

Тема: Схема передачи информационных сообщений при осуществлении перевозочного процесса

Практическое занятие №1

Тема: Схема передачи информационных сообщений при осуществлении перевозочного процесса

Цель: научиться составлять схему передачи информационных сообщений при осуществлении перевозочного процесса.

Оборудование: ПЭВМ, локальная сеть.

Задания:

- 1) пользуясь исходными данными, определить круг работников, участвующих в осуществлении перевозочного процесса;
- 2) составить схему передачи информационных сообщений при осуществлении перевозочного процесса на предприятиях железнодорожного транспорта.

Исходные данные:

Таблица

Тип железнодорожной станции и средства связи

Номер варианта	Средства связи	Тип железнодорожной станции	Тип сообщения
1	2	3	4
1	Автоблокировка	Сортировочная односторонняя	209
2	Автоблокировка	Грузовая	206
3	Автоблокировка	Участковая	208
4	Автоблокировка	Сортировочная двухсторонняя	200
5	Автоблокировка	Сортировочная односторонняя	202
6	Автоблокировка	Грузовая	209
7	Автоблокировка	Участковая	206
8	Автоблокировка	Сортировочная двухсторонняя	208
9	Автоблокировка	Сортировочная односторонняя	200
10	Автоблокировка	Грузовая	208
11	Автоблокировка	Участковая	202
12	Автоблокировка	Сортировочная двухсторонняя	203
13	Автоблокировка	Сортировочная односторонняя	205
14	Автоблокировка	Грузовая	202
15	Автоблокировка	Участковая	209
16	Автоблокировка	Сортировочная двухсторонняя	208

Методика формирования информационных сообщений 200-206, 208, 209

Информационные сообщения состоят из **служебной и информационной фраз** и разделяются на следующие категории:

- **сообщение 200 - отправление со станции;**
- **сообщение 201 - прибытие на станцию;**
- **сообщение 202 - проследование станции;**
- **сообщение 203 - расформирование поезда;**
- **сообщение 204 - «бросание поезда» ;**
- **сообщение 205 - готовность к отправлению;**
- **сообщение 206 - продвижение пассажирских поездов;**
- **сообщение 208 - объединение или разъединение поездов;**
- **сообщение 209 -изменение индекса поезда.**

Служебная фраза сообщений 200, 201, 202, 205:

(:201 608807 2112 6300 11 6255 608544 16 06 11 30 01/03 1:)

201 - код сообщения.

608807 - железнодорожная станция передачи сообщения.

2112- номер поезда.

6300 11 6255 - индекс поезда.

608544 - направление: для 200, 205 - «куда»; 201 - «откуда»; 202 –« куда », «откуда + «куда» (при наличии у станции более 2-х направлений).

16 06 11 30 - дата и время (день, месяц, часы, мин.).

01/03 - парк/путь.

1 - признак отцепки локомотива (только для сообщений 201 и 204): 0- работы с локомотивами и бригадами нет; 1 - отцепка локомотивов; 2 - смена локомотивных бригад; 3 - отцепка толкача; 4 - отцепка вторых локомотивов

Примечания:

1) в случае временной задержки продвижения поезда время задержки указывается через «+» после кода причины задержки;

2) показатель парк/путь является обязательным и должен иметь ненулевые значения.

Информационная фраза сообщений 200, 201, 202, 204, 205, 206:

522 34921 1 15 24 6110 02918 `ТИХОМИРОВ:)

522 – код серии локомотива.

34921 – номер локомотива с признаком секции. Признак секции набивается последним знаком и принимает следующие

значения: 0 – при следовании односекционного локомотива; 1 – при следовании двухсекционного локомотива в заводском исполнении, или одной секции «А» двухсекционного локомотива, или локомотива-гибрида, у которого головная секция «А»; 2 – при следовании одной секции «Б» двухсекционного локомотива или локомотива-гибрида, у которого головная секция «Б».

1 – код состояния локомотива: 1 – движение в голове поезда; 2 – движение в двойной тяге; 3 – движение в пересылке (в двойной тяге для перевозки); 4 – одиночное следование; 5 – следование в ремонт (одиночным порядком); 6 – подталкивание фактическое; 7 – одиночное следование от подталкивания; 8 – работа локомотивов, соединенных для работы по смет-радио.

15 24 – время явки бригады (часы, мин.).

6110 02918 – код депо приписки и табельный номер машиниста.

Примечание: Количество информационных фраз должно соответствовать количеству локомотивов и бригад, следующих резервом.

Коды состояния локомотивов в этом случае должны быть различными. **Например, при двойной тяге:**

ИВАНОВ

СМИРНОВ:)

Если одна бригада следует с двумя локомотивами (второй в пересылке), то у второго локомотива повторяются сведения о бригаде, например:

533 16381 1 11 15'ИВАНОВ

533 26661 3 11 15'СМИРНОВ:)

Оформление локомотивов, следующих резервом, происходит следующим образом. На локомотивы, следующие резервом, передаются сообщения 200, 201, 202.

Резервные локомотивы, следующие без вагонов, а также локомотивы от грузовых поездов с прицепленными к ним не более 10-ю физическими вагонами имеют номер поезда 4301 – 4398.

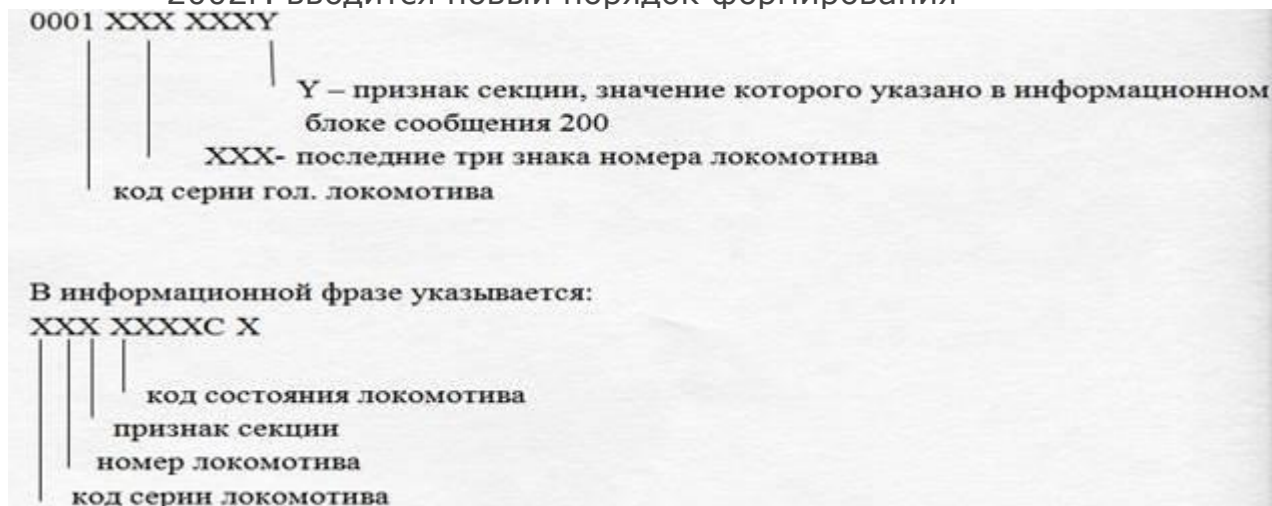
Если к локомотиву, следующему резервом, прицеплена группа вагонов, то индекс поезда в сообщении – обычный.

Пример:

(:200 608807 4351 6203 01 6076 608544 07 04 10 20

СМИРНОВ:)

Если локомотив следует без вагонов, то на основании тел. МПС номер Д-6131 от 4 июня 2002г. вводится новый порядок формирования



индекса:

Пример:

(:200 612802 4351 0001 530 3111 519105 18 06 07 00

Попов:)

Пример сообщения 206 с локомотивной фразой:

(: 206 62726 639 1 62846 10 04 00 41 01/01:

504 08880 1 20 30 6110 01899 `ВЕДЕНИИ Н:)

Транзитный поезд.

Транзитный поезд.

Поезд своего формирования.

Код 34 - прицепка группы вагонов из одного поезда к другому.

В служебной фразе в качестве индекса объединенного поезда и в первой информационной фразе указывается индекс поезда, к которому делается прицепка. Во второй информационной фразе указывается индекс поезда, от которого отцепляется группа вагонов для объединения, и номера головного и хвостового вагонов отцепляемой группы.

Например:

(:208 608807 3120 6203 66 6076 34 12 11 09 10 - первый транзитный поезд.

Краткие теоретические сведения

Информация передается в виде сообщений от некоторого источника информации к ее приемнику посредством канала связи между ними. Источник посылает передаваемое сообщение, которое кодируется в передаваемый сигнал. Этот сигнал посылается по каналу связи. В результате в приемнике появляется принимаемый сигнал, который декодируется и становится принимаемым сообщением. Передача информации по каналам связи часто сопровождается воздействием помех, вызывающих искажение и потерю информации.

Совокупность устройств, предметов или объектов, предназначенных для передачи информации от одного из них, именуемого источником, к другому, именуемому приемником, называется каналом информации или информационным каналом.

Примером канала может служить почта. Информация, закодированная в виде текста, помещается в конверт, поступает в почтовый ящик, извлекается оттуда и перевозится в почтовое отделение, где сортируется (вручную или машиной). Далее информация перемещается с помощью поезда (самолета, теплохода и с п.) в почтовое отделение пункта назначения, сортируется и доставляется адресату. Таким образом, почтовый канал включает в себя: конверт (предмет), транспорт и сортировочные машины (устройства), почтовых работников. Информация, помещенная в этот канал остается неизменной.

Ещё один пример-компьютер. Отдельные его системы передают информацию с помощью сигналов. Компьютер – устройство для обработки информации, он не создает из «ничего» информацию, а преобразует то, что в него введено. Компьютер является информационным каналом с преобразованием информации: информация поступает с внешних устройств (клавиатура, диск, микрофон), преобразуется во внутренний код и обрабатывается, преобразуется в вид, пригодный для восприятия внешним выходным устройством (монитором, печатающим устройством, динамиками и др.), и передается на них.

В настоящее время понятие информационной технологии не только на железнодорожном транспорте, но повсеместно во всем мире связывают с использованием электронных средств передачи и обработки информации. Поэтому распространение получило понимание информационной технологии как системы сбора, хранения, обработки, передачи, представления и использования информации, основанной на применении средств электроники и вычислительной техники.

Во второй половине прошлого века для переработки информации на железнодорожном транспорте начали применяться ЭВМ. Был создан Главный вычислительный центр (ГВЦ), который обрабатывал информацию в масштабе всей сети дорог и являлся основным предприятием по внедрению вычислительной техники на железнодорожном транспорте. На дорогах стали функционировать дорожные информационно-вычислительные центры. Были разработаны и внедрены интегрированные системы обработки данных (ИСОД).

Благодаря ИСОД обеспечивается однократное формирование и многократное использование набора исходных данных, единая схема исходных данных, интеграция процедур преобразования информации, единая нормативно-справочная база. В итоге на базе ГВЦ и информационно-вычислительного центра (ИВЦ) дорог с использованием ИСОД появилась комплексная автоматизированная система управления железнодорожным транспортом (АСУЖТ).

Автоматизированная система управления представляет собой человеко-машинную систему, обеспечивающую автоматический сбор и обработку информации, необходимой для оптимизации управления в различных сферах человеческой деятельности. АСУЖТ включает в себя совокупность административных, технологических и экономико-математических методов, средств вычислительной техники и связи, позволяющих аппарату управления эффективно управлять железнодорожным транспортом.

АСУЖТ представляет собой трехуровневую автоматизированную систему. На ее верхнем уровне автоматизируются функции департаментов ОАО «РЖД», а также создан Центр управления перевозками (ЦУП). На втором уровне функционирует АСУ дороги, автоматизируются функции дорожных служб и дорожного диспетчерского центра. На нижнем уровне функционируют АСУ сортировочной железнодорожной станции (АСУСС),

грузовой (АСУГС), АСУ других линейных предприятий. Именно здесь зарождается основная первичная информация, которая затем обрабатывается в АСУЖТ.

Когда использовались ЭВМ только вычислительных центров, осуществлялась лишь автоматизация локальных процессов. Качественно новый уровень переработки и использования информации достигается только тогда, когда компьютерная техника приходит непосредственно на рабочие места управленцев и участников перевозочного процесса. Это осуществилось благодаря появлению персональных компьютеров и созданию на их базе автоматизированных рабочих мест (АРМ).

В состав АРМ входят персональный компьютер с клавиатурой и монитором, печатающее устройство (принтер), аппаратура приема-передачи информации, а при необходимости и другие технические средства (телефонный коммутатор, магнитофон и др.).

В первую очередь создаются АРМы для тех оперативных работников, у которых регистрируются первичные данные о ходе перевозочного процесса - дежурных по железнодорожной станции, маневровых диспетчеров, операторов станционных технологических центров, дежурных по локомотивным депо, товарных кассиров.

Управление технологическим процессом работы железнодорожных станций основывается на своевременной, достоверной и полной информации о поездах, грузах, вагонах, размещении их на железнодорожных путях железнодорожной станции и т. д. Организации информационного обеспечения перевозочного процесса сопряжена с выполнением большого и сложного комплекса операций. Поэтому крупные железнодорожные станции, связанные с формированием и расформированием поездов, значительной грузовой работой, оснащены автоматизированными системами управления, сосредоточенными в станционных технологических центрах (СТЦ).

В СТЦ большая часть информации и документов перерабатывается на ЭВМ. Оператор СТЦ принимает, обрабатывает информацию о составах прибывших поездов, вагонах и грузах и другие информационные сообщения. Оператор СТЦ составляет и корректирует телеграмму-натурный лист поезда, обеспечивает на АРМ СТЦ передачу достоверной информации о ходе перевозочного процесса в автоматизированные системы управления перевозочным процессом и работой железнодорожной станции, получает справочную информацию и технологические документы на поезда, кодирует данные о грузах и грузополучателях, передает информацию на отправляемые поезда.

Входной информацией, необходимой для функционирования АРМ ряда профессий станционных работников, является предварительная информация о подходе поездов, где о каждом вагоне представлены сведения, содержащиеся в натурном листе поезда, а также сообщения об отправлении поезда с соседней железнодорожной станции, состояние железнодорожных путей, сигналов, положение стрелок на железнодорожные станции. Эти данные вводятся автоматически, если АРМ подключено к дорожному ИВЦ, к напольным устройствам автоматики и телемеханики, в частности к электрической централизации стрелок.

АРМ поездного диспетчера обеспечивает отображение поездного положения на участке, ведение графика исполненного движения, составление прогнозного графика на 3 часа вперед, выдачу по запросу диспетчера варианта пропуска поезда по участку, данных о любом поезде и др. При этом реализуется автоматический съем информации из рельсовых цепей при наличии устройств диспетчерского контроля или диспетчерской централизации. При автоблокировке источником информации является АРМ дежурного по железнодорожной станции, подключаемое к АРМу поездного диспетчера через концентратор информации.

Компьютеры используются в качестве АРМ отдельных лиц либо всего оперативного персонала подразделения. В последнем случае они объединяются в локальные вычислительные сети с единой базой данных посредством сетевых адаптеров. Любая такая сеть должна иметь орган управления (так называемый файловый сервер), включенные в сеть рабочие станции (АРМы), источник бесперебойного питания и другое вспомогательное оборудование.

В современных условиях информационная технология перевозки грузов - это управление процессом транспортировки на основе точных знаний о потребностях и данных о ходе перевозочного процесса в режиме реального времени, когда задействованные мощности и ресурсы максимальны адекватны объему выполняемой в данный период работы, а принимаемые решения отвечают не локальным, а общеотраслевым экономическим интересам.

К средствам реализации информационных технологий относят аппаратные (технические) и программные средства, аппаратно-программные комплексы и автоматизированные информационные системы. Автоматизированная информационная система - это совокупность технических (аппаратных) и программных средств, а также работающих с ними пользователей (персонала), обеспечивающая ввод, передачу, хранение, обработку и представление информации.

На железнодорожном транспорте информационные системы базируются на информационно-вычислительных центрах дорог (ИВЦ) и сети телекоммуникаций для передачи данных.

Эффект от внедрения информационных технологий будет тем выше, чем больше исходной информации вводится в автоматическом режиме, а не вручную. В управлении перевозочным процессом к числу первоочередных относится информация о фактическом поступлении вагона с железнодорожного пути необщего пользования, об отправлении с железнодорожной станции, о прибытии на попутную железнодорожную станцию или на железнодорожную станцию назначения, о поступлении на пункт выгрузки.

Служебная фраза сообщения 206 выглядит следующим образом:

(:206 30847 82 1 26781 24 05 12 40 03/02 1 0 19:)

206 – код сообщения.

30847 – железнодорожная станция передачи сообщения.

82 – номер поезда.

1 – код операции.

26781 – направление следования

24 05 12 40 – дата и время.

03/02 – номер парка, пути.

1 – количество полных суток опоздания.

0 – признак отцепки локомотива.

19 – количество пассажирских вагонов.

Код операции может принимать значения: 1 - прибытие; 2 - отправление; 3 - проследование без остановки. При следовании поезда по станции без смены локомотива или бригады в сообщении 206 можно давать код операции 3/XX - проследование с остановкой, где XX - минуты стоянки.

Номер направления следования может состоять из 5 или 6 знаков , последний знак - контрольный: 1 - откуда прибыл поезд; 2 и 3 – куда отправится поезд.

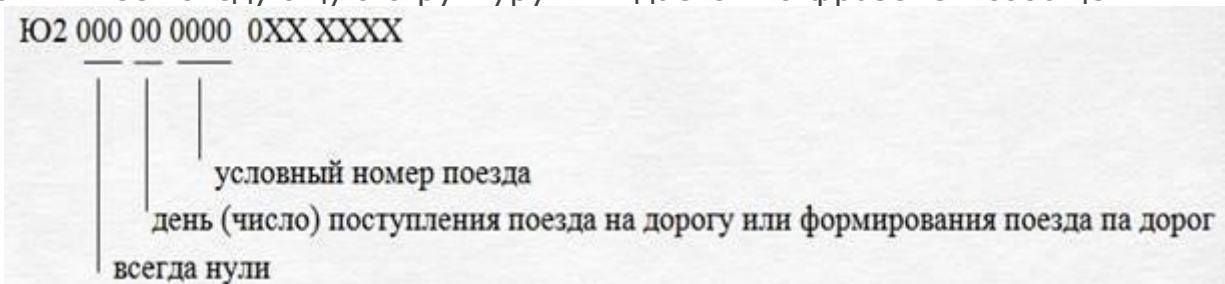
Количество полных суток опоздания проставляется в случае опоздания поезда более чем на 24 часа. Дополнительно в этой графе указывается 9 в случае досрочного прибытия (отправления) поезда (т.е. ранее установленного расписания).

Признак работы с локомотивом может принимать значения: 0-работы нет (может быть опущен); 1 -отцепка локомотива; 2 – смена бригад.

Последние четыре показателя служебной фразы при их нулевых значениях могут не проставляться.

Информационная фраза сообщения 206 аналогична сообщению 200.

При поступлении сообщения 206 и ЭВМ формируется индекс пассажирского поезда, который имеет следующую структуру и выдается во фразе Ю2 сообщения



497:

Пример сообщения 206 с локомотивной фразой:

(: 206 62726 639 1 62846 10 04 00 41 01/01:

504 08880 1 20 30 6110 01899 `ВЕДЕНИИ Н:)

Структура сообщения 208 имеет следующий вид:

(:208 30847 1902 3084 06 2800 5 1205 14 20 08/03

208 - код сообщения.

30847 - железнодорожная станция передачи сообщения.

1902 3084 06 2800 - номер, индекс поезда: при объединении - полученные в результате объединения; при разъединении- номер, индекс разъединяемого поезда.

5 - код операции (1-2 знака).

12 05 14 20 - дата, время (день, месяц, часы, мин.).

08/03 - парк/путь.

Коды операций.

Код 31 - объединение станцией прибывших транзитных поездов в один новый поезд своего формирования. В служебной

фразе сообщения 208 в индексе объединенного поезда станция формирования равняется станции объединения поездов. В информационных фразах указываются номера объединенных поездов, их индексы, номера головных и хвостовых вагонов.

Например:

(208 608807 2233 6088 60 9999 31 03 10 14 00 - поезд своего формирования.

Транзитный поезд.

Б6 8888 23242951 2700001:) - транзитный поезд.

Код 33- прицепка к транзитному поезду поезда своего формирования.

В качестве индекса объединенного поезда в служебной и первой информационной фразах указывается индекс транзитного поезда. Во второй информационной фразе группе вагонов присваивается условный номер поезда - 9999, а в качестве индекса - ж/д станция формирования, которая равняется ж/д станции объединения поездов.

Например:

(:208 608807 3222 6203 45 6076 33 29 09 13 20

Транзитный поезд.

Поезд своего формирования.

Код 34 - прицепка группы вагонов из одного поезда к другому.

В служебной фразе в качестве индекса объединенного поезда и в первой информационной фразе указывается индекс поезда, к которому делается прицепка. Во второй информационной фразе указывается индекс поезда, от которого отцепляется группа вагонов для объединения, и номера головного и хвостового вагонов отцепляемой группы.

Например:

(:208 608807 3120 6203 66 6076 34 12 11 09 10 - первый транзитный поезд.

Б6 6076 4256732 4424113 - группа вагоном первого транзитного поезда.

Группа вагонов второго транзитного поезда.

Структура сообщения 209 имеет следующий вид:

(:209 617007 2552 6190 60 6182 1260115456700:)

209 - код сообщения.

6 17007 - железнодорожная станция передачи сообщения.

2552 - номер поезда.

6190 60 6182 - старый индекс.

1 - причина изменения: 1 - переадресовка поезда; 3 - корректировка номера поезда, если он указан неправильно; 9 - указание стыка сдачи для порожних регулировочных маршрутов.

20 01 15 45 - дата, время изменения.

6700 - новый индекс.

Например, переадресовка поезда с железнодорожной станции назначения Аткарск (6088) на железнодорожную станцию назначения Ртищево (6076):

(:209 608807 2617 6203 06 6088 113 12 10 34 6076:)

Порядок выполнения

Определить круг работников, участвующих в осуществлении перевозочного процесса

Составить схему передачи информационных сообщений при осуществлении перевозочного процесса на заданных железнодорожных станциях.

Содержание отчета

Выводы:

1. Практическая работа №3. Подключение и настройка сетевого адаптера. Подключение и настройка модема.

Цели: обобщить и систематизировать знания по теме «Подключение и настройка сетевого адаптера. Подключение и настройка модема.»; научиться определять параметры сетевого адаптера, настраивать и устанавливать его.

Краткие теоретические сведения

Сетевые карты должны отвечать определенным требованиям в зависимости от того, в какие серверы они устанавливаются. Серверы с сетевыми картами можно разделить на три категории:

1. Сети, в состав которых входит не более 10 станций, используют серверы-десктопы. На таких небольших серверах функционирует небольшое количество информации: не очень объемная база данных или бухгалтерские программы, могут храниться архивы. С таким количеством информации может легко справиться обычная PCI-карта, обеспечивающая скорость 10 или 100 Мбит/сек.

2. Большие локальные сети, насчитывающие 200-300 рабочих станций, используют LAN-серверы. Это более высокий класс серверов, способных выполнять гораздо больший объем работы. Они обладают возможностью разделения файлов и печати, обеспечения межсервисных коммуникаций, функционирования электронной почты. Сетевые карты для больших серверов должны отвечать более высоким требованиям.

3. Суперсерверы, обслуживающие тысячи пользователей, выполняют все приложения, даже самые ресурсоемкие. Такие серверы могут обслуживать бизнес-процессы крупных предприятий. Суперсерверы используют сетевые карты, сравнимые по своим возможностям с сетевыми процессорами.

Исходя из своего предназначения, сетевая карта должна отвечать определенным требованиям. Производительность карты определяется тремя составляющими: микросхемным, конструктивным и программным уровнем карты.

Чем выше микросхемный уровень карты, тем больше она может выполнять функций, ранее выполнявшихся процессором. Серверные сетевые карты, разгружая процессор, значительно оптимизируют работу сети.

Конструктивный уровень карты определяется тем, сколько сегментов сети к нему

возможно присоединить. Чем больше портов имеет карта, тем производительнее ее работа. Конструкция карты должна учитывать, что сеть работает в круглосуточном режиме. При замене карты не должна останавливаться работа сервера.

Программное обеспечение карты позволяет значительно расширить функции управления сетью, вести мониторинг, разделять трафик, группировать порты в логические каналы.

2. Задание к работе

1. Определите тип сетевой карты (тип шины, тип среды для передачи данных).

Осмотрите сетевую карту. Определите тип шины, к которой она подключается (для этого посмотрите на ту часть сетевой карты, которая имеет контакты):

- карта подключается к шине PCI (Peripheral Component Interconnect - соединение периферийных компонент), если длина контактной пластины менее 10 см;
- карта подключается к шине ISA (Industry Standard Architecture - стандартная промышленная архитектура), если длина контактной пластины более 10 см.

Определите тип физической среды, с которой работает сетевая карта. Посмотрите на металлическую пластину, к которой крепится карта.

Круглый коннектор свидетельствует о том, что эта карта для коаксиального кабеля; разъем RJ-45 - для работы с витой парой.

Визуально определите на карте наличие микросхемы для загрузки компьютера по сети.

2. Установите сетевой адаптер в компьютер.

Выключите компьютер и откройте системный блок.

Вставьте сетевую карту в соответствующий разъем на материнской плате и закрепите ее в корпусе.

Закройте системный блок и включите компьютер.

В процессе загрузки ОС определяет подключенное оборудование. Если сетевая карта соответствует стандарту Plug and Play, то она будет найдена ОС и автоматически настроена. Если ОС не сможет определить установленную сетевую карту, то потребуется вручную установить ее драйвера.

Проверьте установку сетевой карты:

- откройте диалоговое окно Диспетчер устройств (Пуск/Панель управления/Система/Оборудование/Диспетчер устройств);
- раскройте список Сетевые платы.

Если в этом списке есть название адаптера, то установка прошла успешно.

3. Изучите параметры сетевого адаптера.

Откройте окно параметров сетевого адаптера (воспользуйтесь Диспетчером устройств).

Определите физический (MAC, Medium Access Control - управление доступом к носителю) адрес сетевой карты помощью команды ipconfig:

- запустите консоль (командную строку) любым способом (например, Пуск/Программы/Стандартные/Командная строка);
- введите команду ipconfig с параметром all;
- в полученном списке найдите строку Физический адрес.

Физический адрес и будет являться MAC-адресом сетевого адаптера.

Например, выведенный системой список может выглядеть так:

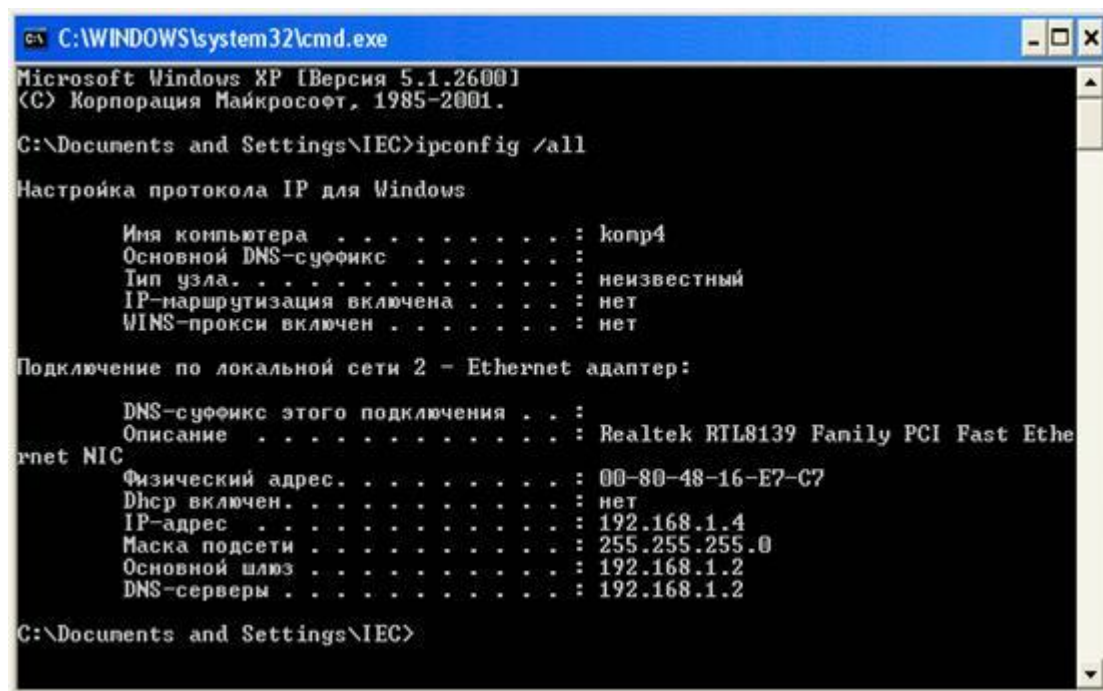


Рисунок 1. Результат работы команды ipconfig /all

3.Контрольные вопросы:

1. Классификация сетевых адаптеров.
2. Основные характеристики сетевых адаптеров.
3. Основные функции сетевых адаптеров.

Практическая работа №4

Сети Ethernet и Fast Ethernet

Цель работы: Изучить структуру сети Ethernet.
Теоретические основы

Наибольшее распространение среди стандартных сетей получила сеть *Ethernet*. Впервые она появилась в 1972 году (разработчиком выступила известная фирма Xerox). Сеть оказалась довольно удачной, и вследствие этого ее в 1980 году поддержали такие крупнейшие компании, как DEC и Intel (объединение этих компаний назвали DIX по первым буквам их названий). Их стараниями в 1985 году сеть *Ethernet* стала международным стандартом, ее приняли крупнейшие международные организации по стандартам: комитет 802 IEEE (Institute of Electrical and Electronic Engineers) и ECMA (European Computer Manufacturers Association).

Стандарт получил название IEEE 802.3 (по-английски читается как "eight oh two dot three"). Он определяет множественный доступ к моноканалу типа шина с обнаружением конфликтов и контролем передачи, то есть с уже упоминавшимся методом доступа CSMA/CD. Этому стандарту удовлетворяли и некоторые другие сети, так как уровень его детализации не высок. В результате сети стандарта IEEE 802.3 нередко были несовместимы между собой как по конструктивным, так и по электрическим характеристикам. Однако в последнее время стандарт IEEE 802.3 считается стандартом именно сети *Ethernet*.

Основные характеристики первоначального стандарта IEEE 802.3:

- топология – шина;
- *среда передачи* – коаксиальный кабель;
- скорость передачи – 10 Мбит/с;
- максимальная длина сети – 5 км;
- максимальное количество абонентов – до 1024;
- длина сегмента сети – до 500 м;
- количество абонентов на одном сегменте – до 100;
- метод доступа – CSMA/CD;
- передача узкополосная, то есть без модуляции (моноканал).

Строго говоря, между стандартами IEEE 802.3 и *Ethernet* существуют незначительные отличия, но о них обычно предпочитают не вспоминать.

Сеть *Ethernet* сейчас наиболее популярна в мире (более 90% рынка), предположительно таковой она и останется в ближайшие годы. Этому в немалой степени способствовало то, что с самого начала характеристики, параметры, протоколы сети были открыты, в результате чего огромное число производителей во всем мире стали выпускать аппаратуру *Ethernet*, полностью совместимую между собой.

В классической сети *Ethernet* применялся 50-омный коаксиальный кабель двух видов (толстый и тонкий). Однако в последнее время (с начала 90-х годов) наибольшее распространение получила версия *Ethernet*, использующая в качестве *среды передачи* витые пары. Определен также стандарт для применения в сети оптоволоконного кабеля. Для учета этих изменений в изначальный стандарт IEEE 802.3 были сделаны соответствующие дополнения. В 1995 году появился дополнительный стандарт на более быструю версию *Ethernet*, работающую на скорости 100 Мбит/с (так называемый *Fast Ethernet*, стандарт IEEE 802.3u), использующую в качестве *среды передачи* витую пару или оптоволоконный кабель. В 1997 году появилась и версия на скорость 1000 Мбит/с (Gigabit Ethernet, стандарт IEEE 802.3z).

Помимо стандартной топологии шина все шире применяются топологии типа пассивная звезда и пассивное дерево. При этом предполагается использование репитеров и репитерных концентраторов, соединяющих между собой различные части (сегменты) сети. В результате может сформироваться древовидная структура на сегментах разных типов (рис 1).

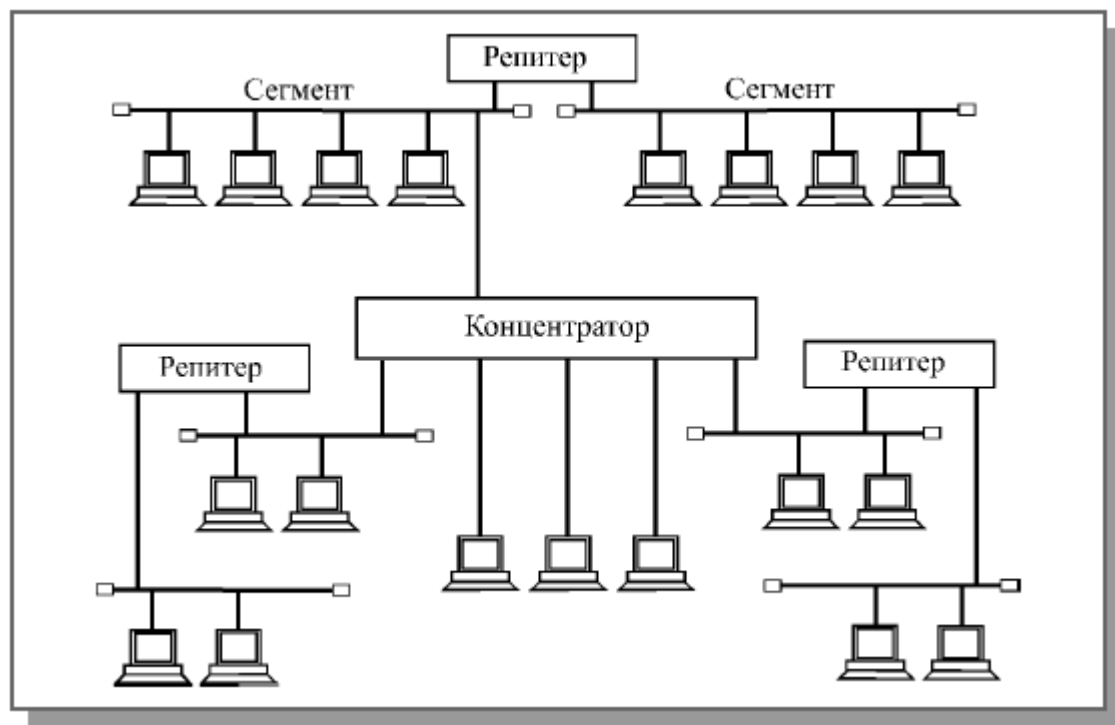


Рис. 1. Классическая топология сети Ethernet

В качестве сегмента (части сети) может выступать классическая шина или единственный абонент. Для шинных сегментов используется коаксиальный кабель, а для лучей пассивной звезды (для присоединения к концентратору одиночных компьютеров) – витая пара и оптоволоконный кабель. Главное требование к полученной в результате топологии – чтобы в ней не было замкнутых путей (петель). Фактически получается, что все абоненты соединены в физическую шину, так как сигнал от каждого из них распространяется сразу во все стороны и не возвращается назад (как в кольце).

Максимальная длина кабеля сети в целом (максимальный путь сигнала) теоретически может достигать 6,5 километров, но практически не превышает 3,5 километров.

В сети *Fast Ethernet* не предусмотрена физическая топология шина, используется только пассивная звезда или пассивное дерево. К тому же в *Fast Ethernet* гораздо более жесткие требования к предельной длине сети. Ведь при увеличении в 10 раз скорости передачи и сохранении *формата пакета* его минимальная длина становится в десять раз короче. Таким образом в 10 раз уменьшается допустимая величина двойного времени прохождения сигнала по сети (5,12 мкс против 51,2 мкс в *Ethernet*).

Для передачи информации в сети *Ethernet* применяется стандартный манчестерский код.

Доступ к сети *Ethernet* осуществляется по случайному методу CSMA/CD, обеспечивающему равноправие абонентов. В сети используются пакеты переменной длины со структурой, представленной на [рис.2](#). (цифры показывают количество байт)



Рис 2. Структура пакета сети Ethernet

Длина кадра *Ethernet* (то есть пакета без преамбулы) должна быть не менее 512 битовых интервалов или 51,2 мкс (именно такова предельная величина двойного времени прохождения в сети). Предусмотрена индивидуальная, групповая и широковещательная адресация.

В пакет *Ethernet* входят следующие поля:

- Преамбула состоит из 8 байт, первые семь представляют собой код 10101010, а последний байт – код 10101011. В стандарте IEEE 802.3 восьмой байт называется признаком начала кадра (SFD – Start of Frame Delimiter) и образует отдельное поле пакета.
- Адреса получателя (приемника) и отправителя (передатчика) включают по 6 байт и строятся по стандарту, описанному в разделе "Адресация пакетов" лекции 4. Эти адресные поля обрабатываются аппаратурой абонентов.
- Поле управления (L/T – Length/Type) содержит информацию о длине поля данных. Оно может также определять тип используемого протокола. Принято считать, что если значение этого поля не больше 1500, то оно указывает на длину поля данных. Если же его значение больше 1500, то оно определяет тип кадра. Поле управления обрабатывается программно.
- Поле данных должно включать в себя от 46 до 1500 байт данных. Если пакет должен содержать менее 46 байт данных, то поле данных дополняется байтами заполнения. Согласно стандарту IEEE 802.3, в структуре пакета выделяется специальное поле заполнения (pad data – незначащие данные), которое может иметь нулевую длину, когда данных достаточно (больше 46 байт).
- Поле контрольной суммы (FCS – Frame Check Sequence) содержит 32-разрядную циклическую контрольную сумму пакета (CRC) и служит для проверки правильности передачи пакета.

Таким образом, минимальная длина кадра (пакета без преамбулы) составляет 64 байта (512 бит). Именно эта величина определяет максимально допустимую двойную задержку распространения сигнала по сети в 512 битовых интервалов (51,2 мкс для *Ethernet* или 5,12 мкс для *Fast Ethernet*). Стандарт предполагает, что преамбула может уменьшаться при прохождении пакета через различные сетевые устройства, поэтому она не учитывается. Максимальная длина кадра равна 1518 байтам (12144 бита, то есть 1214,4 мкс для *Ethernet*, 121,44 мкс для *Fast Ethernet*). Это важно для выбора размера буферной памяти сетевого оборудования и для оценки общей загрузки сети.

Выбор формата преамбулы не случаен. Дело в том, что последовательность чередующихся единиц и нулей (101010...10) в манчестерском коде характеризуется тем, что имеет переходы только в середине битовых интервалов (см. раздел 2.6.3), то есть только информационные переходы. Безусловно, приемнику просто настроиться (синхронизоваться) при такой последовательности, даже если она по какой-то причине укорачивается на

несколько бит. Последние два единичных бита преамбулы (11) существенно отличаются от последовательности 101010...10 (появляются переходы еще и на границе битовых интервалов). Поэтому уже настроившийся приемник легко может выделить их и детектировать тем самым начало полезной информации (начало кадра).

Для сети *Ethernet*, работающей на скорости 10 Мбит/с, стандарт определяет четыре основных типа сегментов сети, ориентированных на различные *среды передачи* информации:

- 10BASE5 (толстый коаксиальный кабель);
- 10BASE2 (тонкий коаксиальный кабель);
- 10BASE-T (витая пара);
- 10BASE-FL (оптоволоконный кабель).

Наименование сегмента включает в себя три элемента: цифра "10" означает скорость передачи 10 Мбит/с, слово BASE – передачу в основной полосе частот (то есть без модуляции высокочастотного сигнала), а последний элемент – допустимую длину сегмента: "5" – 500 метров, "2" – 200 метров (точнее, 185 метров) или тип линии связи: "Т" – витая пара (от английского "twisted-pair"), "F" – оптоволоконный кабель (от английского "fiber optic").

Точно так же для сети *Ethernet*, работающей на скорости 100 Мбит/с (*Fast Ethernet*) стандарт определяет три типа сегментов, отличающихся типами *среды передачи*:

- 100BASE-T4 (счетверенная витая пара);
- 100BASE-TX (сдвоенная витая пара);
- 100BASE-FX (оптоволоконный кабель).

Здесь цифра "100" означает скорость передачи 100 Мбит/с, буква "Т" – витую пару, буква "F" – оптоволоконный кабель. Типы 100BASE-TX и 100BASE-FX иногда объединяют под именем 100BASE-X, а 100BASE-T4 и 100BASE-TX – под именем 100BASE-T.

Подробнее особенности аппаратуры *Ethernet*, а также алгоритма управления обменом CSMA/CD и алгоритма вычисления циклической контрольной суммы (CRC) будут рассмотрены далее в специальных разделах курса. Здесь следует отметить только то, что сеть *Ethernet* не отличается ни рекордными характеристиками, ни оптимальными алгоритмами, она уступает по ряду параметров другим стандартным сетям. Но благодаря мощной поддержке, высочайшему уровню стандартизации, огромным объемам выпуска технических средств, *Ethernet* выгодно выделяется среди других стандартных сетей, и поэтому любую другую сетевую технологию принято сравнивать именно с *Ethernet*.

Развитие технологии *Ethernet* идет по пути все большего отхода от первоначального стандарта. Применение новых *сред передачи* и коммутаторов позволяет существенно увеличить размер сети. Отказ от *метода управления* CSMA/CD (при полнодуплексном режиме обмена) дает возможность резко повысить эффективность работы и снять ограничения с длины сети. Тем не менее, все новые разновидности сети также называются сетью *Ethernet*.

Расчет pdv

Для упрощения расчетов обычно используются справочные данные IEEE, содержащие значения задержек распространения сигналов в повторителях, приемопередатчиках и различных физических средах. В табл. 1 приведены данные, необходимые для расчета значения PDV для всех физических стандартов сетей Ethernet. Битовый интервал обозначен как bt.

Таблица 1. Данные для расчета значения PDV

Тип сегмента	База левого сегмента, bt	База промежуточного сегмента, bt	База правого сегмента, bt	Задержка среды на 1 м, bt	Максимальная длина сегмента, м
10Base-5	11,8	46,5	169,5	0,0866	500
10Base-2	11,8	46,5	169,5	0,1026	185
10Base-T	15,3	42,0	165,0	0,113	100
10Base-FB	—	24,0	—	0,1	2000
10Base-FL	12,3	33,5	156,5	0,1	2000
FOIRL	7,8	29,0	152,0	0,1	1000
AUI (> 2 м)	0	0	0	0,1026	2+48

Комитет 802.3 старался максимально упростить выполнение расчетов, поэтому данные, приведенные в таблице, включают сразу несколько этапов прохождения сигнала. Например, задержки, вносимые повторителем, состоят из задержки входного трансивера, задержки блока повторения и задержки выходного трансивера. Тем не менее в таблице все эти задержки представлены одной величиной, названной базой сегмента. Чтобы не нужно было два раза складывать задержки, вносимые кабелем, в таблице даются удвоенные величины задержек для каждого типа кабеля.

В таблице используются также такие понятия, как левый сегмент, правый сегмент и промежуточный сегмент. Поясним эти термины на примере сети, приведенной на рис. 3. Левым сегментом называется сегмент, в котором начинается путь сигнала от выхода передатчика конечного узла. На примере это сегмент 1. Затем сигнал проходит через промежуточные сегменты 2-5 и доходит до приемника наиболее удаленного сегмента 6, который называется правым. Именно здесь в худшем случае происходит столкновение кадров и возникает коллизия, что, и подразумевается в таблице.

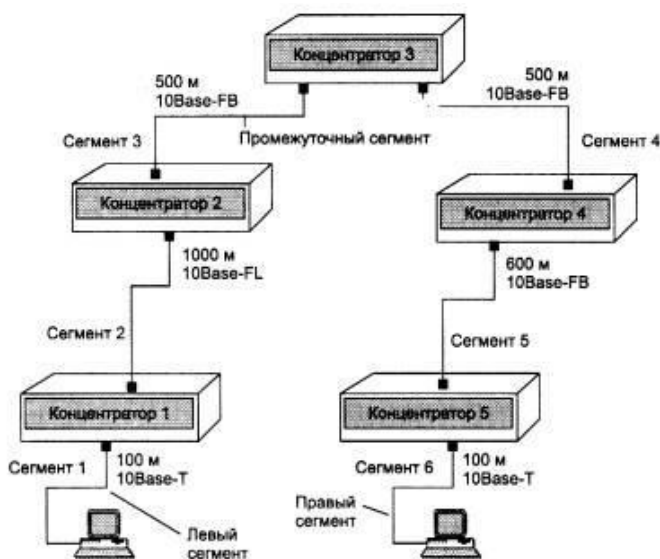


Рис. 3. Пример сети Ethernet, состоящей из сегментов различных физических стандартов

С каждым сегментом связана постоянная задержка, названная базой, которая зависит только от типа сегмента и от положения сегмента на пути сигнала (левый, промежуточный или правый). База правого сегмента, в котором возникает коллизия, намного превышает базу левого и промежуточных сегментов.

Кроме этого, с каждым сегментом связана задержка распространения сигнала вдоль кабеля сегмента, которая зависит от длины сегмента и вычисляется путем умножения времени распространения сигнала по одному метру кабеля (в битовых интервалах) на длину кабеля в метрах.

Расчет заключается в вычислении задержек, вносимых каждым отрезком кабеля (приведенная в таблице задержка сигнала на 1 м кабеля умножается на длину сегмента), а затем суммировании этих задержек с базами левого, промежуточных и правого сегментов. Общее значение PDV не должно превышать 575.

Так как левый и правый сегменты имеют различные величины базовой задержки, то в случае различных типов сегментов на удаленных краях сети необходимо выполнить расчеты дважды: один раз принять в качестве левого сегмента сегмент одного типа, а во второй - сегмент другого типа. Результатом можно считать максимальное значение PDV. В нашем примере крайние сегменты сети принадлежат к одному типу - стандарту 10Base-T, поэтому двойной расчет не требуется, но если бы они были сегментами разного типа, то в первом случае нужно было бы принять в качестве левого сегмент между станцией и концентратором 1, а во втором считать левым сегмент между станцией и концентратором 5.

Приведенная на рисунке сеть в соответствии с правилом 4-х хабов не является корректной - в сети между узлами сегментов 1 и 6 имеется 5 хабов, хотя не все сегменты являются сегментами 10Base-FB. Кроме того, общая длина сети равна 2800 м, что нарушает правило 2500 м.

Расчетная формула PDV:

(Т зад. баз. лев.+L1*Т зад. среды) + ... + (Т зад. баз. прав.+L6*Т зад. среды) =PDV<575bt

Для расчетов вам понадобятся данные комитета IEEE 802.3 о задержках сигналов в повторителях и кабельных сегментах (табл. 1,2):

Рассчитаем значение PDV для нашего примера.

Левый сегмент 1/ 15,3 (база) + 100 * 0,113= 26,6.

Промежуточный сегмент 2/33,5 + 1000 * 0,1 = 133,5.

Промежуточный сегмент 3/ 24 + 500 * 0,1 = 74,0.

Промежуточный сегмент 4/24 + 500 * 0,1 = 74,0.

Промежуточный сегмент 5/ 24 + 600 * 0,1 = 84,0.

Правый сегмент 6/165 + 100 * 0,113 = 176,3.

Сумма всех составляющих дает значение PDV, равное 568,4.

Так как значение PDV меньше максимально допустимой величины 575, то эта сеть проходит по критерию времени двойного оборота сигнала, несмотря на то, что ее общая длина составляет больше 2500 м, а количество повторителей - больше 4-х.

Расчет pvv

Чтобы признать конфигурацию сети корректной, нужно рассчитать также уменьшение межкадрового интервала повторителями, то есть величину PVV.

Для расчета PW также можно воспользоваться значениями максимальных величин уменьшения межкадрового интервала при прохождении повторителей различных физических сред, рекомендованными IEEE и приведенными в табл. 2.

Таблица 2. Сокращение межкадрового интервала повторителями

Тип сегмента	Передающий сегмент, bt	Промежуточный сегмент, bt
10Base-5 или 10Base-2	16	11
10Base-FB	—	2
10Base-FL	10,5	8
10Base-T	10,5	8

В соответствии с этими данными рассчитаем значение PVV для нашего примера.

Левый сегмент 1 10Base-T: сокращение в 10,5 bt.

Промежуточный сегмент 2 10Base-FL: 8.

Промежуточный сегмент 3 10Base-FB: 2.

Промежуточный сегмент 4 10Base-FB: 2.

Промежуточный сегмент 5 10Base-FB: 2.

Сумма этих величин дает значение PW, равное 24,5, что меньше предельного значения в 49 битовых интервала.

Практическая часть

Рассчитать PDVиPVVсогласно своему варианту.

Варианты	Тип сегмента		Тип кабеля	Длина сегмента
1	Сегмент 1		10Base-5	50
	Сегмент 2		10Base-2	100
	Сегмент 3		10Base-T	50
	Сегмент 4		10Base-FB	1000
	Сегмент 5		10Base-FL	1000
	Сегмент 6		FOIRL	200
2	Сегмент 1		10Base-5	10
	Сегмент 2		10Base-2	10
	Сегмент 3		10Base-T	50
	Сегмент 4		10Base-FB	50
	Сегмент 5		10Base-FL	120
	Сегмент 6		FOIRL	500
3	Сегмент 1		10Base-5	600
	Сегмент 2		10Base-2	400
	Сегмент 3		10Base-T	300
	Сегмент 4		10Base-FB	50
	Сегмент 5		10Base-FL	10
	Сегмент 6		FOIRL	25
4	Сегмент 1		10Base-5	250
	Сегмент 2		10Base-2	100
	Сегмент 3		10Base-T	300
	Сегмент 4		10Base-FB	600
	Сегмент 5		10Base-FL	50
	Сегмент 6		FOIRL	500
5	Сегмент 1		10Base-5	1000
	Сегмент 2		10Base-2	300
	Сегмент 3		10Base-T	100
	Сегмент 4		10Base-FB	500
	Сегмент 5		10Base-FL	300
	Сегмент 6		FOIRL	500
6	Сегмент 1		10Base-5	100
	Сегмент 2		10Base-2	20
	Сегмент 3		10Base-T	30
	Сегмент 4		10Base-FB	356
	Сегмент 5		10Base-FL	1000
	Сегмент 6		FOIRL	500
7	Сегмент 1		10Base-5	300
	Сегмент 2		10Base-2	100
	Сегмент 3		10Base-T	1000
	Сегмент 4		10Base-FB	50
	Сегмент 5		10Base-FL	100
	Сегмент 6		FOIRL	1000
8	Сегмент 1		10Base-5	500
	Сегмент 2		10Base-2	300
	Сегмент 3		10Base-T	400
	Сегмент 4		10Base-FB	90

	Сегмент 5	10Base-FL	300	
	Сегмент 6	FOIRL	50	
9	Сегмент 1	10Base-5	100	
	Сегмент 2	10Base-2	120	
	Сегмент 3	10Base-T	300	
	Сегмент 4	10Base-FB	400	
	Сегмент 5	10Base-FL	100	
	Сегмент 6	FOIRL	1000	
10	Сегмент 1	10Base-5	1000	
	Сегмент 2	10Base-2	300	
	Сегмент 3	10Base-T	100	
	Сегмент 4	10Base-FB	100	
	Сегмент 5	10Base-FL	400	
	Сегмент 6	FOIRL	900	
11	Сегмент 1	10Base-5	300	
	Сегмент 2	10Base-2	65	
	Сегмент 3	10Base-T	70	
	Сегмент 4	10Base-FB	1000	
	Сегмент 5	10Base-FL	1000	
	Сегмент 6	FOIRL	500	
12	Сегмент 1	10Base-5	300	
	Сегмент 2	10Base-2	145	
	Сегмент 3	10Base-T	1000	
	Сегмент 4	10Base-FB	100	
	Сегмент 5	10Base-FL	105	
	Сегмент 6	FOIRL	1000	
13	Сегмент 1	10Base-5	1000	
	Сегмент 2	10Base-2	400	
	Сегмент 3	10Base-T	100	
	Сегмент 4	10Base-FB	50	
	Сегмент 5	10Base-FL	50	
	Сегмент 6	FOIRL	80	
14	Сегмент 1	10Base-5	100	
	Сегмент 2	10Base-2	105	
	Сегмент 3	10Base-T	300	
	Сегмент 4	10Base-FB	100	
	Сегмент 5	10Base-FL	85	
	Сегмент 6	FOIRL	1000	
15	Сегмент 1	10Base-5	100	
	Сегмент 2	10Base-2	50	
	Сегмент 3	10Base-T	45	
	Сегмент 4	10Base-FB	100	
	Сегмент 5	10Base-FL	40	
	Сегмент 6	FOIRL	300	
16	Сегмент 1	10Base-5	300	
	Сегмент 2	10Base-2	105	
	Сегмент 3	10Base-T	50	

	Сегмент 4	10Base-FB	1000
	Сегмент 5	10Base-FL	1000
	Сегмент 6	FOIRL	500
17	Сегмент 1	10Base-5	300
	Сегмент 2	10Base-2	105
	Сегмент 3	10Base-T	1000
	Сегмент 4	10Base-FB	50
	Сегмент 5	10Base-FL	85
	Сегмент 6	FOIRL	1000
18	Сегмент 1	10Base-5	1000
	Сегмент 2	10Base-2	300
	Сегмент 3	10Base-T	300
	Сегмент 4	10Base-FB	50
	Сегмент 5	10Base-FL	450
	Сегмент 6	FOIRL	50
19	Сегмент 1	10Base-5	1000
	Сегмент 2	10Base-2	123
	Сегмент 3	10Base-T	367
	Сегмент 4	10Base-FB	300
	Сегмент 5	10Base-FL	105
	Сегмент 6	FOIRL	250
20	Сегмент 1	10Base-5	270
	Сегмент 2	10Base-2	50
	Сегмент 3	10Base-T	85
	Сегмент 4	10Base-FB	100
	Сегмент 5	10Base-FL	300
	Сегмент 6	FOIRL	400
21	Сегмент 1	10Base-5	1000
	Сегмент 2	10Base-2	123
	Сегмент 3	10Base-T	367
	Сегмент 4	10Base-FB	300
	Сегмент 5	10Base-FL	105
	Сегмент 6	FOIRL	1000
22	Сегмент 1	10Base-5	300
	Сегмент 2	10Base-2	105
	Сегмент 3	10Base-T	50
	Сегмент 4	10Base-FB	1000
	Сегмент 5	10Base-FL	1000
	Сегмент 6	FOIRL	500
23	Сегмент 1	10Base-5	600
	Сегмент 2	10Base-2	200
	Сегмент 3	10Base-T	400
	Сегмент 4	10Base-FB	50
	Сегмент 5	10Base-FL	50
	Сегмент 6	FOIRL	10
24	Сегмент 1	10Base-5	1000
	Сегмент 2	10Base-2	105

	Сегмент 3	10Base-T	100
	Сегмент 4	10Base-FB	560
	Сегмент 5	10Base-FL	105
	Сегмент 6	FOIRL	300
25	Сегмент 1	10Base-5	65
	Сегмент 2	10Base-2	70
	Сегмент 3	10Base-T	1000
	Сегмент 4	10Base-FB	1000
	Сегмент 5	10Base-FL	500
	Сегмент 6	FOIRL	150
26	Сегмент 1	10Base-5	300
	Сегмент 2	10Base-2	85
	Сегмент 3	10Base-T	100
	Сегмент 4	10Base-FB	1000
	Сегмент 5	10Base-FL	1000
	Сегмент 6	FOIRL	100
27	Сегмент 1	10Base-5	80
	Сегмент 2	10Base-2	85
	Сегмент 3	10Base-T	1000
	Сегмент 4	10Base-FB	10
	Сегмент 5	10Base-FL	105
	Сегмент 6	FOIRL	1000
28	Сегмент 1	10Base-5	345
	Сегмент 2	10Base-2	400
	Сегмент 3	10Base-T	200
	Сегмент 4	10Base-FB	34
	Сегмент 5	10Base-FL	67
	Сегмент 6	FOIRL	50
29	Сегмент 1	10Base-5	56
	Сегмент 2	10Base-2	57
	Сегмент 3	10Base-T	57
	Сегмент 4	10Base-FB	377
	Сегмент 5	10Base-FL	39
	Сегмент 6	FOIRL	236
30	Сегмент 1	10Base-5	236
	Сегмент 2	10Base-2	53
	Сегмент 3	10Base-T	18
	Сегмент 4	10Base-FB	233
	Сегмент 5	10Base-FL	53
	Сегмент 6	FOIRL	104

Контрольные вопросы.

1. Характеристики сети Ethernet.
2. Структура пакета Ethernet.
3. Методы расчёты и физический смысл PDV PVV

Практическая работа №5

Расчет сети Fast Ethernet

Цели работы:

Цель данной работы — изучение принципов технологий Ethernet и Fast Ethernet и практическое освоение методик оценки работоспособности сети, построенной на базе технологии Fast Ethernet.

Теоретические сведения

Технология Ethernet. Спецификация сети Ethernet была предложена фирмами DEC, Intel и Xerox (DIX) в 1980 г., и несколько позже на ее основе появился стандарт IEEE 802.3.

Первые версии Ethernet v1.0 и Ethernet v2.0 в качестве среды передачи использовали только коаксиальный кабель. Стандарт IEEE 802.3 позволяет в качестве среды передачи использовать также витую пару и оптоволокно. В 1995 г. был принят стандарт IEEE 802.3u (Fast Ethernet) со скоростью 100 Мбит/с, а в 1997 г. — IEEE 802.3z (Gigabit Ethernet — 1000 Мбит/с). Осенью 1999 г. принят стандарт IEEE 802.3ab — Gigabit Ethernet на витой паре категории 5.

В обозначениях Ethernet (10BASE2, 100BASE-TX и др.) первый элемент обозначает скорость передачи данных в Мбит/с; второй элемент BASE означает, что используется прямая (немодулированная) передача; третий элемент обозначает округленное значение длины кабеля в сотнях метров (10BASE2 — 185 м, 10BASE5 — 500 м) или тип среды передачи (T, TX, T2, T4 — витая пара; FX, FL, FB, SX и LX — оптоволокно; CX — твинаксиальный кабель для Gigabit Ethernet).

В основе Ethernet лежит *метод множественного доступа к среде передачи с прослушиванием несущей и обнаружением коллизий — CSMA/CD*

- (*Carrier Sense with Multiple Access and Collision Detection*), реализуемый адаптерами каждого узла сети на аппаратном или микропрограммном уровне:

- все адаптеры имеют устройство доступа к среде (MAU) — трансивер, В подключенный к общей (разделяемой) среде передачи данных;
- каждый адаптер узла перед передачей информации прослушивает В линию до момента отсутствия сигнала (несущей);
- затем адаптер формирует кадр (frame), начинающийся с синхронизирующей преамбулы, за которой следует поток двоичных данных в само-синхронизирующемся (манчестерском) коде;
- другие узлы принимают посланный сигнал, синхронизируются В по преамбуле и декодируют его в последовательность бит;
- окончание передачи кадра определяется обнаружением приемником В отсутствия несущей;
- в случае обнаружения *коллизии* (столкновения двух сигналов от разных узлов) передающие узлы прекращают передачу кадра, после чего В через случайный промежуток времени (каждый через свой) осуществляют В повторную попытку передачи после освобождения линии; при очередной В неудаче делается следующая попытка (и так до 16 раз), причем интервал В задержки увеличивается;
- коллизия обнаруживается приемником по нестандартной длине В кадра, которая не может быть меньше 64 байт, не считая преамбулы;
- между кадрами должен обеспечиваться временной зазор (*межкадровый или межпакетный промежуток, IPG — inter-packet gap*) длительностью В 9,6 мкс — узел не имеет права начать передачу раньше, чем через интервал В IPG, после определения момента пропадания несущей.

Определение 1. *Домен коллизий* — группа узлов, связанных общей средой (кабелями и повторителями) передачи.

Протяженность домена коллизий ограничивается временем распространения сигнала между наиболее удаленными друг от друга узлами.

Определение 2. *Диаметр домена коллизий* — расстояние между двумя наиболее удаленными друг от друга оконечными устройствами.

Определение 3. *Битовый интервал* — время, необходимое для передачи одного бита.

Битовый интервал в Ethernet (при скорости 10 Мбит/с) составляет 0,1 мкс.

Технология Fast Ethernet. В технологии Fast Ethernet величина битового интервала составляет 0,01 мкс, что дает десятикратное увеличение скорости передачи данных. При этом формат кадра, объем переносимых в кадре данных и механизм доступа к каналу передачи данных остались без изменения по сравнению с Ethernet.

В Fast Ethernet используется среда передачи данных для работы на скорости 100 Мбит/с, которая в спецификации IEEE 802.3u имеет обозначения «100BASE-T4» и «100BASE-TX» (витая пара); «100BASE-FX» и «100BASE-SX» (оптоволокно).

Правила построения сети

Первая модель сети Fast Ethernet. Модель представляет собой, по сути, набор правил построения сети (табл. Л.1):

- - длина каждого сегмента витой пары должна быть меньше 100 м;
- - длина каждого оптоволоконного сегмента должна быть меньше 412 м;
- - если используются кабели МП (Media Independent Interface), то каждый из них должен быть меньше 0,5 м;
- - задержки, вносимые кабелем МП, не учитываются при оценке временных параметров сети, так как они являются составной частью задержек, вносимых оконечными устройствами (терминалами) и повторителями.

Таблица Л. 1

Предельно допустимый диаметр домена коллизий в Fast Ethernet

Тип повторителя	Все сегменты ТХ или Т4	Все сегменты FX	Сочетание сегментов В (Т4 и ТХ/FX)	Сочетание сегментов В (ТХ и FX)
Сегмент, соединяющий два узла без повторителей	100	412,0		
Один повторитель класса I	200	272,0	231,0	260,8
Один повторитель класса II	200	320,0	—	308,8
Два повторителя класса II	205	228,0	—	216,2

Стандартом определены два класса повторителей:

- повторители класса I выполняют преобразование входных сигналов В в цифровой вид, а при передаче снова перекодируют цифровые данные В в физические сигналы; преобразование сигналов в повторителе требует В некоторого времени, поэтому в домене коллизий допускается только один В повторитель класса I;
- повторители класса II немедленно передают полученные сигналы без В всякого преобразования, поэтому к ним можно

подключать только сегменты, В использующие одинаковые способы кодирования данных; можно использовать не более двух повторителей класса II в одном домене коллизий.

Вторая модель сети Fast Ethernet. Вторая модель содержит последовательность расчетов временных параметров сети при полудуплексном режиме обмена данными. Диаметр домена коллизий и количество сегментов в нем ограничены временем двойного оборота, необходимым В для правильной работы механизма обнаружения и разрешения коллизий В (табл. Л.2).

Таблица Л2

Временные задержки компонентов сети Fast Ethernet

Компонент	Удельное время двойного оборота, би/м	Максимальное время двойного оборота, би
Пара терминалов TX/FX	-	100
Пара терминалов T4	-	138

Компонент	Удельное время двойного оборота, би/м	Максимальное время двойного оборота, би
Пара терминалов T4 и TX/PX	-	127
Витая пара категории 3	1,14	114(100 м)

Витая пара категории 4	1,14	114(100 м)
Витая пара категории 5	1,112	111,2 (100 м)
Экранированная витая пара	1,112	И 1,2 (100 м)
Оптоволокно	1,0	412 (412 м)
Повторитель класса I	-	140
Повторитель класса II, имеющий порты типа TX/RX	-	92
Повторитель класса II, имеющий порты типа T4	-	67

Время двойного оборота рассчитывается для наихудшего (в смысле рае-пространения сигнала) пути между двумя узлами домена коллизий. Расчет выполняется путем суммирования временных задержек в сегментах, повторителях и терминалах.

Для вычисления времени двойного оборота нужно умножить длину сегмента на величину удельного времени двойного оборота соответствующего сегмента. Определив времена двойного оборота для всех сегментов наихудшего пути, к ним нужно прибавить задержку, вносимую парой оконечныхВ узлов и повторителями. Для учета непредвиденных задержек к полученному результату рекомендуется добавить еще 4 битовых интервала (би)В и сравнить результат с числом 512. Если полученный результат не превышает 512 би, то сеть считается работоспособной.

Пример расчета конфигурации сети Fast Ethernet. На рис. Л.28 приведен пример одной из предельно допустимых конфигураций сети Fast Ethernet.

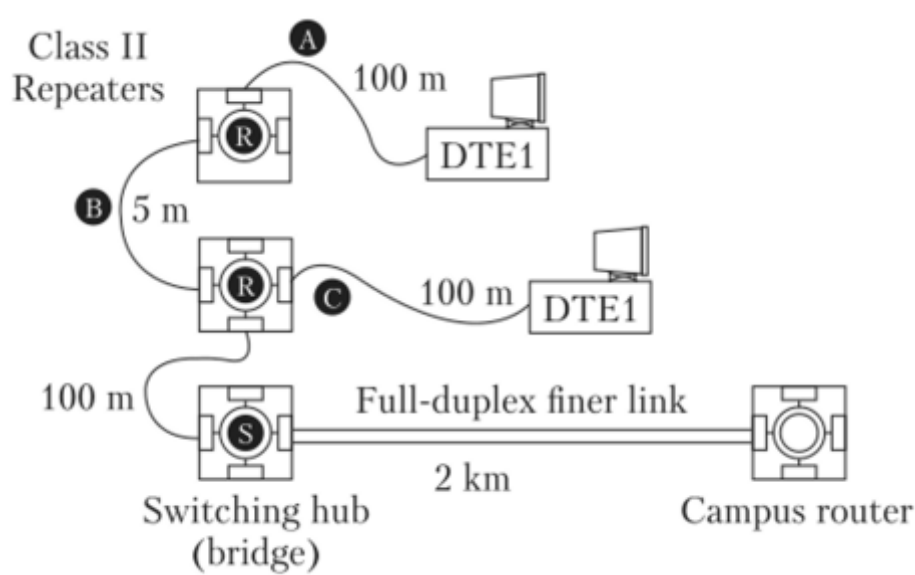


Рис. Л.28. Пример допустимой конфигурации сети Fast Ethernet

Диаметр домена коллизий вычисляется как сумма длин сегментов А (100 м), В (5 м) и С (100 м) и равен 205 м. Длина сегмента, соединяющего В повторители, может быть более 5 м, если при этом диаметр домена коллизий не превышает допустимый для данной конфигурации предел. Коммутатор (switching hub), входящий в состав сети (см. рис. Л.28), считается В окончательным устройством, так как коллизии через него не распространяются. В Поэтому 2-километровый сегмент оптоволоконного кабеля, соединяющий В этот коммутатор с маршрутизатором (router), не учитывается при расчете В диаметра домена коллизий сети Fast Ethernet. Сеть удовлетворяет правилам первой модели.

Проверим теперь ее по второй модели. Наихудшие пути в домене коллизий: от DTE1 к DTE2 и от DTE1 к коммутатору (switching hub). Оба пути состоят из трех сегментов на витой паре, соединенных двумя повторителями класса II. Два сегмента имеют предельно допустимую длину 100 м. В Длина сегмента, соединяющего повторители, равна 5 м.

Предположим, что все три рассматриваемых сегмента являются сегментами 100BASE-TX и в них используется витая пара категории 5. В табл. Л.3 приведены величины времени двойного

оборота для рассматриваемыхВ путей (см. рис. Л.28). Сложив числа из второго столбца этой таблицы, получим 511,96 би — это и будет время двойного оборота для наихудшего пути.

Таблица Л.3

Время двойного оборота сети *Fast Ethernet*

Компонент пути	Время двойного оборота, би
Пара терминалов с интерфейсами 'ГХ	100
Сегмент на витой паре категории 5 (100 м)	111,2
Сегмент на витой паре категории 5 (100 м)	111,2
Сегмент на витой паре категории 5 (5 м)	5,56
Повторитель класса II	92
Повторитель класса II	92

Следует заметить, что в данном случае нет страхового запаса в 4 би, так как в этом примере используются наихудшие значения задержекВ (см. табл. Л.2). Реальные временные характеристики компонентов FastВ Ethernet могут отличаться в лучшую сторону.

Задание для выполнения

Требуется оценить работоспособность 100-мегабитной сети Fast Ethernet в соответствии с первой и второй моделями. Конфигурации сетиВ приведены в табл. Л.4. Топология сети представлена на рис. Л.29—Л.30.

Таблица Л.4

Варианты заданий

№	Сегмент 1	Сегмент 2	Сегмент 3	Сегмент 4	Сегмент 5	Сегмент 6
1	100BAS ETX, 100 м	100BAS ETX, 95 м	100BAS ETX, 80 м	100BAS ETX, 5 м	100BAS ETX, 100 м	100BAS ETX, 100 м

№	Сегмент 1	Сегмент 2	Сегмент 3	Сегмент 4	Сегмент 5	Сегмент 6
2	ЮОВА БЕ-ТХ, 15 м	ЮОВА БЕ-ТХ, 5 м	ЮОВА ЭЕ-ТХ, 5 м	100В АБЕ- ЕХ, 400 м	ЮОВА БЕ-ТХ, 10 м	ЮОВА БЕ-ТХ, 4 м
3	ЮОВА БЕ-ТХ, 60 м	ЮОВА БЕ-ТХ, 95 м	ЮОВАБ Е-ТХ, 10 м	ЮОВА БЕ-ТХ, 10 м	ЮОВА БЕ-ТХ, 90 м	ЮОВА БЕ-ТХ, 95 м

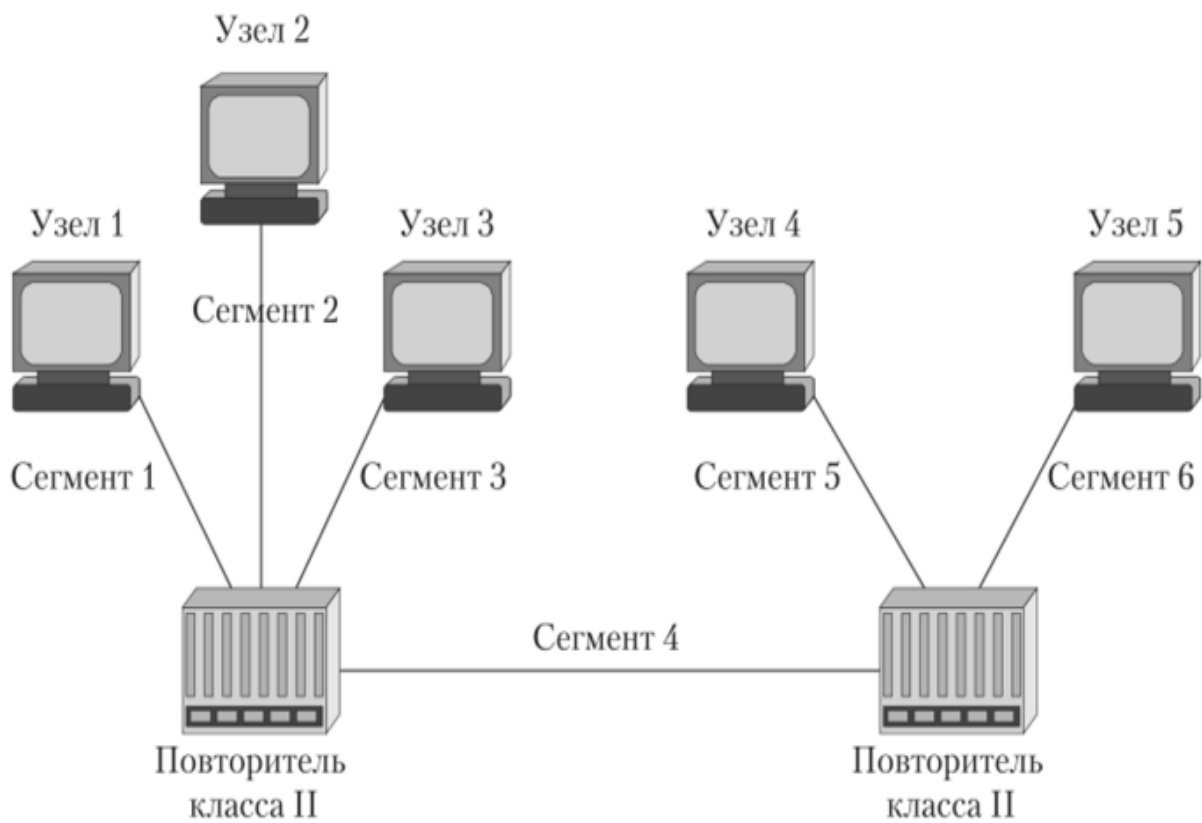


Рис. Л.29. Топология сети 1

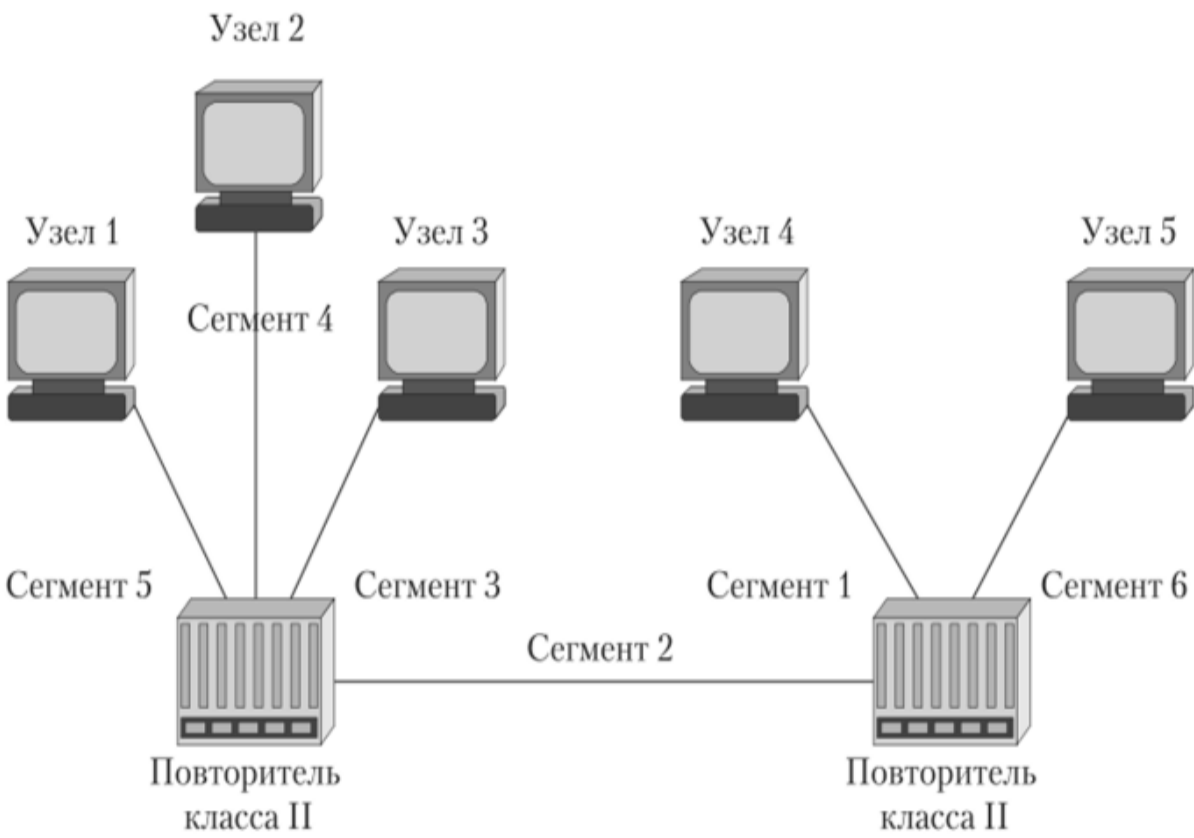


Рис. Л.30. Топология сети 2

Выводы:

Практическая работа №6

Тема: Аппаратное и программное обеспечение сетей ЭВМ. Установка и первичная настройка сетевого ПО.

Цель работы: Изучение состава аппаратного и программного обеспечения сетей ЭВМ. Получение практических навыков базовой настройки сетевой системы.

Задание 1: Охарактеризовать назначение, маркировку, функции и параметры следующего коммуникационного оборудования:

- Повторитель
- Концентратор
- Коммутатор
- Кабельная система «Витая пара»
- Оптоволоконный кабель
- Маршрутизатор
- Брандмауэр
- Сетевая плата
- Модем
- Мост

Повторитель - устройство для соединения сегментов одной сети, обеспечивающее усиление и формирования сигналов. Оперирует на физическом уровне модели OSI. Позволяет расширять сеть по расстоянию и количеству подключенных узлов.

Концентратор - это сетевое устройство, предназначенное для объединения нескольких устройств в общий сигнал.

Функции концентраторов:

1. Объединение сегментов с разными физическими средами в один локальный сегмент.
2. Автосегментация портов.
3. Совместно используют периферийное устройства.

Коммутатор – многопортовое устройство, обеспечивающее высокоскоростную коммутацию пакетов между портами.

Функции концентраторов:

1. Объединяет различные сетевые устройства такие, как компьютеры, серверы, подключенные к ним, в единый сегмент сети.
2. Анализ MAC- адреса порта-отправителя и отправка данных на другой порт, а так же этом формирование таблиц.

Кабельная система «Витая пара» используется в телефонных системах, локальных сетях, в передачи данных на дальние расстояния телефонных и телевизионных сигналов.

Существуют 2 вида «Витай пары»: экранированная витая пара и неэкранированная витая пара.

Неэкранированная витая пара широко используется в ЛВС, максимальная длина сегмента составляет 100 м. Неэкранированная витая пара состоит из двух изолированных медных провода. Разделяются на категории 1-5, 5е, 6, 6а и 7.

Экранированная витая пара имеет медную обмотку, обеспечивающая большую защиту, так же провода перемотаны фольгой. Экранированная витая пара обладает прекрасной изоляцией, защищающий данные от внешних помех. Кабели разделяются на типы (Type1-Type9).

Оптоволоконный кабель.

Информация передаются с помощью световых сигналов.

Каждое стеклянное оптоволокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон с отдельными коннекторами.

Маршрутизатор - устройство, предназначенное для построения компьютерной сети и обеспечения стабильной ее работы, транслирующее пакеты данных между различными элементами сети.

Функции маршрутизатора:

1. Подключение локальных сетей (LAN) к территориально-распределенным сетям (WAN).
2. Соединение нескольких локальных сетей.

Маршрутизаторы работают на третьем или седьмом уровне модели OSI.

Брандмауэр - средство, с помощью которого осуществляется процесс разграничения доступа к компьютеру через интернет. Различают два типа брандмауэров: программные и аппаратные.

Функции брандмауэров:

1. Обеспечивает безопасность компьютера.
2. Осуществляет взаимодействие с какими-либо сетевыми программами, которые установлены на компьютер.

Сетевая плата - специализированный компонент компьютера, обеспечивающий связь и передачу данных между несколькими компьютерами в сети.

Модем - коммуникационное устройство, позволяющее передавать бинарные (цифровые) данные по аналоговой телефонной линии.

Он осуществляет преобразование данных с компьютера в последовательность дискретных (разнотипных) сигналов и их отправку по аналоговой телефонной линии. На другом конце они расшифровываются принимающим модемом путем аналого-цифрового преобразования.

Мост - средство передачи кадров между двумя (или больше) сегментами. Мост анализирует заглавие кадра – его интересуют MAC-адрес источника и получателя. Мост прослушивает кадры, которые приходят, и составляет таблицы MACC-адресов узлов, подключенных к этим портам.

Задание 2: Охарактеризовать сетевые операционные системы согласно вариантам по следующей схеме:

- 1) платность,
- 2) доступ к исходному коду,
- 3) многоплатформенность,
- 4) мультизадачность,
- 5) количество пользователей,
- 6) функции управления сетью,
- 7) интерфейс работы,
- 8) потребляемые ресурсы

MS DOS.

- Бесплатная;
- Закрытый исходный код;
- Одноплатформенная;
- Однозадачная с элементами многозадачности
- Однопользовательская;
- Несетевая;

Window XP.

- Платная;
- Закрытый исходный код;
- Многоплатформенная;
- Мультизадачная;
- Многопользовательская;
- Общий доступ к подключению Интернет;
- Интерес работы: рабочий стол, меню, панели инструментов, программные окна, диалоговые окна, вторичные окна;
- Потребляемые ресурсы: тактовая частота ЦП – 233 МГц, объем ОП – 128 Мбайт, разрешение видеокарты 600х800, свободное место на диске для установки Windows XP – от 1536 Мбайт, столько же для программного обеспечения.

Windows 7.

- Платная;
- Закрытый исходный код;
- Многоплатформенная;
- Мультизадачная;
- Многопользовательская;
- Общий доступ к подключению Интернет;
- Интерес работы: рабочий стол, меню, панели инструментов, программные окна, диалоговые окна, вторичные окна;
- Потребляемые ресурсы: тактовая частота ЦП – 1ГГц и выше, 1Гб или 2Гб ОЗУ, 16 Гб или 20 Гб свободного места на жестком диске.

Windows 8.1.

- Платная;
- Закрытый исходный код;
- Многоплатформенная;
- Мультизадачная;
- Многопользовательская;
- Общий доступ к подключению Интернет;
- Интерес работы: введена «чудо-кнопка», возможность открыть рабочий стол сразу же после входа в систему;
- Потребляемые ресурсы: тактовая частота ЦП – 1ГГц и выше, 1Гб или 2Гб ОЗУ, 16 Гб или 20 Гб свободного места на жестком диске.

Ответы на контрольные вопрос:

- 1. Что такое компьютерная сеть?

Компьютерная сеть - совокупность узлов (компьютеров, терминалов, периферийных устройств), имеющих возможность информационного взаимодействия друг с другом с помощью специального коммуникационного оборудования и программного обеспечения.

- 2. Что входит в аппаратное обеспечение сетей?

- компьютеров;
- коммуникационного оборудования;
- операционных систем;
- сетевых приложений.

3. Функции и характеристики коммуникационного оборудования?

- средства линий передачи данных
- средства соединения линий передачи с сетевым оборудованием узлов
- средства увеличения дистанции передачи данных - репитеры, модемы и пр.
- средства повышения емкости линий передачи (мультиплексирования)
- средства управления информационными потоками в сети

4. Что такое активное оборудование сетей?

Активное оборудование сети – устройство, которому необходимо подача энергии для генерации сигналов.

К активному оборудованию относят интерфейсные карты, повторители, концентраторы.

5. Что такое пассивное оборудование сетей?

Пассивное оборудование сети – устройство, которому не требуется подача энергии.

К пассивному оборудованию относят кабели, соединительные разъемы, коммутационные панели.

6. Что такое вспомогательное оборудование сетей?

Вспомогательное оборудование - устройства бесперебойного питания, кондиционирования воздуха и аксессуары - монтажные стойки, шкафы, кабелепроводы различного вида.

7. Что называют операционной системой?

Сетевая ОС - ОС, которая обеспечивают пользователям распределенный доступ к сетям ЭВМ.

8. Что входит в группу прикладного программного обеспечения?

В группу прикладного ПО входят: сетевые операционные системы; сетевые драйвера, протоколы, службы и другое дополнительное программное обеспечение сетевых интерфейсов; прикладное сетевое программное обеспечение.

9. По каким критериям можно охарактеризовать сетевую операционную систему?

- 1) платность,
- 2) доступ к исходному коду,
- 3) многоплатформенность,
- 4) мультизадачность,
- 5) количество пользователей,
- 6) функции управления сетью,
- 7) интерфейс работы,
- 8) потребляемые ресурсы

10. Что такое технология «клиент-сервер»?

«Клиент – сервер» - топология, обеспечивающая совместный доступ пользователей к определенному типу ресурсов.

11. Что такое виртуальная машина? Ее назначение?

Виртуальная машина - программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы и исполняющая программы для target-платформы на host-платформе.

При помощи виртуальной машины можно установить любую систему на любую платформу. При этом получим полнофункциональную систему с доступом в локальную сеть и интернет.

Виртуальная машина имеет свой жесткий диск, процессор, выделенную оперативную память, графический адаптер и т.д. Всеми этими ресурсами делится с ней физическая машина.

Выводы:

Практическая работа №7

Тема работы: «Построение схемы компьютерной сети».

Цель работы: получение практических навыков построения локальных компьютерных сетей.

Краткие теоретические сведения

Локальная компьютерная сеть - это комплекс программного обеспечения и устройств, объединяющих абонентов, находящихся на незначительной дистанции друг от друга. Как правило, такие системы используются в границах одного предприятия или здания.

Типы локальных сетей

Данные линии принято разделять на 2 вида:

- Сети, для которых характерно централизованное управление, характеризующиеся общей политикой безопасности применимой ко всем пользователям
- Одноранговые сети – сети, в которых все пользователи самостоятельно определяют какую информацию и ресурсы они будут представлять в целях общего пользования, а компьютеры являются полностью равноправными и могут быть одновременно, как клиентом, так и сервером.

Основной задачей локальных вычислительных сетей является реализация совместного доступа всех пользователей к данным, устройствам и программам. Таким образом, клиентам системы доступно выполнять операции одновременно, а не поочередно. Помимо этого, локальные линии решают вопросы:

- Обработки и хранения данных;
- Передачи результатов информации пользователям;
- Контроля выполнения проектов.

Главные составляющие локальной сети

Локальная компьютерная сеть не может полноценно функционировать без специального оборудования. Для нее основными составляющими являются:

- Пассивное оборудование: коммутационные панели, монтажные шкафы, информационные розетки, кабели, кабельные каналы;

- Периферийные устройства и компьютеры: принтеры, серверы, рабочие станции, сканеры;
- Активное оборудование: маршрутизаторы, коммутаторы (свитчи), специальные медиаконвекторы.

В зависимости от того, как будет построена сеть, какой протяженностью и согласно каким требованиям, комплекс устройств при монтаже может существенно меняться.

Преимущества пользования локальной сетью

Такой тип системы решает множество вычислительных и информационных задач в пределах одного предприятия. Поэтому для организации компьютерная сеть локального типа является необходимой в силу нескольких ее преимуществ:

- Система обеспечивает хранение всех данных персонального характера на диске файлового сервера. Это дает возможность осуществлять одновременную работу всеми клиентами, обновлять данные в сетевых программных продуктах и при этом пользоваться информацией, защищенной на уровне файлов и каталогов.
- Локальная сеть способствует обмену информацией между всеми компьютерами, находящимися в системе.
- Каждый клиент имеет доступ к глобальной сети при условии наличия специального коммутационного узла.
- Такая вычислительная сеть обеспечивает полноценную печать информации всеми пользователями на общественных принтерах.
- Локальная система позволяет хранить программные продукты (графические редакторы, таблицы, системы управления базами данных) на дисках файлового сервера в единственном экземпляре.

Требования предъявляемые локальным вычислительным сетям

В настоящее время IT-компаниями создано большое количество локально-вычислительных сетей, которые различаются алгоритмами работы, структурой организации, топологиями, размерами. Они эксплуатируются в разных странах мира, но требования, предъявляемые к ним, являются общепринятыми.

- Надежность. Одно из главных свойств, нацеленное сохранить полное и частичное функционирование при поломке нескольких узлов.

- Скорость. Важнейшее свойство, характеризующееся наличием высокоскоростных каналов передачи данных.
- Адаптация. Свойство локально-вычислительной сети, направленное на расширение: рабочие станции устанавливаются в том месте, где это потребуется.

Локальная сеть – важный элемент любого современного предприятия, без которого невозможно добиться максимальной производительности труда. Однако чтобы использовать возможности сетей на полную мощность, [необходимо их правильно настроить](#), учитывая также и то, что расположение подсоединенных компьютеров будет влиять на производительность ЛВС.

Понятие топологии Топология локальных компьютерных сетей – это месторасположение рабочих станций и узлов относительно друг друга и варианты их соединения. Фактически это архитектура ЛВС. Размещение компьютеров определяет технические характеристики сети, и выбор любого вида топологии повлияет на:

- Разновидности и характеристики сетевого оборудования.
- Надежность и возможность масштабирования ЛВС.
- Способ управления локальной сетью.

Таких вариантов расположения рабочих узлов и способов их соединения много, и количество их увеличивается прямо пропорционально повышению числа подсоединенных компьютеров. Основные топологии локальных сетей – это "звезда", "шина" и "кольцо".

Факторы, которые следует учесть при выборе топологии

До того как окончательно определиться с выбором топологии, необходимо учесть несколько особенностей, влияющих на работоспособность сети. Опираясь на них, можно подобрать наиболее подходящую топологию, анализируя достоинства и недостатки каждой из них и соотнеся эти данные с имеющимися для монтажа условиями.

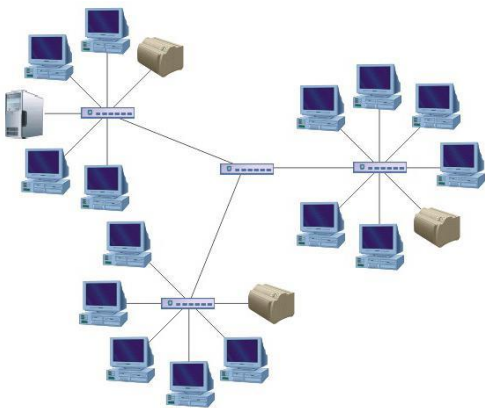
Работоспособность и исправность каждой из рабочих станций, подсоединенных к ЛВС. Некоторые виды топологии локальной сети целиком зависят от этого.

- Исправность оборудования (маршрутизаторов, адаптеров и т. д.). Поломка сетевого оборудования может как полностью нарушить работу ЛВС, так и остановить обмен информацией с одним компьютером.
- Надежность используемого кабеля. Повреждение его нарушает передачу и прием данных по всей ЛВС или же по одному ее сегменту.
- Ограничение длины кабеля. Этот фактор также важен при выборе топологии. [Если кабеля в](#)

наличии **немного**, можно выбрать такой способ расположения, при котором его потребуется меньше.

Топологии «звезда»

Этот вид расположения рабочих станций имеет выделенный центр – сервер, к которому подсоединены все остальные компьютеры. Именно через сервер происходят процессы обмена данными. Поэтому оборудование его должно быть более сложным.



Достоинства:

- Топология локальных сетей "звезда" выгодно отличается от других полным отсутствием конфликтов в ЛВС – это достигается за счет централизованного управления.

- Поломка одного из узлов или повреждение кабеля не окажет никакого влияния на сеть в целом.

- Наличие только двух абонентов, основного и периферийного, позволяет упростить сетевое оборудование.

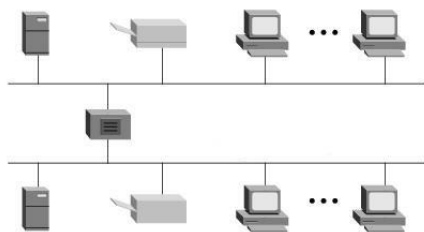
- Скопление точек подключения в небольшом радиусе упрощает процесс контроля сети, а также позволяет повысить ее безопасность путем ограничения доступа посторонних.

Недостатки:

- Такая локальная сеть в случае отказа центрального сервера полностью становится неработоспособной.

- Стоимость "звезды" выше, чем остальных топологий, поскольку кабеля требуется гораздо больше.

Топология «шина»



В этом способе соединения все рабочие станции подключены к единственной линии – коаксиальному кабелю, а данные от одного абонента отсылаются остальным в режиме полудуплексного обмена. Топологии локальных сетей подобного вида предполагают наличие на каждом конце шины специального терминатора, без которого сигнал искажается.

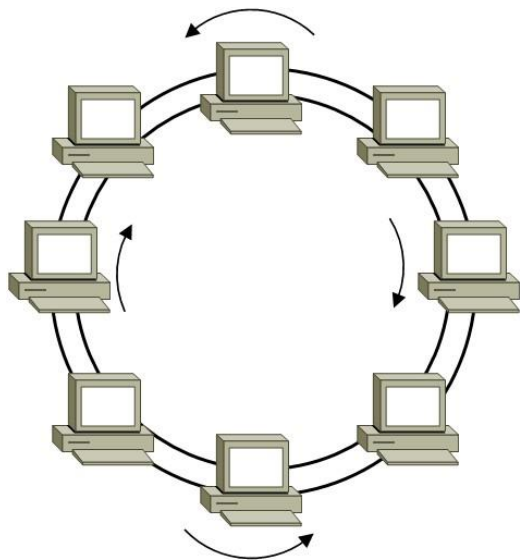
Достоинства:

- Все компьютеры равноправны.
- Возможность легкого масштабирования сети даже во время ее работы.
- Выход из строя одного узла не оказывает влияния на остальные.
- Расход кабеля существенно уменьшен.

Недостатки:

- Недостаточная надежность сети из-за проблем с разъемами кабеля.
- Маленькая производительность, обусловленная разделением канала между всеми абонентами.
- Сложность управления и обнаружения неисправностей за счет параллельно включенных адаптеров.
- Длина линии связи ограничена, потому эти виды топологии локальной сети применяют только для небольшого количества компьютеров.

Характеристики топологии «кольцо»



Такой вид связи предполагает соединение рабочего узла с двумя другими, от **одного из них принимаются данные**, а второму передаются. Главной же особенностью этой топологии является то, что каждый терминал выступает в роли ретранслятора, исключая возможность затухания сигнала в ЛВС.

Достоинства:

- - Быстрое создание и настройка этой топологии локальных сетей.
- - Легкое масштабирование, требующее, однако, прекращения работы сети на время установки нового узла.
- - Большое количество возможных абонентов.
- - Устойчивость к перегрузкам и отсутствие сетевых конфликтов.
- - Возможность увеличения сети до огромных размеров за счет ретрансляции сигнала между компьютерами.

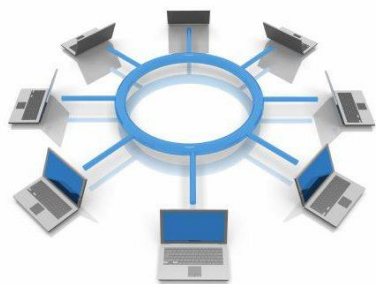
Недостатки:

- Ненадежность сети в целом.
- Отсутствие устойчивости к повреждениям кабеля, поэтому обычно предусматривается наличие параллельной резервной линии.
- Большой расход кабеля.

Типы локальных сетей

Выбор топологии локальных сетей также следует производить, основываясь на имеющемся типе ЛВС. Сеть может быть представлена двумя моделями: одноранговой и иерархической. Они не очень отличаются функционально, что позволяет при необходимости переходить от одной из них к другой. Однако несколько различий между ними все же есть. Что касается одноранговой модели, ее применение рекомендуется в ситуациях, когда возможность организации большой сети отсутствует, но создание какой-либо системы связи все же необходимо. Рекомендуется создавать ее только для небольшого числа компьютеров. Связь с централизованным управлением обычно применяется на различных предприятиях для контроля рабочих станций.

Одноранговая сеть



Этот тип ЛВС подразумевает равноправие каждой рабочей станции, распределяя данные между ними. Доступ к информации, хранящейся на узле, может быть разрешен либо запрещен его пользователем. Как правило, в таких случаях топология локальных компьютерных сетей «шина» будет наиболее подходящей. Одноранговая сеть подразумевает доступность ресурсов рабочей станции остальным пользователям. Это означает возможность редактирования документа одного компьютера при работе за другим, удаленной распечатки и

запуска приложений.

Достоинства однорангового типа ЛВС:

- Легкость реализации, монтажа и обслуживания.
- Небольшие финансовые затраты.

Такая модель исключает надобность в покупке дорогого сервера.

Недостатки:

- Быстродействие сети уменьшается пропорционально увеличению количества подсоединенных рабочих узлов.
- Отсутствует единая система безопасности.

- Доступность информации: при выключении компьютера данные, находящиеся в нем, станут недоступными для остальных.
- Нет единой информационной базы.

Иерархическая модель

Наиболее часто используемые топологии локальных сетей основаны именно на этом типе ЛВС. Его еще называют «клиент-сервер». Суть данной модели состоит в том, что при наличии некоторого количества абонентов имеется один главный элемент – сервер. Этот управляющий компьютер хранит все данные и занимается их обработкой.

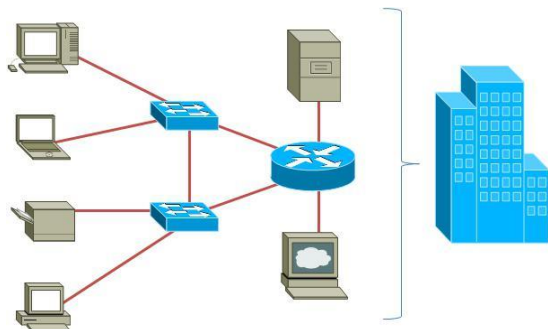
Достоинства:

- Отличное быстродействие сети.
- Единая надежная система безопасности.
- Одна, общая для всех, информационная база.
- Облегченное управление всей сетью и ее элементами.

Недостатки:

- Необходимость наличия специальной кадровой единицы – администратора, который занимается мониторингом и обслуживанием сервера.
- Большие финансовые затраты на покупку главного компьютера.

Наиболее часто используемая конфигурация (топология) локальной компьютерной сети в иерархической модели – это «звезда».



Выбор топологии (компоновка сетевого оборудования и рабочих станций) является исключительно важным моментом при организации локальной сети. Выбранный вид связи должен обеспечивать максимально эффективную и безопасную работу ЛВС. Немаловажно также уделить внимание финансовым затратам и возможности дальнейшего расширения сети. Найти рациональное решение

– непростая задача, которая выполняется благодаря тщательному анализу и ответственному подходу. Именно в таком случае правильно подобранные топологии локальных сетей обеспечат максимальную работоспособность всей ЛВС в целом.

Практическая часть

Задание 1

1.
- Описать одноранговую локальную сеть с топологией линейная шина.
2.
- Описать одноранговую локальную сеть с топологией звезда.
3.
- Описать локальную сеть с топологией кольцо.

Схема локальной сети

недостатки

преимущества

количество компьютеров

оборудование

выводы

Задание 2

- 1.Создать схему соединения компьютерной сети согласно своему заданию.
- 2.Описать построенную топологию.

Варианты заданий:

№	Сервер	ПК	Топология	
			Сервер	ПК
1	4	6	Общая шина	Кольцо

2	3	7	Звезда	Звезда
3	4	5	Звезда	Кольцо
4	6	5	Звезда	Общая шина
5	3	7	Кольцо	Звезда
6	6	3	Звезда	Кольцо

Контрольные вопросы

1. Что такое локальная компьютерная сеть?
2. Дайте определение топологии.
3. Какие топологии вы знаете? Дайте им краткое описание.
4. Какие основные составляющие ЛВС вы знаете?
- 5.

Выводы:

Практическое занятие № 8:

Тема: Расчет количества АРМ работников сортировочной (участковой, грузовой)

Цель:

1. Рассчитать количество работников сортировочной станции, необходимых для работы на автоматизированных рабочих местах (АРМ)
2. Ознакомиться с функциональными возможностями АРМ работников сортировочной станции.
3. Научиться определять потребное количество автоматизированных рабочих мест (АРМ) для сортировочных станции в зависимости от их типа.
4. Определить количество АРМ работников сортировочной станции согласно индивидуального задания.

Ход выполнения Практического занятия № 8:

1. Выписать из Таблицы №3 вариант задания согласно индивидуального номера задачи.
2. В зависимости от типа сортировочной станции и объёма её работы (Таблица №3) определить количество АРМ работников, связанных с прибытием и отправлением поездов.

Указания к выполнению задач: 1-40 (Таблица №3).

Для определения количества автоматизированных рабочих мест работников СТЦ на сортировочной станции применим формулу:

$$N_{ам}^{тк} = \frac{N_n \cdot n_{в} \cdot K_{зн} \cdot K_n \cdot K_{пв} \cdot (2 - \alpha)}{n_{см} \cdot T_{см} \cdot P_{оп}} \quad (\phi-1)$$

Где:

N_n - количество прибывающих поездов, (поездов / сутки);

$n_{в}$ - среднее число вагонов в составе поезда;

$K_{зн}$ - среднее число знаков, приходящихся на один вагон;

K_n - коэффициент, учитывающий неравномерность прибытия поездов;

$K_{пв}$ - коэффициент повторного ввода, учитывающий исправление ошибок, равный 1,25;

$n_{см}$ - число смен работы станции, принимается равным 2;

$T_{см}$ - продолжительность смены, равная 12 часам;

$P_{оп}$ - производительность оператора, (знаков/час);

α - доля информации, поступающей по межмашинному обмену.

Кроме того, необходимо предусмотреть по одному автоматизированному рабочему месту для каждой сортировочной горки

(АРМ дежурного по горке – $N_{ам}^Г$), каждого парка отправления (АРМ дежурного по станции – $N_{ам}^{ДСП}$) и один АРМ маневрового диспетчера - $N_{ам}^М$

Общее число автоматизированных мест на сортировочной станции составит:

$$N_{ам}^{об} = N_{ам}^{тк} + \alpha \cdot N_{ам}^Г + b \cdot N_{ам}^{ДСП} + N_{ам}^М \quad (\phi-2)$$

Таблица 3. Технические параметры сортировочных станций

номер вариан	Тип сортировоч	Число прибываю	Средн ее	Средне е число	Коэффициен т	Производитель ность	Доля информаци	Количество сортировоч	Количес тво
-----------------	-------------------	-------------------	-------------	-------------------	-----------------	------------------------	-------------------	--------------------------	----------------

та	ной станции щих поездов	число вагонов в составе	знаков на вагон строке натурного листа	неравномерности прибытия поездов	оператора	и, поступающей на станцию по межмашинному обмену	ных горок	парков отправления
	N_n	n_{ε}	K_{3H}	K_H	$n_{оп}$	α	a	b
	односторонняя			1,22		0,50		1
	двухсторонняя			1,28		0,55		
	односторонняя			1,27		0,51		
	двухсторонняя			1,25		0,56		
	односторонняя			1,26		0,52		
	двухсторонняя			1,24		0,57		
	односторонняя			1,20		0,53		
	двухсторонняя			1,23		0,58		
	односторонняя			1,29		0,54		
	двухсторонняя			1,21		0,59		
	односторонняя			1,22		0,50		
	двухсторонняя			1,28		0,55		
	односторонняя			1,27		0,51		
	двухсторонняя			1,25		0,56		
	односторонняя			1,26		0,52		
	двухсторонняя			1,24		0,57		
16	односторонняя			1,20		0,53		
	двухсторонняя			1,23		0,58		
	односторонняя			1,29		0,54		
	двухсторонняя			1,21		0,59		
	односторонняя			1,22		0,50		
	двухсторонняя			1,28		0,55		

односторон няя	1,27	0,51
двухсторон няя	1,25	0,56
односторон няя	1,26	0,52
двухсторон няя	1,24	0,57
односторон няя	1,20	0,53
двухсторон няя	1,23	0,58
односторон няя	1,29	0,54
двухсторон няя	1,21	0,59
односторон няя	1,22	0,50
двухсторон няя	1,28	0,55
односторон няя	1,27	0,51
двухсторон няя	1,25	0,56
односторон няя	1,26	0,52
двухсторон няя	1,24	0,57
односторон няя	1,20	0,53
двухсторон няя	1,23	0,58
односторон няя	1,29	0,54
двухсторон няя	1,21	0,59
приме односторо р нняя	1,18	0,45

Пример выполнения Практического занятия № 8:

Определить необходимое число АРМ для односторонней сортировочной станции, параметры которой заданы в строке «**пример**» Таблицы №3.

1. Определяем количество АРМ работников СТЦ по формуле (ф-1):

$$N_{\text{ам}}^{\text{тк}} = \frac{N_{\text{п}} \cdot n_{\text{в}} \cdot K_{\text{зн}} \cdot K_{\text{н}} \cdot K_{\text{пв}} \cdot (2 - \alpha)}{n_{\text{см}} \cdot T_{\text{см}} \cdot \Pi_{\text{оп}}} =$$
$$= \frac{95 \cdot 52 \cdot 30 \cdot 1,18 \cdot 1,25 \cdot (2 - 0,45)}{2 \cdot 12 \cdot 4800} = 2,94 \approx 3$$

2. Общее число АРМ на сортировочной станции будет определено по формуле (ф-2):

$$N_{\text{ам}}^{\text{об}} = N_{\text{ам}}^{\text{тк}} + a \cdot N_{\text{ам}}^{\text{г}} + b \cdot N_{\text{ам}}^{\text{дсп}} + N_{\text{ам}}^{\text{м}} =$$
$$= 3 + 1 \cdot 1 + 1 \cdot 1 + 1 = 6$$

Выводы:

Практическая работа № 9

“Создание простейшей базы данных в Microsoft Access 2010

Ввод и сортировка записей”.

Цель работы:

- научиться создавать таблицы с помощью Шаблонов таблиц и Конструктора таблиц;
- осуществлять правильный ввод данных в таблицы;
- научиться выполнять сортировку записей в таблице.

Задание 1. Создание пустой базы данных с помощью шаблонов таблиц.

Порядок работы

1. Запустите программу СУБД Microsoft Access. Для этого выполните: *Пуск – Все программы – Microsoft office – Microsoft office Access 2010*).
2. Перед Вами откроется окно следующего вида (*Рисунок 1*):

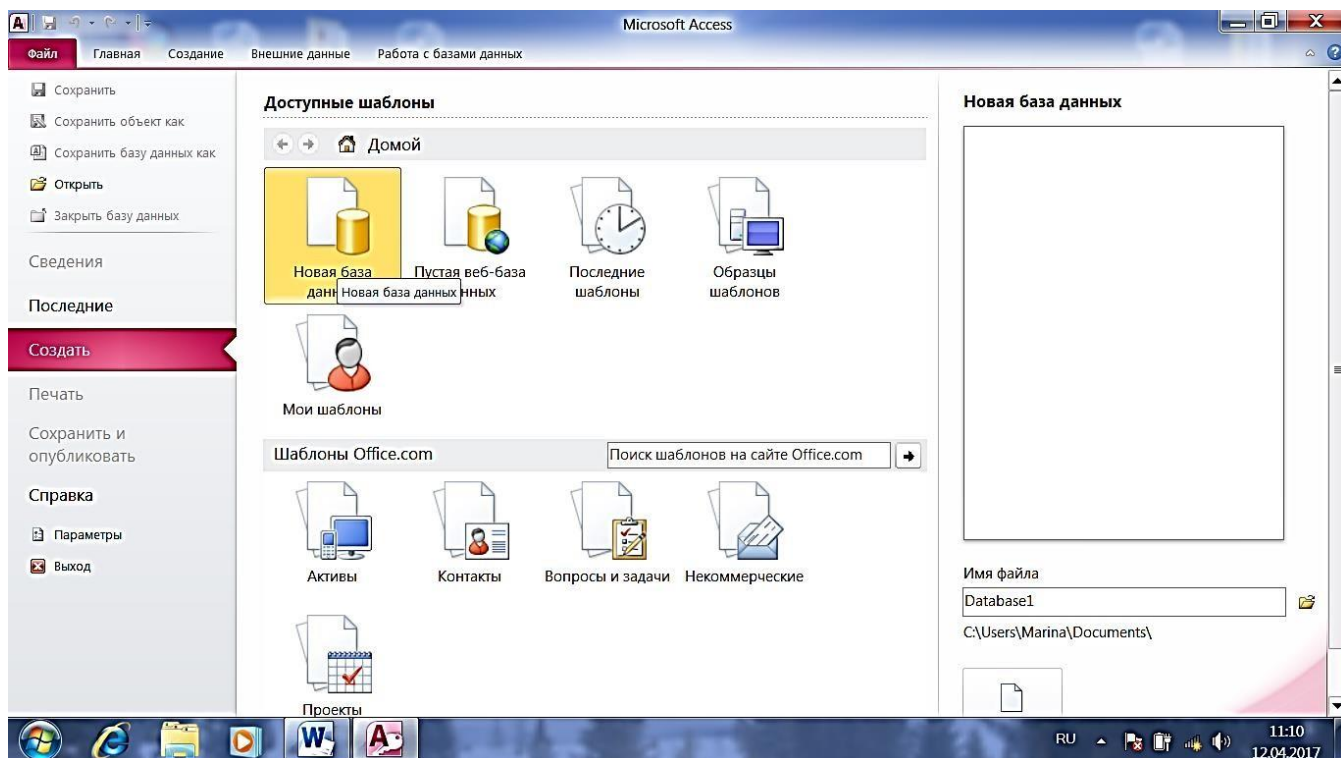


Рис. 1

3. Выберите значок *Новая база данных*. Затем введите имя файла – *База работников* и нажмите кнопку *Создать*. Перед Вами откроется окно следующего вида (см. Рисунок 2).

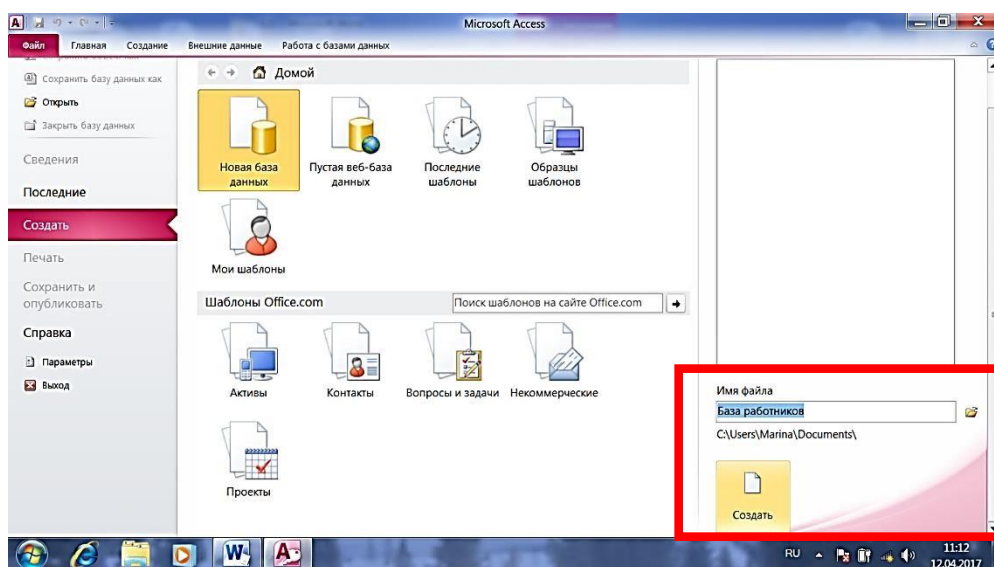
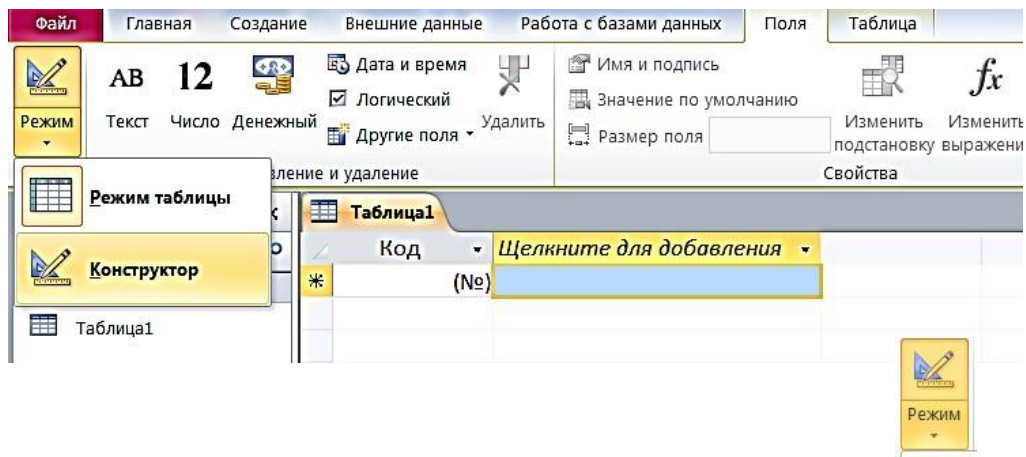
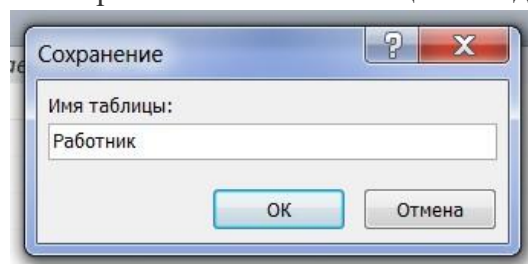


Рис. 2

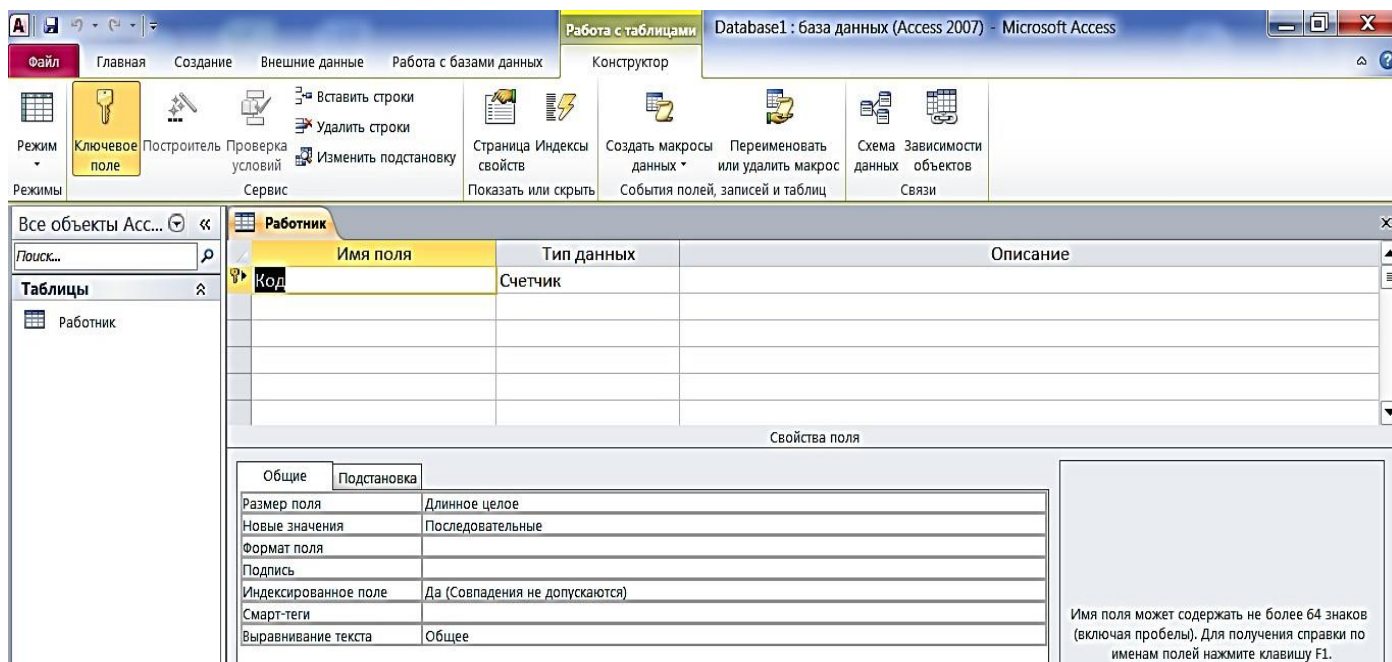
4. Откроется окно:



5. Щелкните ЛКМ по кнопке **Режим** на одноименном ярлычке
6. Откроется контекстное меню – выберите пункт **Конструктор**
7. Откроется окно с запросом об имени таблицы – введите имя **Работник**



8. Откроется окно

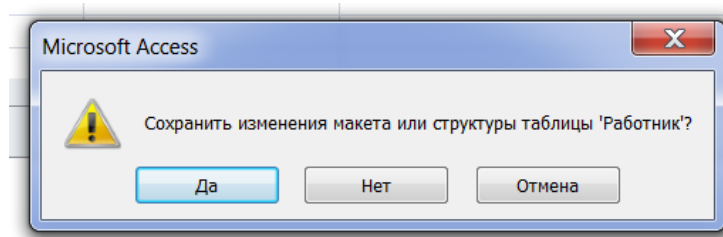


9. Создайте структуру БД, присвоив полям имена и определив тип данных:

Организация	текст	20
Фамилия	текст	15

Имя	текст	10
Адрес электронной почты	текст	15
Должность	текст	15
Рабочий телефон	число	6
Домашний телефон	число	6
Мобильный телефон	число	11
Номер факса	число	6
Адрес	текст	15
Город	текст	15
Республика	текст	15
Индекс	число	6
Страна	текст	10

10. Задав все поля, закройте окно – значок . На экране появится окно с запросом:



11. Щелкните ЛКМ по кнопке **Да**. Окно закроется и в области слева вы увидите имя вашей таблицы. Сделайте двойной щелчок ЛКМ по имени таблицы (таблица откроется) и заполните ее следующими данными

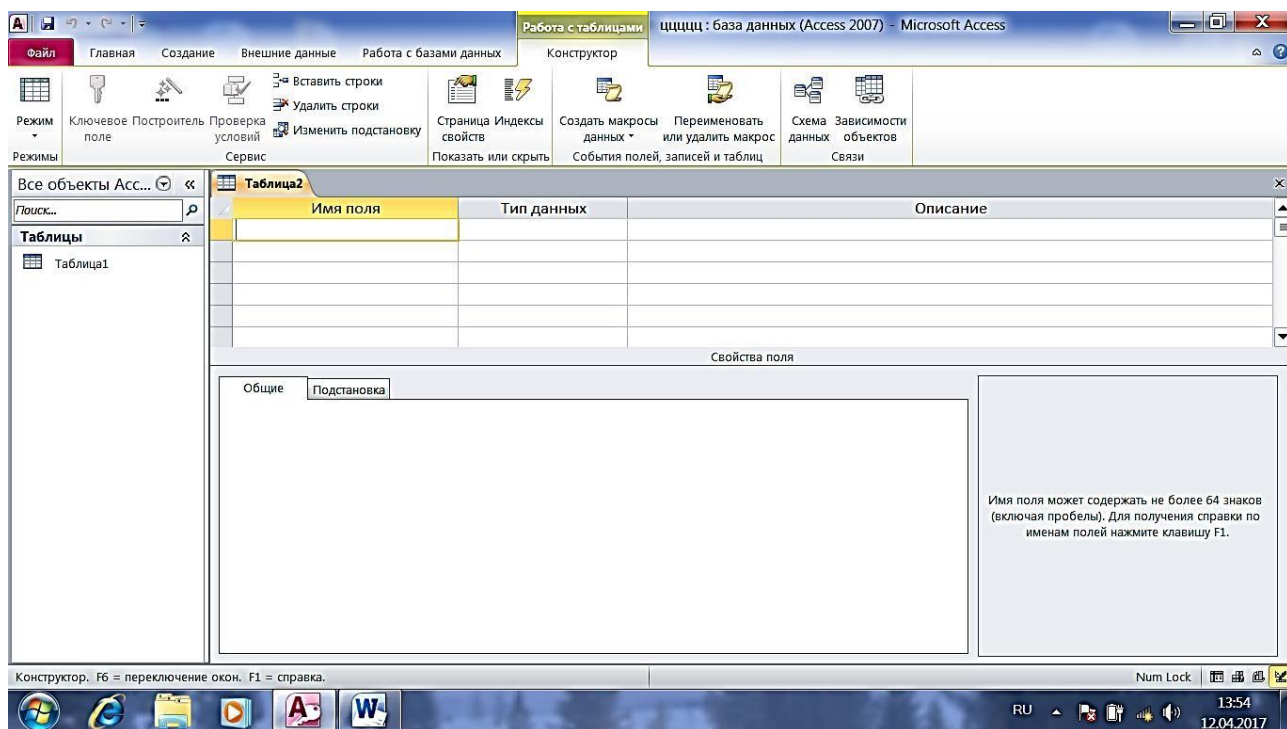
		Адрес										
Органи-	Код	Фамилия	Имя	электронной	Долж-	Рабочий	Домашний	Мобильный	Номер	Адрес	Город	Респу-
зация				почты	ность	телефон	телефон	телефон	факса			блика
												Индекс
1	Растр	Иванов	Сергей	ivanov@mail.ru	инженер	516987	265414	898294586232	264589	Гоголя,18	Лениногорск	Татарстан
2	Иволга	Сидоров	Дмитрий	sid@rambler.ru	электрик	264578	514589	890856423783	264578	Куйбышева,10	Лениногорск	Татарстан
3	Голден	Петров	Иван	Petr@rambler.ru	р	568989	214589	870554268972	564278	Ленина, 12	Лениногорск	Татарстан
4	Лайма	Никитина	Элина	nikita@mail.ru	бухгалтер	265578	214583	891745678891	265578	Шашина,30	Лениногорск	Татарстан
5	Рубин	Сергеева	Мария	serg@mail.ru	директор	568374	245889	893745698750	264582	Кирова, 58	Лениногорск	Татарстан

Таблица 1.

12. В данной таблице отсортируйте столбец “Организация” по алфавиту: выделите столбец →  → .

Задание 2. Создание пустой базы данных с помощью конструктора таблиц. Порядок работы

1. Создадим таблицу под именем “Студент” с помощью конструктора таблиц. Для этого выполните команду: *Создание – конструктор таблиц*.



Перед Вами откроется окно (Рисунок 3):

Рисунок 3

2. Заполните столбец *Имя поля* следующими данными (заголовками столбцов): *КодСтудент*, *Фамилия*, *Имя*, *Отчество*, *Адрес*, *Номер телефона*, *Специализация* соответственно типы данных для полей: *КодСтудент* – СЧЕТЧИК, *Фамилия*, *Имя*, *Отчество*, *Должность*, *Адрес*, *Специализация* – ТЕКСТОВЫЙ, *Номер телефона* – ЧИСЛОВОЙ.

Выводы:

Практическая работа № 10

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ АСЕМБЛЕР

Цели работы: ознакомиться с основами программирования на языке ассемблер. Научиться создавать, компилировать и компоновать программы на языке ассемблер с помощью пакета TASM.

Набор регистров

В программах на языке ассемблер регистры микропроцессора (МП), изображенные на рисунке 1.1, используются очень интенсивно.



Регистры
ния

Рисунок 1.1 — Основные регистры 32-разрядных процессоров

Регистры общего назначения (РОН) без каких-либо ограничений могут использоваться для хранения операндов логических и арифметических операций, компонентов адреса, указателей на ячейки памяти:

EAX/AX/AH/AL (Accumulator register) — *аккумулятор* — применяется для хранения промежуточных данных;

EBX/BX/BH/BL (Base register) — *базовый регистр* — применяется для хранения базового адреса некоторого объекта в памяти;

ECX/CX/CH/CL (Count register) — *регистр-счетчик* — применяется в командах, производящих некоторые повторяющиеся действия;

EDX/DX/DH/DL (Data register) — *регистр данных* — так же, как и регистр *EAX/AX/AH/AL*, он хранит промежуточные данные.

Следующие два регистра используются для поддержки цепочечных команд (подробно рассматриваются в лабораторной работе № 3):

ESI/SI (Source Index register) — *индекс источника* — содержит текущий адрес элемента в цепочке-источнике;

EDI/DI (Destination Index register) — индекс приемника — содержит текущий адрес элемента в цепочке-приемнике.

Следующие регистры предназначены для работы со стеком:

ESP/SP (Stack Pointer register) — регистр указателя стека — содержит указатель вершины стека в сегменте стека. Этот регистр не следует использовать явно для хранения каких-либо операндов программы;

EBP/BP (Base Pointer register) — регистр указателя базы кадра стека — предназначен для организации произвольного доступа к данным внутри стека.

Сегментные регистры предназначены для аппаратной поддержки структурной организации программы в виде отдельных частей, называемых *сегментами*. Сегмент представляет собой независимый блок памяти фиксированного размера. Операционная система размещает сегменты программы в оперативной памяти, после чего помещает их адреса в соответствующие сегментные регистры.

МП поддерживает следующие типы сегментов:

сегмент кода — содержит машинные команды, для доступа к нему служит регистр *CS (Code Segment register)* — *сегментный регистр кода*;

сегмент стека — для доступа к нему служит регистр *SS (Stack Segment register)* — *сегментный регистр стека*;

сегмент данных — содержит обрабатываемые программой данные, для доступа к нему служит регистр *DS (Data Segment register)* — *сегментный регистр данных*;

дополнительные сегменты данных — их адреса должны содержаться в регистрах *ES, GS, FS*.

Регистры состояния и управления содержат информацию о состоянии МП и программы. К этим регистрам относятся: *регистр флагов EFLAGS/FLAGS* и *регистр указатель команд EIP/IP*. Используя эти регистры, можно получать информацию о результатах выполнения команд и влиять на состояние МП. На рисунке 1.2 показано содержимое регистра *EFLAGS*, а в таблице 1.1 — назначение отдельных флагов.

←----- EFLAGS -----→																←----- FLAGS -----→															
31	...	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00								
0	...	ID	VIP	VIF	AC	VM	RF	0	NT	IOPL	OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF									

Рисунок 1.2 — Содержимое регистра EFLAGS

Указатель команд *EIP/IP* содержит адрес следующей выполняемой команды относительно содержимого регистра *CS*. Регистр *EIP/IP* непосредственно недоступен

программисту, но загрузка и изменение его значения производятся командами условных и безусловных переходов, вызова и возврата из процедур, а также вызова и возврата из прерываний.

Таблица 1.1 — Назначение флагов регистра EFLAGS

Флаг	Название	Назначение
1	2	3
CF	Флаг переноса	= 1 — арифметическая операция произвела перенос из старшего бита результата = 0 — переноса не было
PF	Флаг паритета	= 1 — 8 младших разрядов результата содержат четное число единиц = 0 — нечетное
AF	Дополнительный флаг переноса	Для команд, работающих с BCD-числами: = 1 — перенос из разряда 3 в 4-й при сложении или заем в разряд 3 из 4-го при вычитании = 0 — не было переносов или заемов
ZF	Флаг нуля	= 1 — результат нулевой = 0 — результат ненулевой
SF	Флаг знака	= 1 — старший бит результата равен 1 = 0 — старший бит результата равен 0
TF	Флаг трассировки	При пошаговой работе МП: = 1 — МП генерирует прерывание с номером 1 после выполнения каждой машинной команды = 0 — обычная работа
IF	Флаг прерывания	= 1 — аппаратные прерывания разрешены = 0 — аппаратные прерывания запрещены
DF	Флаг управления	Определяет направление обработки цепочек: = 1 — от конца к началу = 0 — от начала к концу
OF	Флаг переполнения	= 1 — произошел перенос в старший или заем из старшего знакового бита результата = 0 — не было переноса или заема
IOPL	Уровень привилегий ввода-вывода	Для контроля доступа к командам ввода-вывода в защищенном режиме работы МП
NT	Флаг вложенности задачи	Для фиксации в защищенном режиме работы МП того факта, что одна задача вложена в другую
RF	Флаг возобновления	Используется при обработке прерываний от регистров отладки
VM	Флаг виртуального 8086	= 1 — МП в режиме виртуального 8086 = 0 — МП в реальном или защищенном режиме
AC	Флаг контроля выравнивания	Разрешение контроля выравнивания при обращениях к памяти
VIF	Флаг виртуального прерывания	Появился в МП Pentium. В V-режиме является аналогом флага IF

1	2	3
VIP	Флаг отложенного виртуального прерывания	Появился в МП Pentium. = 1 — отложенное прерывание, используется при работе в V-режиме совместно с флагом VIF
ID	Флаг идентификации	Если программа может установить этот флаг, то МП поддерживает инструкцию CPUID

Типы данных

С точки зрения размерности, МП аппаратно поддерживает следующие основные типы данных, изображенные на рисунке 1.3:

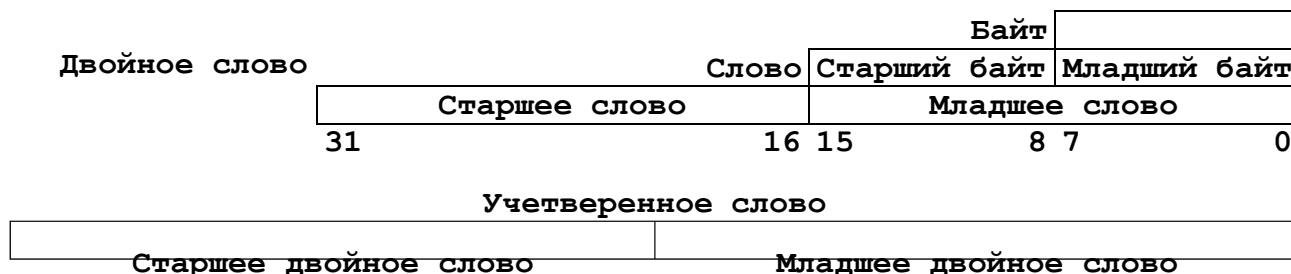
байт — восемь последовательно расположенных бит;

слово — два байта, имеющих последовательные адреса. Слово делится на *младший байт* и *старший*. Младший байт всегда хранится по меньшему адресу, который является *адресом слова*;

двойное слово — четыре байта, расположенных по последовательным адресам. Двойное слово состоит из *младшего слова* и *старшего*. Младшее слово хранится по меньшему адресу, который является *адресом двойного слова*;

учетверенное слово — восемь байт, расположенных по последовательным адресам. Учетверенное слово делится на *младшее двойное слово* и *старшее двойное слово*. Младшее двойное слово хранится по меньшему адресу, который является *адресом учетверенного слова*;

128-битный упакованный тип данных — появился в МП Pentium III. Для работы с ним были введены специальные команды.



63

32 31

0

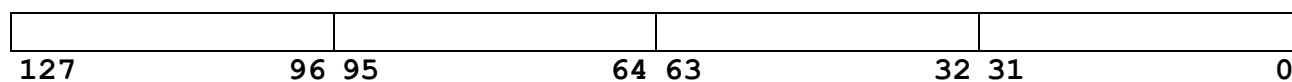
Упакованный тип данных

Рисунок 1.3 — Основные типы данных микропроцессора

Кроме трактовки типов данных с точки зрения их разрядности, МП поддерживает их логическую интерпретацию, изображенную на рисунке 1.4:

целый тип со знаком — двоичное значение со знаком размером 8/16/32 бита. Знак содержится соответственно в 7/15/31 бите (ноль — положительное число, единица — отрицательное). Отрицательные числа представляются в до-полнительном коде. Числовые диапазоны приведены в таблице 1.2;

целый тип без знака — двоичное значение со знаком размером 8/16/32 бита. Числовые диапазоны для этого типа данных приведены в таблице 1.2;

ближний указатель — 32-разрядный логический адрес, представляющий собой относительное смещение в байтах от начала сегмента;

дальний указатель — 48-разрядный логический адрес, состоящий из двух частей: 16-разрядной сегментной части — селектора и 32-разрядного смещения;

цепочка — некоторый непрерывный набор байт, слов или двойных слов длиной до 4 Гбайт (подробно рассматривается в лабораторной работе № 3);

битовое поле — непрерывная последовательность бит. Каждый бит является независимым и может рассматриваться как отдельная переменная;

неупакованный двоично-десятичный тип — байтовое представление десятичной цифры от 0 до 9. Числа хранятся как байтовые значения без знака по одной цифре в каждом байте (в младшей тетраде);

упакованный двоично-десятичный тип — представление двух десятичных цифр от 0 до 9. Каждая цифра хранится в своей тетраде (старшая — в старшей, младшая — в младшей);

типы данных с плавающей точкой — специальные типы данных для обработки чисел с плавающей точкой в математическом сопроцессоре (подробно рассматриваются в лабораторной работе № 5);

типы данных MMX-расширения (Pentium MMX/II) — представляют собой совокупность упакованных целочисленных элементов определенного размера (подробно рассматриваются в лабораторной работе № 6);

типы данных XMM-расширения (Pentium III) — представляют собой совокупность упакованных элементов с плавающей точкой фиксированного размера.

Таблица 1.2 — Диапазоны целых чисел

Размерность, бит	Без знака	Со знаком
8	0 — 255	–128 — +127
16	0 — 65 535	–32 768 — +32 767
32	0 — 4 294 967 295	–2 147 483 648 — +2 147 483 647

Целые без знака

Байт

Двойное слово

Слово

Целые со знаком

Байт Зн

Слово Зн

Двойное слово

Зн

31 15 7 0

Цепочка

...

До 4 Гбайт 15 7 0

Битовое поле

31 15 7 0

Указатель ближнего типа

СМЕЩЕНИЕ

31 0

Указатель дальнего типа

СЕЛЕКТОР СМЕЩЕНИЕ

47 31 0

Неупакованный двоично-десятичный тип

0000 BCD 0000 BCD 0000 BCD

15 7 0

Упакованный двоично-десятичный тип

BCD BCD BCD BCD BCD BCD BCD BCD

15 7 0

Типы данных сопроцессора

Типы данных MMX-расширения

Типы данных XMM-расширения

Рисунок 1.4 — Основные логические типы данных микропроцессора

Структура программы

Программа на языке ассемблера состоит из строк, содержащих следующие поля:

метка : команда/директива операнды ;комментарий

Все эти поля необязательны. Метка может быть любой комбинацией символов английского алфавита, цифр и символов «_», «\$», «@», «?». Цифра не может быть первым символом метки. Большие и маленькие буквы по умолчанию не различаются, но различие можно включить, задав соответствующую опцию в командной строке ассемблера. В поле команды может располагаться команда процессора, которая будет транслирована в исполнимый код, или ди-

ректива, которая не приводит к генерации кода, а управляет работой самого ассемблера. В поле операндов располагаются требуемые командой или директивой операнды. Текст от символа «;» не анализируется ассемблером и используется в качестве комментария.

Если метка располагается перед командой процессора, сразу после нее ставится символ «:», например:

```
some_loop:
    lodsw                ; считать слово из строки в ax
    mov     ax, 7        ; если это 7 - выйти из цикла
loopne
    some_loop
```

Если метка стоит перед директивой — двоеточие не ставится. Рассмотрим директивы, работающие напрямую с метками и их значениями.

Директива *label* определяет идентификатор *name* и задает его тип *type*:

```
name        label    type
```

Тип может быть одним из следующих: *byte* — байт, *word* — слово, *dword* — двойное слово, *fword* — 6 байт, *qword* — учетверенное слово, *tbyte* — 10 байт, *near* — ближняя метка, *far* — дальняя метка. Идентификатор получает значение, равное адресу следующей команды или следующих данных. С помощью директивы *label* удобно организовывать доступ к одним и тем же данным, как к байтам, так и к словам, определив перед данными две метки с разными типами.

Директива *equ* присваивает идентификатору *name* значение, которое определяется как результат выражения *expression*:

```
name        equ      expression
```

Это может быть целое число, адрес или любая строка символов, например:

```
truth       equ      1
message1    equ      'Try again$'
var2        equ      [si + 4]

    cmp      ax, truth ; cmp ax, 1
    db       message1 ; db 'Try again$'
    mov      ax, var2  ; mov ax, [si + 4]
```

Директива *equ* чаще всего используется для введения параметров, общих для всей программы.

С помощью директивы = идентификатор может принимать только целочисленные значения. Кроме того, идентификатор может быть переопределен позже, например:

```
buf_size = 1024
```

Директивы распределения памяти

Определение переменных в общем виде записывается следующим образом:

name *d** *value*

где *d** — одна из псевдокоманд: *db* — определить байт, *dw* — определить слово (2 байта), *dd* — определить двойное слово (4 байта), *df* — определить 6 байт (дальний указатель), *dq* — определить учетверенное слово (8 байт), *dt* — определить 10 байт (80-битные типы данных, используемые FPU).

Поле значения может содержать одно или несколько чисел, символов строк (взятых в одиночные или двойные кавычки), операторов *?* и операторов *dup*, разделенных запятыми, например:

```
text_string    db    'Hello world!', 0
number         dw    7
table          db    0,1,2,3,4,5,6,7,8,9
float_number   dd    3.5e7
```

Если вместо точного значения указан знак *?*, переменная считается неинициализированной, например:

```
var1           dq    ?
```

Если требуется заполнить участок памяти повторяющимися данными, используется специальный оператор *dup*, имеющий формат

count dup (value)

Например, определение

```
table_512wdw   512 dup(?)
```

создает массив из 512 неинициализированных слов.

Структуры объявляются с помощью директивы *struc*, которая позволяет определить структуру данных аналогично структурам в языках высоко-го уровня:

```
struct_name    struc
                fields
ends
```

где *fields* — любой набор псевдокоманд определения переменных или структур. В дальнейшем для создания объектов такой структуры в памяти используют имя структуры:

name struct_name <values>

Для чтения и записи в элемент структуры используется оператор «.», на-пример:

```
point struc                                ;определение структурных dw
    0                                     ;два слова со значениями

    y dw 0                               ;по умолчанию 0, 0
    color db 3 dup(?)                   ;и три байта для цвета
ends

cur_point point <1,1,1,255,255,255> ;инициализация
...

mov ax, cur_point.x ;обращение к слову x
```

Организация программы на ассемблере

Набор допустимых команд указывается в самом начале программы. По умолчанию ассемблер использует набор команд процессора 8086 и выдает сообщения об ошибках, если встречается команда, которую этот процессор не поддерживает. Для разрешения использования команд более новых процессоров предлагаются следующие директивы:

```
.186          ;команды 80186;
.286          ;непривилегированные команды 80286;
.386          ;непривилегированные команды 80386;
.486          ;непривилегированные команды 80486;
.586          ;непривилегированные команды P5 (Pentium);
.686          ;непривилегированные команды P6 (Pentium);
.8087         ;команды FPU 8087;
.287          ;команды FPU 80287;
.387          ;команды FPU 80387;
.487          ;команды FPU 80486;
.587          ;команды FPU 80586;
.MMX          ;команды IA MMX;
.K3D          ;команды AMD 3D;
```

Модель памяти в общем виде задается директивой

`.model model_name, language, modifier`

где *model_name* — одно из значений, перечисленных в таблице 2.1; *language* — необязательный операнд, принимающий значения *C*, *PASCAL*, *BASIC*, *FORTRAN*, *SYSCALL* и *STDCALL*. Если он указан, ассемблер считает, что все процедуры рассчитаны на вызов из программ на соответствующем языке высо-

кого уровня. *Modifier* — необязательный операнд, принимающий значения *NEARSTACK* (по умолчанию) или *FARSTACK*.

Таблица 2.1 — Модели памяти

Модель	Описание
TINY	Код, данные и стек размещаются в одном и том же сегменте размером 64Кбайт
SMALL	Код — в одном сегменте, а данные и стек — в другом
COMPACT	Код — в одном сегменте, а данные могут располагаться в нескольких сегментах. Используются дальние указатели
MEDIUM	Код — в нескольких сегментах, а все данные — в одном. Для доступа к данным используется только смещение, а для вызова подпрограмм — команды дальнего вызова процедур
LARGE, HUGE	И код, и данные могут занимать несколько сегментов
FLAT	То же, что и <i>TINY</i> , но используются 32-разрядные адреса. Максимальный размер сегмента — 4 Мбайт

Директивы сегментации определяют сегменты программы. Как отмечалось ранее, каждая программа состоит из сегмента кода, сегмента данных и сегмента стека.

Сегменты описываются директивами *segment* и *ends*:

```
segment_name      segment      attributes
...
ends
```

здесь *attributes* — атрибуты сегмента. На практике чаще всего используют упрощенные директивы определения сегментов:

```
.stack size      ;сегмент стека в size байт
                  ; (по умолчанию 1024)
.data            ;обычный сегмент данных
.const          ;сегмент неизменяемых данных
.code           ;основной сегмент кода
```

Глобальные объявления необходимы для определения идентификаторов, общих для разных модулей. Директива

```
public language   name
```

делает идентификатор *name* доступным из других модулей программы. Директива

```
extrn language    name:type
```

описывает идентификатор *name*, определенный в другом модуле (с помощью *public*). Тип *type* должен соответствовать типу идентификатора в том модуле, в котором он был определен. Директива

```
global language      name:type
```

действует, как *public* и *extrn* одновременно. Если идентификатор *name* находится в том же модуле, он становится доступным для других модулей (действует как *public*). Если идентификатор не описан, то он считается внешним и выполняется действие, аналогичное действию директивы *extrn*.

Выражения

Выражение — это набор идентификаторов, чисел или строк, связанных друг с другом операторами. Все выражения вычисляются в ходе ассемблирования программы, так что в полученном коде используются только значения.

В выражениях используются операторы следующих типов: арифметические операторы: + (плюс), – (минус), * (умножение), / (целочисленное деление), *mod* (остаток от деления), например:

```
mov al, 90 mod 7      ; mov al, 6
```

логические операторы: *and* (И), *not* (НЕ), *or* (ИЛИ), *xor* (исключающее ИЛИ), *shl* (сдвиг влево), *shr* (сдвиг вправо), например:

```
mov ax, 1234h and 4321h ; mov ax, 0220h
```

операторы сравнения: *eq* (равно), *ge* (больше или равно), *gt* (больше), *le* (меньше или равно), *lt* (меньше), *ne* (не равно). Результат действия каждого из этих операторов — единица, если условие выполняется, и ноль — если не выполняется;

операторы адресации: *seg expr* — сегментный адрес *expr*; *offset expr* — смещение *expr*; *type ptr expr* — переопределение типа *expr* в соответствии с типом *type*; *sizeof var/type* — число байт, занимаемых переменной *var* или типом *type*, например:

```
mov dx, offset msg      ; занести в dx
                        ; смещение переменной msg
mov ax, sizeof word      ; занести в ax размер типа word
dword ptr [si], 0        ; записать 4 байта нулей
                        ; по адресу ds:si
```

другие операторы: *()* (круглые скобки) — задают приоритет вычислений,

[] (квадратные скобки) — косвенная адресация.

Другие директивы

Директива

```
include      file_name
```

вставляет в текст программы текст файла аналогично команде препроцессора языка C *#include*. Обычно используется для включения файлов, содержащих определения констант, структур и макросов. Директива

```
#include lib  file_name
```

указывает компоновщику имя дополнительной библиотеки или объектного файла, который потребуется для компоновки данной программы.

Формат команд и режимы адресации

В машинную команду МП явно или неявно входят следующие элементы:

поле префиксов — элемент команды, который уточняет либо модифицирует действие этой команды в следующих аспектах: замена сегмента, изменение размерности адреса, изменение размерности операнда, указание на необходимость повторения данной команды;

поле кода операции, определяющее действие данной команды;

поле операндов; содержит от 0 до 3 элементов.

Важной особенностью машинных команд является то, что они не могут манипулировать одновременно двумя операндами, находящимися в оперативной памяти. По этой причине возможны только следующие сочетания операндов в команде:

регистр — регистр;

регистр — память;

память — регистр;

регистр — непосредственный операнд;

память — непосредственный операнд.

Из перечисленных сочетаний операндов наиболее часто употребляются *регистр — память* и *память — регистр*. Ввиду их важности рассмотрим их подробнее, и в частности способы адресации операндов в памяти.

Прямая адресация — простейший вид адресации операнда в памяти. Прямая адресация может быть двух типов:

относительная — используется в командах условного и безусловного перехода, например:

```
jz mm1      ;переход на метку mm1, если флаг zf = 1
mov al, 32
```

...

mm1:

абсолютная — использует несколько форм такой адресации, например:

```
mov ax, dword ptr [0000]; записать слово по адресу  
; ds:0000 в регистр ax
```

Но такая адресация применяется редко, обычно используемым ячейкам в программе присваиваются символические имена, например:

```
.data  
var1          dw 20h  
...  
add var1, ax   ; сложить содержимое переменной var1 и  
; регистра ax, результат поместить в  
; переменную var1 (ее адрес ds:0000)
```

Косвенная базовая (регистровая) адресация — адрес операнда находится в любом из регистров общего назначения, кроме *SP/ESP* и *BP/EBP*. Синтаксически в команде этот режим адресации выражается заключением имени регистра в квадратные скобки []. Например, команда

```
mov bx, [eax]
```

помещает в регистр *BX* содержимое слова по адресу из сегмента данных со смещением, хранящимся в регистре *EAX*.

Этот режим очень удобен для организации циклических вычислений и для работы с различными структурами данных типа таблиц и массивов.

Косвенная базовая (регистровая) адресация со смещением является дополнением предыдущей и предназначена для доступа к данным с известным смещением относительно некоторого базового адреса. Например, команда

```
mov bx, [eax + 10h]
```

пересылает в регистр *BX* слово из области памяти по адресу: содержимое *EAX* + 10h. Следующий пример иллюстрирует инициализацию массива последовательными значениями:

```
mas          db 5 dup(0)          ; массив из 5 слов  
...  
mov edi, 5mm1:  
    dec eax          ; eax = eax - 1  
    mov mas[eax], al ; mas[eax] = al
```

```
jnz mm1      ;повторяем, пока eax не станет равным 0
```

Косвенная индексная адресация со смещением очень похожа на косвенную базовую адресацию со смещением, но обладает той особенностью, что позволяет *масштабировать* содержимое индексного регистра. Например, в команде

```
mov ax, mas[esi*2]
```

значение эффективного адреса второго операнда вычисляется выражением $MAS + (ESI) * 2$. Возможность масштабирования существенно помогает для организации индексации массивов элементов размером 1, 2, 4 или 8 байт.

Косвенная базовая индексная адресация похожа на косвенную базовую адресацию со смещением. Эффективный адрес формируется как сумма содержимого двух любых регистров общего назначения: базового и индексного. Например:

```
mov eax, [esi][edx] ; эффективный адрес = (esi) + (edx)
```

Косвенная базовая индексная адресация со смещением является дополнением косвенной базовой индексной адресации. Эффективный адрес формируется как сумма трех составляющих: содержимого базового регистра, содержимого индексного регистра и значения смещения. Например, команда

```
mov eax, [esi + 5][edx]
```

пересылает в регистр *eax* двойное слово по адресу: $(ESI) + 5 + (EDX)$, а команда

```
add ax, array[esi][ebx]
```

производит сложение содержимого регистра *AX* с содержимым слова по адресу:

$ARRAY + (ESI) + (EBX)$.

3 Создание первой программы

Традиционно первая программа для освоения нового языка программирования — программа, выводящая на экран текст «*Hello world!*».

Наберите в любом текстовом редакторе следующий текст:

```
.model small
.stack
.data
Message      db 'Hello world!', 13, 10, '$'
.code begin:
```

```
mov ax, @data ;@data — адрес сегмента данных
mov ds, ax
;ds указывает на сегмент данных
```

```

mov  ah, 9           ;функция печати строки DOS
mov  dx, offset Message ;dx указывает на Message int  21h
                           ;вывод на дисплей «Hello world!» mov
ah, 4ch             ;функция завершения программы в DOSint
21h                 ;завершить программу

end begin

```

Сохраните файл под именем *hello_1.asm*.

Ассемблирование программы

Прежде чем получить исполнимый файл, сначала нужно вызвать ассемблер, для того чтобы скомпилировать программу в объектный файл с именем *hello_1.obj*, набрав в командной строке следующую команду:

```
tasm hello_1.asm
```

С ассемблерными программами также можно работать из интегрированных сред разработки (Turbo C, Borland C++ 3.1), но создание полноценных программ на ассемблере требует некоторой их перенастройки.

Компоновка программы

После успешного ассемблирования следующим шагом является компоновка. Для компоновки используется компоновщик *tlink*. В командной строке необходимо ввести

```
tlink hello_1.obj
```

В результате при отсутствии ошибок будет получен исполнимый файл с именем *hello_1.exe*. Если его выполнить, на экране появится строка «*Hello world!*» и программа завершится.

Модификация первой программы

Изменим программу таким образом, чтобы она могла принять из внешнего мира байт входной информации (в данном случае с клавиатуры):

```
.model small
```

```
.stack
```

```
.data
```

```
TimePrmpt          db    'Уже день? (д/н) $'
```

```

GoodMorningMsg      db 'Доброе утро!', 13, 10, '$'
GoodAfternoonMsg    db 'Добрый день!', 13, 10, '$'

```

.code start:

```

mov  ax, @data      ;в ax адрес сегмента данных
mov  ds, ax         ;ds указывает на сегмент данных
mov  dx, offset TimePrmpt ;dx указывает на подсказку
mov  ah, 9          ;функция вывода строки DOS

int  21h            ;вывод на дисплей подсказки
mov  ah, 1          ;функция приема символа DOS
int  21h            ;прием одного символа

cmp  al, 'д'        ;введен символ 'д'?
jz   is_afternoon   ;да, время за полдень
cmp  al, 'Д'        ;введен символ 'Д'
jnz  is_morning     ;нет, время до полудня
is_afternoon:
mov  dx, offset GoodAfternoonMsg ;дневное приветствие
jmp  display_greeting

is_morning:
mov  dx, offset GoodMorningMsg ;утреннее приветствие
display_greeting:
mov  ah, 9          ;функция вывода строки DOS
int  21h            ;вывод на дисплей приветствия
mov  ah, 4ch        ;функция завершения программы в DOS
int  21h            ;конец программы, выход в DOS

end start

```

4

Порядок выполнения работы

Для выполнения данной работы необходимо следующее:

- а) изучить программную модель 32-разрядного микропроцессора, набор регистров, типы данных;
- б) изучить принципы построения программ на языке ассемблер, директивы, определение переменных, режимы адресации;
- в) набрать в любом текстовом редакторе пример программы, приведенной в разд. 3. Сохранить в файле с расширением *.asm*;
- г) полученный файл откомпилировать с помощью утилиты *tasm.exe*, а с помощью компоновщика *link.exe* получить исполнимый файл, как указано в подразд. 3.1 и 3.2;
- д) то же самое проделать с примером программы из подразд. 3.3;е) получить индивидуальное задание у преподавателя.

Работа оценивается по результатам выполнения индивидуального задания, а также по знанию теоретической части лабораторной

