**Национальный государственный университет**

**физической культуры спорта и здоровья имени П.Ф. Лесгафта**

**г.Санкт-Петербург**

Л.А. Заварухина

ИНФОРМАТИКА

(лекции)

**Санкт-Петербург**

**2009**

Содержание

|  |  |  |  |
| --- | --- | --- | --- |
|  | [Введение](#предисловие) |  | 3 |
| [Лекция 1](#введение) | Информатика – предмет, задачи, история |  | 5 |
| [Лекция 2](#история) | История развития вычислительной техники. Организация,  хранение и передача данных в компьютере. Файловая структура, организации информации в компьютере |  | 22 |
| [Лекция 3](#аппаратное) | Аппаратное обеспечение компьютера |  | 32 |
| [Лекция 4](#программное) | Программное обеспечение компьютера |  | 48 |
| [Лекция 5](#Сети) | Компьютерные сети |  | 63 |
| [Лекция 6](#безопасность) | Безопасность информации |  | 79 |
|  | [Вопросы](#вопросы) для закрепления материала по лекционному курсу. |  | 85 |

***Лекция 5***

**Программное обеспечение компьютера**

**План лекции:**

1. Уровни программного обеспечения
2. Базовое программное обеспечение
3. Системное программное обеспечение
4. Служебное программное обеспечение
5. Прикладное программное обеспечение

**Новые термины и понятия:** *программа, программное обеспечение, базовое программное обеспечение, системное программное обеспечение, служебное программное обеспечение, прикладное программное обеспечение, операционная система, файловый менеджер.*

**1.** **Уровни программного обеспечения**

Возможности компьютера как технической основы системы обработки данных связаны с используемым программным обеспечением.

***Программа*** (program, routine) – упорядоченная последовательность команд (инструкций) компьютеру для решения задачи. Конечная цель любой компьютерной программы – управление аппаратными средствами.

Пример программы написанной на языке FOXPRO 8.0

CLEAR

STORE 1 TO i

STORE 0 TO rez

STORE 0 TO x

@ 5,20 SAY "Программа для вывода таблицы умножения до 10 на заданное число"

@ 7,30 say "Введите число для таблицы умножения" get x picture "9"

read

IF x <=0 OR x>10

@ 9,20 say "Можно вводить числа только от 1 до 9"

@ 7,30 say "Введите число для таблицы умножения" get x

read

endif

DO WHILE i<=10

rez= x\*i

@ 11+i,30 say x picture "9"

@ 11+i,32 say " \* "

@ 11+i,36 say i picture "99"

@ 11+i,39 say " = "

@ 11+i,43 say rez picture "99"

i=i+1

ENDDO

Блок схема программы

Заголовок

Ввод

данных

Проверка правильности ввода данных

Нет

Да

Выполнение расчета и вывод ответа

Проверка условия окончания

Нет

Да

Окончание

программы

**Основные элементы программирования:**

1. Ввод данных
2. Вычисления, действия
3. Ветвление
4. Условный и безусловный переход
5. Цикл
6. Вывод результатов
7. Массивы
8. Подпрограммы

И т.д.

***Программное обеспечение*** – совокупность программ обработки данных.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и непрерывном взаимодействии. Несмотря на то, что программное и аппаратное обеспечение рассматриваются раздельно, нельзя забывать, что между ними существует диалектическая связь, и раздельное рассмотрение их является условным.

Существует несколько уровней программного обеспечения. Эти уровни взаимодействуют между собой. Они представляют пирамидальную конструкцию. Каждый последующий уровень опирается на программное обеспечение предшествующих уровней, при этом каждый вышележащий уровень повышает функциональность всей системы. Так, например, вычислительная система с программным обеспечением базового уровня не способна выполнять большинство функций, но позволяет установить системное программное обеспечение.

**Различают четыре уровня программного обеспечения:**

* базовый уровень;
* системный уровень;
* служебный уровень
* прикладной уровень.

Следует отметить, что труд Н.А.Бернштейна (1947) "О построении движений" выделяет пять уровней, на которых "строятся" движения. Более низкие, базовые уровни отвечают за простейшие движения, более высокие – за сложные движения, однако, функционирование более высоких уровней невозможно без низких уровней.

**2. Базовое программное обеспечение**

Самый низкий уровень программного обеспечения представляет ***базовое программное обеспечение***. Оно отвечает за взаимодействие с базовыми аппаратными средствами. Как правило, программные средства этого уровня входят непосредственно в состав базового оборудования и хранятся в специальных микросхемах, называемых *постоянными запоминающими устройствами* (ПЗУ – Read Only Memory, ROM). Программы и данные записываются ("прошиваются") в микросхемы ПЗУ на этапе производства и не могут быть изменены в процессе эксплуатации. Комплект программ находящихся в ПЗУ образует ***базовую систему ввода-вывода*** (***BIOS*** – Basic Input Output System). Основное назначение BIOS на этапе загрузки компьютера – проверить работоспособность системы.

В тех случаях, когда изменение базовых программных средств во время эксплуатации является технически целесообразным, вместо микросхем ПЗУ применяют перепрограммируемые постоянные запоминающие устройства (ППЗУ). В этом случае содержание ППЗУ можно изменять.

**Функции и назначения базовой системы ввода-вывода**

BIOS самый близкий к аппаратуре компонент.

Основная функция BIOS заключается в *управлении стандартными внешними и внутренними устройствами:*

* монитором
* клавиатурой
* дисководами
* принтером
* таймером

Вспомогательные функции реализуются при включении ПК на этапе "загрузки".

* *тестирование* аппаратного обеспечения, в том числе оперативной памяти. В случае обнаружения неисправности выводится индикация
* *инициализация векторов прерывания* нижнего уровня (ранжирование устройств ПК по степени значимости, «важности»)
* *поиск* сначала на гибком, затем на жестком диске *программы*-*загрузки ОС* и загрузка ОС в оперативную память

**3. Системное программное обеспечение**

***Системное программное обеспечение*** (system software) – минимальный набор программных средств, обеспечивающих работу компьютера. Системный уровень – переходный. Программы, работающие на этом уровне, обеспечивают взаимодействие прочих программ компьютерной системы с программами базового уровня и непосредственно с аппаратным обеспечением, то есть выполняют "посреднические" функции.

К системному программному обеспечению относятся:

* операционная система
* операционные оболочки
* файловые менеджеры

***Операционная система*** (operating system) – набор программ, обеспечивающих работоспособность компьютерной системы, управление аппаратурой и прикладными программами, интерфейс с пользователем.

На IBM-совместимых компьютерах устанавливаются следующие операционные системы:

* MS-DOS (Microsoft Disk Operating System, фирма Microsoft) (в настоящее время устарела)
* Windows 95, Windows 98, Windows 2000; Windows XP (фирма Microsoft)
* UNIX (фирма Bell Laboratories)
* Linux (разработчик Линус Торвальдс)

На компьютерах Macintosh чаще всего устанавливается Mac OS 10.

***Сетевая операционная система*** – комплекс программ, обеспечивающий обработку, передачу и хранение данных в сети. Сетевая ОС предоставляет пользователю различные виды сетевых служб (управление файлами, электронная почта, процессы управления сетью и др.), поддерживает работу в абонентских системах.

В настоящее время наибольшее распространение имеют:

* Novel NetWare (фирма Novell)
* Windows NT, Windows Server (фирма Microsoft)
* LAN Server (Local Area Network)

***Операционные-оболочки и файловые менеджеры.*** Во времена господства на IBM-совместимых компьютерах операционной системы MS-DOS пользователю было очень трудно с ней взаимодействовать. Это было связано с тем, что диалог с операционной системой проводился в виде ввода в компьютер команд с клавиатуры. Пользователь должен был помнить формат (правила записи) каждой команды, что вызывало определенные затруднения. Пользовательский интерфейс также оставлял желать лучшего. В связи с необходимостью облегчить пользователю взаимодействие с операционной системой появились ***операционные оболочки*** – специальные программы, предназначенные для облегчения общения пользователя с командами операционной системы. Самой популярной программой-оболочкой стала программа **Norton Commander** (фирма Symantec). Кроме того, в нашей стране использовались также другие операционные оболочки: **Windows 3.1 – Windows 3.11** (фирма Microsoft); **DOS Navigator** (Rit Research Labs); **FAR** (Евгений Рошаль).

С появлением операционных систем, имеющих графический интерфейс, например, **Windows 95,** потребность в таких программах-оболочках отпала, однако те удобства, которые были предоставлены пользователям при базовых приемах работы с файлами (копирование, перемещение, переименование) были настолько велики, что появились программы под названием файловые менеджеры. В нашей стране наиболее распространена программа Windows Commander (рис.3.1).

**4****.Служебное программное обеспечение**

Расширением системного программного обеспечения компьютера является набор сервисных, дополнительно устанавливаемых программ, которые можно классифицировать по функциональному признаку следующим образом:

* программы-утилиты
* антивирусные программы
* программы-архиваторы
* языки программирования

***Программы-утилиты.*** Основное назначение служебных программ этого класса состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы. Во многих случаях они используются для расширения или улучшения функций системных программ. Некоторые служебные программы изначально включаются в состав операционной системы, но большинство служебных программ являются для операционной системы внешними и служат для расширения ее функций.

В настоящее время широко распространены следующие программы-утилиты: **Norton Utilities**, **SiSoft Sandra** (рис. 3.2), **Fix-It,** дающие исчерпывающую информацию о всех компонентах компьютера.

***Антивирусные программы* –** программы, позволяющие обнаруживать и уничтожать компьютерные вирусы.

***Компьютерный вирус*** – программа, способная самопроизвольно присоединяться к другим программам, создавать свои копии, внедрять их в файлы, системные области компьютера и в вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе на компьютере.

В настоящее время наиболее популярными антивирусными программами являются: **Dr Web, Norton Antivirus, Антивирус Касперского (KAV или AVP).** Программа **Dr Web** (рис. 3.3)в 2000 году была признана лучшей в мире.

***Программы – архиваторы*** позволяют преобразовать информацию таким образом, при котором уменьшается избыточность в ее представлении и соответственно требуется меньший объем памяти для хранения.

***Архивация*** (упаковка) – помещение исходных файлов в архивный файл в сжатом или несжатом виде.

***Разархивация*** (разупаковка) – процесс восстановления файлов из архива в точно таком виде, какой они имели до загрузки в архив.

В настоящее время самыми распространенными программами–архиваторами являются программы WinRAR (рис. 3.5) и WinZip.

Особенностью современных программ-архиваторов является создание самораспаковывающихся архивов. Архивы такого рода создаются в виде exe – файлов.

***Языки программирования*** (Инструментальное ПО) служат для написания новых программ.

Программы для первых компьютеров приходилось писать на *машинном языке,* т.е. в кодах, непосредственно воспринимаемых компьютером. Это было очень тяжелой, малопроизводительной и кропотливой работой, в ходе которой можно было весьма легко ошибиться. Для облегчения процесса программирования в начале 50-х годов были разработаны системы, позволяющие писать программы не на машинном языке, а с использованием мнемонических обозначений машинных команд, имен точек программы и т.д. Такой язык для написания программ называется *автокодом,* или языком *ассемблера.* Программы на ассемблере очень просто переводятся в машинные команды, это делается с помощью специальной программы, которая также называется ассемблером. Ассемблер и сейчас часто используется при программировании в тех случаях, когда требуется достичь максимального быстродействия и минимального размера программ либо наиболее полно учесть в программе особенности компьютера.

Однако написание программ на языке ассемблера все же весьма трудоемко. Для этого программист должен очень хорошо знать систему команд соответствующего компьютера, а в ходе работы ему приходится бороться не столько со сложностями решаемой задачи, сколько с переводом необходимых действий в машинные команды. Поэтому и после появления ассемблеров многие исследователи продолжали попытки облегчить процесс программирования, «научив» компьютеры понимать более удобные для человека языки составления программ. Такие языки стали называть *языками программирования высокого уровня,* а языки ассемблера и другие машинно-ориентированные языки — *языками низкого уровня.* Программы на языках высокого уровня либо преобразуются в программы, состоящие из машинных команд (это делается с помощью специальных программ, называемых *трансляторами* или *компиляторами),* либо интерпретируются с помощью программ-*интерпретаторов.*

Языки высокого уровня позволили значительно упростить процесс написания программ, так как они ориентированы на удобство описания решаемых с их помощью задач, а не на особенности какого-то конкретного компьютера. Разумеется, для каждой программы на языке высокого уровня искусный программист может написать на языке ассемблера более компактную и быстродействующую программу для выполнения тех же функций, однако эта работа является весьма трудоемкой, поэтому она имеет смысл только в особых случаях.

Первый коммерчески используемый язык программирования высокого уровня **Фортран** был разработан в 1958 году в фирме IBM под руководством Джона Бэкуса. Этот язык был предназначен, прежде всего, для научных вычислений и он (в усовершенствованном варианте) до сих пор широко используется в данной области. Для других применений было разработано множество различных языков высокого уровня, но широкое распространение получили лишь немногие из них, в частности **Си** и **Си++**, **Паскаль, Бейсик, Лого, Форт, Лисп, Пролог**, в последнее время — **Java**, и др.

**Краткая история и классификация языков программирования**

Первые языки программирования были очень примитивными и мало чем отличались от формализованных упорядоченных последовательностей единиц и нулей, понятных компьютеру. Использование таких языков было крайне неудобно с точки зрения программиста, так как он должен был знать числовые коды всех машинных команд, должен был сам распределять память под команды программы и данные.

Для того, чтобы облегчить общение человека с ЭВМ, были созданы языки программирования типа Ассемблер. Переменные величины стали изображаться символическими именами. Числовые коды операций заменились на мнемонические обозначения, которые легче запомнить. Язык программирования приблизился к человеческому языку, и отдалился от языка машинных команд.

Один из первых языков программирования – **Фортран** (**For**mula **Tran**slation) был создан в середине 50-х годов. Благодаря своей простоте и тому, что на этом языке накоплены большие библиотеки программ Фортран, и в наши дни он остается одним из самых распространенных. Он используется для инженерных и научных расчетов, для решения задач физики и других наук с развитым математическим аппаратом.

Для решения экономических задач был создан язык программирования - **Кобол**.

Расширение областей применения ЭВМ влечет за собой создание языков, ориентированных на новые сферы применения: **Снобол** – алгоритмический язык для обработки текстовой информации. **Лисп** - алгоритмический язык для обработки символов. Лисп находит широкое применение в исследованиях по созданию искусственного интеллекта.

В 1968 году был объявлен конкурс на лучший язык программирования для обучения студентов. Победителем стал язык **Алгол-68**, но широкого распространения не получил. Для этого конкурса Никлаус Вирт создал язык **Паскаль**, достаточно простой, удобный, с наличием мощных средств структурирования данных. Хотя Паскаль был разработан как язык для обучения программированию, он впоследствии получил широкое развитие и в настоящее время считается одним из самых используемых языков. Для обучения младших школьников Самуэлем Пайпертом был разработан язык **Лого**. Он отличается простотой и богатыми возможностями.

Широкое распространение в школах в качестве обучающего языка получил язык **Бейсик**, позволяющий взаимодействовать с ЭВМ в режиме непосредственного диалога. Спустя много лет после изобретения Бейсика, он и сегодня самый простой для освоения из десятков языков общецелевого программирования.

Необходимость разработки больших программ, управляющих работой ЭВМ, потребовала создания специального языка программирования **СИ** в начале 70-х г. Он является одним из универсальных языков программирования. В отличие от Паскаля, в нем заложены возможности непосредственного обращения к некоторым машинным командам и к определенным участкам памяти компьютера. Си широко используется как инструментальный язык для разработки операционных систем, трансляторов, баз данных и других системных и прикладных программ. Си – это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью, и переносимостью. Во многих случаях программы, написанные на Си, сравнимы по скорости с программами, написанными на языке Ассемблера. При этом они имеют лучшую наглядность, и их более просто сопровождать. Си сочетает эффективность и мощность в относительно малом по размеру языке.

Появление функционального программирования привело к созданию языка **Пролог.** Этот язык программирования разрабатывался для задач анализа и понимания естественных языков на основе языка формальной логики и методов автоматического доказательства теорем.

В 80-х годах 20 века был создан язык **Ада.** Этот язык в дополнение к классическим свойствам, обеспечивает программирование задач реального времени и моделирования параллельного решения задач.

Существуют различные **классификации языков программирования**. По наиболее распространенной классификации все языки программирования делят на языки **низкого, высокого и сверхвысокого** уровня.

В группу языков **низкого уровня** входят машинные языки и языки символического кодирования (**Автокод, Ассемблер**). Операторы этого языка – это те же машинные команды, но записанные мнемоническими кодами, а в качестве операндов используются не конкретные адреса, а символические имена. Все языки низкого уровня ориентированы на определенный тип компьютера, т. е. являются машинно-зависимыми. **Машинно-ориентированные языки** – это языки, наборы операторов и изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.).

Следующую, существенно более многочисленную группу составляют языки программирования **высокого уровня**. Это **Фортран, Алгол, Кобол, Паскаль, Бейсик, Си, Пролог и т.д**. Эти языки машинно-независимы, т.к. они ориентированы не на систему команд той или иной ЭВМ, а на систему операндов, характерных для записи определенного класса алгоритмов. Однако программы, написанные на языках высокого уровня, занимают больше памяти и медленнее выполняются, чем программы на машинных языках.

К языкам **сверхвысокого уровня** можно отнести лишь **Алгол-68 и APL.** Повышение уровня этих языков произошло за счет введения сверхмощных операций и операторов.

Другая классификация делит языки на **вычислительные** и языки **символьной обработки.** К первому типу относят Фортран, Паскаль, Алгол, Бейсик, Си, ко второму типу - Лисп, Пролог, Снобол и др.

В современной информатике можно выделить два основных направления развития языков программирования: **процедурное** и **непроцедурное**.

Процедурное программирование возникло на заре вычислительной техники и получило широкое распространение. В процедурных языках программа явно описывает действия, которые необходимо выполнить, а результат задается только способом получения его при помощи некоторой процедуры, которая представляет собой определенную последовательность действий.

Среди процедурных языков выделяют в свою очередь **структурные** и **операционные** **языки**. В структурных языках одним оператором записываются целые алгоритмические структуры: ветвления, циклы и т.д. В операционных языках для этого используются несколько операций. Широко распространены следующие структурные языки: Паскаль, Си, Ада, ПЛ/1. Среди операционных известны Фортран, Бейсик, Фокал.

**Непроцедурное** (**декларативное**) программирование появилось в начале 70-х годов 20 века, но стремительное его развитие началось в 80-е годы, когда был разработан японский проект создания ЭВМ пятого поколения, целью которого явилась подготовка почвы для создания интеллектуальных машин. К непроцедурному программированию относятся **функциональные** и **логические** языки.

В **функциональных** **языках** программа описывает вычисление некоторой функции. Обычно эта функция задается как композиция других, более простых, те в свою очередь разлагаются на еще более простые и т.д. Один из основных элементов в функциональных языках - **рекурсия**, то есть вычисление значения функции через значение этой же функции от других элементов. Присваивания и циклов в классических функциональных языках нет.

В **логических** **языках** программа вообще не описывает действий. Она задает данные и соотношения между ними. После этого системе можно задавать вопросы. Машина перебирает известные и заданные в программе данные и находит ответ на вопрос. Порядок перебора не описывается в программе, а неявно задается самим языком. Классическим языком логического программирования считается Пролог. Построение логической программы вообще не требует алгоритмического мышления, программа описывает статические отношения объектов, а динамика находится в механизме перебора и скрыта от программиста.

Можно выделить еще один класс языков программирования - **объектно**-**ориентированные** **языки** **высокого** **уровня.** На таких языках не описывают подробной последовательности действий для решения задачи, хотя они содержат элементы процедурного программирования. Объектно-ориентированные языки, благодаря богатому пользовательскому интерфейсу, предлагают человеку решить задачу в удобной для него форме. Примером такого языка может служить язык программирования визуального общения Object Pascal.

**Языки описания сценариев**, такие как Perl, Python, Rexx, Tcl и языки оболочек UNIX, предполагают стиль программирования, весьма отличный от характерного для языков системного уровня. Они предназначаются не для написания приложения с нуля, а для комбинирования компонентов, набор которых создается заранее при помощи других языков. Развитие и рост популярности Internet также способствовали распространению языков описания сценариев. Так, для написания сценариев широко употребляется язык Perl, а среди разработчиков Web-страниц популярен JavaScript.

**5. Прикладное программное обеспечение**

Программное обеспечение прикладного уровня представляет собой комплекс прикладных программ, с помощью которых на данном рабочем месте решаются конкретные задачи. Спектр этих задач чрезвычайно широк – от производственных до творческих, от обучающих до развлекательных.

Существуют различные классификации прикладного программного обеспечения. Рассмотрим основные классы прикладного программного обеспечения, необходимые для решения научных и образовательных задач.

* офисные пакеты
* графические редакторы
* программы распознавания текста
* программы обработки данных;
* программы-переводчики
* мультимедиа-программы
* программы для работы в Интернете

***Офисные пакеты***

Данный класс включает перечень прикладных программ, предназначенных для широкого использования. В этот класс входят:

* текстовые редакторы – программы, предназначенные для работы с текстом;
* табличные процессоры – программы, предназначенные для работы с различного рода таблицами, построения графиков и т.д.
* системы управления базами данных – программы, предназначенные для разработки и хранения локальных баз данных на автономных компьютерах
* средства презентационной графики – программы, позволяющие создавать слайд-шоу и мультимедиа-презентации.
* программы-органайзеры – программы, позволяющие планировать рабочее время, составлять протоколы встреч, расписаний, ведения записной и телефонной книжки;
* средства поддержки электронной почты.

Наиболее распространенным пакетом этого класса является пакет Microsoft Office различных версий (97, 2003, 2007).

***Графические редакторы***

Это обширный класс программ, предназначенных для создания и (или) обработки графической информации. В данном классе различают:

* растровые редакторы;
* векторные редакторы;
* программные средства для создания и обработки трехмерной графики (3D - редакторы).

*Растровые редакторы* применяют в тех случаях, когда графический объект представлен в виде комбинации точек, образующих растр и обладающих свойствами яркости и света. Такой подход эффективен в тех случаях, когда информация о цвете элементов, составляющих объект важнее, чем информация об их форме. Это характерно для фотографий. Примером растровых редакторов являются: Adobe Photoshop, Microsoft Photo Editor, Photo Plus, FotoCanvas, Paint и др. Форматы: JPG, BMP, TIFF и др.

*Векторные редакторы* отличаются от растровых способом представления данных об изображении. Элементарным объектом векторного изображения является не точка, а линия. Такой подход характерен для чертежно-графических работ, в которых форма линий имеет большее значение, чем информация о цвете отдельных точек, ее составляющих. В векторных редакторах каждая линия рассматривается как математическая кривая третьего порядка и, соответственно, представляется не комбинацией точек, а математической формулой. Такое представление намного компактнее, чем растровое. Примером векторной графики являются программы: CorelDRAW 10, Adobe Dimensions, Adobe Illustrator, Macromedia Free Hand.

Редакторы *трехмерной графики* используются для создания трехмерных композиций. Они имеют две характерные особенности. Во-первых, они позволяют гибко управлять взаимодействием свойств поверхности изображаемых объектов со свойствами источников освещения, и, во-вторых, позволяют создавать трехмерную анимацию. Примерами программ трехмерной графики являются 3DS Max (3D Studio), Maya 3D Canvas и др.

***Программы распознавания текста***

Ввести со сканера текст в компьютер несложно, однако редактировать такой текст невозможно, так как страница с текстом представляет собой графический файл – обычную картинку. Однако существуют программы, способные перевести сканированный текст из графического в текстовый формат. Эти программы называются программами распознавания текста или OCR (Optical character Recognition). Примерами таких программ являются FineReader и GineiForm.

***Программы обработки данных***

Данный класс включает программные продукты, обеспечивающие независимо от предметной области и функций информационных систем математические, статистические и другие методы решения задач.

В настоящее время разработано большое количество программ, осуществляющих статистическую обработку данных. К этим программам относятся: STATGRAPHICS plus for Windows, STATISTICA, SPSS, S-PLUS PRO 2000.

Более мощные пакеты, включающие в себя как решение различных вычислительных, математических и графических задач: MathCad 2001 Pro, 3D Grapher, Matlab 6, Mathematica 4.1, др.

***Программы-переводчики*** позволяют осуществлять автоматизированный перевод текста с одного языка другие. В настоящее время наиболее распространены программы-переводчики PROMT, STILUS, Сократ.

***Мультимедиа-программы***

Самый обширный класс программных продуктов, который в настоящее время очень динамично развивается. В настоящее время очень широко представлены следующие направления этого класса:

* обучающие программы, например, «Анатомия», «Вышивание» и др
* энциклопедии и энциклопедические словари, например, Большая Энциклопедия Кирилла и Мефодия, Большой энциклопедический словарь, Britannica, Microsoft Encarta, Толковый словарь Даля, Толковый словарь Ожегова, Передвижники, Классическая музыка
* сборники эстрадной и классической музыки, содержащие текстовую, графическую и видеоинформацию об исполнителях: Romantic Hits, Madonna
* компьютерные игры: 3D-Action (Doom, Quake); Adventure (Tomb Raider); Квесты (Гарри Потер); стратегии (Starkraft, Warkraft, Age of Empire), гонки (Need for Speed)
* различные видеофильмы

***Программы для работы в Интернете***

* электронная почта (OutlookExpress)
* броузеры (Intrnet Explorer, Netscape communicator)
* поисковые машины (Rambler, Яndex, Yahoo, Altavista)
* редакторы Web-страниц (Word, Front Page Express)
* программы для IP-телефонии

**Литература**

1. Информатика: Учебник /Под ред. Н.В.Макаровой.– М.: Финансы и статистика, 1998.–c.
2. Леонтьев В., Турецкий Д. Новейшая энциклопедия программ.– М.: Олма-Пресс, 2002.- С. 846 с.

**Лекция 6.**

**Компьютерные сети**

**Обзор компьютерных сетей**

Итак, что же такое сеть и зачем она нужна? Прежде всего, вы должны знать, что компьютерная сеть — это способ объединения компьютеров, причем таким образом, чтобы один из них мог общаться с любым другим и разделять свои ресурсы, такие, например, как данные, хранящиеся в общей базе данных. Компьютер сам по себе «знает», как проводить расчеты и как общаться с вами. Обычно компьютер общается с вами, отображая информацию на экране, а вы общаетесь с ним с помощью клавиатуры и мыши.

Сегодня существует множество других видов связи между компьютерами и людьми, например с помощью динамика и микрофона, но экран, клавиатура и мышь остаются самыми распространенными средствами такой связи.

Компьютеры не могут общаться между собой так же, как с людьми. Для этого и нужны сети.

**Типы сетей**

Существуют два основных типа сетей: локальная сеть (Local Area Network — LAN), в которой компьютеры расположены приблизительно в одном и том же месте, например, в одной организации или в одном здании, и глобальная сеть (Wide Area Network — WAN), в которой компьютеры удалены на значительные расстояния. Глобальные сети часто являются сетями сетей, объединяющими несколько локальных сетей в одну. Сейчас появилась еще одна разновидность сетей — так называемая *городская* сеть (Metropolitan Area Network — MAN), которая является своеобразным гибридом локальной и глобальной. В связи со все более и более увеличивающимися скоростями передачи данных эта сеть получает большое распространение.

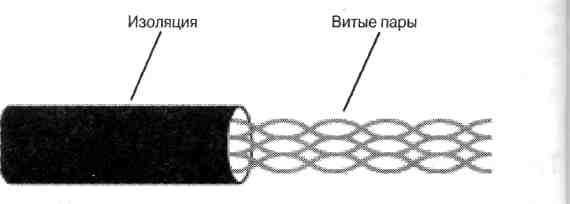
Компьютерные сети развивались довольно бурно и быстро. В зависимости от типа сети компьютеры поначалу соединялись с помощью телефонного или коаксиального кабеля (аналогичным кабелем подключается телевизор к внешней антенне). Сейчас практически везде в локальных сетях используется медный кабель, который часто называют «витая пара», либо оптический кабель. Витая пара — это, попросту говоря, несколько пар скрученных проводов, помещенных в один общий кабель. Этот кабель обычно экранирован и изолирован от внешних воздействий: электромагнитных волн и т.п..

**Коаксиальный кабель**

Изоляция Диэлектрик

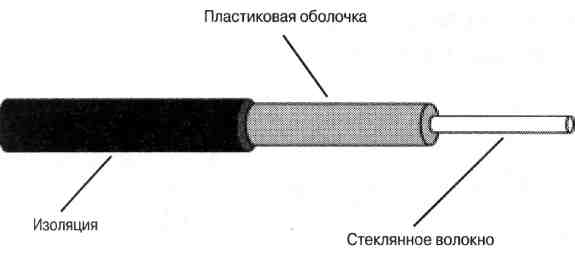


**Витая пара**



Существует 5 категорий витой пары, каждая из которых соответствует определенным параметрам. Чаще всего сейчас используют витую пару пятой категории, так как она позволяет передавать данные на довольно большие расстояния, с хорошей скоростью, и меньше всего подвержена внешним помехам.

Оптическое волокно — это тонкий и гибкий кабель, по которому данные передаются с помощью световых волн. Такой тип кабеля позволяет передавать данные на расстояния, превышающие 1 километр, без потери качества передаваемого сигнала. По своему внешнему виду оптический кабель похож на коаксиальный. Он состоит из толстого стеклянного волокна, вокруг которого оплетена пластиковая изоляция, не позволяющая выйти лучу света за пределы центрального волокна, и, наконец, все это еще раз оплетено защитной пластиковой изоляцией. Если кабель состоит из нескольких оптических волокон, то они соединены в один пучок и оплетены дополнительной пластиковой изоляцией.



*Оптический кабель*

Зачем вообще необходимы компьютерные сети? Ответ на этот вопрос каждый может легко найти сам, но давайте приведем несколько довольно веских причин.

* Во многих компаниях возникает необходимость в обмене информацией между служащими. Если компьютеры соединены в сеть, то обмен такой информацией существенно упрощается.
* Если ваша компания имеет громадную базу данных, к которой вам необходим доступ, вы можете подключить свой компьютер к компьютеру, на котором эта база данных находится, с помощью сети.
* E-mail (электронная почта) — это один из самых популярных и современных видов интерактивного общения. Использование электронной почты во многом напоминает традиционную почту, но только ее намного удобнее читать и обрабатывать. Например, большинство программ электронной почты позволяет вам автоматически сортировать письма в вашем электронном почтовом ящике по дате получения или адресам людей, отправивших вам эти сообщения.
* Возможно, что к одному из компьютеров подсоединено оборудование (например, принтер), которое необходимо пользователям других компьютеров. Это можно осуществить с помощью сети. Например, вместо того, чтобы покупать принтер для каждого компьютера, можно купить один или несколько принтеров и поместить их в компьютерную сеть.
* Компьютерные игры в мире сетей перешли на качественно новый уровень, ведь гораздо приятнее побродить по виртуальному миру не с бесчувственной железкой, а с приятелем, который может находиться даже в другом городе.

**Принцип передачи данных по локальной сети**

Для передачи данных в локальных сетях используются кадры, в которых собирается вся информация, необходимая для их доставки в запрашиваемое место. Вообще, *кадр —* это блок данных, передаваемый по сети. Размер кадра в байтах или битах и его структура зависят от протокола физического уровня, который используется в сети, например, Ethernet, Token Ring и т.д. Кадр можно сравнить с обычным конвертом: никого не удивит, если десятый конверт будет иметь те же размеры, что и девятый, — 16x11,5 см. Однако его полезная нагрузка может варьироваться от размера, содержимого, срочности и т.д. Зная только размер конверта, также нельзя сказать, как он будет перенаправлен по месту назначения. В мире сетей за передачу кадров отвечает *протокол.*

Однако в конверте может быть достаточно видимой информации, чтобы определить, кто его отправил и кому. Эта информация используется либо для отправки конверта адресату, либо для оповещения отправителя о том, что конверт не может быть доставлен.

Продолжим аналогию. Если вы знаете размер конвертов, можно создать механизм для запаковывания их в бандероли. Таким образом, стандартизация размера конвертов очень важна для механизма пересылки, так как в этом случае можно обработать все поступившие конверты, независимо от источника их создания.

Сети, по сути, являются таким механизмом, и для них необходима такая же стандартизация. Стандартизация гарантирует, что различные компоненты сети, изготовленные различными производителями, смогут работать друг с другом. В этих стандартах также описываются общие правила преобразования кадров при переходе из сети одного типа в другой, например из Ethernet в Token Ring.

За все эти стандарты, поддерживающиеся в современных высокоскоростных сетях, отвечает институт инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers — IEEE). Работа по стандартизации началась в феврале 1980 года, когда IEEE создал комитет, отвечающий за стандарты для локальных и городских сетей, который еще иногда называют *проектом 802.*

**Что нужно знать для создания компьютерной сети**

Работая с компьютерной сетью, вы будете встречаться с определенными понятиями, названиями и множеством аббревиатур (например, LAN и WAN, с которыми вы уже познакомились). Ниже приведены некоторые из них.

* **Анонимный FTP** (File Transfer Protocol — протокол передачи файлов). Протокол, позволяющий пользователям передавать файлы с одного компьютера на другой. Такой тип доступа называется анонимным, потому что для него не требуется определенной записи на FTP-сервере, в качестве имени и пароля используются соответственно слово *anonymous* и адрес электронной почты (зачастую любой). Правда, и доступ к серверу предоставляется очень ограниченный.
* **Аппаратный адрес. (MAC адрес).** Его также называют физическим адресом. Это адрес, присвоенный определенному сетевому устройству. Имеет вид шестизначного числа в шестнадцатеричном формате (например, 00-E0-6F-88-D4-25).
* **Брандмауэр** (Firewall). Аппаратное или программное устройство, защищающее сеть и управляющее доступом к ней. Брандмауэр препятствует поступлению нежелательных данных внутрь сети и позволяет только определенной информации покидать сеть.
* **Витая пара.** Физическая среда передачи данных, состоящая из пар медных проводов, переплетенных по спирали. Переплетение проводов в таком кабеле необходимо для уменьшения помех и интерференции от соседних проводов.
* **Виртуальный канал.** Канал, который для пользователя представлен в виде простого прямого соединения, но на самом деле реализующийся более сложными методами.
* **Выделенная линия.** Постоянное соединение, предоставляемое телекоммуникационной компанией для доступа к Интернет или интранет. Для соединения по выделенной линии не нужно звонить, как при модемном соединении.
* **Интранет.** Набор технологий, использующих серверы из частной сети.
* **Клиент.** Обычно так называют программу, которая работает в паре с другой программой — сервером, которая находится где-то в сети. Клиент, или клиентская программа, отправляет серверу по сети запрос для выполнения каких-либо задач.
* **Коллизия.** Результат одновременной попытки нескольких компьютеров получить доступ к физической среде сети.
* **Концентратор** (Hub). Устройство, соединяющее другие устройства (например, компьютеры) обычно в топологию «звезда».
* **Маршрутизатор** (Router). Представляет собой устройство, «принимающее решение», по какому из нескольких сетевых путей отправить данные, которые через него проходят.
* **Маршрутная таблица.** Каталог, хранящийся в памяти маршрутизатора, в котором содержатся адреса других сетей или устройств и способы их достижения.
* **Модем** (Модулятор-Демодулятор). Коммуникационное устройство, преобразующее цифровой сигнал в аналоговый (при передаче данных) и аналоговый в цифровой (при приеме данных).
* **Мост.** Устройство, соединяющее несколько локальных сетей. Мосты часто используются для сегментирования сети, тем самым увеличивая полосу пропускания данных в отдельных ее сегментах.
* **Пакет.** Набор битов, состоящих из адреса, данных и контрольной информации, который передается как единое целое.
* **Полудуплекс** (Half-duplex). Метод двухсторонней передачи данных, при котором информация в определенный момент времени может передаваться только в одном направлении.
* **Полный дуплекс** (Full-duplex). Метод передачи данных с возможностью одновременной передачи информации в обоих направлениях (отправка и получение).
* **Переход** (Нор). Термин в маршрутизации, означающий переход данных от одного маршрутизатора к другому, прежде чем будет достигнут пункт назначения.
* **Поставщик услуг Интернет (Провайдер)** (Internet Service Provider — ISP). Компания, предоставляющая прямой доступ к Интернет.
* **Повторитель** (Repeater). Устройство, позволяющее увеличить длину локальной сети или расстояние между устройствами, подключенными к ней. Повторитель усиливает сигнал, попадающий в него, прежде чем передать его дальше.
* **Протокол.** Набор правил, описывающих метод передачи информации по сети. Протоколы управляют форматом, временем передачи данных и исправлением ошибок, возникающих при передаче.
* **Сервер.** Программа, выполняющая команды клиента и выдающая определенный результат. Часто сервером называют компьютер, на котором работает серверное программное обеспечение.
* **Сетевой администратор.** Человек, который управляет сетью, компьютерами, подключенными к ней, и активным сетевым оборудованием.

**♦ Топология.** Физическая структура и организация сети. Наиболее популярными топологиями являются шинная, древовидная, «кольцо» и «звезда».

* **Управление сетью.** Работа по управлению сетью (как правило, выполняемая сетевым администратором) для поддержки ее нормальной работы. Управление сетью разбито на пять категорий: поддержка производительности, устранение неполадок, ведение учетных записей пользователей, поддержка безопасности и конфигурирование сети.
* **Файловый сервер.** Компьютер, подключенный к сети и предоставляющий большое дисковое пространство, службы доступа к нему для пользователей. Файловые серверы часто настроены таким образом, чтобы к ним могли получить доступ только определенные пользователи или группы пользователей.
* **Шлюз** (Gateway). Сетевое устройство, производящее преобразование протоколов при передаче информации между разнотипными сетями. Шлюзы часто используются для доступа к глобальной сети из локальной.
* **ADSL** (Asymmetrical Digital Subscriber Line — ассиметричная цифровая клиентская линия). Высокоскоростная модемная технология, благодаря которой предоставляется, например, доступ к Интернет по существующей телефонной линии. При таком соединении в сторону клиента достигается скорость передачи данных до 8 Мбит/с, в сторону поставщика услуг — до 1,5 Мбит/с.
* **ASCII** (American Standard Code for Information Interchange — американский стандарт кода для обмена информацией). Набор символов, в котором каждая буква, цифра или контрольный символ состоит из 7-битовой последовательности.
* **DNS** (Domain Name Service — служба доменных имен). Служба, преобразующая строчные адреса (например, [**www.yahoo**](http://www.yahoo). **com)** в понятные компьютеру IP-адреса (например, 182.156.12.24).
* **Ethernet.** Наиболее часто использующаяся среда локальных сетей, позволяющая работать со скоростями 10, 100 (Fast Ethernet) и 1000 Мбит/с (Gigabit Ethernet).
* **Frame Relay.** Метод использования виртуальных каналов для передачи данных между сетями, подключенными к глобальной сети.
* **IP** (Internet Protocol — протокол Интернет). Сетевой протокол, использующий адресную и определенную контрольную информацию для того, чтобы данные могли перемещаться по сети в нужных направлениях. IP-адрес необходим для маршрутизации и состоит из четырех десятичных чисел от 0 до 255 (например, 192.168.35.20).
* **IPX** (Internetwork Packet Exchange — обмен внутри сетевыми пакетами). Набор протоколов, разработанный компанией Novell Inc. и использующийся компьютерными системами, подключенными к сети под управлением операционной системы Novell Netware. Является эквивалентом протокола IP в наборе протоколов TCP/IP.
* **Ping.** Программа, использующаяся для определения доступности удаленного компьютера путем отправки нескольких специальных эхо пакетов и ожидания ответа.
* **RJ45.** Стандартный 8-контактный штекер (иногда его называют «джек»). Такой штекер очень похож на обычный 6-контактный штекер (RJ11), использующийся в телефонной сети.
* **TCP/IP.** Набор протоколов, использующихся для передачи данных в интранет и Интернет. TCP и IP — это два протокола. Протокол TCP (Transmission Control Protocol — протокол управления передачей) отвечает за надежную передачу данных. Протокол IP отвечает за непрерывную передачу данных.

**Сетевые топологии**

Топологии локальных сетей можно рассматривать либо с физической, либо с логической точки зрения. *Физическая топология* определяет геометрическое расположение элементов, из которых состоит сеть. Топология — это не просто карта сети, а теоретическое и в какой-то мере графическое описание формы и структуры локальной сети.

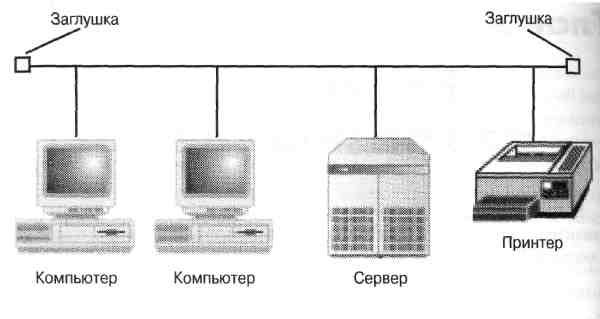
*Логическая топология* определяет возможные связи между объектами сети, которые могут общаться друг с другом. Такой тип топологии удобно использовать, когда необходимо определить, какие пары объектов сети могут обмениваться информацией, и имеют ли эти пары физическое соединение друг с другом.

**Базовые топологии**

Существуют три базовые физические топологии: линейная, кольцеобразная и звездообразная. При ***линейной топологии*** все элементы сети подключены друг за другом при помощи одного кабеля. Концы такой сети должны быть затерминированы при помощи небольших заглушек, которые часто так и называют — *терминаторы*.

Обычно при такой топологии используется один кабель, и в ней нет никакого дополнительного, активного сетевого оборудования, которое позволяет соединять компьютеры и другие объекты сети. Все подключенные к такой сети устройства «слушают» сеть и принимают только те проходящие пакеты, которые предназначены для них, остальные — игнорируют. Отсутствие дополнительного активного оборудования (например, повторителей) делает такие сети простыми и недорогими. Однако недостаток линейной топологии заключается в ограничениях по размеру сети, ее функциональности и расширяемости. Такие сети можно встретить в домах и небольших офисах.

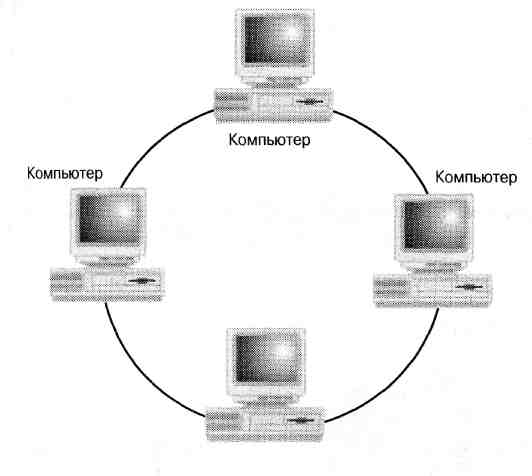
***Линейная топология сети***



При ***кольцеобразной***топологии каждая рабочая станция соединяется с двумя своими ближайшими соседями. Такая взаимосвязь образует локальную сеть в виде петли или кольца. Данные передаются по кругу в одном направлении, а каждая станция играет роль повторителя, который принимает и отвечает на пакеты, адресованные ему, и передает другие пакеты следующей рабочей станции «вниз».

В оригинальной кольцеобразной сети все ее объекты подключались друг к другу, и такое подключение должно было быть замкнутым, т.е. образовывать кольцо. Преимуществом такой топологии было предсказуемое время реагирования сети. Чем больше устройств было в кольце, тем дольше сеть реагировала на запросы. Но самый большой ее недостаток заключался в том, что при выходе из строя хотя бы одного устройства переставала функционировать вся сеть. Однако такую топологию можно улучшить, подключив все сетевые устройства через концентратор. Визуально такое «подправленное» кольцо физически кольцом уже не является, но в подобной сети данные все равно передаются по кругу.

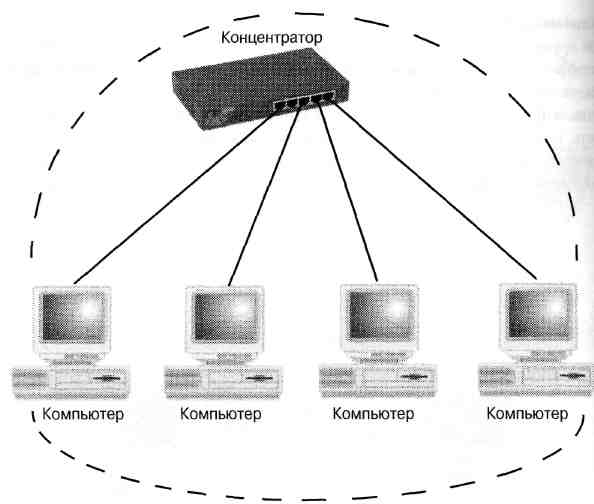
**Классическая кольцеобразная топология**



На рисунке ниже сплошными линиями обозначены физические соединения, а пунктирными — направление передачи данных. Таким образом, подобная сеть имеет логическую кольцевидную топологию, тогда как физически представляет собой звезду.

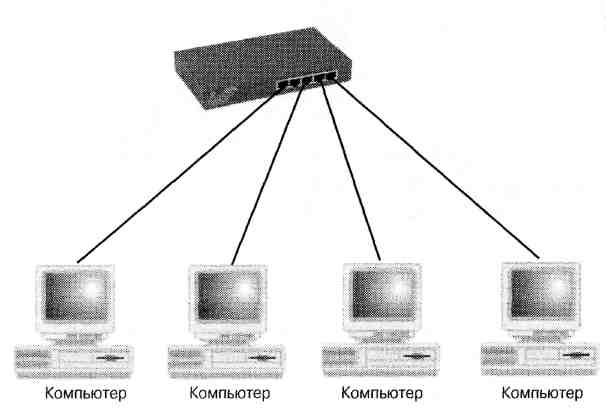
В сетях со ***звездообразной*** топологией рабочие станции подключаются к центральным устройствам — *концентраторам*. В отличие от кольцеобразной топологии, физической или виртуальной, каждое устройство в звездообразной топологии получает доступ к сети независимо от других, и общая скорость работы сети ограничена только пропускной способностью концентратора.

**Логическая кольцевидная топология при физической звездообразной**



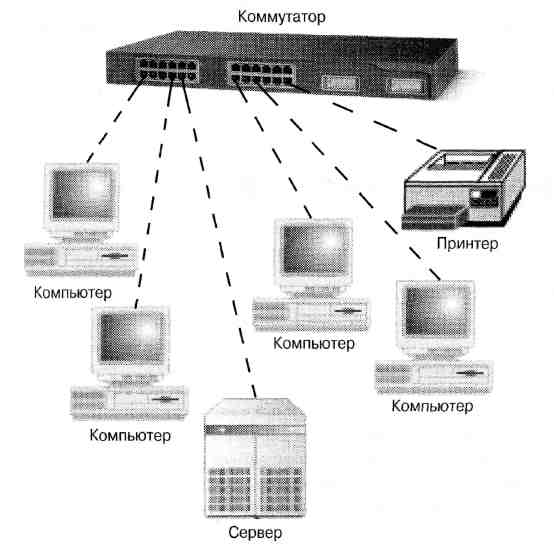
**Классическая звездообразная топология**

Концентратор



Звездообразная топология является доминирующей в современных локальных сетях. Такие сети довольно гибкие, легко расширяемые и относительно недорогие по сравнению с более сложными сетями, в которых строго фиксированы методы доступа устройств к сети. Таким образом, «звезды» вытеснили устаревшие и редко использующиеся линейные и кольцеобразные топологии. Более того, они стали переходным звеном к последнему виду топологии — *коммутируемой звезде.*

*Коммутатор* — это многопортовое, активное сетевое устройство. Коммутатор «запоминает» аппаратные (или MAC) адреса подключенных к нему устройств и создает временные пути от отправителя к получателю, по которым и передаются данные. В обычной локальной сети с коммутируемой топологией имеется несколько соединений с коммутатором. Каждый порт и устройство, которое к нему подключено, имеет свою собственную пропускную способность (или, попросту, скорость передачи данных).



Коммутаторы могут значительно улучшить производительность сетей и делают это двумя способами. Во-первых, они увеличивают общую пропускную способность, которая доступна для данной сети. Например, в 8-портовом коммутаторе может быть 8 отдельных соединений, поддерживающих скорость до 10 Мбит/с каждое. Соответственно, общая пропускная способность такого устройства составляет 80 Мбит/с. Во-вторых, коммутаторы увеличивают производительность сети, уменьшая количество устройств, которые могут заполнить всю пропускную способность одного из их сегментов. В одном таком сегменте могут находиться только два устройства: собственно сетевое устройство рабочей станции и порт коммутатора. Таким образом, за полосу пропускания в 10 Мбит/с могут «соперничать» всего два устройства, а не восемь (при использовании обыкновенного 8-портового концентратора, который не предусматривает такого разделения полосы пропускания на сегменты).

Какую же топологию выбрать для вашей сети? На этот вопрос сложно дать четкий ответ. Рассмотренные топологии являются всего лишь «кирпичиками» для построения хорошей сети. Их можно расширять, комбинировать и смешивать. Выбирайте и используйте ту топологию, которая лучше всего соответствует требованиям по производительности сети ваших клиентов. Скорее всего, идеальная топология будет представлять собой комбинацию базовых топологий.

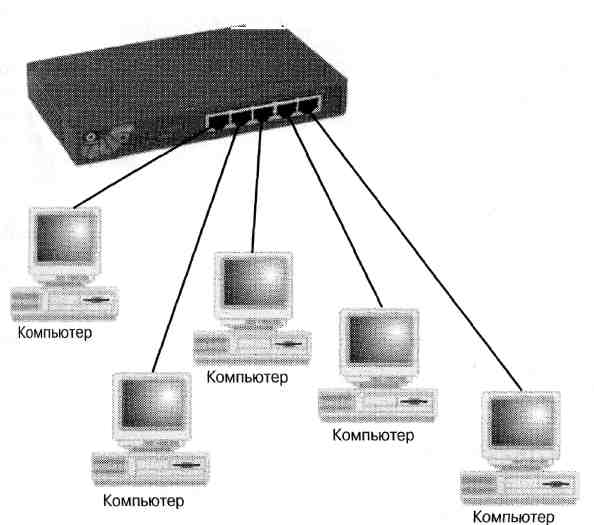
**Одноранговые и серверные сети**

Начиная разрабатывать и строить локальную сеть, необходимо четко знать, насколько сложной она будет, и как она может расширяться, а значит, усложняться в будущем. Сети можно еще разделить на два типа — одноранговые и серверные. *Одноранговая* сеть — это простейшая сеть, в которой используются рабочие станции и простейшее сетевое оборудование (концентраторы и/или коммутаторы). Такие сети очень легко настраиваются, однако они имеют некоторые ограничения. Например, если для такой сети необходим выход в Интернет, придется использовать одну из рабочих станций в качестве программного маршрутизатора. Но дома или в небольшом офисе такая сеть вполне подойдет.

*Серверные* сети стоят на более высоком уровне. Как правило, в них используется выделенный сервер (т.е. довольно мощный компьютер с определенным программным обеспечением), который управляет доступом пользователей к сети, разрешает или запрещает передачу данных в определенные точки сети и выполняет еще множество других задач. Такие сети устанавливаются в крупных компаниях с несколькими сотнями и даже тысячами пользователей, которыми гораздо легче управлять централизованно. Например, если в сети крупной компании установлен сервер Microsoft Windows Server 2003, то системному администратору не придется бегать по всему офису и устанавливать необходимые обновления программного обеспечения на всех рабочих станциях. Используя Windows Server 2003, он может сделать это, не вставая со своего рабочего места.

**Одноранговая сеть**

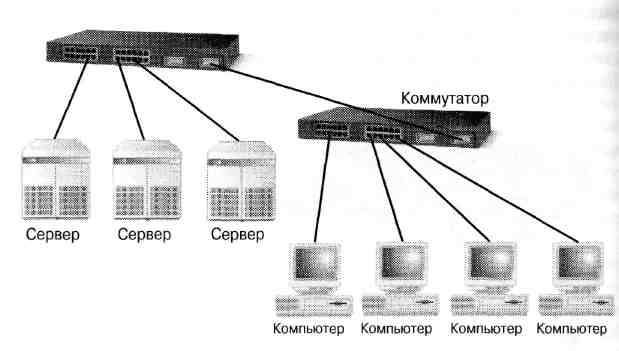
Концентратор



После этого нарисуйте на бумаге приблизительную схему сети, и вам будет легче определить, какое нужно оборудование, чтобы в будущем расширить сеть.

**Серверная сеть**

Коммутатор



При планировании сети не нужно забывать о ее стоимости. Как бы ни была богата компания, в которой строится сеть, бюджет на данный проект будет ограничен. Часто приходится изменять самый идеальный план сети из-за отсутствия необходимых средств.

**Заключение**

Итак, топология является важнейшим фактором улучшения общей производительности локальной сети. Базовые топологии — линейная, кольцеобразная и звездообразная — могут применяться в любой комбинации, и такие комбинации можно смело придумывать самостоятельно. Важно понимать, что сильные и слабые стороны каждой топологии влияют на желаемую производительность локальной сети и зависят от существующих технологий. Необходимо добиться равновесия между реальным расположением сети (например, в нескольких зданиях), возможностями используемого кабеля, путями его прокладки и даже его типом.

Удачно выбранная топология зависит от уровня необходимой пользователям производительности сети и от таких параметров, как стоимость, будущий рост и ограничения в используемых технологиях. Однако, основную сложность представляет определение зависимости между требованиями пользователей и параметрами производительности сети.

**Роль Windows Server 2003**

Серверы могут играть множество ролей в сетевом окружении типа клиент-сервер. Одни серверы сконфигурированы для проведения аутентификации, другие — для запуска приложений. Некоторые предоставляют сетевые услуги, позволяющие пользователям общаться друг с другом или находить другие серверы и ресурсы в сети. Как системный администратор, вы должны знать типы серверов и те функции, которые они выполняют в сети. Ниже перечислены типы таких серверов.

* **Контроллер домена** (Domain controller (Active Directory)). На контроллерах домена хранятся данные о пользователях и устройствах сети. Вся эта информация обозначается одним словом — *каталог*(от англ. *directory).* Такие серверы также отвечают за взаимосвязь между доменами, включая процесс регистрации пользователей в сети, аутентификацию и поиск в каталоге. Если на компьютер под управлением Windows Server 2003 установить систему Active Directory (активный каталог), предназначенную для обеспечения управления, защиты, доступа и разработки компонентов сети, этот компьютер становится контроллером домена.

В сети с работающим сервером Windows Server 2003 все серверы в домене, которые не являются контроллерами домена, называются *рядовыми серверами* (member server). Серверы, которые не связаны с доменом, называются *серверами рабочих групп* (workgroup server).

* **Файловый сервер** (File Server). На файловом сервере пользователям предоставляется централизованное место, где они могут хранить и обмениваться файлами. Например, если пользователям понадобился важный файл, скажем, план проекта, они могут получить его на файловом сервере, а не передавать между своими компьютерами.
* **Принт-сервер** (Print Server). Принт-сервер представляет собой централизованное место в сети, где пользователи могут распечатывать документы. На этом сервере также могут находиться клиентские программы с обновленными драйверами для принтеров. Кроме того, он управляет очередью печати, а также безопасностью.
* **DNS-сервер** (DNS server). Система доменных имен (Domain Name System — DNS) является службой стандартных имен в Интернет и сетях TCP/IP. Служба DNS позволяет клиентским компьютерам в вашей сети регистрироваться и распознавать имена доменов. Компьютер, который сконфигурирован для работы службы DNS, является DNS-сервером. Для работы системы Active Directory в сети должен быть DNS-сервер.
* **Сервер приложений** (Application server). Сервер приложений предоставляет ключевую инфраструктуру и службы для приложений, находящихся в системе. Обычный сервер приложений выполняет следующие функции:
* Распределение ресурсов (например, распределение соединений к базе данных).
* Управление распределенными транзакциями.
* Асинхронная связь с программами, обычно с помощью цепочки сообщений.
* Предоставление автоматического интерфейса для Web-службы XML (Extensible Markup Language — расширенный язык разметки) для доступа к бизнес-объектам.
* Поиск сбойных приложений и поддержка работоспособности программ.
* Использование встроенной службы безопасности.
* Информационный сервер Интернет (IIS) предоставляет средства и выполняет функции, необходимые для простого управления безопасным Web-сервером. Если вы планируете разместить Web- и FTP-узлы в рамках IIS, сконфигурируйте сервер как сервер приложений.
* **Терминальный сервер** (Terminal server). Терминальный сервер предоставляет удаленным компьютерам доступ к Windows-приложениям, работающим под управлением системы Windows Server 2003. Благодаря терминальному серверу можно устанавливать приложения в одном месте на одном сервере. После этого сразу несколько пользователей могут получить доступ к этим приложениям, не устанавливая их на свои компьютеры. Пользователи могут запускать программы, сохранять файлы и пользоваться ресурсами сети с удаленного сервера так, как если бы все эти ресурсы находились на их собственных компьютерах.

**Лекция 6**

**Безопасность информации**

***Резервирование данных***

В процессе эксплуатации компьютера по самым разным причинам возможны (и часто происходят) порча и потеря информации, находящейся на жестких дисках компьютера. Это может быть вызвано ошибочными действиями пользователей, некорректной работой программ, сбоями в электропитании, авариями жестких дисков, компьютерными вирусами, пожарами и т.д. Единственный надежный способ предотвращения потери информации и соответствующих (иногда очень существенных) потерь времени и денег — это создание *резервных копий* данных, то есть копий, позволяющих восстановить данные при их повреждении или утрате. Процесс создания резервных копий обычно называется *резервированием.*

Создание резервных копий на дискетах целесообразно только для тех пользователей, у которых имеется лишь несколько Мбайт, в крайнем случае, несколько десятков Мбайт данных, подлежащих копированию (остальное — это, скажем, программы или данные, уже имеющиеся на дистрибутивных дискетах или компакт-дисках).

В качестве носителей для резервных данных в наше время чаще всего используют CD, DVD, Флэш память, карты памяти и внешние винчестеры.

***Ограничение доступа***

Для обеспечения безопасности данных очень полезно применение простых, но эффективных мер по ограничению доступа к этим данным.

Следует устанавливать защиту от записи на дискетах и других съемных дисках с файлами, которые не надо изменять — это позволит избежать их случайного уничтожения, а также заражения и порчи вирусами.

Не следует допускать к работе на компьютере без присмотра посторонних лиц, особенно если они имеют свои дискеты. Наиболее опасны любители компьютерных игр — они способны удалить файлы, если для их игры не хватает места, они могут не только не соблюдать никаких мер предосторожности от вирусов, но и игнорировать любые предупреждения антивирусных средств. Так, очень часто причиной заражения компьютера вирусом являлась принесенная на дискете игра, в которую кто-то поиграл 5-10 минут на компьютере.

Если желательно предотвратить доступ к компьютеру в отсутствие его владельца, можно использовать реализованную в BIOS многих компьютеров возможность установки пароля. Это не позволит загрузить компьютер, не указав пароль. Аналогично можно поступить и при загрузке ОС.

В том случае, если избежать доступа случайных лиц к компьютеру невозможно (например, в учебном центре), целесообразно все или почти все программы, находящиеся на жестком диске компьютера, располагать на логическом диске, защищенном от записи. Создать такие диски можно, например, с помощью программы DiskLock или с помощью программно-аппаратного комплекса Sheriff (который, к тому же, защищает компьютер от вирусов).

***Хранение конфиденциальных и секретных данных***

Если на компьютере, к которому может иметь доступ более одного человека, необходимо держать конфиденциальные данные, их следует хранить в зашифрованной форме — скажем, в защищенном паролем архиве программ PKZIP или ARJ, или на защищенном паролем диске (NDisk), созданном программой DiskReet из комплекса Norton Utilities. Впрочем, оба способа ненадежны против серьезного взлома — существуют даже программы, взламывающие пароли, и эти программы легко доступны. Так что для действительно секретных данных или данных, представляющих серьезный коммерческий интерес, такие методы не годятся. Здесь следует предпринимать особые, комплексные, схемы защиты, которые выходят за пределы нашей лекции.

***Защита от компьютерных вирусов***

***Компьютерные вирусы*** — это специально написанные программы, которые могут записывать (внедрять) свои копии (возможно, измененные) в компьютерные программы, расположенные в исполняемых файлах, системных областях дисков, драйверах, документах и т.д., причем эти копии сохраняют возможность к самопроизвольному «размножению». Программа или иной объект, содержащие вирус, называются *зараженными.* Зараженными могут быть исполняемые файлы, программы начальной загрузки жесткого диска или дискеты, файлы драйверов, командные файлы DOS, документы Word для Windows, электронные таблицы в формате Excel и т.д.

Многие вирусы не только «заражают» другие программы или иные объекты, но и выполняют различные вредные действия, иногда — очень разрушительные: форматирование дисков, шифрование данных на дисках, изменение случайно выбранных блоков данных на жестком диске и т.д. Но даже те вирусы, которые не портят данные сознательно, могут сильно мешать работе. Поэтому если на Вашем компьютере есть сколько-нибудь ценные данные, а не только DOOM, Quake, Dune и другие средства убить время, следует обязательно применять антивирусные средства и предпринимать некоторые элементарные меры предосторожности, позволяющие предотвратить заражение вирусом.

***«Ремонтный набор»***

Все программное обеспечение, расположенное на жестком диске компьютера, может оказаться недоступным, если компьютер вдруг перестанет загружаться с жесткого диска. В таких случаях приходится загружать компьютер с компакт-диска, выяснять и устранять причину повреждения, работая со съемными носителями информации.

При заражении компьютера вирусом обычно загрузить компьютер с жесткого диска вполне возможно, но делать этого не следует, так как при этом, скорее всего, будет запущен вирус, который может что-то испортить, и при наличии которого всякое лечение от вируса будет бесполезно (вылеченные объекты будут снова заражаться). Так что вся работа по лечению компьютера должна выполняться при работе только со съемными носителями информации.

*Рекомендуемый «Ремонтный набор».*

1. Загружаемый CD (DVD) с установленной ОС (лучше копия, а оригинал – храниться)
2. ОС DOS с дисковыми (HDD) утилитами и файловыми менеджерами.
3. Программы для работы с HDD установленной ОС.
4. Комплект дистрибутива основных прикладных программ и утилит.
5. Для учебных классов набор учебных программ на сменном носителе.

Когда зараженная программа начинает работу, то сначала управление получает вирус. Вирус находит и «заражает» другие программы или иные объекты, а также может выполнить какие-нибудь вредные действия. Затем вирус передает управление той программе, в которой он находится, и она работает так же, как обычно. Тем самым внешне работа зараженной программы выглядит так же, как и незараженной.

***Резидентные вирусы.*** Многие разновидности вирусов устроены так, что при запуске зараженной программы вирус остается резидентно, т.е. до перезагрузки OS, в памяти компьютера. В этом случае вирус может вплоть до перезагрузки компьютера заражать программы и выполнять вредные действия на компьютере. При этом такие вирусы обычно стараются обеспечить свою активизацию и после перезагрузки компьютера. Имеются даже вирусы, выживающие после выключения компьютера или нажатия кнопки «Reset» с последующей загрузкой с чистой системной дискеты!

***Назначение вирусов***

Большинство вирусов не выполняет каких-либо действий, кроме своего распространения (заражения других программ, дисков и т.д.) и, иногда, выдачи каких-либо сообщений или иных эффектов («приколов»), придуманных автором вируса: проигрывания музыки, перезагрузки компьютера, выдачи на экран разных рисунков, блокировки или изменения функций клавиш клавиатуры, замедления работы компьютера, создания видеоэффектов и т.д.. Однако, сознательной (или по недомыслию) порчи информации эти вирусы не осуществляют. Такие вирусы условно называются *неопасными.* Впрочем, и эти вирусы способны причинить большие неприятности (например, перезагрузка компьютера каждые несколько минут вообще не дадут Вам работать).

Однако около трети всех видов вирусов **портят данные на дисках** — или сознательно, или из-за содержащихся в вирусах ошибок, скажем, из-за не вполне корректного выполнения некоторых действий. Если порча данных происходит лишь эпизодически и не приводит к тяжелым последствиям (например, портится лишь .СОМ-файлы при заражении, если размер этих файлов более 64000 байт), то вирусы называются малоопасными. Если же порча данных происходит часто или вирусы причиняют значительные разрушения, то такие вирусы называются опасными.

Вирусы могут читать поля, где хранятся пароли для Интернет-соединений и Интернет-кошельков и пересылать их отправителю вируса. В этом случае работа вируса заканчивается потерей денег – это различного рода **черви и трояны.** Наилучший способ борьбы с ними - не сохранять пароли на компьютере, а запоминать их или записывать отдельно.

С появлением Интернет появилась большая группа вирусов - *несанкционированная реклама.* Эти вирусы с какой-то периодичностью выводят на экран какую-либо рекламу или загружают сайт с рекламой через Интернет.

**Типы вирусов.**

*Файловые вирусы.* Компьютерные вирусы отличаются друг от друга по тому, в какие объекты они внедряются, иначе говоря, что они заражают. Впрочем, некоторые вирусы заражают сразу несколько видов объектов.

Большинство вирусов распространяется, заражая исполняемые файлы, т.е. файлы с расширениями имени .СОМ и .ЕХЕ, а также оверлейные файлы (то есть вспомогательные программные файлы, загружаемые при выполнении других программ). Такие вирусы называются *файловыми.* Вирус в зараженных исполняемых файлах начинает свою работу при запуске той программы, в которой он находится.

Еще один широко распространенный вид вирусов внедряется в начальный сектор дискет или логических дисков, где находится загрузчик операционной системы, или в начальный сектор жестких дисков, где находится таблица разбиения жесткого диска и небольшая программа, осуществляющая загрузку с одного из разделов, указанных в этой таблице. Такие вирусы называются *загрузочными,* или *бутовыми* (от слова *boot* — загрузчик). Эти вирусы начинает свою работу при загрузке компьютера с зараженного диска. Загрузочные вирусы всегда являются резидентными и заражают вставляемые в компьютер дискеты. Встречаются загрузочные вирусы, заражающие также и файлы — *файлово-загрузочные вирусы.*

*Замечания*. Дискеты, через которые распространяются загрузочные вирусы, вовсе не обязаны быть системными. Ведь на любой дискете в начальном секторе имеется загрузчик ОС, но на системной дискете он находит файлы ОС, загружает их и передает им управление, а на несистемной дискете — не находит и выводит сообщение «Non-system disk» или что-то в этом роде.

2. Для заражения компьютера загрузочным вирусом достаточно всего один раз оставить зараженную дискету в дисководе А: в момент перезагрузки компьютера. При этом вирус заразит жесткий диск компьютера. И после этого при загрузке с жесткого диска компьютера будет запускаться вирус.

Некоторые вирусы умеют заражать драйверы, то есть файлы, указываемые в предложении DEVICE или DEVICEHIGH файла CONFIG.SYS.

Вирус, находящийся в драйвере, начинает свою работу при загрузке данного драйвера из файла CONFIG.SYS при начальной загрузке компьютера. Обычно заражающие драйверы вирусы заражают также исполняемые файлы или загрузочные сектора дискет, поскольку иначе им не удавалось бы распространяться — ведь драйверы очень редко переписывают с одного компьютера на другой. Иногда заражение драйверов используется в качестве этапа в стратегии распространения вируса. Например, вирус JEWS-2339 при загрузке из исполняемого файла заражает все драйверы, обнаруженные в CONFIG.SYS, а при загрузке зараженного драйвера становится резидентным и заражает все файлы на дискетах (а на жестком диске, наоборот, лечит).

Летом 1995 года появилась новая разновидность вирусов — вирусы, заражающие документы Word для Windows версий 6.0 и 7.0. Вследствие распространенности редактора Word для Windows такие документы имеются почти на каждом компьютере. Долгое время заражение файлов документов считалось невозможным, так как документы не содержали исполняемых программ. Однако программисты фирмы Microsoft встроили в документы Word для Windows мощный язык макрокоманд WordBasic. При этом макрокоманды не видны в редактируемом документе — но именно в них может содержаться вирус.

**Резюме:**

1. Любая информация, находящаяся на компьютере, может быть найдена открыта и расшифрована. Это вопрос времени и денег (желания).
2. Наиболее ценная информация должна находиться на сменных носителях вне компьютера.
3. Приватная информация передаваемая через интернет должна быть закодирована, а так же передаваться частями по разным каналам и в разное время.
4. Антивирусные программы рекомендуется обновлять как минимум еженедельно, а лучше ежедневно. Также желательно иметь возможность сканировать информацию различными антивирусами.

**Защита информации в сетях**

1. Система допусков и паролей на различных уровнях
2. Ограничения доступа к сети
3. Администрирование
4. Шифрование информации
5. Использование спецканалов связи.
6. Программная защита несанкционированного доступа как к данной сети, так и к обращениям из данной сети.
7. Антивирусная защита

Вопросы для закрепления материала по лекционному курсу.

**Информатика. Предмет и задачи**

1. Информатика. Предмет и задачи информатики как науки.
2. Основные объекты предмета Информатика
3. Понятие интерфейса. Типы интерфейсов.
4. Информационные революции.
5. Понятия: информация, данные, методы
6. Единицы измерения информации
7. Способы создания, хранения и передачи информации
8. Представление данных в компьютере.
9. Национальные программы создания информационного общества
10. Внедрение информационных технологий в повседневную жизнь.
11. Спорные проблемы 4- ой информационной революции
12. Место и значение Интернет для информации.
13. Позиционные системы счисления
14. Логические основы ЭВМ
15. Индекс ISI и факторы на него влияющие.
16. История развития вычислительной техники
17. История развития ИТ в России.
18. Тезаурусная мера информации.

**Аппаратное обеспечение**

1. Мониторы. Основные технические характеристики цветных мониторов:
2. Принтеры. Основные технические характеристики принтеров:
3. Сканеры. Основные технические характеристики сканеров:
4. Центральный процессор. Основные технические характеристики центральных процессоров:
5. Основные внутренние устройства компьютера.
6. Устройство «МОДЕМ» и его назначение и классификация:
7. Базовая, стандартная и расширенная конфигурация компьютера.
8. Винчестер. Основные технические характеристики винчестеров
9. Основные шины компьютера. Основные технические характеристики
10. Оперативная память компьютера. Основные технические характеристики
11. Постоянное запоминающее устройство. Основные технические характеристики
12. Материнская плата. Основные технические характеристики
13. Корпус компьютера. Основные технические характеристики
14. Основные устройства для чтения сменных устройств памяти. Основные технические характеристики
15. Видеокарта. Основные технические характеристики
16. Звуковая карта и акустические системы. Основные технические характеристики
17. Основные устройства ввода информации в компьютер. Основные технические характеристики
18. Основные устройства вывода информации из компьютера. Основные технические характеристики
19. Основные порты обмена компьютера. Основные технические характеристики
20. Основная конфигурация ПК.

**Программное обеспечение**

1. Понятие программы, программное обеспечение, классификация программного обеспечения
2. Базовое программное обеспечение. ППЗУ
3. Основные функции BIOS.
4. Запуск BIOS и тестирование АО.
5. С какими устройствами работает BIOS.
6. Системное программное обеспечение. Существующие варианты ОС.
7. Основные функции ОС и отличие ОС друг от друга.
8. Файловые менеджеры.
9. Служебные и стандартные программы.
10. Прикладные программы.
11. Инструментальные программы. Их классификации.
12. Основные понятия языков программирования, структуры и типы данных языка программирования
13. Офисные пакеты программ на примере MS Office.
14. Интернет программы и почтовые программы.
15. Программы для работы со звуком изображением и музыкальные программы.
16. Тестовые и отладочные и настроечные программы.
17. Антивирусы, архиваторы, альбомы.
18. Программы для просмотра графических файлов и графические редакторы.
19. Специальное программное обеспечение.
20. Программы работы с текстом, конвертаторы
21. Базы данных и СУБД. Классификация.
22. Основные этапы создания баз данных.
23. Основные элементы реляционных баз данных.
24. БД Access. Структура.
25. Информационная модель объекта. Блок схема.
26. Основные элементы программирования.

**Файловая структура**

1. Структура долговременной памяти компьютера.
2. Понятие «файл», «папка».
3. Понятие «устройство памяти».
4. Правила написания имён файлов папок и устройств.
5. Дерево каталогов, правила создания и назначение.
6. Атрибуты файла.
7. Понятие «кластер». Фактический размер файла и размер файла на устройстве. Размер кластера.
8. Физическая структура дисковой памяти.
9. Понятие форматирование. FAT.
10. Двоичная система счисления.
11. Физический диск, логический диск, сектор.
12. Полное имя файла, путь к файлу, расширение файла.

**Локальные и глобальные сети ЭВМ**

1. Основные типы сетей их назначение и характеристики.
2. Способы соединения (подключения) компьютеров к сети.
3. Топология компьютерных сетей
4. Коммуникационное оборудование.
5. Сетевой сервис и сетевые стандарты.
6. Программы для работы в сети Интернет
7. IP адрес , WWW адрес , DNS служба
8. InterNet и WWW
9. Правила организации имён в www и электронной почте.
10. Одноранговые и серверные сети. Основные типы серверов.
11. Защита информации в ПК.
12. Защита информации в локальных сетях.
13. Защита информации в беспроводных сетях.
14. Основные способы подключения к интернет и их характеристики.