100	010		
	10110011	00100100	011		
	10110011	00100100	100		
	10110011	00100100	101		
	10110011	00100100	110		
Маска подсетей	11111111	11111111	111	00000	00000000
Десятичная с точкой запись маски	255	255	224		0

Правильное назначение маски имеет большое значение при организации обмена. Если, например, два сегмента сети, соединенные маршрутизатором (рис. 5), оба будут иметь маску 255.255.0.0, то любая станция одного сегмента будет воспринимать любую станцию другого сегмента как абонента, находящегося с ним в одной сети 131.107. Поэтому данные на маршрутизатор передаваться не будут и межсегментный обмен будет невозможен.

Если изменить маску и взять ее в виде 255.255.255.0, то правый и левый сегменты будут восприниматься как различные сети с адресами 131.107.100 и 131.107.33 и все проблемы снимутся.



Рис. 5

Назначение маски не по умолчанию несколько видоизменяет адресацию по сравнению с делением сетей на классы. Действительно, если для IP-адреса 129.64.134.5 взята маска 255.255.128.0, то при записи в двоичной форме получится

	Двоичное		Десятичное с точкой
IP-адрес	10000001 01000000 1	0000110 00000101	129. 64.134.5
Маска	11111111 11111111 1	0000000 00000000	255.255.128.0
Номер сети	10000001 01000000 1		129. 64.128.0
Номер узла		0000110 00000101	0. 0. 6.5

Можно заметить, что по структуре IP-адреса сеть принадлежала бы (без маски) классу В, так как старший октет $129 > 128$. По виду маски она напоминает больше сеть класса С, но ею не является из-за того, что величина старшего октета $129 < 192$.

Установки локальной вычислительной сети

Для организации локальной вычислительной сети необходимо, чтобы компьютеры имели в своем составе сетевые платы и были соединены сетевым шнуром. Каждому компьютеру в сети *обязательно* должен быть присвоен свой IP-адрес (при организации сети по IP-протоколу), сетевое имя и установлен доступ к ресурсам компьютера (дискам и принтерам). Эти действия выполняются с использованием значка *Сетевое окружение/Свойства/Подключение по локальной сети/Свойства*.

В списке компонент выбирается TCP/IP. Затем необходимо зайти на вкладку *Свойства*, где устанавливается адрес компьютера и маска подсети (рис. 6, 7). В IP-адресе компьютера последняя цифра обозначает порядковый номер компьютера в сети.

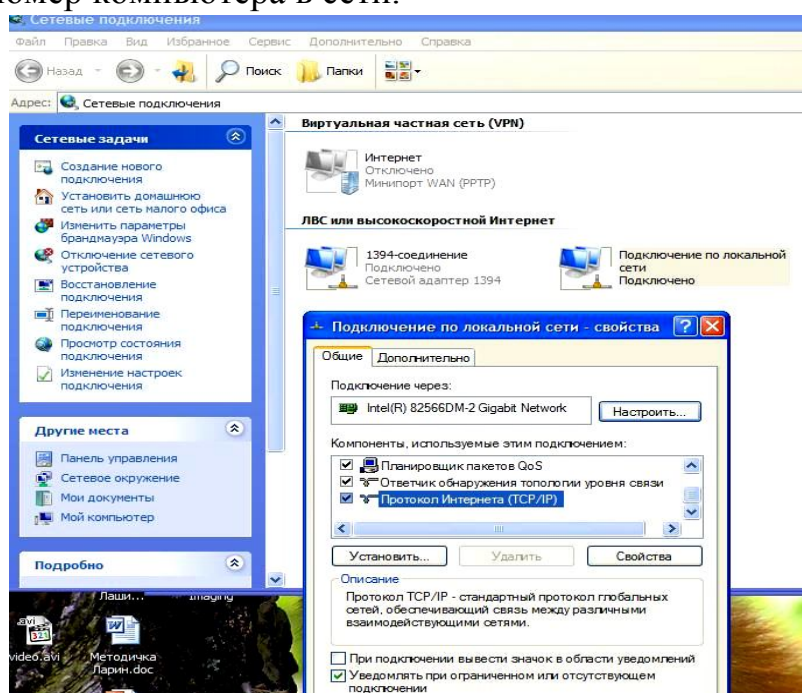


Рис. 6

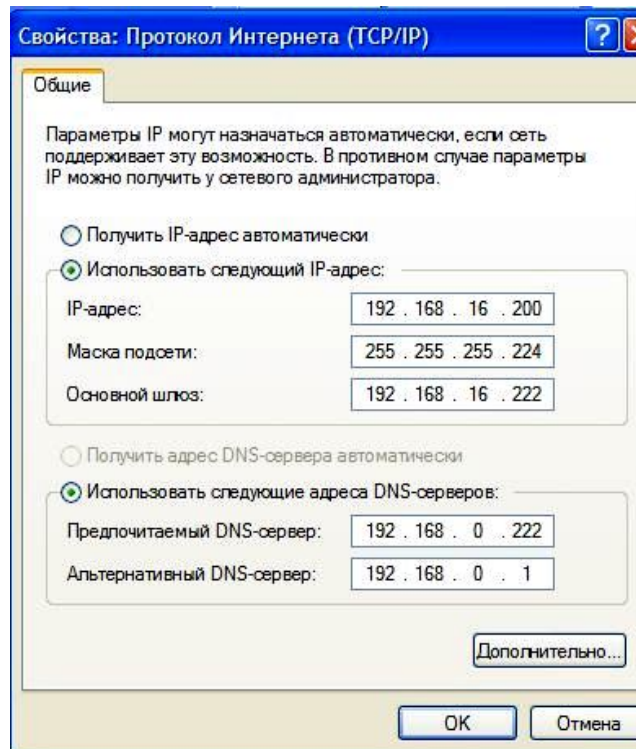


Рис. 7

Проверить сетевые установки можно командой **ipconfig** в режиме ввода команд. Для перехода в этот режим необходимо набрать *Пуск/ Выполнить/ cmd*, после этого вводить нужные команды. Один из вариантов ответа на команду **ipconfig** приведен на рисунке 8.

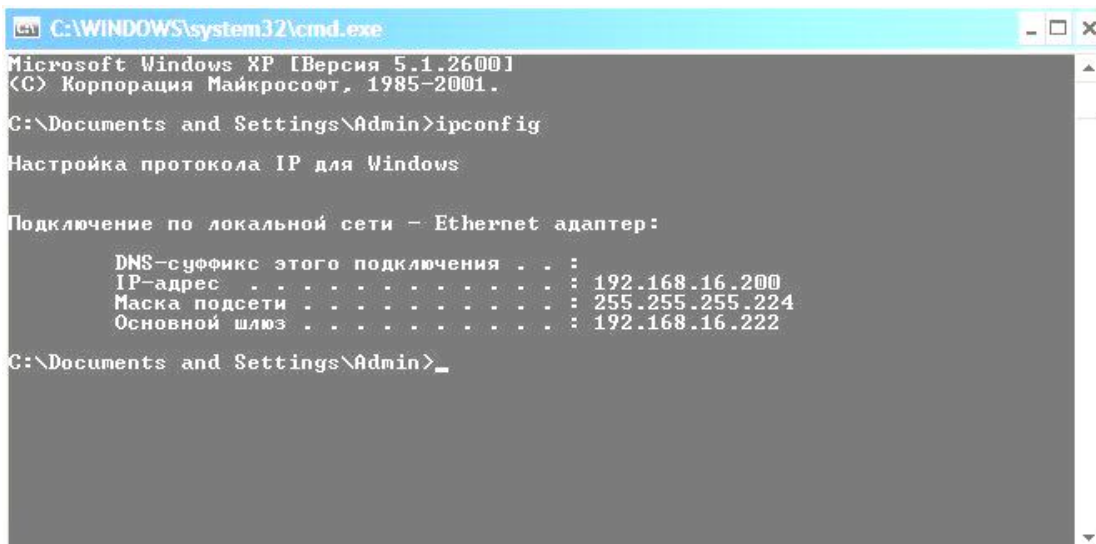


Рис. 8

Сетевые возможности Windows

Работа с сетевыми ресурсами и использование ресурсов вашего компьютера другими пользователями организована так же просто, как и работа с локальными ресурсами. В качестве локальных ресурсов компьютера ис-

пользуются диски, папки и принтеры. Их можно сделать доступными по сети другим пользователям в качестве сетевых ресурсов.

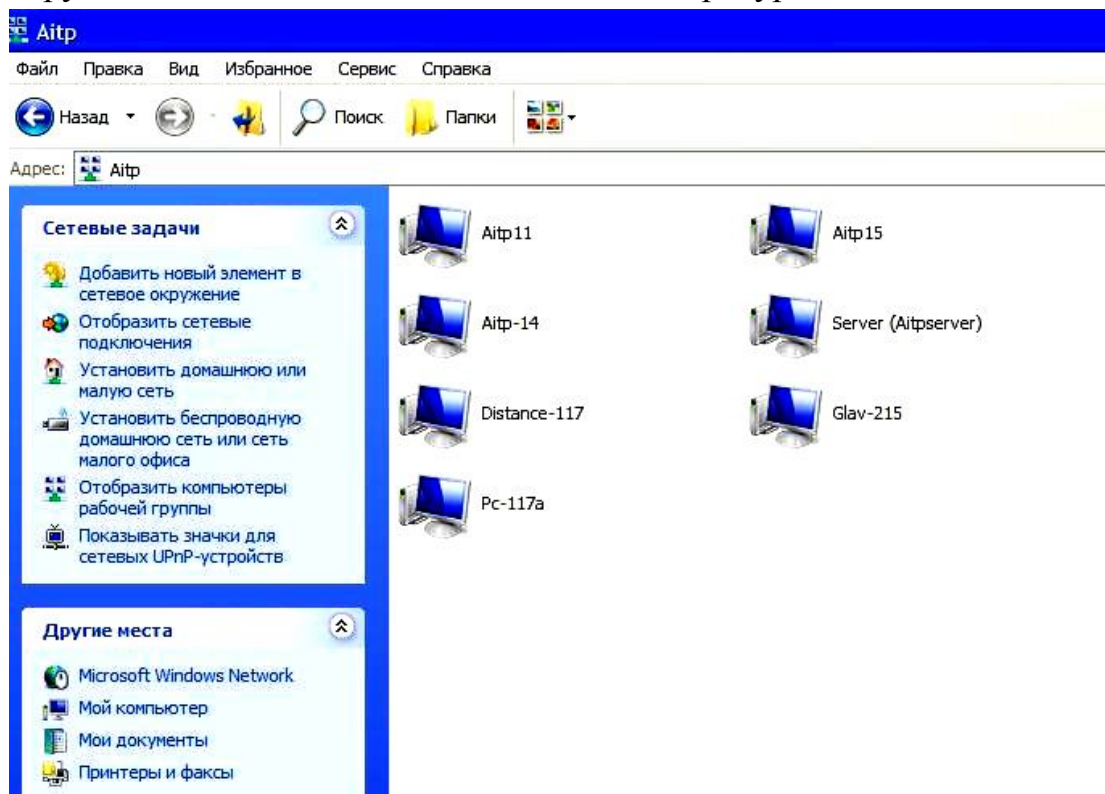


Рис. 9

Папка *Сетевое окружение* предоставляет в распоряжение пользователя сетевые ресурсы точно так же, как и приложение *Мой компьютер* открывает доступ к ресурсам локальной вычислительной машины. Двойной щелчок по значку *Сетевое окружение* открывает окно папки, в котором находятся значки всех компьютеров рабочей группы, состав которой определяется администратором сети.

На рис. 9 показана папка *Сетевое окружение* для локальной вычислительной сети при использовании значка под названием *Отобразить компьютеры рабочей группы*.

Для доступа к данным на любом компьютере сети, который рассматривается как удаленное хранилище совместно используемых файлов, достаточно произвести двойной щелчок по его значку. В результате в правой области окна с названием рабочей группы будут отображены все доступные в локальной сети папки выбранного компьютера.

Уровни доступа

Диски и папки могут быть назначены для совместного использования как с предоставлением всех прав, так и в режиме *Только чтение*. В последнем случае пользователю предоставляется возможность работы с хранящимися в папке программами и документами, но не разрешается сохранять их в этой папке. Для сохранения документа, извлеченного из сетевой

папки, открытой только для чтения, в качестве места назначения необходимо указывать локальную папку.

Если папка предназначена для совместного использования с полным доступом, то пользователь может делать с хранящимися там программами и документами то же самое, что и с файлами, хранящимися в локальных папках. При отсутствии ограничений доступа на уровне файлов пользователь может открывать, сохранять, переименовывать, удалять, перемещать и копировать объекты из этих папок так, словно они находятся на компьютере пользователя.

Совместное использование папок в сети

Для того чтобы сделать хранящуюся на собственном компьютере папку доступной для других пользователей сети, необходимо выбрать в контекстном меню объекта команду *Общий доступ и безопасность*. Меню вызывается щелчком правой кнопкой мыши по значку нужной папки.

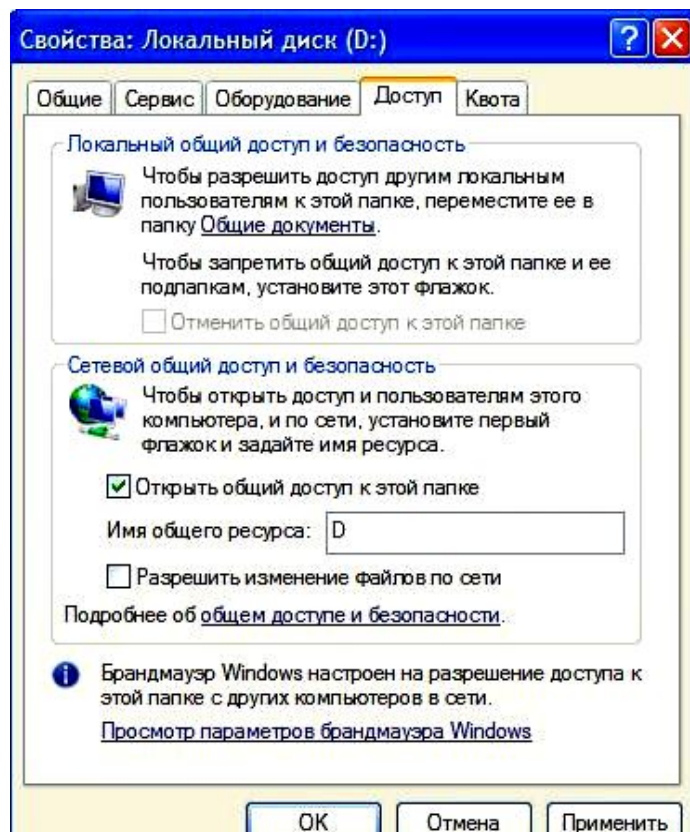


Рис. 10

Установка флажка *Открыть общий доступ к этой папке* (рис. 10) делает выбранную папку (или устройство) видимыми в сети. Можно установить имя общего ресурса, отличающееся от его имени на компьютере. Флажком *Разрешить изменение файлов по сети* открывается полный доступ к ресурсу или только чтение.

Изменение доступа к сетевым ресурсам возможно только из папки *Мой компьютер*. Если входить на свой компьютер из папки *Сетевое окружение*, то здесь будет отключена возможность изменения доступа.

Подключение сетевых дисков

Для того чтобы работать с сетевой папкой или сетевым диском как со своим, локальным, можно в папке *Сетевое окружение* вызвать контекстное меню нужного сетевого ресурса и выбрать пункт *Подключить сетевую папку*.

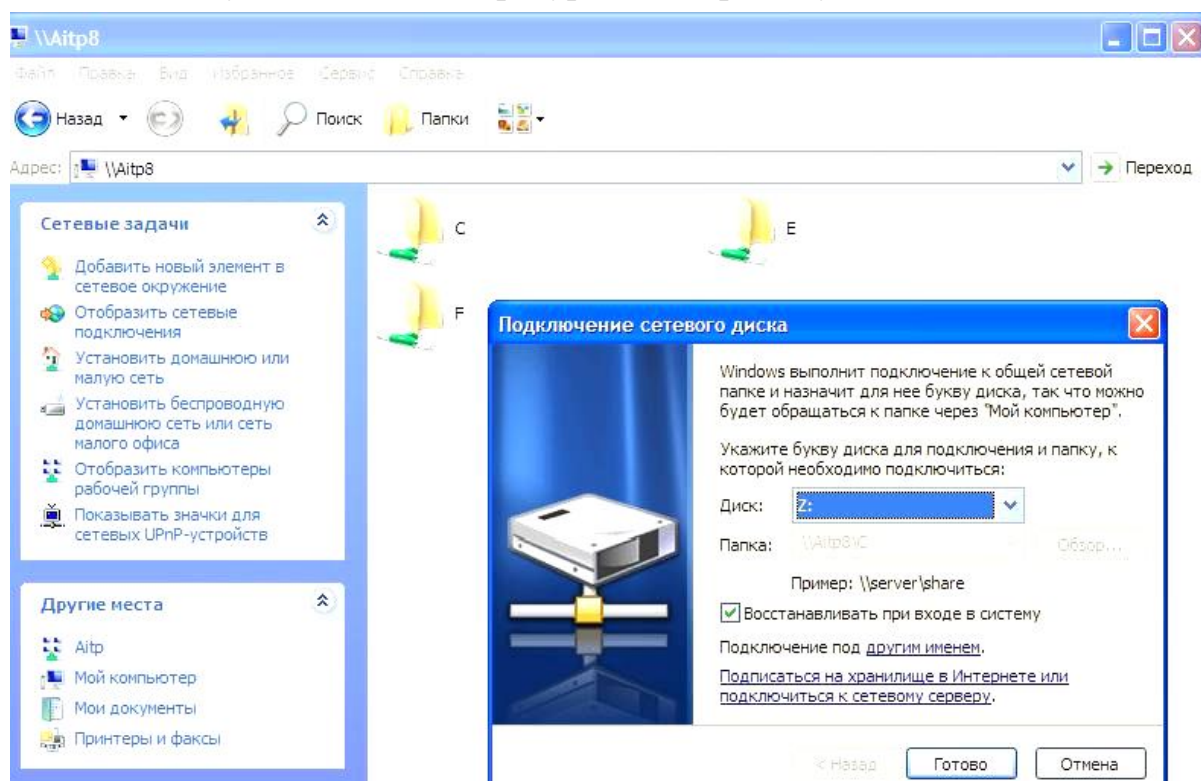


Рис. 11

вой диск. Из предложенного списка имен выбирается имя для сетевого диска или папки. После этого обращаться к сетевому ресурсу можно из папки *Мой компьютер*, где наряду с локальными дисками и папками появится новый ресурс с указанным именем.

После окончания работы с сетевым диском или папкой надо из его контекстного меню выбрать команду *Отключить сетевой диск*.

Задание на лабораторную работу

1. Войти в локальную сеть через значок *Сетевое окружение*.
2. Открыть содержимое любого компьютера сети.
3. Создать в папке *Студент/grXXXX* папку для совместного использования ее в сети. Установить доступ к папке различными способами. Усвоить разницу в доступе к папкам, возможности работы с такими папками.

4. Подключить папку или диск соседнего компьютера как сетевой ресурс.
5. Создать в своей папке текстовый файл и попытаться его скопировать на соседние компьютеры в сетевые папки с различным уровнем доступа.
6. Открыть сетевые папки других компьютеров с различным уровнем доступа и изменить содержимое находящихся там файлов.
7. Отключить сетевой диск.
8. В меню *Пуск/Выполнить* набрать команду CMD для перехода в командный режим ввода, запустить программу IPCONFIG для просмотра установленного IP- адреса своего компьютера и назначенной маски подсети.

Лабораторная работа № 2

СОЗДАНИЕ WEB-СТРАНИЦЫ СРЕДСТВАМИ ЯЗЫКА HTML

Цель работы: знакомство с простейшими элементами языка HTML, получение навыков создания Web-страниц.

Долгое время Интернет оставался уделом специалистов, обменивавшихся технической документацией и сообщениями электронной почты. С увеличением числа узлов и пользователей появилась необходимость в создании новых служб для удовлетворения потребностей в обмене информацией. В 1993 году была создана служба World Wide Web (WWW), основанная на пользовательском протоколе передачи данных HTTP и на особом формате представления данных – HTML. Документы, выполненные в этом формате, получили название Web-страниц.

Одновременно с введением концепции WWW была представлена программа Mosaic, обеспечивающая отправку запросов и прием сообщений в формате HTML. Эта программа стала первым в мире *Web-браузером*, т.е. программой для просмотра Web-страниц.

Служба WWW и Web-пространство

Служба WWW является одной из множества действующих в Интернете служб и включает в себя три компонента, из которых два – программы и один – данные. Со стороны Интернета WWW основывается на программах, которые называют *Web-серверами*. Со стороны пользователя WWW основан на программах, которые называются *Web-браузерами*. Работая в паре, серверы и браузеры обеспечивают доступ к Web-документам (их называют *Web-страницами*).

В операционную систему Windows встроен клиент службы WWW – это браузер Microsoft Internet Explorer. Его задача, как и у любого браузера, – прием Web-страницы и воспроизведение ее на компьютере клиента, т.е. браузер должен понимать стандартный код HTML, с помощью которого задано форматирование Web-страниц.

Web-страница представляет собой текстовый документ, предназначенный для просмотра в окне браузера. Экранное представление Web-страницы формируется с помощью специальных команд языка HTML, интерпретируемых браузером. Этими командами можно обеспечить отображение рисунков, звукозаписей, видеороликов и других объектов в документе.

При создании Web-страницы возможны два основных подхода. Первый состоит в формировании документа вручную с использованием редактора и непосредственно команд HTML [4]. Этим путем можно реализовать все, что позволяет язык HTML. Автоматизировать работу можно с помощью специальных *редакторов HTML* – обычных текстовых редакторов, содержащих дополнительные средства для быстрого ввода команд HTML и проверки их правильности.

Второй способ создания Web-страницы состоит в использовании редакторов WYSIWYG (What You See Is What You Get – *что видите, то и имеете*). В этом случае знать язык HTML необязательно. Приемы создания документа подобны используемым в текстовом процессоре. Текст размещается на странице, после чего к нему применяют команды форматирования. Редактор автоматически формирует команды HTML, а страница немедленно отображается на экране примерно так, как ее будут наблюдать посетители Web-узла в Интернете. Качество кода HTML здесь обеспечивается не автором страницы, а программой. Это иногда приводит к некорректному преобразованию страницы в код HTML. Однако такой путь позволяет сократить время и силы на разработку Web-страниц.

Язык HTML (Hyper Text Markup Language) – это язык *разметки гипертекста*. Разметка заключается в том, что в обычный текст *добавляются* специальные команды HTML, описывающие то, как он должен выглядеть. Если из текста выбросить все эти команды, то получается в точности исходный текст документа. Гипертекст – это расширенный текст, содержащий дополнительные элементы. Главный из них – это *гипертекстовая ссылка* (гиперссылка), щелчок на которой позволяет перейти к другому документу или другому фрагменту того же самого документа. Вставные объекты (рисунки, видеоролики и т.д.) также рассматриваются как элементы гипертекста.

Описание документа с помощью языка HTML

Команды языка HTML называются *тегами* и записываются в виде последовательности букв, заключенных в угловые скобки: `<тег>`. Теги могут быть одиночными (в этом случае тег дает разовый эффект в том месте, где он располагается) или парными. Парный тег состоит из открывающего и закрывающего тегов. Закрывающий тег содержит ту же последовательность букв, что и открывающий, но им предшествует косая черта: `</тег>`. Область действия парного тега охватывает ту часть документа, которая заключена между открывающим и закрывающим тегами. После появления открывающего тега в документе должен обязательно встретиться и закрывающий.

`<H1>Заголовок</H1>` - парный тег определяет заголовок первого уровня.

`
` - одиночный тег, указывающий, что последующий текст должен начинаться с новой строки. Тег `</BR>` не существует.

Тег может содержать *атрибуты*, которые задают дополнительные (или обязательные параметры), определяющие способ применения тега. Атрибуты включаются в открывающий тег или непосредственно в одиночный тег. Отделяются друг от друга атрибуты с помощью пробелов.

Атрибуты бывают двух типов. Для части атрибутов надо указывать только имя. Для других должно быть указано значение. Тип значения у разных атрибутов различен – это может быть число, имя файла, произвольная символьная строка и прочее. Например:

`<HR SIZE="5" NOSHADE>` - одиночный тег HR задает горизонтальную линейку, атрибут NOSHADE не требует указания значения и определяет отображение линейки в плоском одноцветном виде, атрибут SIZE задает толщину линейки, равную 5 пикселям.

Чтобы указать значение атрибута, надо после его имени поставить знак равенства, затем привести нужное значение, заключив его в двойные кавычки.

Имена тегов и атрибутов содержат только латинские буквы, причем в любом регистре (можно вперемешку).

Основные структурные элементы HTML

Весь документ представляет собой один большой элемент, определяемый тегом `<HTML>`. Внутри он состоит из двух основных разделов: раздела заголовков (элемент, который определяется тегом `<HEAD>`) и тела документа (элемент с тегом `<BODY>`). Раздел заголовков содержит ряд элементов, описывающих свойства документа в целом (параметры шрифта и т.д.). Он, в частности, должен содержать элемент `TITLE`, заключающий в себе общий заголовок документа. Внутри тела документа располагается

все его содержание. Именно сюда помещаются теги, определяющие способ оформления документа в окне браузера.

Все перечисленные структурные теги – парные. Таким образом, минимальный полный документ должен выглядеть примерно так:

```
<HTML>
<HEAD>
<TITLE>
Все о кафедре АИТП
</TITLE>
</HEAD>
<BODY>
Кафедра АИТП входит в состав факультета АИТУ
</BODY>
</HTML>
```

После создания такого документа и сохранения его в виде файла с расширением `htm`, можно открыть его с помощью браузера Internet Explorer. Откроется окно документа. Содержимое элемента `TITLE` (и всего раздела заголовков) не рассматривается как содержимое документа и выводится только в строке заголовка окна браузера (рис. 12).

Открывающий тег `<BODY>` может содержать группу атрибутов, оказывающих влияние на внешний вид документа, например, можно задать цвета различных элементов документа. Атрибуты записываются внутри открывающегося тега, отделяются от него и друг от друга пробелами.

Задание цвета выполняется одним из двух способов. Значение нужного атрибута может являться наименованием цвета (например, `ТЕХТ="RED"`) или его шестнадцатеричным кодом (`ТЕХТ="#FF0000"`). Удобнее использовать шестнадцатеричный код цвета, состоящий из трех пар шестнадцатеричных цифр, которым предшествует символ `#`. Первые две цифры описывают насыщенность красного цвета, следующие – зеленого, третья пара – синего. Так, код `#6495ED` (васильковый цвет) имеет насыщенность красного 100 (`#64` в шестнадцатеричной системе), зеленого – 149 (`#95`), синего – 237 (`#ED`).

Максимальное значение насыщенности любого компонента – 255 (`#FF` шестнадцатеричное).

В теге `<BODY>` можно задать следующие цветовые атрибуты:

- **BGCOLOR**- определяет фоновый цвет страницы;
- **ТЕХТ** – определяет цвет текста на странице;
- **LINK** – определяет цвет для непросмотренных ссылок;
- **ALINK** – определяет цвет активной ссылки (на которую наведен указатель мыши или которая выделена с помощью клавиши TAB);

- **VLINK** – определяет цвет просмотренных ссылок, для которых осуществлялся переход по щелчку.

Атрибуты указываются **внутри** открывающегося тега **<BODY>**, отделяясь от него пробелом. Например, тег

<BODY TEXT="RED" BGCOLOR="GREEN">

задает красный цвет текста и зеленый цвет для фона страницы.

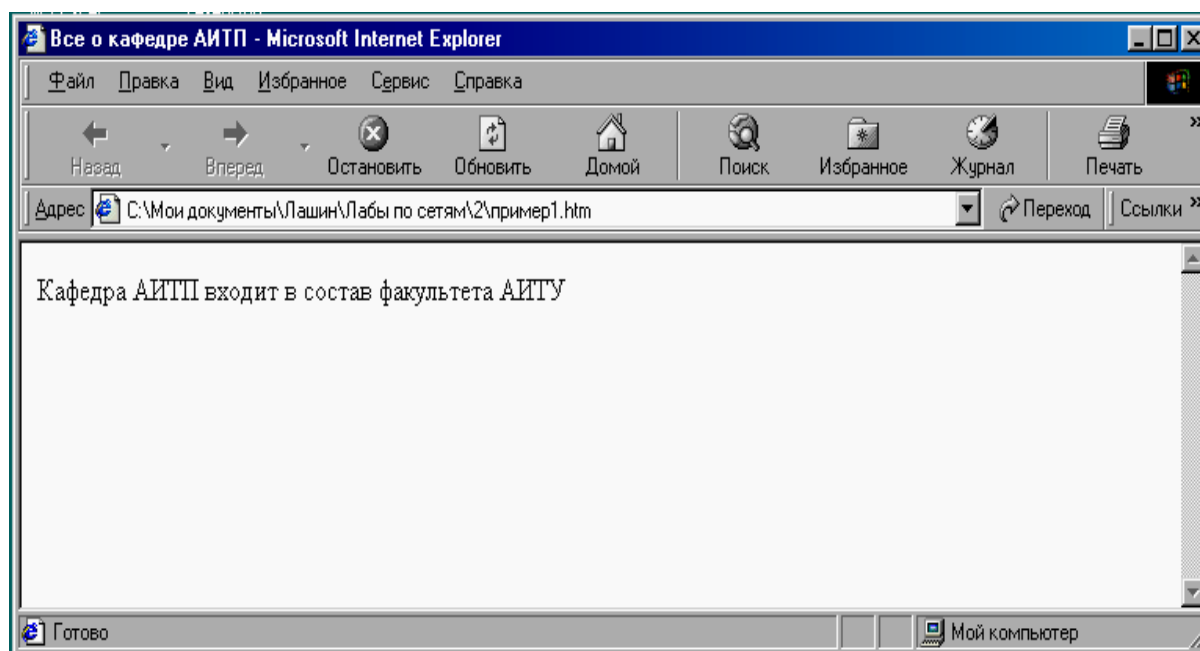


Рис. 12

Средства представления содержания документа HTML

1. Заголовки

Служат для указания границ функциональных или логических разделов документа. Например, заголовок – это функциональный раздел, а новый абзац – раздел логический.

Для заголовков можно задать шесть различных уровней, определяемых однотипно записываемыми тегами от **<H1>** до **<H6>**. Все эти теги парные.

При отображении заголовков в браузерах используется более крупный шрифт (обычно полужирный), постепенно уменьшающийся по мере снижения уровня заголовка (возрастания цифры после H).

Необязательный атрибут **ALIGN** тега заголовка может принимать одно из четырех значений: **LEFT**, **RIGHT**, **CENTER** и **JUSTIFY**, соответствующих выравниванию по левому краю, по правому краю, по центру и по ширине. Если атрибут не задан, то выравнивание идет по левому краю.

2. Абзацы

Отдельный абзац текста задается с помощью тега **<P>**. Это тег парный, но использование закрывающего тега не обязательно. Считается, что он располагается там, где начинается новый блочный элемент. Тег **<P>** может содержать только атрибут выравнивания **ALIGN**.

Задание начала нового абзаца обязательно включать в текст, так как клавиша Enter не создает нового абзаца. В исходном HTML-файле нажатие этой клавиши делает перевод строки, а в окне браузера перевод строки не будет осуществлен, если отсутствует тег **<P>**.

Начало новой строки можно задать также с использованием одиночного тега **
**.

3. Логические разделы

Разбиение текста на логические разделы осуществляется с помощью элемента **DIV**. Он аналогичен абзацу, но предназначен для задания одинакового способа выравнивания для группы блочных элементов. Выравнивание задается атрибутом **ALIGN=**, но не поддерживается выравнивание по ширине. Тег **<DIV>** парный, закрывающий тег **</DIV>** обязателен.

Пример (рис. 13):

```
<HEAD>
<TITLE>Создание абзаца</TITLE>
</HEAD>
<BODY>
<DIV ALIGN="CENTER">
```

<P>Все последующие абзацы центрируются, благодаря атрибутам, определяющим раздел документа

<P ALIGN="RIGHT">Для конкретного абзаца настройку можно изменить с помощью атрибутов в открывающемся теге

```
<P>Для всех остальных абзацев действуют настройки раздела
</DIV>
</BODY>
</HTML>
```

4. Списки

Списки являются блочными элементами особого рода. Возможны пять видов списков: нумерованные, маркированные, списки определений, списки каталогов и списки меню. Рассмотрим только первые два вида списков.

Маркированные списки задаются парным тегом ****. Открывающий тег может включать атрибут **TYPE=**, описывающий тип маркера:

- **TYPE="DISC"** – круглый черный маркер (по умолчанию);
- **TYPE="SQUARE"** – квадратный черный маркер;
- **TYPE="CIRCLE"** – маркер в виде колечка.

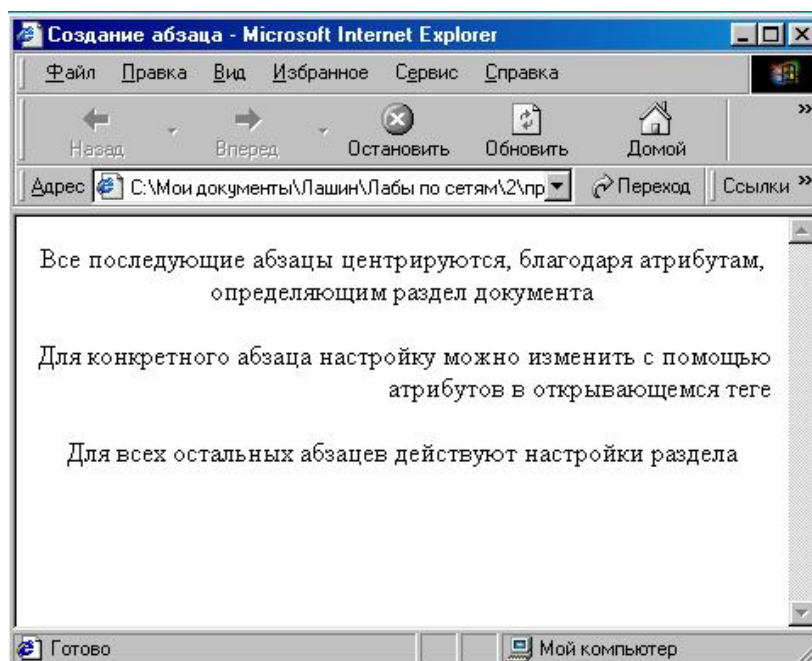


Рис. 13

Нумерованные списки оформляются с помощью парного тега ****. Возможные значения атрибута **TYPE=** задают способ нумерации списка:

- **TYPE="1"** – арабскими цифрами (1, 2, 3 ...по умолчанию);
- **TYPE="a"** – латинскими буквами нижнего регистра;
- **TYPE="A"** – латинскими буквами верхнего регистра;
- **TYPE="i"** – римскими цифрами с использованием нижнего регистра;
- **TYPE="I"** – римскими цифрами с использованием верхнего регистра.

Для нумерованных списков можно использовать атрибут **START=**, значением которого является число, указывающее номер первого элемента списка.

Элементы маркированных и нумерованных списков отделяются друг от друга тегом ****. Этот тег парный, но закрывающий указывать не обязательно. Каждый элемент списка выводится на отдельной строке в соответствии с заданным типом выравнивания, установленным для раздела, содержащего список.

Создание гиперссылок

Гиперссылки представляют собой текстовые элементы, позволяющие осуществить переход к другому документу или части текущего документа, если он содержит большое количество разделов. Отличие их от дру-

гих элементов в том, что гиперссылка не может быть вложена в другую гиперссылку.

Создание гиперссылки производится с помощью элемента **якоря**, который задается парным тегом `<A>`. Закрывающий тег обязателен. Текст гиперссылки представляет собой содержимое элемента якоря.

При создании гиперссылки открывающий тег якоря должен обязательно содержать атрибут **HREF=**. В качестве значения этого атрибута указывается адрес URL (*Uniformed Resource Locator – унифицированный указатель ресурса*), по которому произойдет переход в случае щелчка на гиперссылке. Этот адрес заключается в кавычки. URL-адрес определяет адрес любого файла, имеющегося в мировой компьютерной сети. Например, тег

` Переход по ссылке `

задает появление в окне гипертекста названия «Переход по ссылке», щелчок по которому открывает файл `myfile.htm`, находящийся в той же папке, что и исходный файл.

Гиперссылка отображается в браузере с использованием синего цвета и подчеркивания. При наведении указателя мыши на ссылку, он меняет форму (изображение кисти руки с вытянутым пальцем). В строке состояния браузера при этом появляется адрес, на который указывает ссылка. Ссылка может быть на любой файл как данного компьютера, так и других компьютеров сети. Адрес URL должен содержать полный путь к файлу из каталога, в котором хранится создаваемая Web-страница.

Ссылка не обязательно должна указывать на другую Web-страницу. Она может использовать любой из стандартных протоколов Интернета, распознаваемых браузером. Операция, выполняемая при щелчке мыши на ссылке, зависит от типа протокола и типа документа, указанного в ссылке.

Если текст ссылки недостаточно информативен, то можно указать **вспомогательный заголовок**, который задается в теге `<A>` с помощью атрибута **TITLE=**. Значением этого атрибута является текстовая строка, заключенная в кавычки. Обычно текст вспомогательного заголовка появляется во всплывающем окне. Например,

` Переход по ссылке`.

При создании документа, содержащего несколько страниц, связанных гиперссылками, обычно рекомендуется помещать все файлы страниц в одну отдельную папку. Тогда при переносе папки на другой компьютер не возникает проблемы с редактированием гиперссылок.

Внутренние гиперссылки

Позволяют быстро перейти к определенному месту внутри любого документа.

Чтобы можно было перейти внутри какого-либо документа, он должен содержать *именованный якорь*, задающийся с помощью тега `<A>`, но для него обязательным является не атрибут `HREF=`, а атрибут `NAME=`. Значением этого атрибута может быть произвольная строка, содержащая только латинские буквы и цифры.

Например, можно поставить якорь с именем `CONTENTS` около заголовка «Содержание» в тексте документа следующим образом:

```
<A NAME="CONTENTS">Содержание</A>
```

Именованный якорь является парным тегом, который должен иметь имя и обязательно должен быть закрыт. Значение имеет только точка размещения открывающегося тега – сюда произойдет переход по ссылке.

Чтобы сослаться на место в документе, надо указать имя, присвоенное именованному якорю, в адресе URL. Оно указывается после имени файла документа и отделяется от него символом `#`. Для перехода по ссылке внутри создаваемой Web-страницы имя файла указывать не надо:

```
<A HREF="#"#CONTENTS">Переход к содержанию</A>
```

- в результате выполнения этого тега в тексте документа появится текст гиперссылки «Переход к содержанию», нажатие на которую откроет часть документа с заголовком «Содержание».

Добавление графических элементов

Основной способ вставки рисунков в документ – использование тега ``. Его обязательным атрибутом является `SRC=`, значение которого – адрес URL файла изображения. Например, тег

```
<IMG SRC="picture.gif">
```

добавляет в текст документа картинку из файла `picture.gif`, расположенного в том же каталоге, что и текущий создаваемый документ. Атрибуты `WIDTH=ширина` и `HEIGHT=высота` задают ширину и высоту рисунка в пикселях и указываются внутри тега `IMG`. Например, тег

```
<IMG SRC="picture.gif" WIDTH=150 HEIGHT=150 >
```

задает размеры рисунка в 150 пикселей по вертикали и горизонтали. При изменении размеров рисунка необходимо знать исходные пропорции, иначе в результате выполнения тега рисунок может «вытянуться» по вертикали или горизонтали.

Включение графических изображений в текст Web-страницы увеличивает объем данных, которые надо загружать при загрузке страницы браузером. В некоторых браузерах отключен режим загрузки графики. Чтобы дать понятие о том, что содержит незагруженное изображение, используется *альтернативный текст*, который задается атрибутом `ALT=`. Значение этого атрибута – строка текста, «заменяющего» изображение:

```
<IMG SRC="tree.jpg" ALT="[Дерево, растущее за моим окном]">
```


Вместо рисунка дерева появляется пустая рамка, содержащая альтернативный текст, если отключен режим отображения графики. При включении отображения графики альтернативный текст появляется в окне всплывающей подсказки при наведении на рисунок маркера мыши. Обычно альтернативный текст дополнительно заключают в квадратные скобки.

Атрибут **ALIGN=**, размещенный в теге **IMG**, позволяет задать размещение рисунка в тексте документа. Значениями этого атрибута могут являться:

- **MIDDLE** или **CENTER** – выравнивание по центру;
- **LEFT** – выравнивание по левому полю;
- **RIGHT** – выравнивание по правому полю;
- **TOP** – выравнивание по верхней границе;
- **BOTTOM** – выравнивание по нижней границе.

Атрибуты **VSPACE=значение** и **HSPACE=значение** обозначают горизонтальный и вертикальный отступ. Они используются для того, чтобы задать отступ от краев изображения до текста в пикселях.

Выделение частей текста изменением параметров шрифта

Если возникает необходимость выделения части текста, можно воспользоваться специальными тегам, позволяющими изменить шрифт на полужирный или курсив. Для этих целей предназначены парные теги:

**** текст **** - текст между парными тегам выделяется полужирным шрифтом;

<I> текст **</I>** - текст выделяется курсивом;

<U> текст **</U>** - текст выделяется подчеркиванием.

Если необходимо одновременно выделить текст полужирным шрифтом и курсивом, используется один из вариантов кода:

<I> текст **</I>** или **<I>** текст **</I>**

Для изменения цвета и размера шрифта текста в документе можно воспользоваться парным тегом **** текст ****, где в качестве атрибутов задаются цвет (атрибут **COLOR=**) и размер шрифта (атрибут **SIZE= ±n**, где n – целое число, определяющее увеличение или уменьшение размера шрифта в пунктах, где пункт равен 1/72 дюйма или 0,353 мм). Увеличение или уменьшение задается относительно размера, установленного в браузере по умолчанию. Атрибуты указываются после слова **FONT** через пробел.

Например, при использовании следующей записи в гипертекстовом документе

****Команда **<I>** Спартак **</I>** победила в соревнованиях по ****хоккею!**** слова «Команда Спартак» будут отображаться синим цветом и шрифтом, увеличенным на 3 пункта. Дополнительно слово «Спартак» выделится по-

лужирным шрифтом и курсивом. Фраза «победила в соревнованиях по » останется в стандартном виде, а часть текста «хоккею!» выделится полужирным шрифтом.

Команда *Спартак* победила в соревнованиях по хоккею!

Бегущая строка

Бегущие строки используются для украшения страниц и привлечения внимания пользователя к ее тексту. Для создания такой строки используется тег `<MARQUEE> текст </MARQUEE>`. На место слова **текст** подставляется любое слово или фраза, которые надо заставить «бежать» по экрану компьютера. Стандартный вариант расположения текста в бегущей строке – когда вставленная фраза появляется из-за правого края окна браузера и, пройдя через всю страницу, скрывается за левым краем. Направление движения бегущей строки можно менять с использованием атрибута направления `DIRECTION=LEFT`.

Чтобы задавать характер движения надписи по странице, применяется атрибут `BEHAVIOR=`, значениями которого могут быть:

- **SCROLL** – стандартное движение от правого края к левому. Этот атрибут организует бесконечный цикл. Число циклов можно ограничить, указав атрибут `LOOP=n`, где **n** указывает число повторений цикла;
- **SLIDE** – надпись один раз пробегает от правого края к левому и там остается;
- **ALTERNATE** – движение от правого края страницы к левому и обратно. Бесконечный цикл.

Для указания ширины бегущей строки используется атрибут `WIDTH=n`, где **n** указывает значение ширины в пикселях или процентах от общей ширины видимой части страницы.

Аналогично можно задать высоту бегущей строки атрибутом `HEIGHT=n`, где величина **n** – в пикселях.

Можно окрасить поверхность бегущей строки в какой-либо цвет:

`<MARQUEE BGCOLOR=n> текст </MARQUEE>`,

где **n**, как обычно при задании цвета, можно указать в виде шестнадцатеричного числа или названия цвета на английском.

Величину и цвет шрифта пробегающей надписи можно изменить, если перед тегом бегущей строки поставить тег изменения шрифта с атрибутами цвета и размера.

`<MARQUEE> текст </MARQUEE>`

В теге задания бегущей строки может быть использовано любое количество и сочетание атрибутов, разделенных между собой пробелами.

Задание на лабораторную работу

В папке СТУДЕНТ/GrXXX создать в *Блокноте* текст на языке HTML для Web-страницы своей бригады. Текст сохранить в файле с указанием типа htm. Страница должна содержать любую интересную информацию о работе бригады. Кроме того, на каждого участника бригады должна быть своя страничка, где он рассказывает о себе (можно с иллюстрациями своих достижений). В основной странице бригады должны быть ссылки на страницы участников бригады и возвраты обратно на бригадную страницу. С использованием списков каждый человек должен составить свое резюме как о будущем специалисте.

Отчет о работе должен содержать текст на языке HTML созданных Web-страниц.

Лабораторная работа № 3

РАБОТА С WEB-СТРАНИЦАМИ В РЕДАКТОРЕ WORD 2003

Цель работы: создание Web-страниц стандартными средствами MS Office.

Прямое кодирование с помощью языка HTML теперь уже не единственный способ создания Web-страниц. Текстовый редактор Word сегодня является мощным средством создания Web-документов. Редактор использует для оформления страниц все, что можно, например списки стилей CSS(Cascading Style Sheets), применяемые как средство форматирования; активные сценарии, встраиваемые для организации некоторых нестандартных элементов, и другие средства, которые не относятся непосредственно к элементам HTML. Поэтому самостоятельно разобраться, что и как генерирует редактор Word, довольно сложно человеку, слабо знающему языковые особенности HTML.

С помощью программы Word предполагается, главным образом, создание отдельных страниц или небольших узлов объемом не более десятка страниц. При использовании программы Word необходимо также вручную контролировать размещение в нужных каталогах всех файлов, используемых на Web-узле, - иногда примененная адресация может не работать после переноса файлов узла на Web-сервер.

Представление документа как Web-страницы

Если есть готовый документ, который надо опубликовать в Интернете, то редактор Word 2003 служит конвертором имеющегося документа в

формате HTML. Для сохранения имеющегося документа в формате HTML дается команда *Файл/Сохранить как Веб-страницу*.

При первом сохранении документа желательно сначала по кнопке *Сервис* заполнить пункт *Параметры Веб-документа*. Открывшееся окно содержит четыре вкладки (рис. 14).

- Вкладка *Общие* позволяет выбрать браузеры, на которые данная страница рассчитана, искусственно ограничить используемые средства HTML, указать, можно ли применять списки CSS.

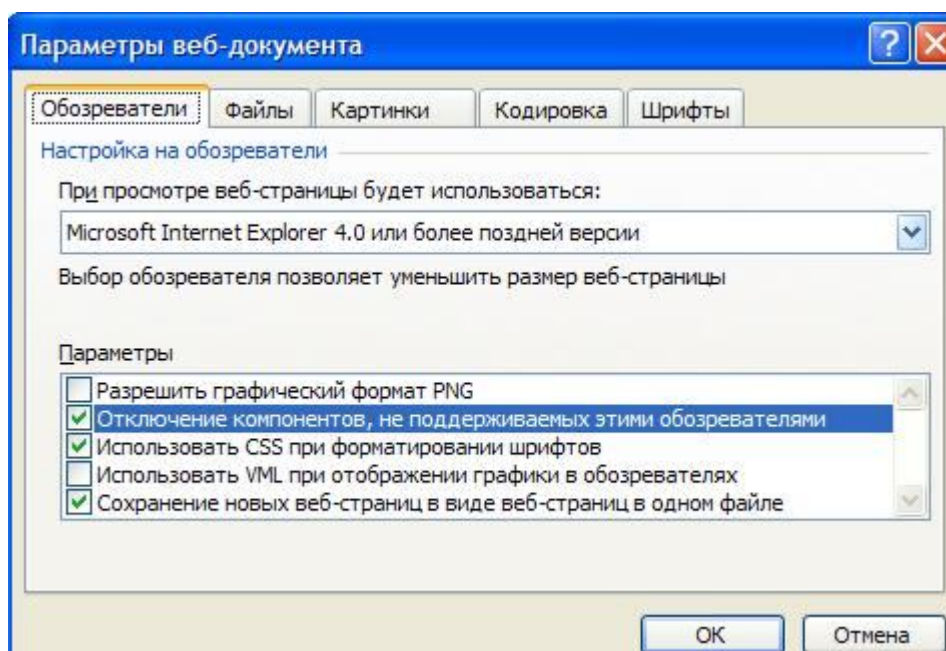


Рис. 14

- Вкладка *Файлы* определяет способ хранения созданной Web-страницы. Если установить флажок *Создать папку для вспомогательных файлов*, то графические иллюстрации, мультимедийные объекты и прочие дополнительные файлы записываются в отдельную папку, вложенную в ту же папку, где хранится создаваемая страница. Если флажок сброшен, все файлы, относящиеся к данной странице, записываются в одну и ту же папку.

Если установить флажок *По возможности использовать длинные имена файлов*, то все длинные имена Web-страницы и вспомогательных файлов сохраняются. В противном случае эти имена приводятся к системе 8.3 (восемь символов на имя и три – на расширение имени).

Установленный флажок *Обновлять ссылки при сохранении* указывает, что при выборе каталога, содержащего вспомогательные файлы, вно-

сятся необходимые поправки в ссылки на эти файлы, чтобы они правильно отражали относительное расположение файлов в созданных каталогах.

- Вкладка *Картинки* определяет способ хранения изображений.
- Вкладка *Кодировка* определяет кодировку символов, используемую при сохранении Web-страницы.

Использование шаблонов для создания документов

Для создания новой Web-страницы можно воспользоваться меню *Файл/ Создать/ Веб-страница*. Вся информация нового документа будет сохранена в файле с типом *htm*.

Имеется возможность создавать новые документы с использованием существующих шаблонов и мастеров создания документов. Для этого необходимо из меню *Файл/ Создать* выбрать категорию **Шаблоны/ На моем компьютере** (рис. 15).

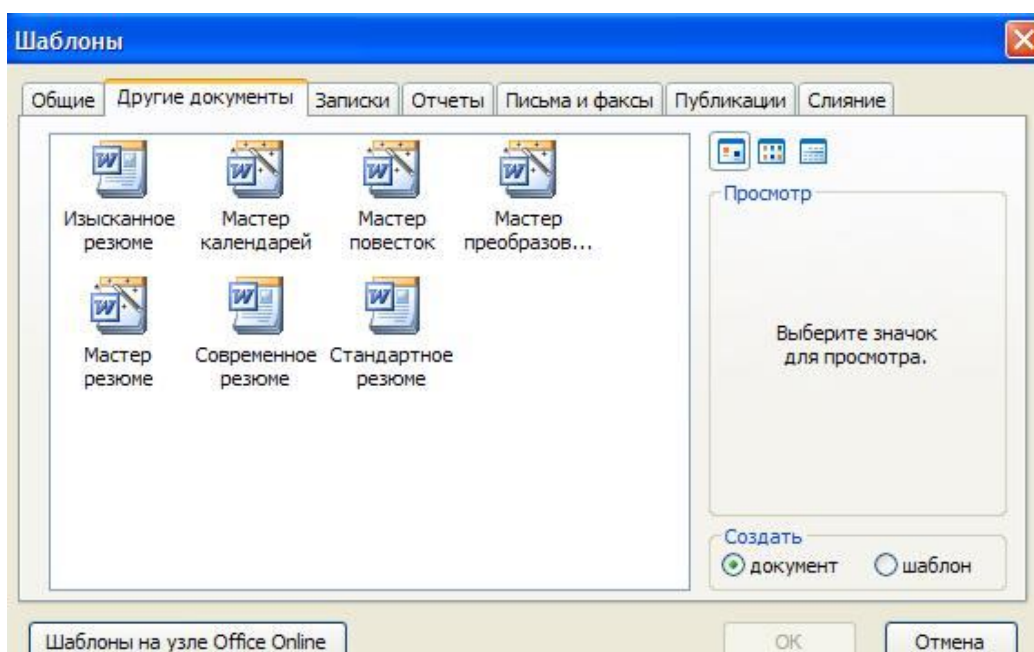


Рис. 15

Вкладки содержат шаблоны различных стандартных документов, используемых в электронном документообороте. Вкладка *Другие документы* используется для создания календарей, повесток и резюме выбором соответствующего шаблона или с использованием *Мастеров*.

Стандартные подготовленные шаблоны для резюме следующие.

- *Изысканное резюме.*
- *Современное резюме.*
- *Стандартное резюме.*

Структура этих страниц позволяет рассказать о себе. Каждая страница состоит из нескольких разделов, которые требуется заполнить персональными данными. Отличия шаблонов только в оформлении и структурном содержании.

Для создания собственного резюме можно воспользоваться *Мастером резюме* (рис. 16).

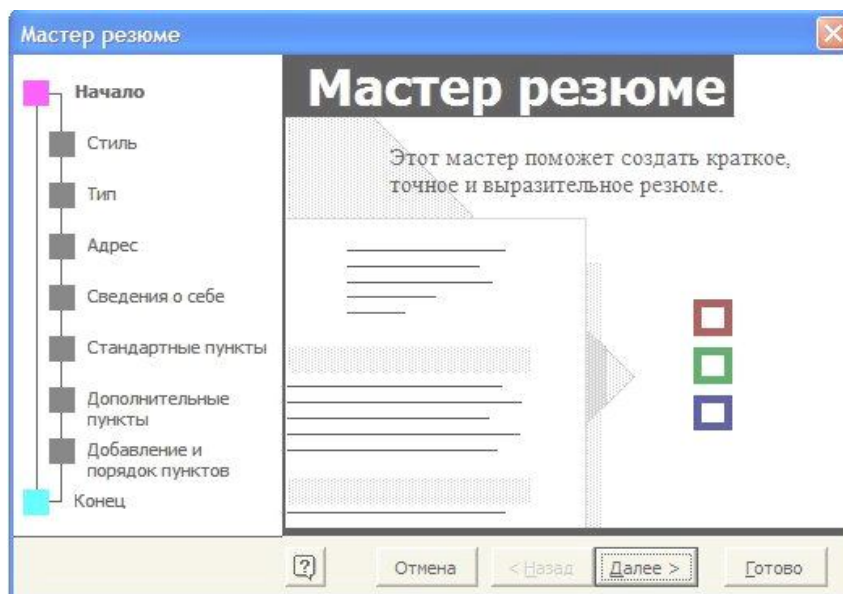


Рис. 16

Процесс получения резюме разбит на 7 шагов:

1. стиль;
2. тип;
3. адрес;
4. сведения о себе;
5. стандартные пункты;
6. дополнительные пункты;
7. добавление и порядок пунктов.

На 1-м шаге предлагается выбрать стиль размещения информации в документе (рис. 17).

На 2-м шаге необходимо определить тип размещения информации в резюме: обычное, хронологическое, функциональное или профессиональное.

3 шаг – задание адреса проживания, электронной почты и других сведений (рис. 18).

4 шаг – указание, какие сведения о себе необходимо представить в резюме (рис. 19).

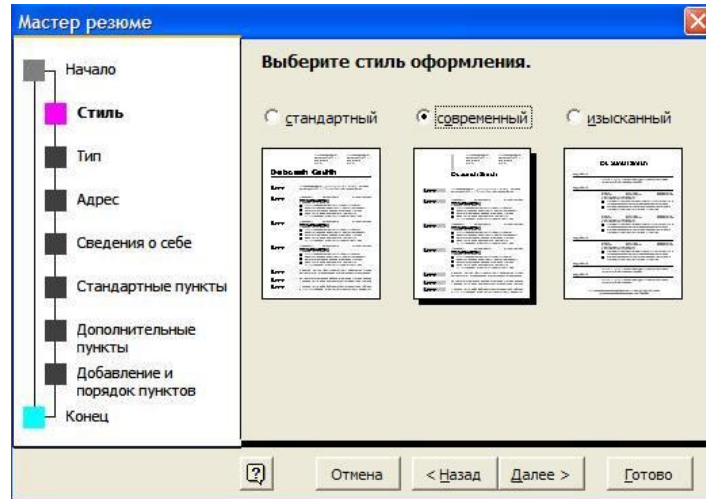


Рис. 17

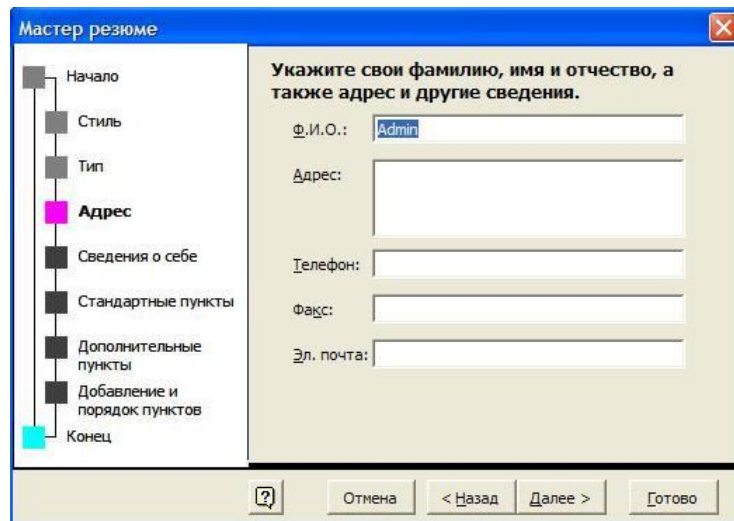


Рис. 18

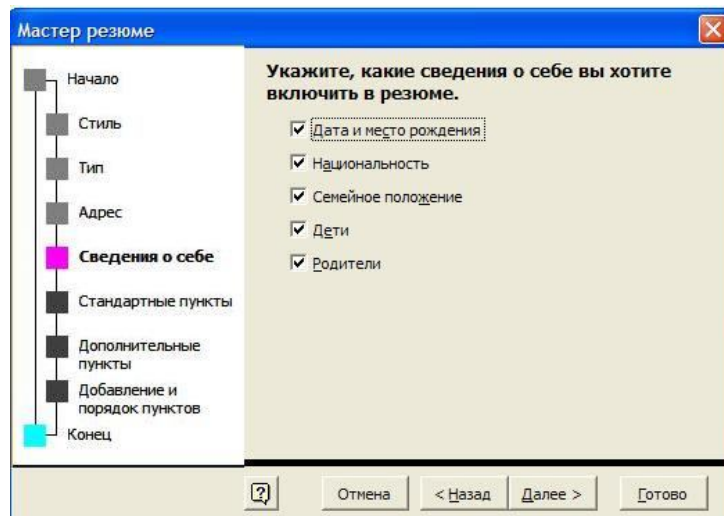


Рис. 19

5 шаг позволяет выбрать из списка стандартных пунктов те, что будут включены в резюме (рис. 20).

На 6 шаге определяется, какие дополнительные пункты необходимо включить в резюме (рис. 21).

7 шаг позволяет ввести дополнительные пункты из предложенного списка сведений в резюме (рис. 22).

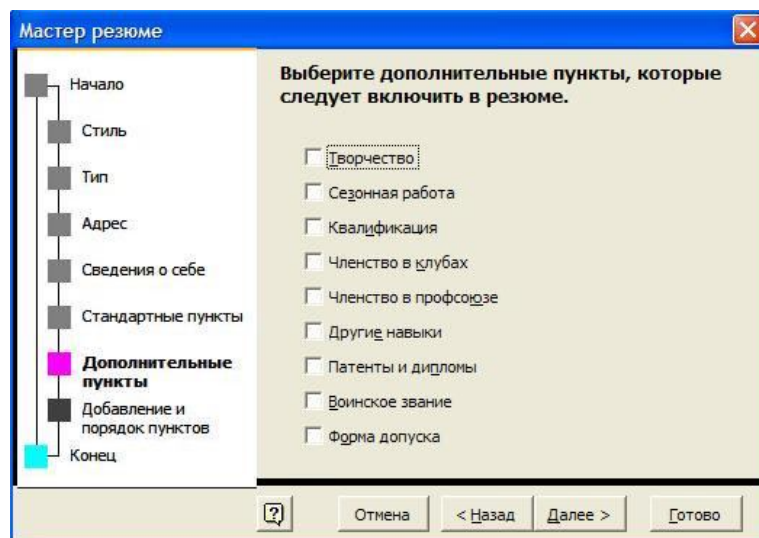


Рис. 20

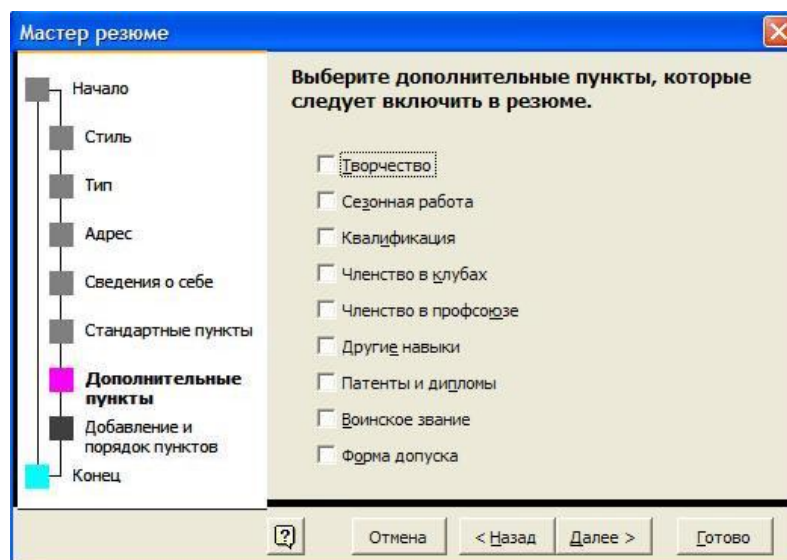


Рис. 21

В результате прохождения всех шагов получается документ, содержащий все необходимые сведения для резюме и имеющий тот стиль представления данных, который выбран пользователем.

Аналогично работают *Мастер календарей* и *Мастер служебных записок*.

Остальные вкладки предназначены для получения записок, отчетов, писем и факсов, создания публикаций.

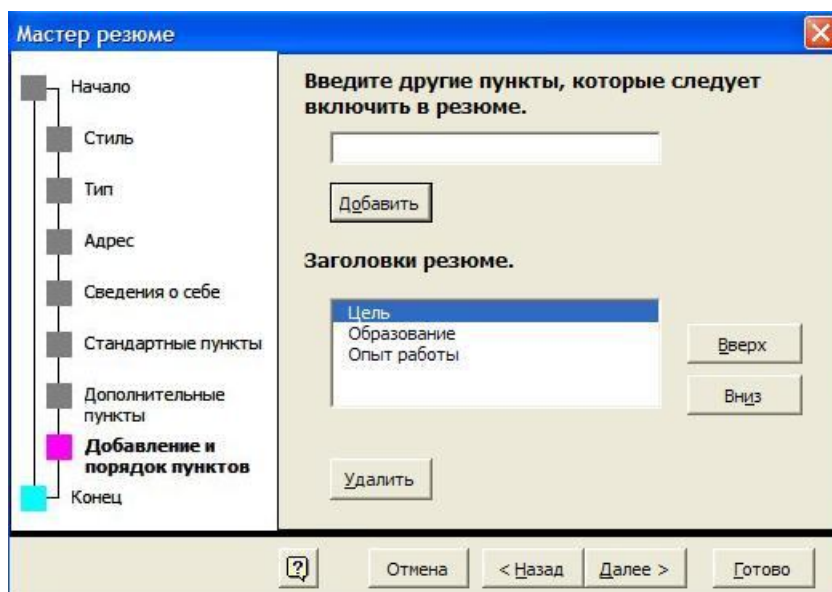


Рис. 22

Задание на лабораторную работу

1. В редакторе Word в обычном текстовом виде с использованием средств оформления текста создать основную страницу бригады и страницы для членов бригады, вставить гиперссылки для перехода на страницы участников бригады и возврата на основную страницу. Сохранить полученные документы как Веб-страницы. Просмотреть полученные результаты в окне браузера Internet Explorer.

2. Создать резюме для членов своей бригады с использованием шаблонов редактора Word .

Библиографический список

1. Компьютерные сети. Принципы, технологии, протоколы / В.Г.Олифер, Н.А.Олифер. СПб: Питер, 2000.
2. Джамса К., Коуп К. Программирование для Internet в среде Windows. СПб: Питер, 1996.
3. Microsoft TCP/IP. Учебный курс. М.: ИТД «Русская редакция», 1999.
4. Евсеев Г. , Симонович С. . Новейший самоучитель по работе в ИНТЕРНЕТЕ. – М.: ДЕСС КОМ, Информ-Пресс, 2000.

Содержание

Лабораторная работа № 1. Обмен данными в локальной вычислительной сети (ЛВС). Настройки	1
Лабораторная работа № 2. Создание Web-страницы средствами языка HTML	14
Лабораторная работа № 3. Работа с Web-страницами в редакторе WORD 2003	25
Библиографический список	31

Информационные сети и телекоммуникации

Составители: К у з ь м и н а Екатерина Михайловна
К у л и ч е н к о Татьяна Александровна
Л а ш и н Виктор Александрович

Редактор Н.А. Орлова
Корректор С.В. Макушина

Подписано в печать 10.02.14. Формат бумаги 60x84 1/16.

Бумага газетная. Печать трафаретная. Усл. печ. л. 2.0.

Тираж 40 экз. Заказ

Рязанский государственный радиотехнический университет.

390005, Рязань, ул. Гагарина, 59/1.

Редакционно- издательский центр РГРТУ.