

**Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
"Кубанский государственный аграрный университет"
(КубГАУ)**

Кафедра компьютерных технологий и систем

С.В. Лаптев

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Курс лекций для студентов специальности
21.03.02120700.62 - "Землеустройство и кадастры"
(прикладной и академический бакалавриат)

Краснодар
20173

УДК 004.42 (073.8)

ББК 32.973.26-018

А-45

Рецензенты:

Барановская Т.П. – профессор, доктор экономических наук, заведующая кафедрой системного анализа и обработки информации Кубанского государственного аграрного университета (г. Краснодар);

Степанов В.В. – профессор, доктор технических наук, заведующий кафедрой вычислительной техники и автоматизированных систем (ВТиАС) Кубанского государственного технологического университета (г. Краснодар)

Информационные технологии: Курс лекций для студентов специальности 21.03.02 "Землеустройство и кадастры" 230201.65 (прикладной и академический бакалавриат)- "Информационные системы и технологии" и 080801.65—"Прикладная информатика (по областям)"/ С. В. Лаптев, – Краснодар: ФГОУ ВПО КубГАУ, 2017~~1~~. – 135~~4~~2 с.

Составлен в соответствии с рабочей программой дисциплины **Информационные технологии** для студентов второго курса специальности 21.03.02 230201.65 – “Землеустройство и кадастры Информационные системы и технологии” 080801.65—"Прикладная информатика (по областям)".

Работа выполнена по решению методической комиссии факультета прикладной информатики и кафедры компьютерных технологий и систем (протокол №29 от 28.02.2017~~1~~ г.)

Введение.

Включение курсов информационных систем и технологий в учебные планы различных специальностей государственных вузов говорит о безусловной значимости информатики и её составной части - информационной технологии в деятельности специалиста любого профиля.

Предлагаемый курс написан в соответствии с Государственным стандартом по дисциплине «Информационные технологии» и состоит из семнадцати лекций.

Для успешного усвоения изучаемого материала необходимы базовые знания курса «Информатики», который изучается студентами различных специальностей ранее.

Содержимое курса лекций можно разбить на 7 основных разделов.

В первом разделе студенты получают базовые понятия об автоматизированных информационных технологиях и методах оценки информации. Второй раздел посвящен глубокому изучению понятия систем, управления и моделирования в системах. В третьем разделе курса учащиеся получают сведения о базовых информационных технологиях и уровнях их моделирования.

В любой базовой информационной технологии обязательно присутствуют три информационных процесса – обработки, накопления и передачи данных. Четвертый и пятый разделы курса предназначены для детального изучения процессов обработки и накопления данных. В шестом разделе рассмотрены базовые конструкции языка запросов SQL, широко применяющегося для работы с различными базами данных, которые в настоящее время являются основным средством структурирования данных в процессе накопления.

Заключительный, седьмой раздел курса, предназначен для ознакомления с процессом передачи данных. В разделе рассматриваются компьютерные сети, которые в настоящее время широко распространены и динамично развиваются.

Базовые информационные технологии высокого уровня помимо процессов обработки, накопления и передачи данных включают в себя также процесс накопления знаний. Данный процесс в предлагаемом курсе лекций детально не рассматривается

Отформатировано: По ширине

Отформатировано: Шрифт: 16 пт

Отформатировано: По ширине,
Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: По ширине,
Отступ: Первая строка: 1,25 см

Отформатировано: По ширине

ввиду ограниченности часового объема в Государственном стандарте.

Отформатировано: русский

Отформатировано: Шрифт: не полужирный

Отформатировано: По ширине, Отступ: Первая строка: 1,25 см

Лекция 1. Понятие информационной технологии.

1. Возникновение информационных технологий (ИТ).
2. Автоматизированные информационные технологии.
3. Информатика и информационная технология.
Моделирование информационного процесса.

1. Возникновение информационных технологий (ИТ).

Понятие «информационная технология» возникло в последние десятилетия XX века в процессе становления информатики. Особенностью информационных технологий является то, что в ней и предметом, и продуктом труда является информация, а орудиями труда - средства вычислительной техники и связи. Информационная технология как наука о производстве информации возникла именно потому, что информация стала рассматриваться как вполне реальный производственный ресурс наряду с другими материальными ресурсами. Причем производство информации и ее верхнего уровня - знаний оказывает решающее влияние на модификацию и создание новых промышленных технологий.

Под технологией в широком смысле понимают науку о производстве материальных благ, включающей три аспекта: информационный, инструментальный и социальный [39]. *Информационный* аспект включает описание принципов и методов производства, *инструментальный* - орудия труда, с помощью которых реализуется производство, *социальный* - кадры и их организацию. В более узком промышленном смысле технология рассматривается как последовательность действий над предметом труда с целью получения конечного продукта. Например, технология получения стали из железной руды или технология производства сливочного масла из молока.

Понятие „информационная технология“ возникло в последние десятилетия XX века в процессе становления информатики. Особенностью информационных технологий является то, что в ней и предметом, и продуктом труда является информация, а орудиями труда - средства вычислительной техники и связи. Информационная технология как наука о производстве

информации возникла именно потому, что информация стала рассматриваться как вполне реальный производственный ресурс наряду с другими материальными ресурсами. Причем производство информации и ее верхнего уровня - знаний оказывает решающее влияние на модификацию и создание новых промышленных технологий.

За точку отсчета развития человеческой цивилизации обычно принимают время, когда люди начали создавать орудия труда и охоты. Вся последующая история технического прогресса от овладения огнем до открытия ядерной энергии - это история последовательного подчинения человеку все более могущественных сил природы. Задача, решаемая на протяжении тысячелетий, - умножать различными инструментами и машинами мускульную силу человека. В то же время попытки создания инструментов, усиливающих природные возможности человека по обработке информации, начиная от камешков абака и до механического прототипа современной ЭВМ - машины Беббиджа, составляют лишь ничтожную часть в общем потоке развития научно-технического прогресса.

Этапы развития информационных технологий.

Письменность стала первым историческим этапом развития информационной технологии. Истоки этого этапа исследованы почти на 30 тыс. лет в глубь человеческой истории: пещерная живопись - выполненные 25-20 тыс. лет назад наскальные изображения людей и животных; лунный календарь, выгравированный на кости более чем 20 тыс. лет назад; специальным образом обработанные кости с числовыми нарезками, которые 26 тыс. лет назад «должны были служить инструментами для измерения».

Книгопечатание – второй исторический этап развития информационных технологий, позволившее тиражировать информацию и открывшее эру бумажной информационной технологии, занимающей и в настоящее время значительное место.

Изобретение телеграфа, телефона, радио, затем телевидения - третий исторический этап развития информационных технологий, принципиально изменивший и ускоривший процесс передачи информации.

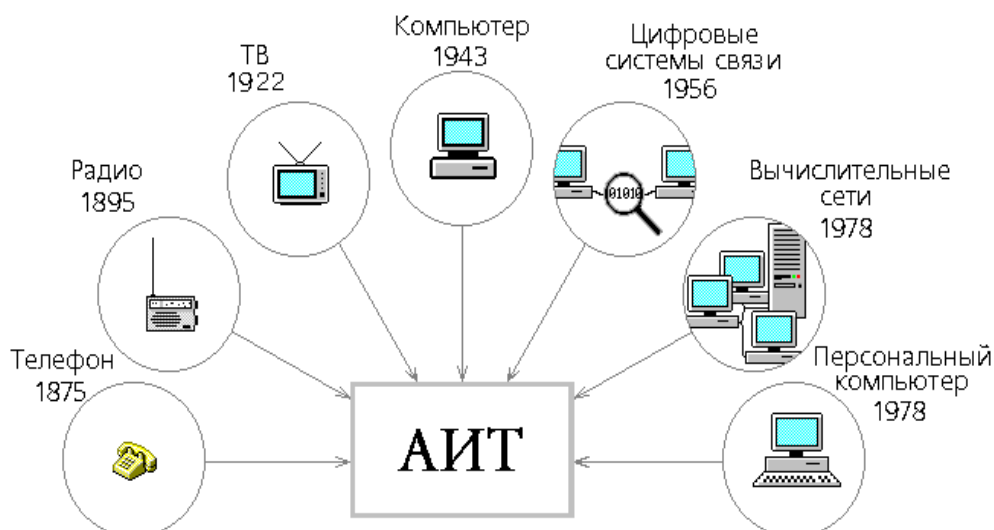
Изобретение микропроцессора и первого персонального компьютера, а затем цифровых систем связи и вычислительных

сетей – четвертый исторический этап развития информационных технологий.

2. Автоматизированные информационные технологии.

Автоматизированные информационные технологии (АИТ) – следствие интеграции достижений человечества в области средств связи, обработки, накопления и отображения информации.

На рисунке [ниже 1.1](#) представлена схема формирования информационных технологий, являющихся следствием интеграции достижений человечества в области средств связи, обработки, накопления и отображения информации.



[Схема формирования информационных технологий](#) **Рисунок**

1.1

Отформатировано: Шрифт: 16 пт

Отформатировано: По левому краю, Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: 16 пт

Основу автоматизированных информационных технологий составляют следующие технические достижения:

1) создание средств накопления больших объемов информации на машинных носителях таких, как магнитные и оптические диски;

2) создание различных средств связи таких, как радио - и телевизионная связь, телекс, телефакс, цифровые системы связи, компьютерные сети, космическая связь, позволяющих

воспринимать, использовать и передавать информацию практически в любой точке земного шара;

3) создание компьютера, и особенно персонального компьютера, позволяющего по определенным алгоритмам обрабатывать и отображать информацию, накапливать и генерировать знания.

Информационные технологии направлены на увеличение степени автоматизации всех информационных операций, и следовательно, ускорения научно-технического прогресса общества.

3. Информатика и информационная технология. Моделирование информационного процесса.

Информатика как наука занимается изучением информационных процессов и методов их автоматизации на основе программно - аппаратных средств вычислительной техники и средств связи. Исторически информатика изучала научную информацию и способы её структуризации, систематизации, хранения и распространения. Появление средств вычислительной техники позволило автоматизировать часть указанных операций. Дальнейшее изучение процессов возникновения, накопления информации, её структуризации, передачи, обработки и представления потребовало создания специального аппарата, позволяющего описывать, анализировать и систематизировать различные фазы информационных процессов. Так возник аппарат информационного моделирования. Наличие частных моделей информационных процессов позволило целенаправленно использовать средства вычислительной техники и связи, которые в свою очередь совершенствовались для большего удовлетворения потребностей информатики. Начиная с 80-х годов нашего столетия, различные фазы преобразования информации стали рассматриваться как единый информационный процесс, направленный на удовлетворение информационных потребностей человечества. В этом проявился выход информатики на глобальный уровень, позволяющий говорить о том, что человечество осознало информацию как ресурс развития общества, а информатику как науку, развитие которой позволит обеспечить полное использование этого ресурса. С информатикой связывают решение

принципиально новых проблем человечества: создание информационной модели мира; расширение творческого аспекта деятельности человека; переход к безбумажной информатике; доступность информационного ресурса каждому члену общества.

В настоящее время информатика приобрела многоаспектный характер. В ней соединены глобальность и конкретность применения, методы формализации и физической реализации.

При моделировании информационного процесса и его фаз, как указывалось, выделяют три уровня: *концептуальный*, на котором описывается содержание и структура предметной области; *логический*, на котором производится формализация модели и *физический*, определяющий способ реализации информационной модели в техническом устройстве.

Трёхуровневый подход может быть целесообразен и при изучении информатики. При таком подходе можно выделить следующие уровни информатики: физический, логический и прикладной (или пользовательский).

На *физическом* уровне информатика изучает аппаратно-программные средства вычислительной техники и средств связи, которые как бы составляют её фундамент и позволяют физически реализовывать её логический и прикладной уровни.

На *логическом* уровне информатики изучается технология переработки информационного ресурса с целью получения новой информации на базе средств ВТ и связи. То есть, логический уровень - это *информационная технология*.

Наконец, третий, *прикладной* уровень информатики характерен изучением вопросов использования информационной технологии при создании и эксплуатации систем, в которых преобладающими процессами являются информационные.

Таким образом, предметом изучения курса «Информационные технологии» являются логический и прикладной уровни информатики. Физический же уровень изучается в курсе «Информатика», который посвящен аппаратным средствам электронной вычислительной техники и базовому программному обеспечению.

Информационная технология (ИТ) имеет свою цель, методы и средства реализации. Кратко их содержание состоит в следующем.

Целью информационной технологии является создание из информационного ресурса качественного информационного продукта, удовлетворяющего требованиям пользователя.

Методами ИТ являются методы обработки и передачи данных.

Средства ИТ - это математические, программные, информационные, технические и другие средства.

При таком определении целей, методов и средств под автоматизированной информационной технологией будем понимать целостную техническую *систему*, обеспечивающую целенаправленное создание, передачу, хранение и отображение информационного продукта (данных, идей, знаний) с наименьшими затратами и в соответствии с закономерностями той социальной среды, где развивается информационная технология [36].

Практическое приложение методов и средств обработки данных может быть различным, поэтому целесообразно выделить глобальную, базовые и конкретные информационные технологии.

Глобальная информационная технология включает модели, методы и средства, формализующие и позволяющие использовать информационные ресурсы общества. *Базовая информационная технология* предназначена для определённой области применения (производство, научные исследования, обучение и т.д.). *Конкретные информационные технологии* реализуют обработку данных при решении функциональных задач пользователей (например, задачи учёта, планирования, анализа).

Лекция 2. Методы оценки информации.

- 1. Статистический подход оценки информации.**
- 2. Семантический подход оценки информации.**
- 3. Структурный подход оценки информации.**
- 4. Прагматический подход оценки информации**

1. Статистический подход оценки информации.

Для того, чтобы оценить и измерить количество информации применяются различные подходы и методы. Среди них выделяются статистический, семантический, прагматический и структурный. Исторически наибольшее развитие получил статистический подход.

Статистический подход изучается в обширном разделе кибернетики, называемом теорией информации. Его основоположником считается К. Шеннон, опубликовавший в 1948г. свою математическую теорию связи. Большой вклад в теорию информации до него сделали ученые Найквист и Хартли. В 1924 и 1928 годах они опубликовали работы по теории телеграфии и передаче информации. Признаны во всем мире исследования по теории информации российских ученых А.Н. Колмогорова, А.Я. Хинчина, В.А. Котельникова, А.А. Харкевича и др.

К. Шенноном было введено понятие количества информации как меры неопределенности состояния системы, снимаемой при получении информации. Количественно выраженная неопределенность состояния получила название энтропии. При получении информации уменьшается неопределенность, т. е. энтропия, системы. Очевидно, что чем больше информации получает наблюдатель, тем больше снимается неопределенность, и энтропия системы уменьшается. При энтропии, равной нулю, о системе имеется полная информация, и наблюдателю она представляется целиком упорядоченной. Таким образом, получение информации связано с изменением степени неосведомленности получателя о состоянии этой системы.

До получения информации ее получатель мог иметь некоторые предварительные (априорные) сведения о системе X .

Оставшаяся неосведомленность и является для него мерой неопределенности состояния (энтропией) системы. Обозначим априорную энтропию системы X через $H(X)$. После получения некоторого сообщения наблюдатель приобрел дополнительную информацию $I(X)$, уменьшившую его начальную неосведомленность так, что апостериорная (после получения информации) неопределенность состояния системы стала $H'(X)$. Тогда количество информации I может быть определено как

$$I(X) = H(X) - H'(X)$$

Другими словами, количество информации измеряется уменьшением (изменением) неопределенности состояния системы.

Если апостериорная энтропия системы обратится в нуль, то первоначально неполное знание заменится полным знанием и количество информации, полученной в этом случае наблюдателем, будет

$$I(X) = H(X),$$

т.е. энтропия системы может рассматриваться как мера недостающей информации.

Если система X обладает дискретными состояниями (т.е. переходит из состояния в состояние скачком), их количество равно N , а вероятность нахождения системы в каждом из состояний $P_1, P_2, P_3, \dots, P_N$ (причём $\sum_{i=1}^N P_i = 1$ и $P_i \leq 1$), то, согласно теореме Шеннона, энтропия системы $H(X)$ равна

$$H(X) = -K_0 \sum_{i=1}^N P_i \log_a P_i.$$

Здесь коэффициент K_0 и основание логарифма a определяют систему единиц измерения количества информации. Логарифмическая мера информации была предложена Хартли для представления технических параметров систем связи как более удобная и более близкая к восприятию человеком, который привык к линейным сравнениям с принятыми эталонами. Например, каждый чувствует, что две однотипные дискеты должны обладать вдвое большей емкостью, чем одна, а два идентичных канала связи должны иметь удвоенную пропускную способность.

Знак минус поставлен для того, чтобы значение энтропии было положительным, так как $P_i \leq 1$ и логарифм в этом случае отрицательный.

Если все состояния системы равновероятны, то есть $P_i = 1/N$, то ее энтропия

$$H(X) = -K_0 \sum_{i=1}^N \frac{1}{N} \log_a \frac{1}{N} = K_0 \log_a N$$

Энтропия H обладает рядом интересных свойств. Вот некоторые из них.

1. $H=0$ только тогда, когда все вероятности P_i , кроме одной, равны нулю, а эта единственная вероятность равна единице. Таким образом, $H=0$ только в случае полной определенности состояния системы.

2. При заданном числе состояний системы N величина H максимальна и равна $K_0 \log_a N$, когда все P_i равны.

Определим единицы измерения количества информации с помощью выражения для энтропии системы с равновероятными состояниями.

Пусть система имеет два равновероятных состояния, то есть $N=2$. Будем считать, что снятие неопределенности о состоянии такой системы дает одну единицу информации, так как при полном снятии неопределенности энтропия количественно равна информации $H=I$. Тогда

$$1 = K_0 \log_a 2$$

Очевидно, что правая часть равенства будет тождественно равна единице информации, если принять $K_0 = 1$ и основание логарифма $a = 2$. В общем случае, при N равновероятных состояний количество информации будет

$$I = \log_2 N$$

Эта формула получила название формулы Хартли и показывает, что количество информации, необходимое для снятия неопределенности о системе с равновероятными состояниями, зависит лишь от количества этих состояний.

Информация о состояниях системы передается получателю в виде сообщений, которые могут быть представлены в различной синтаксической форме, например, в виде кодовых комбинаций, использующих m -различных символов и n разрядов, в каждом из которых может находиться любой из символов. Если код не избыточен, то каждая кодовая комбинация отображает одно из состояний системы. Количество кодовых комбинаций будет $N=m^n$. Подставив это выражение в формулу для I , получим

$$I = n \log_i m$$

Если код двоичный, т.е. используется лишь два символа (0 или 1), то $m=2$ и $I = n$.

В этом случае количество информации в сообщении составит n -двоичных единиц.

Эти единицы называют *битами*. **Binary digit (bit)**- двоичная цифра.

При использовании в качестве основания логарифма числа десять единицы измерения информации могут быть десятичными или *дитами*. Так как

$\log_2 N = \log_{10} N / \log_{10} 2 = 3,33 \log_{10} N$ то десятичная единица составляет примерно 3,33 бита.

Иногда удобно применять натуральное основание логарифма e . В этом случае получающиеся единицы информации называются натуральными или *натами*. Переход от основания a к основанию b требует лишь умножения на $\log_b a$.

Введенная количественная статистическая мера информации широко используется в теории информации для оценки собственной, взаимной, условной и других видов информации. Рассмотрим в качестве примера собственную информацию.

Под *собственной информацией* будем понимать информацию, содержащуюся в данном конкретном сообщении. А конкретное сообщение, как указывалось, дает получателю информацию о возможности существования конкретного состояния системы. Тогда количество собственной информации, содержащееся в сообщении X_i , определяется как

$$I(X_i) = -\log_2 P(X_i)$$

1. Собственная информация неотрицательна.

2. Чем меньше вероятность возникновения сообщения, тем больше информации оно содержит. Именно поэтому неожиданные сообщения так воздействует на психику человека, что содержащееся в них большое количество информации создает информационный психологический удар, иногда приводящий к трагическим последствиям.

3. Если сообщение имеет вероятность возникновения, равную единице, то информация, содержащаяся в нем равна нулю, т.к. заранее известно, что может прийти только это сообщение, а значит, ничего нового потребитель информации не получает.

4. Собственная информация обладает свойством аддитивности, т.е. количество собственной информации нескольких независимых

сообщений равно сумме собственных информаций сообщений. Например, для собственной информации двух сообщений X_i и Y_i может быть записано:

$$I(X_i, Y_i) = -\log_2 P(X_i) - \log_2 P(Y_i) = I(X_i) + I(Y_i).$$

Следует еще раз отметить, что статистический подход к количественной оценке информации был рассмотрен для дискретных систем, случайным образом переходящих из состояния в состояние, и, следовательно, сообщение об этих состояниях так же возникает случайным образом. Кроме того, статистический метод определения количества информации практически не учитывает семантического и прагматического аспектов информации.

2. Семантический подход оценки информации.

Семантический подход определения количества информации является наиболее трудно формализуемым и до сих пор окончательно не определившимся. Наибольшее признание для измерения смыслового содержания информации получила *тезаурусная* мера, предложенная Ю.И. Шнейдером. Идеи тезаурусного метода были сформулированы еще основоположником кибернетики Н. Винером. Для понимания и использования информации ее получатель должен обладать определенным запасом знаний.

Если индивидуальный тезаурус потребителя S_{II} отражает его знания о данном предмете, то количество смысловой информации I_C , содержащееся в некотором сообщении, можно оценить степенью изменения этого тезауруса, произошедшего под воздействием данного сообщения. Очевидно, что количество информации I_C нелинейно зависит от состояния индивидуального тезауруса пользователя, и хотя смысловое содержание сообщения \bar{s} постоянно, пользователи, имеющие отличающиеся тезаурусы, будут получать неодинаковое количество информации.

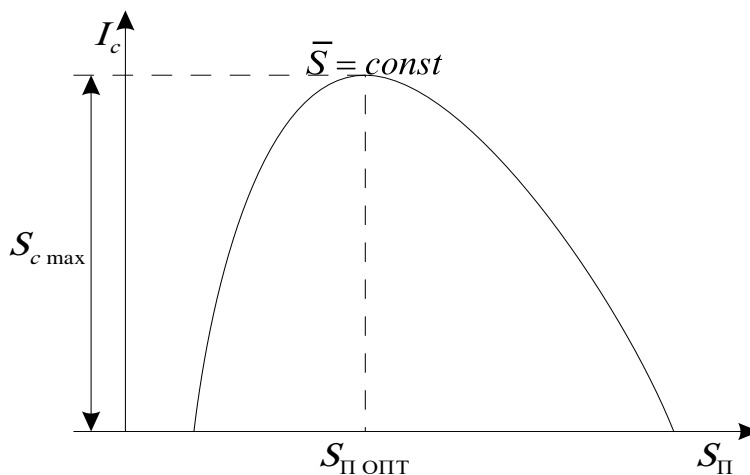
В самом деле, если индивидуальный тезаурус получателя информации близок к нулю ($S_{II} \approx 0$), то в этом случае и количество воспринятой информации равно нулю $I_C = 0$.

Иными словами, получатель не понимает принятого сообщения и, как следствие, для него количество воспринятой информации равно нулю. Такая ситуация эквивалентна

прослушиванию сообщения на неизвестном иностранном языке. Несомненно, сообщение не лишено смысла, однако оно непонятно, а значит, не имеет информативности.

Количество семантической информации I_C в сообщении также будет равно нулю, если пользователь информации абсолютно все знает о предмете, т.е. его тезаурус стремиться к бесконечности, и сообщение не дает ему ничего нового.

Интуитивно мы чувствуем, что между этими полярными значениями тезауруса пользователя существует некоторое оптимальное значение, $S_{\text{Попт}}$, при котором количество информации I_C , извлекаемое из сообщения, становится для получателя максимальным. Эта функция зависимости количества информации I_C от состояния индивидуального тезауруса пользователя $S_{\text{П}}$ приведена на рисунке



Тезаурусный метод подтверждает тезис о том, что информация обладает свойством относительности и имеет, таким образом, относительную, субъективную ценность. Для того, чтобы объективно оценивать научную информацию, появилось понятие общечеловеческого тезауруса, степень изменения которого и определяла бы значительность получаемых человечеством новых знаний.

$$I_{II} = \log \frac{P_1}{P_0}$$

3. Прагматический подход оценки информации.

Прагматический подход определяет количество информации как меры, способствующей достижению поставленной цели. Одной из первых работ, реализующей этот подход, явилась статья А.А. Харкевича. В ней он предлагал принять за меру ценности информации количество информации, необходимое для достижения поставленной цели. Этот подход базируется на статистической теории Шеннона и рассматривает количество информации как приращение вероятности достижения цели. Так, если принять вероятность достижения цели до получения информации равным P_0 , а после её получения P_1 , то прагматическое количество информации I_{II} определяется как

$$I_{II} = \log \frac{P_1}{P_0}$$

Если основание логарифма сделать равным двум, то I_{II} будет измеряться в битах, как и при статистическом подходе.

При оценке количества информации в семантическом и прагматическом аспектах необходимо учитывать и временную зависимость информации. Дело в том, что информация имеет свойство стареть, т.е. ее ценность со временем падает, и важно использовать ее в момент наибольшей ценности.

4. Структурный подход оценки информации.

Структурный подход связан с проблемами хранения, реорганизации и извлечения информации, и по мере увеличения объемов накапливаемой в компьютерах информации приобретает все большее значение.

При структурном подходе абстрагируются от субъективности, относительности ценности информации и рассматривают логические и физические структуры организации информации. С изобретением компьютеров появилась возможность хранить на машинных носителях громадные объемы информации. Но для ее эффективного использования необходимо определить такие структуры организации информации, чтобы существовала

возможность быстрого поиска, извлечения, записи, модификации информационной базы.

При машинном хранении структурной единицей информации является один байт, содержащий восемь бит (двоичных единиц информации). Менее определенной, но так же переводимой в байты, является неделимая единица информации – реквизит.

Реквизиты объединяются в показатели, показатели - в записи, записи - в массивы, из массивов создаются комплексы массивов, а из комплексов - информационные базы. Структурная теория позволяет на логическом уровне определить оптимальную структуру информационной базы, которая затем с помощью определенных средств реализуется на физическом уровне - уровне технических устройств хранения информации. От выбранной структуры хранения зависит такой важный параметр, как время доступа к данным, т.е. структура влияет на время записи и считывания информации, а значит и на время создания и реорганизации информационной базы.

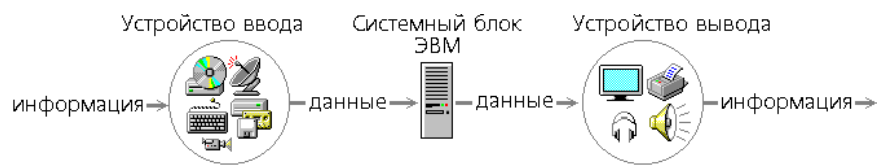
Информационная база совместно с системой управления базой данных (СУБД) формирует автоматизированный банк данных.

Значение структурной теории информации растет при переходе от банков данных к банкам знаний, в которых информация подвергается еще более высокой степени структуризации.

После преобразования информации в машинную форму, ее аналитический и прагматический аспекты как бы уходят в тень, и дальнейшая обработка информации происходит по «машинным законам» одинаковым для информации любого смыслового содержания. Информация в машинном виде, т. е. в форме электрических, магнитных и т.п. сигналов и состояний, носит название *данных*. Для того, чтобы понять их смысловое содержание, необходимо данные снова преобразовать в информацию

Преобразования «информация – данные» производятся в устройствах ввода-вывода ЭВМ.

Рисунок ниже иллюстрирует данный процесс.



Лекция 3. Понятие систем.

1. Основные определения.
2. Общая классификация систем.
3. Классификация систем по временной зависимости.
4. Классификация систем по сложности.

1. Основные определения.

По-гречески система (*systema*) - это целое, составленное из частей. Другими словами *система* - есть совокупность элементов, взаимосвязанных друг с другом и таким образом образующих определённую целостность.

Количество элементов, из которых состоит система, может быть любым, важно, чтобы они были между собой взаимосвязаны. Примеры систем - техническое устройство, состоящее из узлов и деталей; живой организм, состоящий из клеток; коллектив людей; предприятие; государство и т.д. Лекционная аудитория с лектором и студентами - система; каждый студент - тоже система; оборудование аудитории - система; отдельный стол - тоже система. А вот ножка стола - уже не система. Но это с точки зрения макропредставлений. Если же рассматривать ножку стола с точки зрения микропредставлений, то это тоже система, образуемая совокупностью молекул и атомов.

Из вышеперечисленных примеров ясно, что системы очень разнообразны, но все они имеют ряд общих свойств и понятий.

Элемент системы - часть системы, выполняющая определённую функцию (лектор читает лекцию, студенты её слушают и конспектируют, и т.д.). Элемент системы может быть сложным, состоящим из взаимосвязанных частей, т.е. тоже представлять собой систему. Такой сложный элемент часто называют *подсистемой*.

Организация системы - внутренняя упорядоченность и согласованность взаимодействия элементов системы. Организация системы проявляется, например, в ограничении разнообразия состояний элементов в рамках системы (во время лекции не играют в волейбол).

Структура системы - совокупность внутренних устойчивых связей между элементами системы, определяющая её основные

свойства. Например, в иерархической структуре отдельные элементы образуют соподчиненные уровни и внутренние связи образованы между этими уровнями.

Целостность системы - принципиальная несводимость свойств системы к сумме свойств её элементов. В то же время свойства каждого элемента зависят от его места и функции в системе. Так, если вернуться к примеру с лекцией, то рассматривая отдельно свойства лектора, студентов, предметов оборудования аудитории и т.д., нельзя однозначно определить свойства системы, где эти элементы будут совместно использоваться.

Управление - одно из центральных понятий в изучении систем.

Управление можно определить как функцию системы, обеспечивающую либо сохранение её основных свойств, либо её развитие в направлении определённой цели. Следовательно, управление неразрывно связано с системой и без неё не существует.

2. Общая классификация систем.

Классификация систем, как и любая классификация, может производиться по различным признакам. В наиболее общем плане системы можно разделить на материальные и абстрактные.

Материальные системы представляют собой совокупность материальных объектов. Среди материальных систем можно выделить неорганические (технические, химические и т.п.), органические (биологические) и смешанные, содержащие элементы как неорганической, так и органической природы. Среди смешанных систем следует обратить особое внимание на человеко-машинные (эрготехнические) системы, в которых человек с помощью машин осуществляет свою трудовую деятельность.

Важное место среди материальных систем занимают социальные системы с общественными отношениями (связями) между людьми. Подклассом этих систем являются социально - экономические системы, в которых связи между элементами - это общественные отношения людей в процессе производства.

Абстрактные системы - это продукт человеческого мышления: знания, теории, гипотезы и т.п.

3. Классификация систем по временной зависимости.

По временной зависимости различают статические и динамические системы. В статических системах с течением времени состояние не изменяется, в динамических системах происходит изменение состояния в процессе её функционирования. Динамические системы с точки зрения наблюдателя могут быть детерминированными и вероятностными (стохастическими). В детерминированной системе состояние её элементов в любой момент времени полностью определяется их состоянием в предшествующий или последующий моменты времени. Иначе говоря, всегда можно предсказать поведение детерминированной системы. Если же поведение предсказать невозможно, то система относится к классу вероятностных (стохастических) систем. Любая система входит в состав большей системы. Эта большая система как бы окружает её и является для данной системы внешней средой. По тому, как взаимодействует система с внешней средой, различают *закрытые* и *открытые системы*. Закрытые системы не взаимодействуют с внешней средой, все процессы, кроме энергетических, замыкаются внутри системы. Открытые системы активно взаимодействуют с внешней средой, что позволяет им развиваться в сторону совершенствования и усложнения.

4. Классификация систем по сложности. Признаки больших систем.

По сложности системы принято делить на простые, сложные и большие (очень сложные).

Простая система - это система, не имеющая развитой структуры (например, нельзя выявить иерархические уровни).

Сложная система - система с развитой структурой и состоящая из элементов - подсистем, являющихся в свою очередь простыми системами.

Большая система - это сложная система, имеющая ряд дополнительных признаков: наличие разнообразных (материальных, информационных, денежных, энергетических) связей между подсистемами и элементами подсистем; открытость

системы; наличие в системе элементов самоорганизации; участие в функционировании системы людей, машин и природной среды.

Понятие большой системы было введено, как следует из приведённых выше признаков, для обозначения особой группы систем, не поддающихся точному и подробному описанию. Для больших систем можно выделить следующие основные признаки:

Наличие структуры, благодаря которой можно узнать, как устроена система, из каких подсистем и элементов состоит, каковы их функции и взаимосвязи, как система взаимодействует с внешней средой.

Наличие единой цели функционирования, т.е. частные цели подсистем и элементов должны быть подчинены цели функционирования системы.

Устойчивость к внешним и внутренним возмущениям. Это свойство подразумевает выполнение системой своих функций в условиях внутренних случайных изменений параметров и дестабилизирующих воздействий внешней среды.

Комплексный состав системы, т.е. элементами и подсистемами большой системы являются самые разнообразные по своей природе и принципам функционирования объекты.

Способность к развитию. В основе развития систем лежат противоречия между элементами системы. Снятие противоречий возможно при увеличении функционального разнообразия, а это и есть развитие.

Изучение, анализ и синтез больших систем производится на основе *системного подхода*, который предполагает учет основных свойств таких систем.

Лекция 4.

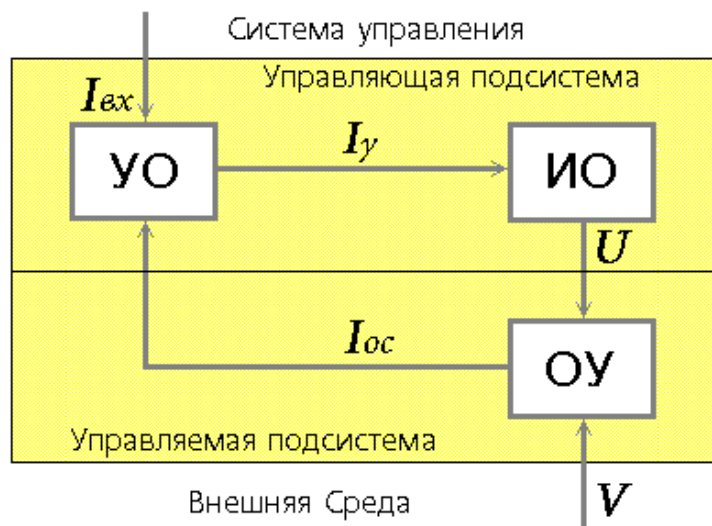
Управление в системах.

1. Структура системы управления.
2. Системы автоматического управления (САУ).
3. Автоматизированные системы управления (АСУ).

1. Структура системы управления.

Процессы управления присущи как живой, так и неживой природе. С управлением мы сталкиваемся в своей жизни повсеместно. Это и государство, которым управляют соответствующие структуры; это ЭВМ, работающая под управлением программы и т.д.

Совокупность объекта управления (ОУ), управляющего органа (УО) и исполнительного органа (ИО) образует систему управления, в которой выделяются две подсистемы: управляющая подсистема (УО и ИО) и управляемая подсистема (ОУ). На рисунке ниже представлена укрупнённая структурная схема системы управления, на которой выделены входящие в неё подсистемы.



В процессе функционирования этой системы управляющий орган (УО) получает информацию I_{oc} о текущем состоянии объекта управления (ОУ) и информацию I_{ex} о том, в каком состоянии *должен* находится объект управления. Отклонения объекта

управления от заданного состояния происходит под воздействием внешних возмущений (v). Результатом сравнения информации $I_{вх}$ и $I_{ос}$ в управляющем органе является возникновение управляющей информации $I_{у}$, которая воздействует на исполнительный орган (ИО). На основе информации $I_{у}$ исполнительный орган вырабатывает управляющее воздействие (U), которое ликвидирует отклонение в объекте управления.

Наиболее сложным звеном в системе управления является управляющий орган. Здесь степень сложности определяется количеством выполняемых функций, т.е. управляющий орган должен уметь производить наибольшее разнообразие действий. Это естественно, т.к. на любое состояние объекта управления управляющий орган должен отреагировать соответствующим образом, своевременно обработав поступившую в него информацию и выработав управляющую информацию.

Как видно из структурной схемы управления для её функционирования необходима *информация*. На приведённой схеме изображены три её потока: $I_{вх}$, $I_{ос}$ и $I_{у}$. Информация $I_{вх}$ сообщает управляющему органу о множестве возможных состояний объекта управления и управляющего органа, а также о том, в каком из состояний должен находиться объект управления при заданных внешних условиях. Информация $I_{ос}$ - это информация *обратной связи*. Понятие обратной связи является фундаментальным в теории управления. В общем случае под обратной связью понимают передачу воздействия с выхода какой-либо системы обратно на её вход. В системах управления обратная связь является информационной, и с её помощью в управляющую подсистему поступает информация о текущем состоянии управляемой подсистемы. Третий информационный поток $I_{у}$ - это информация, возникшая в результате обработки в управляющем органе информации $I_{вх}$ и $I_{ос}$ и управляющая работой исполнительного органа (ИО).

Очень важной компонентой входной информации $I_{вх}$ является информация о *цели управления*, ибо управление *бессмысленно*, если не направлено на достижение определённой *цели*. Если управление наилучшим образом соответствует поставленной цели, то такое управление называется *оптимальным*. Критерием оптимальности управления является некоторая, количественно измеряемая величина, отражающая цель управления.

Математическая запись критерия оптимальности носит название *целевой функции*. При *оптимальном управлении* значение целевой функции достигает экстремума (максимума или минимума в зависимости от критерия оптимальности).

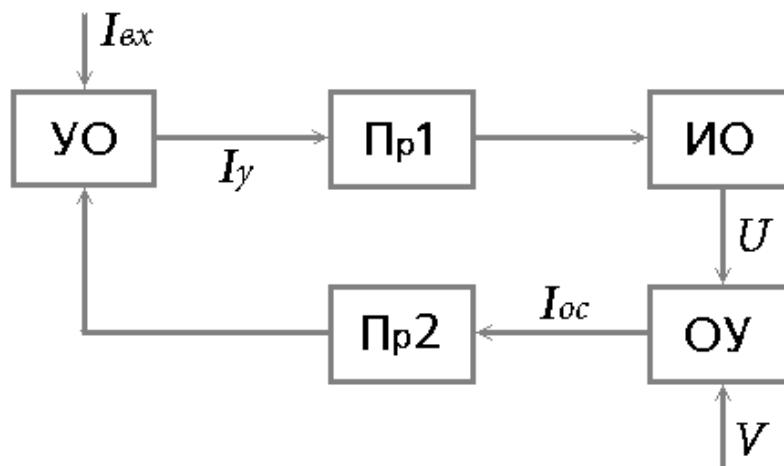
Ярко выраженный целевой информационный характер управления подтверждается кибернетическим его определением: *управление есть процесс целенаправленной переработки информации*.

2. Системы автоматического управления (САУ).

В зависимости от того, в какой системе (простой, сложной, большой) производится управление, различают системы автоматического управления (САУ) и автоматизированные системы управления (АСУ).

Автоматическое управление осуществляется, как правило, в простых системах, в которых заранее известны описание объекта управления и алгоритм управления им. По принципу управления системы автоматического управления могут быть *разомкнутыми* и *замкнутыми*. В *разомкнутых системах* измеряется возмущение, отклоняющее объект от заданного состояния, и вырабатывается воздействие, компенсирующее возникшее возмущение. Такая система не способна длительное время управлять неустойчивым объектом. В *замкнутых системах* реализуется идея обратной связи, благодаря которой информация об отклонении управляемого объекта от заданного состояния позволяет выработать воздействие, возвращающее объект в это состояние.

Благодаря тому, что поведение объекта и алгоритм управления строго заданы, системы автоматического управления могут работать автономно, без участия человека (хотя, конечно, их создание и наблюдение за их функционированием невозможно без человека). На рисунке ниже приведена упрощенная структурная схема замкнутой системы автоматического управления.



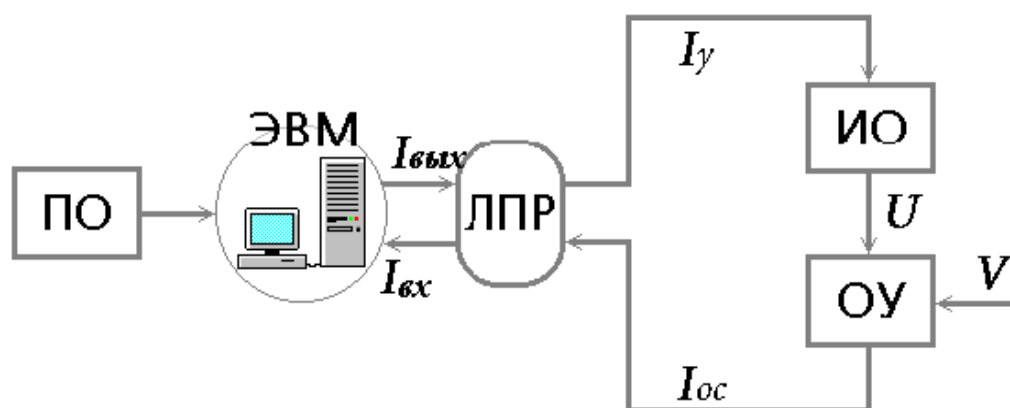
Упрощенная структурная схема замкнутой САУ

Как правило, САУ используются в технических системах, и в качестве управляющего органа (УО) используется компьютер, который с помощью программы (для него это I_{ex}), выдаёт результат обработки информации, обычно физический сигнал. Это - сигнал управления (I_y), который через преобразователь (Пр1) приводит в действие исполнительный орган (ИО), возвращающий объект управления (ОУ) в заданное программой компьютера состояние. Состояние ОУ, меняющееся под воздействием внешних возмущений v , определяет значение сигнала обратной связи (I_{oc}), которое через преобразователь (Пр2) поступает в компьютер (УО). Преобразователи необходимы для изменения уровней или природы проходящих через них сигналов, т.к. элементы системы могут быть различны по своей физической сути.

3. Автоматизированные системы управления (АСУ).

С ростом и усложнением производства объекты управления приобретают характер сложных и больших систем, имеющих большое число элементов и подсистем, связи между которыми не всегда ясны, а критерии функционирования не обладают достаточной чёткостью. В этих условиях использовать результаты теории автоматического управления в полной мере не удаётся, и в контур управления, помимо человека - оператора ЭВМ, действующего по заданным алгоритмам, включается *лицо*,

принимаящее решения (ЛПР). Наличие ЛПР в контуре управления является отличительной чертой автоматизированных систем управления. Автоматизированное управление применяется в том случае, если нет возможности реализовать автоматическое управление. Рисунок ниже иллюстрирует упрощенную структурную схему АСУ:



Упрощенная структурная схема АСУ.

Как видно из рисунка, ЛПР, получив информацию обратной связи $I_{ос}$, осведомляющую его о состоянии объекта управления (ОУ), обращается к ЭВМ (поток $I_{вх}$), имеющей определенное программное обеспечение (ПО) и вырабатывающей рекомендации к принятию решения (поток $I_{вых}$). На основе анализа предложенных ЭВМ альтернатив ЛПР принимает решение, которое в виде управляющей информации (I_y) поступает в исполнительный орган (ИО), переводя его в необходимое состояние.

Например, министр (это - ЛПР), получив информацию о состоянии отрасли (это - ОУ), после обработки всей нужной информации на ЭВМ и просчета наборов вариантов поведения в сложившейся ситуации, принимает решение, которое реализуется аппаратом министерства (это - ИО) в управляемой отрасли производства.

Лекция 5.

Моделирование в информационных системах.

1. Информационные модели.
2. Человек и информационная технология.
3. Процесс принятия решения.

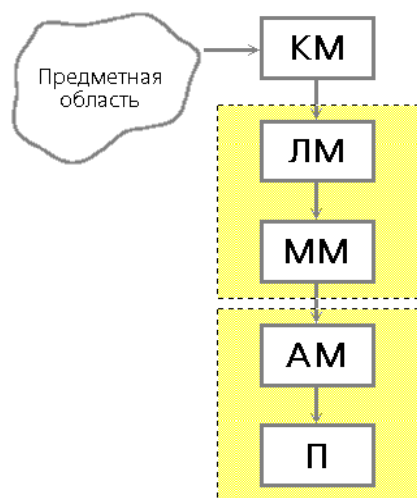
1. Информационные модели.

Важным инструментом исследования систем, да и не только систем, является метод моделирования. Суть этого метода состоит в том, что исследуемый объект заменяется его моделью, т.е. некоторым другим объектом, сохраняющим основные свойства реального объекта, но более удобным для исследования или использования.

Различают *физические* и *абстрактные модели*. При изучении автоматизированных информационных технологий наибольшее распространение получили абстрактные *информационные модели*.

Математические модели представляют собой формализованное описание на языке математики исследуемых объектов и отображают в виде математических отношений взаимосвязи параметров этих объектов. Наличие достаточно полной математической модели объекта позволяет разработать алгоритм управления этим объектом, т.е. создать *алгоритмическую модель*. Если для управления используется ЭВМ, то алгоритмическая модель преобразуется с помощью языков программирования в программу, управляющую работой ЭВМ, а через неё - объектом управления.

Информационная модель - это отражение предметной области в виде информации. *Предметная область* представляет собой часть реального мира, которая исследуется или используется. Отображение предметной области в информационных технологиях представляется информационными моделями нескольких уровней, показанных на рисунке ниже.



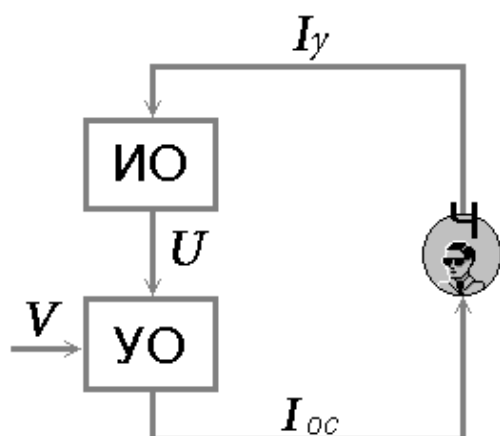
Концептуальная модель (КМ) обеспечивает интегрированное представление о предметной области (например, технологические карты, техническое задание, план производства и т.п.) и имеет слабо формализованный характер. *Логическая модель (ЛМ)* формируется из концептуальной путем выделения конкретной части (скажем, подлежащей управлению), её детализации и формализации. Логическая модель, формализующая на языке математики взаимосвязи в выделенной предметной области, называется *математической моделью (ММ)*. С помощью математических методов математическая модель преобразуется в *алгоритмическую модель (АМ)*, задающую последовательность действий, реализующих достижение поставленной цели управления. На основе АМ создаётся машинная программа (П), являющаяся той же алгоритмической моделью, только представленной на языке, понятном ЭВМ.

Выделение информационных моделей разных уровней абстракции позволяет разделить сложный процесс отображения «предметная область – программа» на несколько итеративных более простых отображений.

2. Человек и информационная технология.

При производстве продуктов труда человек всегда управляет орудиями труда в процессе их воздействия на предмет труда. Скажем, вскапываете вы грядки. В этом случае предмет труда -

земля, орудие труда - лопата, продукт труда - вскопанная грядка. Естественно, перед началом копки вы составляете концептуальную модель, содержащую информацию о том, где, в каком месте огорода будет располагаться будущая грядка, какой она будет ширины, глубины, нужно ли разбивать комья земли и т.д. Когда вы копаете, т.е. воздействуете лопатой (орудием труда) на землю (предмет труда), вы подсознательно сравниваете получающийся результат с хранимой в памяти концептуальной моделью и, в зависимости от результата сравнения, копаете глубже или немного левее, т.е. управляете орудием труда. Не трудно видеть, что перед нами система управления, в которой объектом управления является земля, исполнительным органом - лопата вместе с мускулами человека, управляющим органом - человеческий мозг. Т.е. действует классическая схема системы управления (рисунок ниже).

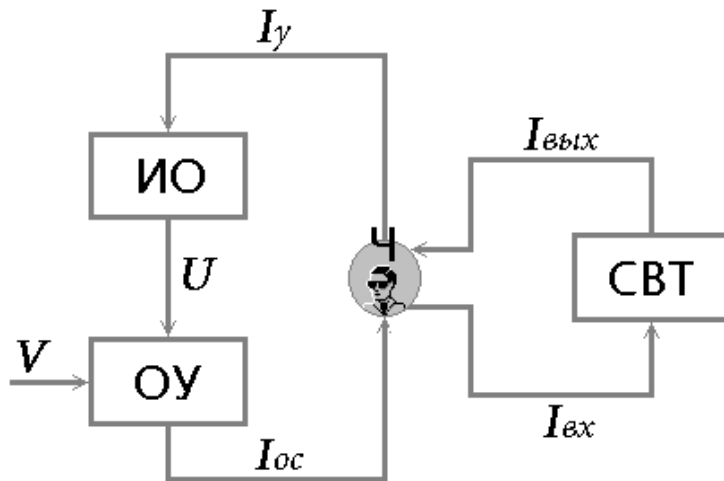


Человек в элементарной системе управления.

С усложнением производства, т.е. объектов управления, и их концептуальных моделей, объёмы информации I_{oc} возрастают и человеческая возможность их переработки в необходимом темпе исчерпываются. Тогда на помощь человеку приходят технические средства ускорения переработки информации, как правило, средства вычислительной техники (рис. 1.10.).

Возникает, таким образом, самостоятельный дополнительный информационный контур, помогающий человеку быстрее обработать осведомляющую информацию I_{oc} и выработать управляющую информацию I_y . Появление контура дополнительной обработки информации (помимо человека) на средствах

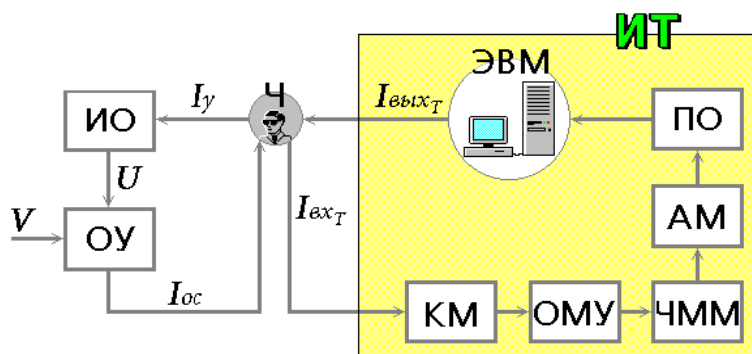
вычислительной техники (СВТ) и есть начало возникновения информационной технологии. Рисунок ниже иллюстрирует данную ситуацию.



Появление контура Информационной технологии.

Совершенствование ЭВМ, программного обеспечения, математических методов и моделей, позволили создать более сложные информационные системы, в которых чётко обозначился контур информационной технологии.

На рисунке ниже проиллюстрирована информационная технология в более сложной информационной системе.



Информационная технология (ИТ) в сложной Информационной системе (ИС).

В общем случае информационная технология состоит из информационных моделей разного уровня абстракции и ЭВМ. На вход ИТ поступает информация от человека $I_{вхТ}$, формируемая на основе информации $I_{ос}$ от объекта управления. Информация $I_{вхТ}$ сравнивается с концептуальной моделью (КМ) объекта управления. Реакция на результат сравнения определяется общей математической моделью управления (ОМУ), декомпозированной на частные математические модели (ЧММ). Набор ЧММ описывает возможные состояния объекта управления (ОУ) и тактику управления в этих состояниях. Эта тактика реализуется через алгоритмические модели, формализованные в программы (ПО) для ЭВМ. В результате ЭВМ выдаёт информацию $I_{выхТ}$, представляющую собой рекомендации по управлению ОУ в данной ситуации.

Таким образом, человек в автоматизированной системе управления является центральным и объединяющим звеном двух контуров: собственно управления (человек – исполнительный орган – объект управления) и информационной технологии (человек – ИТ), как изображено на рисунке выше.

3. Процесс принятия решения.

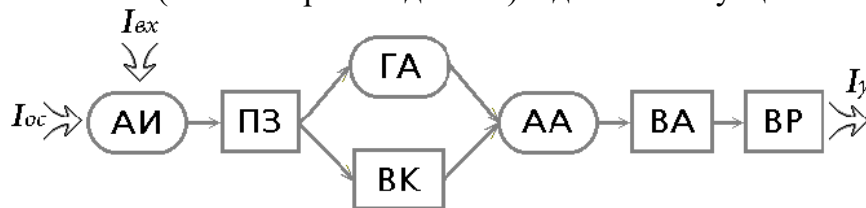
В автоматизированной системе управления, несмотря на наличие контура информационной технологии, ответственность за принятое управляющее решение возлагается на человека - лицо, принимающее решение (ЛПР). Другими словами, решение принимает человек, а информационная технология помогает ему в этом.

Когда ученые выделили из процесса управления стадию принятия решения, то вначале казалось, что для полной автоматизации достаточно разработать математическую модель и реализовать ее в ЭВМ. И тогда АСУ превращается в САУ. Однако, как оказалось, процесс принятия решения человеком очень сложен. Иногда в этот процесс включаются такие механизмы, которые невозможно предусмотреть и, тем более, формализовать. При принятии решения человек может учитывать и такие аспекты, как мораль, традиции, человеческие взаимоотношения. Вот почему при управлении социально-экономическими системами (иначе - при

организационно-экономическом управлении), процесс принятия решения не может быть осуществлен без человека.

На рисунке показана взаимосвязь фаз принятия решения.

Человек, на основе анализа (АИ) осведомляющей информации $I_{ос}$ от объекта управления и информации $I_{вх}$ от концептуальной модели объекта управления, производит постановку задачи (ПЗ), решение которой должно позволить наилучшим образом управлять объектом (скажем производством) в данной ситуации.



Фазы процесса принятия решения

Однако, решений (альтернатив) всегда несколько (если решение всегда одно, то проблемы выбора не существует, а, значит, и теряет смысл сам процесс принятия решения). Поэтому, далее идёт фаза генерации альтернатив (ГА), то есть выдвижение возможных решений задачи (альтернатива – чередоваться, каждая из исключаяющих друг друга возможностей). Как уже говорилось, управление всегда ведётся с определённой целью. Решение поставленной задачи должно согласовываться с общей целью управления и частной целью в данной ситуации. Поэтому выбрать альтернативу невозможно, если нет критерия выбора, отражающего цель управления. Т.о., следующая фаза - выбор критерия (ВК) решения поставленной задачи. На этапе анализа альтернатив (АА) производится их исследование по выбранному критерию, а далее - окончательный выбор одной из альтернатив (ВА), наилучшим образом удовлетворяющей критерию выбора. Выбранная альтернатива дополнительно анализируется и выдаётся окончательное решение (ВР), принимающее в организационных системах вид потока управляющей информации $I_{у}$.

Если рассматривать фазы принятия решения с точки зрения возможности их автоматизации на базе информационной технологии, то в настоящее время, пожалуй, только фазы анализа информации (АИ), генерации альтернатив (ГА) и анализа альтернатив (АА) по выбранному критерию удаётся

автоматизировать в достаточной мере (на рисунке выше изображены овалами). Для этого необходимо, чтобы в ЭВМ находились модели поставленной задачи, с помощью которых возможно было бы быстро просчитать результаты решения по различным альтернативам, исходным данным и критериям. Конечно, для этого желательно, чтобы ЛПР умело использовать средства информационной технологии. В противном случае приходится иметь штат системщиков, аналитиков и т.п. Развитие программно - аппаратных средств ИТ с каждым годом приводит к всё большему упрощению взаимодействия человека с ЭВМ и, таким образом, уменьшает число посредников диалога, что ускоряет и повышает качество принимаемых решений. Большое значение для принятия быстрого и верного решения имеет автоматизация фазы обработки и анализа информации (АИ), поступающей с потоками I_{oc} и I_{vx} . Для принятия решения всегда может потребоваться дополнительная информация, не содержащаяся в потоках I_{oc} и I_{vx} . В этих случаях важную роль принимает информационное обеспечение ЛПР, которое в целях оперативности должно быть организовано с помощью средств ИТ (базы и банки данных).

Лекция 6.

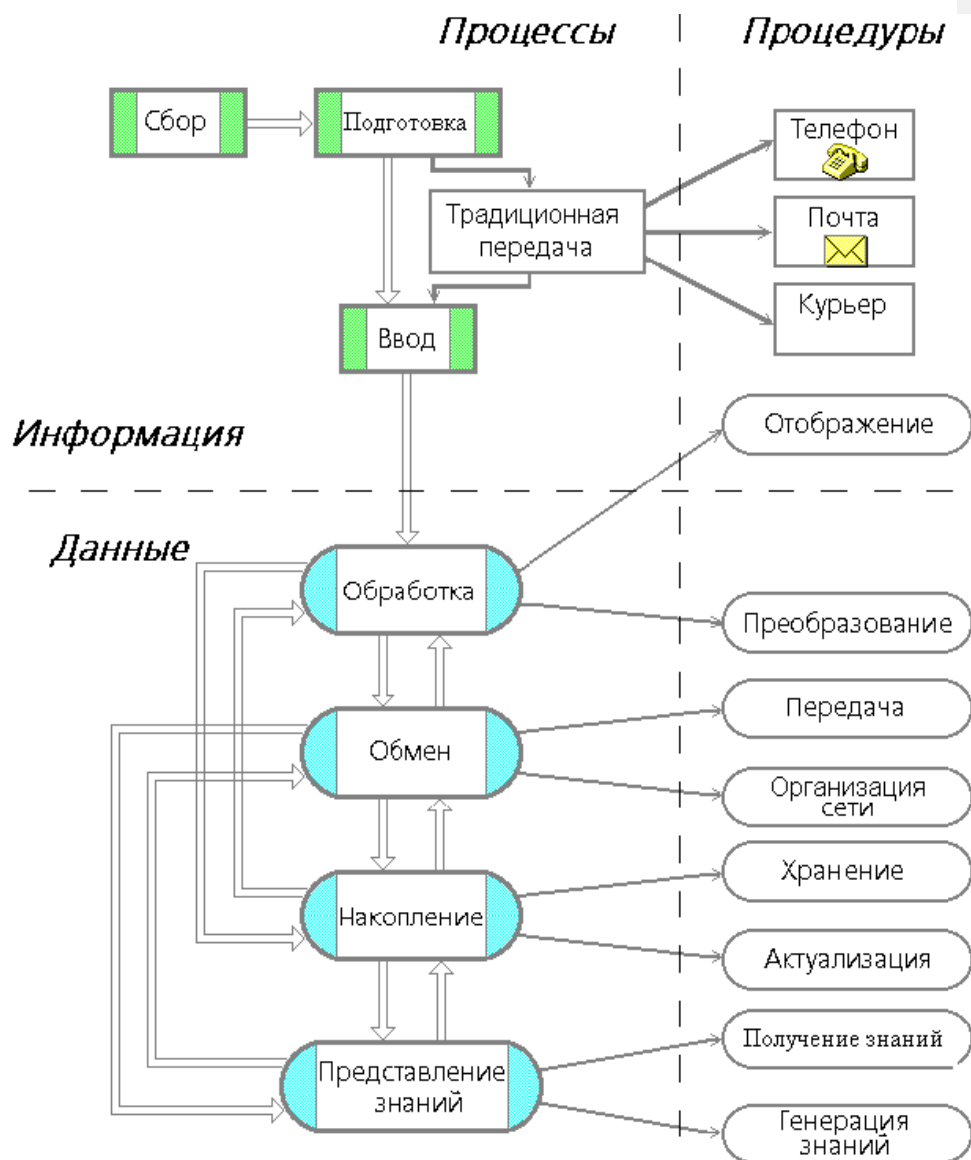
Концептуальный уровень базовой информационной технологии.

- 1. Концептуальная модель базовой информационной технологии.**
- 2. Технология переработки информации концептуального уровня базовой информационной технологии.**
- 3. Информационные процессы переработки данных концептуального уровня базовой информационной технологии.**

- 1. Концептуальная модель базовой информационной технологии.**

При производстве информационного продукта исходный информационный ресурс в соответствии с поставленной задачей подвергается в определенной последовательности различным преобразованиям. Динамика этих преобразований отображается в протекающих при этом информационных процессах. Таким образом, информационный процесс - это процесс преобразования информации. В результате него информация может изменить и содержание, и форму представления, причем, как в пространстве, так и во времени.

Фазы преобразования информации в информационной технологии достаточно многочисленны, и простое их перечисление может привести к потере ощущения целостности технологической системы (за деревьями не увидеть леса). Однако, если провести структуризацию технологии, выделив такие крупные структуры как процессы и процедуры, то концептуальная модель базовой информационной технологии может быть представлена схемой, представленной на рисунке ниже.



Концептуальная модель базовой информационной технологии

На этой схеме в левой части изображены блоки информационных процессов, в правой - блоки процедур. Блок в виде прямоугольника изображает процесс или процедуру, в которых преобладают ручные или традиционные операции. Овальная форма блоков соответствует автоматическим операциям,

производимым с помощью технических средств (ЭВМ и средств передачи данных).

В верхней части схемы информационные процессы и процедуры осуществляют преобразование информации, имеющей человеческую форму представления, т.е. ярко выраженное смысловое содержание. Синтаксический аспект информации находится здесь на втором плане. В этом случае говорят о преобразовании собственно *информации*.

В нижней части схемы производится преобразование *данных*, т.е. информации, представленной в машинном виде. И на этом уровне представления преобладает синтаксический аспект информации.

2. Технология переработки информации концептуального уровня базовой информационной технологии.

Технология переработки информации начинается с формирования информационного ресурса, который после определенных целенаправленных преобразований должен превратиться в информационный продукт.

Формирование информационного ресурса (получение исходной информации) начинается с процесса *сбора* информации, которая должна в информационном плане отразить предметную область, т.е. объект управления или исследования (его характеристики, параметры, состояние и т.п.).

Собранная информация для ее оценки (по полноте, непротиворечивости, достоверности и т.д.) и последующих преобразований должна быть соответствующим образом *подготовлена* (осмыслена и структурирована, например, в виде таблиц).

После подготовки информация может быть передана для дальнейшего преобразования традиционными способами (с помощью телефона, почты, курьера и т.п.), а может быть подвергнута сразу процессу преобразования в машинные данные, т.е. процессу *ввода*. Процессы сбора, подготовки и ввода в информационной технологии организационно-экономических

систем по своей реализации являются в основном ручными (кроме процесса подготовки, который частично может быть автоматизированным). Процесс ввода преобразует информацию в данные, имеющие форму цифровых кодов, реализуемых на физическом уровне с помощью различных физических представлений (электрических, магнитных, оптических, механических и т.д.).

3. Информационные процессы переработки данных концептуального уровня базовой информационной технологии.

Следующие за вводом информационные процессы уже производят *преобразование данных* в соответствии с поставленной задачей. Эти процессы протекают в ЭВМ (или организуются ЭВМ) под управлением различных программ, которые и позволяют так организовать данные, что после вывода из ЭВМ результат обработки представляет собой наполненную смыслом информацию о результате решения поставленной задачи. При преобразованиях данных можно выделить четыре основных информационных процесса и соответствующих им процедур. Это *процессы обработки, обмена, накопления данных и представление знаний*.

Процесс *обработки* данных связан с *преобразованием* значений и структур данных, а также их преобразованием в форму, удобную для человеческого восприятия, т.е. *отображением*. Отображенные данные - это уже информация. Процедуры преобразования данных осуществляются по определенным алгоритмам и реализуются в ЭВМ с помощью набора машинных операций. Процедуры отображения переводят данные из цифровых кодов в изображение (текстовое или графическое) или звук.

Информационный процесс *обмена* предполагает обмен данными между процессами информационной технологии. Из схемы на рисунке выше видно, что процесс обмена связан взаимными потоками данных со всеми информационными процессами на уровне переработки данных. При обмене данными можно выделить два основных типа процедур. Это процедуры *передачи* данных по каналам связи и сетевые процедуры, позволяющие осуществить *организацию* вычислительной *сети*.

Процедуры передачи данных реализуются с помощью операции кодирования-декодирования, модуляции-демодуляции, согласования и усиления сигналов. Процедуры организации сети включают в себя в качестве основных операции по коммутации и маршрутизации потоков данных (трафика) в вычислительной сети. Процесс обмена позволяет с одной стороны передавать данные между источником и получателем информации, а с другой - объединять информацию многих ее источников.

Процесс *накопления* позволяет так преобразовать информацию в форме данных, что удастся ее длительное время хранить, постоянно обновляя, и при необходимости оперативно извлекать в заданном объеме и по заданным признакам. Процедуры процесса накопления, таким образом, состоят в организации *хранения* и *актуализации* данных. *Хранение* предполагает создание такой структуры расположения данных в памяти ЭВМ, которая позволила бы быстро и не избыточно накапливать данные по заданным признакам и не менее быстро осуществлять их поиск. В настоящее время ЭВМ имеет два основных вида запоминающих устройств оперативные (электронные) и внешние (на магнитных и оптических дисках). Их физическая природа и устройство различны, поэтому различаются и возможности по организации структур хранения данных. Можно выделить *операции* по организации *хранения* и поиска данных в *оперативной* и *внешней* памяти ЭВМ. В процессе накопления данных важной процедурой является их *актуализация*. Под актуализацией понимается поддержание хранимых данных на уровне, соответствующем информационным потребностям решаемых задач в системе, где организована информационная технология. Актуализация данных осуществляется с помощью операций добавления новых данных к уже хранимым, корректировки (изменения значений или элементов структур) данных и их уничтожения, если данные устарели и уже не могут быть использованы при решении функциональных задач системы.

Наконец, информационный процесс *представления знаний* включен в базовую информационную технологию как один из основных информационных процессов, поскольку высшим продуктом информационной технологии является знание. Формирование знания как высшего информационного продукта до недавнего времени являлось (да в основе своей является и сейчас) прерогативой человека. Однако, оказать помощь человеку при

решении не формализуемых или трудно формализуемых задач может автоматизированный процесс представления знаний. В этом процессе объединяются процедуры формализации знаний, их накопления в формализованном виде и формальной генерации (вывода) новых знаний на основе накопленных в соответствии с поставленной задачей. Вывод нового знания - это эквивалент решения задачи, которую не удастся представить в формальном виде. Таким образом, *процесс представления знаний* - это процесс, состоящий из процедур *получения* формализованных знаний и процедур *генерации* (вывода) новых знаний из полученных.

К сожалению, практическая реализация процесса представления знаний с помощью ЭВМ еще не достигла достаточно широкого применения в информационных технологиях. Это связано как с продолжающимися поисками форм представления знаний в теории искусственного интеллекта, так и практическими трудностями при создании баз знаний. Тем не менее, развитие теории искусственного интеллекта продолжается и в новом веке процесс представления знаний займет ключевое место в информационных технологиях.

В зависимости от решаемых информационной технологией задач удельный вес и взаимосвязь информационных процессов различны.

Лекция 7.

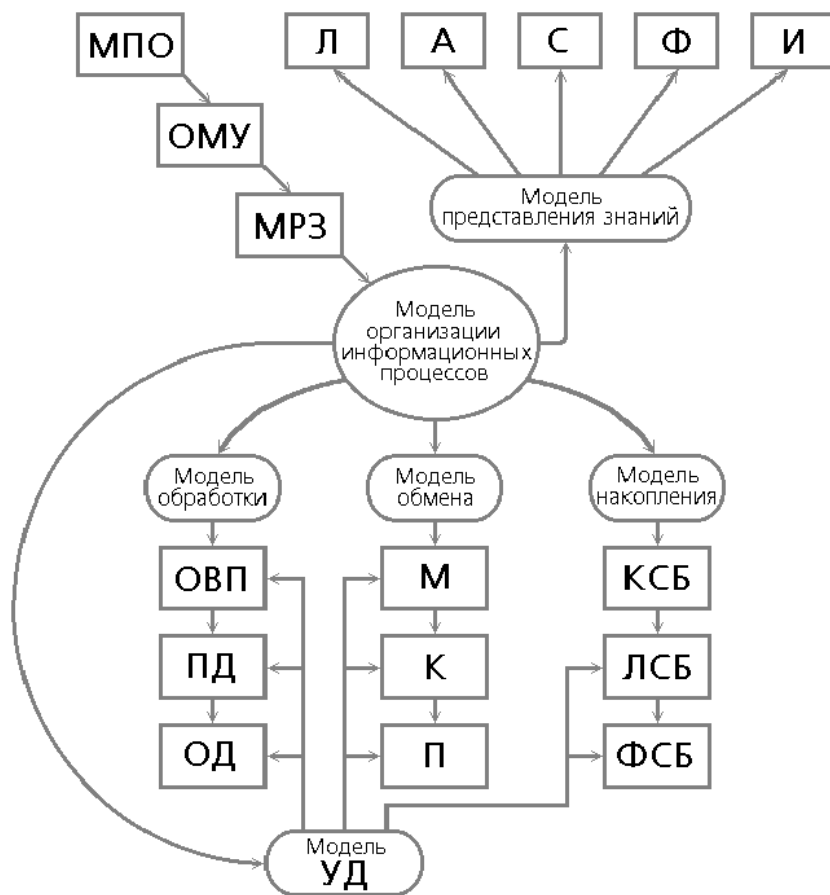
Логический и физический уровни базовой информационной технологии.

- 1. Комплекс моделей логического уровня базовой информационной технологии.**
- 2. Модель организации информационных процессов логического уровня базовой информационной технологии.**
- 3. Взаимосвязь подсистем физического уровня базовой информационной технологии.**

1. Комплекс моделей логического уровня базовой информационной технологии.

Логический уровень информационной технологии, представляется комплексом взаимосвязанных моделей, формализующих информационные процессы при технологических преобразованиях информации и данных. Формализованное (в виде моделей) представление информационной технологии позволяет связать параметры информационных процессов, а это означает возможность реализации управления информационными процессами и процедурами.

На рисунке ниже приведены состав и взаимосвязи моделей базовой информационной технологии. В зависимости от области применения и назначения информационной технологии модели информационных процессов конкретизируются, а некоторые могут и отсутствовать. Например, в настоящее время из-за отсутствия на потребительском рынке информационных технологий относительно недорогих, надежных и простых в эксплуатации интеллектуальных систем, процесс представления знаний в структуре организуемой информационной технологии может отсутствовать. Если, к примеру, информационная технология проектируется на не объединенных в сеть автоматизированных рабочих местах (АРМх), процесс обмена данными и соответственно его модели будут отсутствовать. Однако, наибольший эффект информационная технология дает тогда, когда в ее составе используется весь набор информационных процессов.



Состав моделей базовой информационной технологии

На основе *модели предметной области* (МПО), характеризующей объект управления, создается *общая модель управления* (ОМУ), а из нее вытекают *модели решаемых задач* (МРЗ). Так как решаемые задачи в информационной технологии имеют в своей основе различные информационные процессы, то на передний план выходит *модель организации информационных процессов*, призванная на логическом уровне увязать эти процессы при решении задач управления.

2. Модель организации информационных процессов логического уровня базовой информационной технологии.

При обработке данных формируются четыре основных информационных процесса: обработки, обмена, накопления и представления знаний.

Модель обработки данных включает в себя формализованное описание процедур организации вычислительного процесса, преобразования данных и отображения данных. Под *организацией вычислительного процесса* (ОВП) понимается управление использованием ресурсов компьютера (памяти, процессора, внешних устройств) при решении задач обработки данных. Эта процедура формализуется в виде алгоритмов и программ системного управления компьютером. Комплексы таких алгоритмов и программ получили название операционных систем. Операционные системы выступают в виде посредников между ресурсами компьютера и прикладными программами, организуя непрерывную и безошибочную их работу. Процедуры *преобразования данных* (ПД) на логическом уровне представляют собой алгоритмы и программы обработки данных и их структур. Сюда включаются стандартные процедуры, такие как сортировка, поиск, создание и преобразование статистических и динамических структур данных, а так же нестандартные процедуры, обусловленные алгоритмами и программами преобразования данных при решении конкретных информационных задач. Моделями процедур *отображения данных* (ОД) являются компьютерные программы преобразования данных, представленных машинными кодами, в воспринимаемую человеком информацию, несущую в себе смысловое содержание. В современных ЭВМ данные могут быть отражены в виде текстовой информации, в виде графиков, изображений, звука, с использованием средств мультимедиа, которые интегрируют в компьютере все основные способы отображения.

Модель обмена данными включает в себя формальное описание процедур, выполняемых в вычислительной сети: передачи (П), маршрутизации (М), коммутации (К). Именно эти процедуры и составляют информационный процесс обмена. Для

качественной работы сети необходимы формальные соглашения между ее пользователями, что реализуется в виде протоколов сетевого обмена. В свою очередь *передача* данных основывается на моделях кодирования, модуляции, каналов связи. На основе моделей обмена производится синтез системы обмена данными, при котором оптимизируется топология и структура вычислительной сети, метод коммутации, протоколы и процедуры доступа, адресации и маршрутизации.

Модель накопления данных формализует описание информационной базы, которая в компьютерном виде представляется базой данных. Процесс перехода от информационного (смыслового) уровня к физическому отличается трехуровневой системой моделей представления информационной базы: концептуальной, логической и физической схем. Концептуальная схема информационной базы (КСБ) описывает информационное содержание предлагаемой области, т.е. какая и в каком объеме информация должна накапливаться при реализации информационной технологии. Логическая схема информационной базы (ЛСБ) должна формализовано описать ее структуру и взаимосвязь элементов информации. При этом могут быть использованы различные подходы: реляционный, иерархический, сетевой. Выбор подхода определяет и систему управления базой данных (СУБД). А СУБД, в свою очередь, определяет физическую модель данных - физическую схему информационной базы (ФСБ), описывающую методы размещения данных и доступа к ним на машинных (физических) носителях информации.

В современных информационных технологиях формирование моделей предметной области и решаемых задач производится в основном человеком, что связано с трудностями формализации этих процессов. Но по мере развития теории и практики интеллектуальных систем становится возможным формализовать человеческие знания, на основе которых и формируется вышеуказанные модели.

Модель представления знаний, включенная в систему моделей информационной технологии, позволит проектировщику ИТ в автоматизированном режиме формировать из фрагментов модель предметной области, а так же модели решаемых задач. Наличие этих моделей поможет пользователю в заданной предметной области выбрать необходимую ему модель решаемой задачи и

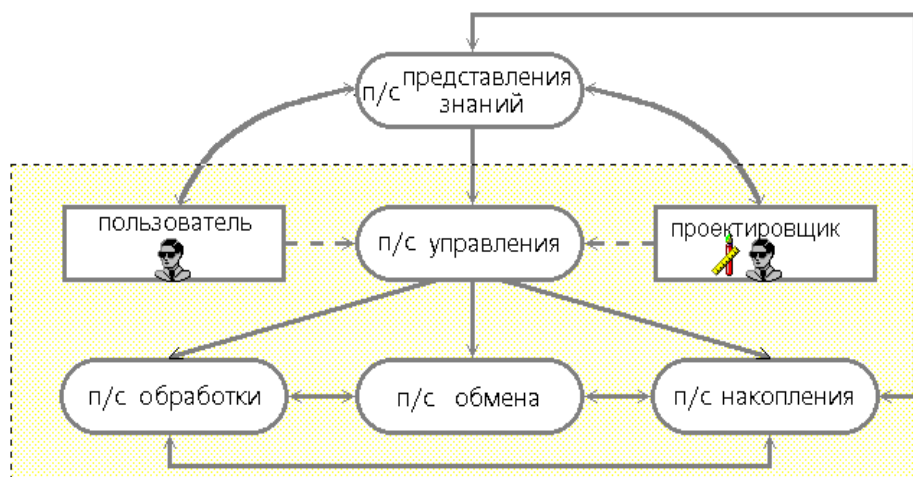
решить ее с помощью информационной технологии. Модель представления знаний может быть выбрана в зависимости от предметной области и вида решаемых задач. Сейчас практически используются такие модели как логические (Л), алгоритмические (А), фреймовые (Ф), семантические (С) и интегральные (И).

Взаимная увязка базовых информационных процессов, их синхронизация на логическом уровне осуществляется через модель управления данными (УД). Так как базовые информационные процессы оперируют данными, то управление данными - это управление процессами обработки, обмена и накопления. Управление процессом обработки данных означает управление организацией вычислительного процесса, преобразованиями и отображениями данных в соответствии с моделью организации информационных процессов, основанной на модели решаемой задачи. При управлении процессом обмена управлению подлежат процедуры маршрутизации и коммутации в вычислительной сети, а так же передачи сообщений по каналам связи. Управление данными в процессе накопления означает организацию физического хранения данных в базе и ее актуализацию, т.е. добавление данных, их корректировку и уничтожение. Кроме того, должны быть подчинены управлению процедуры поиска, группировок, выборок и т.п. На логическом уровне управление процессом накопления - это комплексы программ управления базами данных, получившими название систем управления базами данных (СУБД). Увеличение объемов информации, хранимых в базах данных, при переходе к распределенным базам и банкам данных, управление процессом накопления усложняется и не всегда поддается формализации. Поэтому в ИТ при реализации процесса накопления часто возникает необходимость в человеке - администраторе базы данных, который формирует и ведет модель накопления данных, определяя ее содержание и актуальное состояние.

3. Взаимосвязь подсистем физического уровня базовой информационной технологии.

Физический уровень базовой информационной технологии представляет ее программно-аппаратную реализацию. При этом

стремятся максимально использовать типовые технические средства и программное обеспечение, что существенно уменьшает затраты на создание и эксплуатацию ИТ. С помощью программно-аппаратных средств осуществляются базовые информационные процессы и процедуры в их взаимосвязи и подчинении единой цели функционирования. Таким образом, и на физическом уровне ИТ рассматривается как система, причем большая система, в которой выделяется несколько крупных подсистем. Это - подсистемы, реализующие на физическом уровне информационные процессы: *подсистема (п/с) обработки данных, подсистема обмена данными, подсистема накопления данных, подсистема управления данными и подсистема представления знаний*. С системой информационной технологии взаимодействуют *пользователь и проектировщик* системы. Состав и взаимосвязь подсистем ИТ приведены на рисунке ниже



Состав и взаимосвязь подсистем физического уровня базовой ИТ.

Лекция 8.

Подсистемы физического уровня базовой информационной технологии. Преобразование информации в данные.

- 1. Состав и функции подсистем физического уровня базовой информационной технологии.**
- 2. Процесс преобразования информации в данные.**
- 3. Методы контроля подготовленной информации и ее ввод.**

- 1. Состав и функции подсистем физического уровня базовой информационной технологии.**

Для выполнения функций подсистемы обработки данных используются электронные вычислительные машины различных классов. В настоящее время при создании автоматизированных информационных технологий применяются три основных класса ЭВМ:

на верхнем уровне – большие универсальные ЭВМ (по зарубежной классификации - мэйнфреймы), способные накапливать и обрабатывать громадные объемы информации и используемые как главные ЭВМ;

на среднем уровне – абонентские вычислительные машины (серверы);

на нижнем уровне – персональные компьютеры либо управляющие ЭВМ.

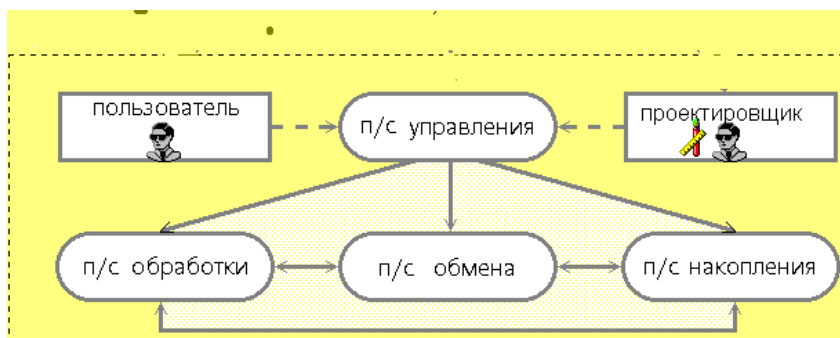
Обработка данных, т.е. их преобразование и отображение, производится с помощью программ решения задач в той предметной области, для которой создана информационная технология.

В подсистему обмена данными входят комплексы программ и устройств, позволяющих реализовать вычислительную сеть и осуществить по ней передачу и прием сообщений с необходимой скоростью и качеством. Физическими компонентами подсистемы обмена служат устройства приема - передачи (модемы, усилители, коммутаторы, кабели, специальные вычислительные комплексы, осуществляющие коммутацию, маршрутизацию и доступ к сетям). Программными компонентами подсистемы являются программы

сетевого обмена, реализующие сетевые протоколы, кодирование-декодирование сообщений и др.

Подсистема накопления данных реализуется с помощью банков и баз данных, организованных на внешних устройствах компьютеров и ими управляемых. В вычислительных сетях, помимо локальных баз и банков, используется организация распределенных банков данных и распределенной обработки данных. Аппаратно-программными средствами этой подсистемы являются компьютеры различных классов с соответствующим программным обеспечением.

Для автоматизированного формирования модели предметной области из ее фрагментов и модели решаемой информационной технологией задачи создается подсистема представления знаний. На стадии проектирования информационной технологии *проектировщик* формирует в памяти компьютера модель заданной предметной области, а так же комплекс моделей решаемых технологией задач. На стадии эксплуатации *пользователь* обращается к подсистеме знаний и, исходя из постановки задачи, выбирает в автоматизированном режиме соответствующую модель решения, после чего через подсистему управления данными включаются другие подсистемы информационной технологии. Реализация подсистем представления знаний производится, как правило, на персональных компьютерах, программирование которых осуществляется с помощью языков программирования. При отсутствии в ИТ подсистемы представления знаний состав и взаимосвязь подсистем ограничивается пунктирным контуром на рисунке ниже.



Взаимосвязь подсистем физического уровня без подсистемы представления знаний базовой ИТ

Подсистема управления данными организуется на компьютерах с помощью подпрограммных систем управления обработкой данных и организации вычислительного процесса, систем управления вычислительной сетью и систем управления базами данных. При больших объемах накапливаемой на компьютере и циркулирующей в сети информации на предприятиях, где внедрена информационная технология, могут создаваться специальные службы такие, как администратор баз данных, администратор вычислительной сети и т.п.

Последовательное рассмотрение концептуального, логического и физического уровней представления информационной технологии позволяет облегчить восприятие при переходе от содержания цели информационной технологии к ее физическому достижению.

2. Процесс преобразования информации в данные.

Базовыми информационными процессами информационной технологии называют процессы обработки и накопления данных, обмена данными и представления знаний, то есть те процессы, которые поддаются формализации, а, следовательно, и автоматизации с помощью ЭВМ и средств связи.

Автоматизированные информационные процессы оперируют машинным представлением информации - данными, и, как информационная технология в целом, могут быть представлены тремя уровнями: концептуальным, логическим и физическим. Однако, прежде, чем превратиться в данные, информация должна быть сначала собрана, соответствующим образом подготовлена, и только после этого введена в ЭВМ, представ в виде данных на машинных носителях информации.

Процесс перевода информации в данные в технологических системах управления может быть полностью автоматизирован, так как для сбора информации о состоянии производственной линии применяются разнообразные электрические датчики, которые уже по своей природе позволяют проводить преобразования физических параметров, вплоть до превращения их в данные, записываемые на машинных носителях информации, без выхода на

человеческий уровень представления. Это оказывается возможным, благодаря относительной простоте и однозначности информации, снимаемой датчиками (давление, температура, скорость и т.п.). В организационно-экономических системах управления осведомляющая о состоянии объекта управления информация семантически сложна, разнообразна и ее сбор не удастся автоматизировать. Поэтому в организационно-экономических системах информационная технология на этапе превращения исходной (первичной) информации в данные в основе своей остается ручной. На рисунке ниже приведена последовательность фаз процесса преобразования информации в данные в информационной технологии организационно-экономических систем управления.



Сбор информации состоит в том, что поток осведомляющей информации, поступающей от объекта управления, воспринимается человеком и переводится в документальную форму (записывается на бумажный носитель информации). Составляющими этого потока могут быть показания приборов (например, пробег автомобиля по спидометру, накладные, акты, ордера, ведомости, журналы, описи и т.п.).

Для перевода потока осведомляющей информации в автоматизированный контур информационной технологии, необходимо собранную информацию передать в места ее ввода в

компьютер, так как часто пункты получения первичной информации от них пространственно удалены.

Передача осуществляется, как правило, традиционно, с помощью курьера, телефона.

Собранная информация для ввода должна быть предварительно *подготовлена*, поскольку модель предметной области, заложенная в компьютер, накладывает свои ограничения на состав и организацию вводимой информации. В современных информационных системах ввод информации осуществляется по запросам программы, отображаемым на экране дисплея, и часто дальнейший ввод приостанавливается, если оператор проигнорирован какой-либо важный запрос. Очень важными на этапах подготовки информации и ввода являются процедуры контроля.

3. Методы контроля подготовленной информации и ее ввод.

Контроль подготовленной и вводимой информации направлен на предупреждение, выявление и устранение ошибок, которые неизбежны, в первую очередь из-за так называемого „человеческого фактора“. Человек устает, его внимание может ослабнуть, кто-то может его отвлечь - в результате возникают ошибки. Ошибки при сборе и подготовке могут быть и преднамеренными. Любые ошибки приводят к искажению вводимой информации, к ее недостоверности, а значит, к неверным результатам обработки, и в конечном итоге к ошибкам в управлении системой. При контроле собранной и подготовленной информации применяют совокупность приемов как ручных, так и формализованных, направленных на обнаружение ошибок. Вообще процедуры контроля полноты и достоверности информации и данных используются при реализации информационных процессов повсеместно и могут быть подразделены *на визуальные, логические и арифметические*.

Визуальный метод широко используется на этапе сбора и подготовки информации и является ручным.

Логический и арифметический, являясь автоматизированными методами, применяется на последующих этапах преобразования данных.

При *визуальном методе* производится зрительный просмотр документа с целью проверки полноты, актуальности, подписей ответственных лиц, юридической законности и т.д.

При *логическом методе* контроля сопоставляются фактические данные с нормативными или с данными предыдущих периодов обработки, проверяется логическая непротиворечивость функционально-зависимых показателей и их групп и т.д.

Арифметический метод контроля включает подсчет контрольных сумм по строкам и столбцам документов, имеющих табличную форму, контроль по формулам, признакам делимости или четности, балансовые методы, повторный ввод и т.п. Для предотвращения случайного или намеренного искажения информации служат и организационные, и специальные мероприятия. Это - четкое распределение прав и обязанностей лиц, ответственных за сбор, подготовку, передачу и ввод информации в системе информационной технологии. Это - и автоматическое протоколирование ввода, и обеспечение санкционированного доступа в контур ИТ.

В настоящее время в нашей стране, как и во всем мире, персональные компьютеры все шире используются на рабочих местах служащих, ответственных за сбор, подготовку и предварительный контроль первичной информации. В этом случае используется автоматизированная подготовка и контроль собранной информации, и таким образом фазы подготовки и ввода объединяются.

Ввод информации при создании информационной технологии в организационно-экономической системе в конечном итоге является ручным - пользователь ЭВМ „набирает“ информацию (алфавитно-цифровую) на клавиатуре, визуально контролируя правильность вводимых символов по отображению на экране дисплея. Каждое нажатие клавиши - это преобразование символа изображенного на ней в электрический двоичный код, т.е. в данное.

Этап ввода - это заключительный этап процесса преобразования исходной информации в данные. Конечно, сейчас есть, помимо клавиатуры, и другие устройства ввода, позволяющие убыстрить и упростить этот трудоемкий и изобилующий ошибками

этап, например, сканеры или устройства ввода с голоса. Однако, указанные устройства, особенно последние, далеки от совершенства и имеют высокую стоимость.

Для решения задач информационной технологии, помимо ввода осведомляющей информации об объекте управления, необходимо так же подготавливать и вводить информацию о структуре и содержании предметной области (т.е. модель объекта управления), а также информацию о последовательности и содержании процедур технологических преобразований для решения поставленных задач (т.е. алгоритмическую модель). Суть подготовки информации такого вида состоит в написании программ и описании структур и данных на специальных формальных языках программирования. Этап разработки и ввода программ в настоящее время автоматизирован благодаря использованию развивающихся многофункциональных систем программирования. С их помощью существенно облегчается процесс создания программ, их отладка и ввод. Тем не менее, сам процесс моделирования, т.е. разработки моделей предметной области, решаемых задач и их алгоритмической реализации, остается творческим, и на этапе разработки информационных технологий в своей основе практически не автоматизируем.

Таким образом, после сбора, подготовки, контроля и ввода исходная информация (документы, модели, программы) превращается в данные, представленные машинными (двоичными) кодами, которые хранятся на машинных носителях и обрабатываются техническими средствами информационной технологии.

Лекция 9.

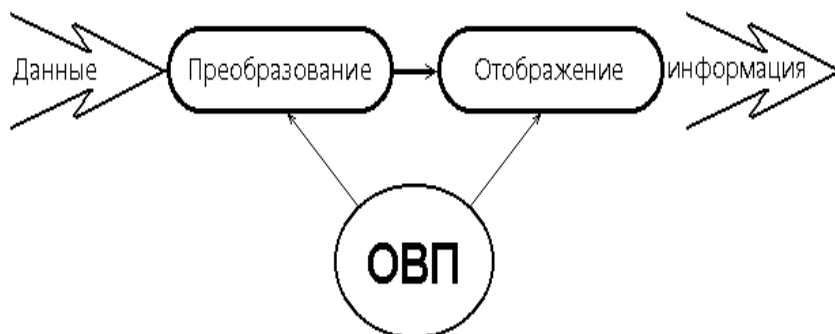
Информационный процесс обработки данных.

1. Организация вычислительного процесса.
2. Классификация архитектур вычислительных систем.
3. Отображение данных.

1. Организация вычислительного процесса.

Процесс обработки данных в информационной технологии преследует определенную цель - решение с помощью ЭВМ вычислительных задач, отображающих функциональные задачи той системы, в которой ведется управление. Для реализации этой цели должны существовать модели обработки данных, соответствующие алгоритмам управления, и воплощенные в машинных программах.

Процесс обработки может быть разбит на ряд связанных между собой процедур: организация вычислительного процесса (ОВП), преобразования данных и отображение данных (рисунок ниже).



Процедуры процесса обработки данных

Содержание процедур процесса обработки данных представляет его концептуальный уровень, модели и методы, формализующие процедуры обработки данных в ЭВМ - логический уровень, а средства аппаратной реализации процедур - физический уровень процесса.

Рассмотрим процедуры обработки данных в информационной технологии.

Процедура ОВП имеет различную функциональную сложность в зависимости от класса и количества решаемых задач, режимов обработки данных, топологии системы обработки данных. В наиболее полном объеме функции организации вычислительного процесса реализуются при обработке данных на больших универсальных машинах (мэйнфреймах), которые, как правило, работают в многопользовательском режиме и обладают большими ресурсами по памяти и производительности. При обработке данных с помощью ЭВМ, в зависимости от конкретного применения информационной технологии, а, значит, и решаемых задач, различают три основных режима: *пакетный, разделения времени, реального времени.*

При *пакетном* режиме обработки задачи (задания), а точнее, программы с соответствующими исходными данными, накапливаются на дисковой памяти ЭВМ, образуя „пакет“. Обработка заданий осуществляется в виде их непрерывного потока. Размещенные на диске задания образуют входную очередь, из которой они выбираются автоматически последовательно или по установленным приоритетам. Входные очереди могут пополняться в произвольные моменты времени. Такой режим, позволяет максимально загрузить ЭВМ, т.к. отсутствуют простои между заданиями. Но с точки зрения скорости этот режим дает задержки в получении решения, так как некоторое время задание простаивает в очереди.

Режим *разделения времени* реализуется путем выделения для выполнения заданий, определенных интервалов времени, называемых квантами. Предназначенные для обработки в этом режиме задания находятся в оперативной памяти ЭВМ одновременно. В течение одного кванта обрабатывается одно задание, затем выполнение первого задания приостанавливается с запоминанием полученных промежуточных результатов и номера следующего шага программы, и в следующий квант обрабатывается второе задание и т. д. Задание при этом режиме находится все время в оперативной памяти вплоть до завершения их обработки. При большом числе одновременно поступающих на обработку заданий возможно, для более эффективного использования оперативной памяти, временно перемещать во внешнюю память только что обрабатывавшееся задание до следующего своего кванта. В режиме разделения времени,

возможно, также реализация диалоговых операций, обеспечивающих непосредственный контакт человека с вычислительной системой.

Режим *реального времени* используется при обработке данных в информационных технологиях, предназначенных для управления физическими процессами. В таких системах информационная технология должна обладать высокой скоростью реакции, чтобы успеть за короткий промежуток времени (лучше бы мгновенно!) обработать поступившие данные и использовать полученные результаты для управления процессом. Поскольку в технологической системе управления потоки данных имеют случайный характер, вычислительная система (ВС) всегда должна быть готова получать входные сигналы и обрабатывать их. Повторить поступившие данные невозможно, поэтому потеря их недопустима

Вычислительная среда, в которой протекает процесс обработки данных, может представлять собой одномашинный комплекс, работающий в режиме разделения времени (многопрограммном режиме), или многомашинный (многопроцессорный), в котором несколько заданий могут выполняться одновременно на разных ЭВМ (процессорах). Но в обоих случаях поток заданий должен подвергаться диспетчерированию, что означает организацию и обслуживание очереди. Задания, поступившие на обработку, накапливаются в очереди входных заданий. Из этой очереди они поступают на обработку в порядке, определяемом используемой системой приоритетов. Результаты решения задач накапливаются в выходные очереди, откуда они рассылаются либо в сеть, либо на устройство отображения, либо на устройство накопления.

На [рисунке ниже](#) изображена схема организации вычислительной многомашинной системы, где упорядочение очереди из потока заданий осуществляется диспетчером Д1, а ее обслуживание ЭВМ осуществляется через диспетчера Д2.

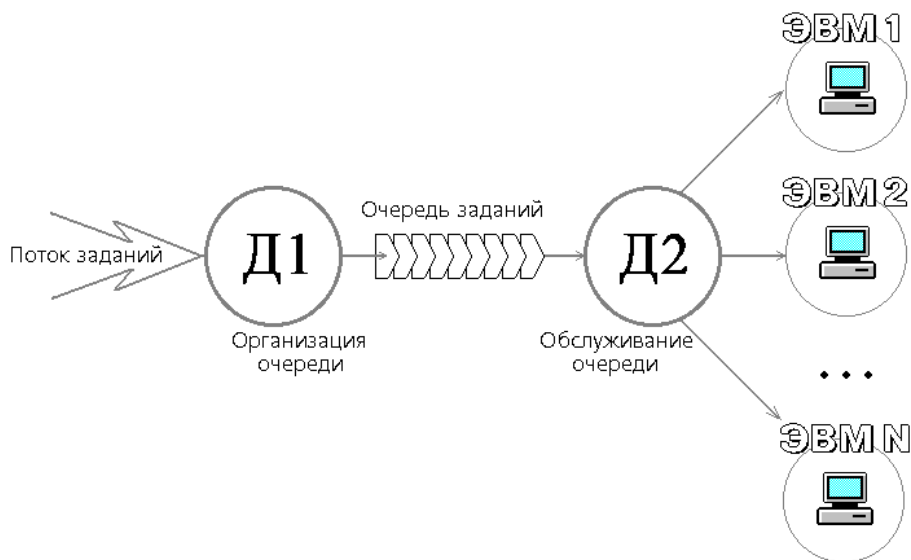


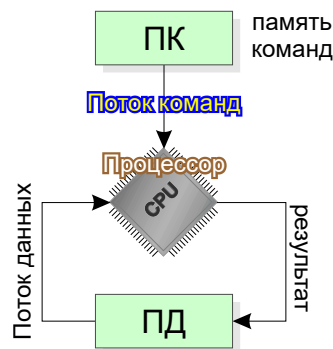
Схема организации обслуживания заданий в многомашинной вычислительной системе.

2. Классификация архитектур вычислительных систем.

Многопроцессорные системы, ориентированные на достижение сверхбольших скоростей работы, содержат десятки или сотни сравнительно простых процессоров с упрощенными блоками управления. Отказ от универсальности применения таких ВС и специализация их на определенном круге задач, допускающих эффективное распараллеливание вычислений, позволяют строить их с регулярной структурой связей между процессорами.

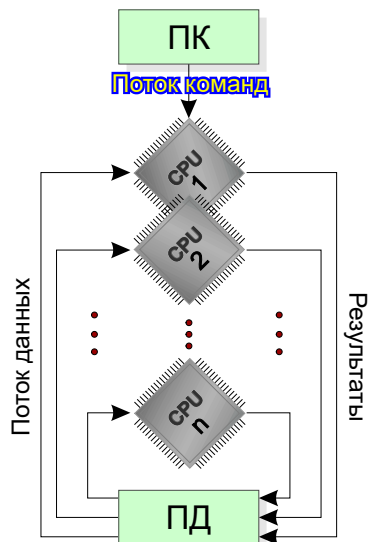
Рассмотрим классификацию вычислительных систем.

Однопроцессорная ЭВМ. Структура обыкновенной однопроцессорной ЭВМ содержит одинарный поток команд и одинарный поток данных (структура *ОКОД* или *SISD*).



Структура ОКОД или SISD

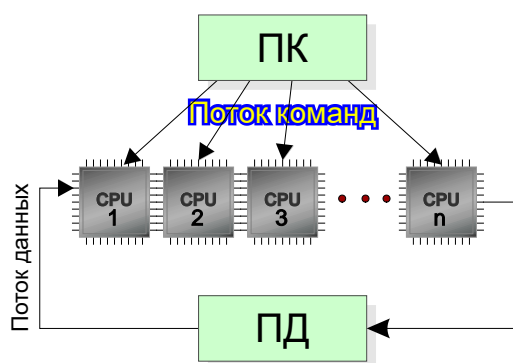
Матричная МПС – структура типа ОКМД (или SIMD).
 Система содержит некоторое число одинаковых сравнительно простых быстродействующих процессоров, соединенных друг с другом и с памятью данных регулярным образом так, что образуется сетка (матрица), в узлах которой размещаются процессоры. В системе имеется несколько потоков данных и один общий поток команд. Здесь возникает сложная задача распараллеливания алгоритмов решаемых задач для обеспечения загрузки процессоров. В ряде случаев эти вопросы лучше решаются в конвейерной системе.



Структура типа ОКМД (или SIMD)

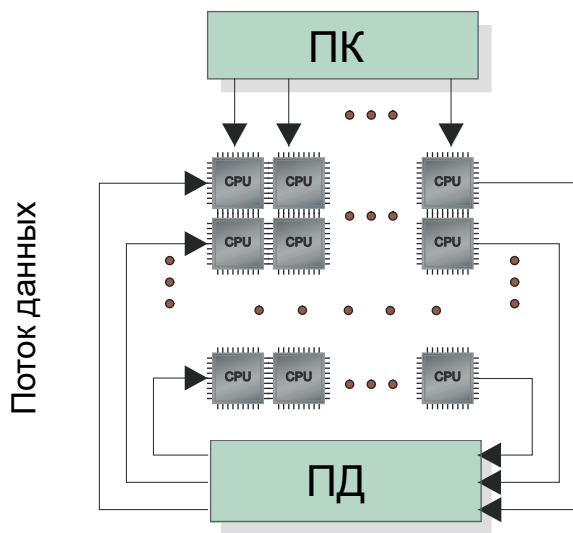
Конвейерная МПС – структура типа МКОД (или MISD).
Система имеет регулярную структуру в виде цепочки последовательно соединенных процессоров или специальных вычислительных блоков (СВБ), так что информация на выходе одного процессора является входной информацией для следующего в конвейерной цепочке.

Процессоры (СВБ) образуют конвейер, на вход которого одинарный поток данных доставляет операнды из памяти. Каждый процессор обрабатывает соответствующую часть задачи, передавая результаты соответствующему процессору, который использует их в качестве исходных данных. Таким образом, решение задач для некоторых исходных данных разворачивается последовательно в конвейерной цепочке. Это обеспечивает подведение к каждому процессору (СВБ) своего потока команд, т.е. имеется множественный поток команд.



Структура типа МКОД (или MISD)

Общий случай МПС – структура типа МКМД (или MIMD). На рисунке ниже представлен общий случай структуры МПС, в которой несколько потоков данных и несколько потоков команд.



Структура типа МКМД (или MIMD)

Существует несколько типов МКМД. К ним относятся:

- мультипроцессорные системы;
- системы с мультиобработкой;
- многомашинные системы;
- компьютерные сети.

3. Отображение данных.

Процедура отображения данных является одной из важнейших в информационной технологии. Без возможности восприятия результата обработки информации человеческими органами чувств этот результат оставался бы вещью в себе (ведь мы не ощущаем машинное представление информации).

Наиболее активным из человеческих органов является зрение, поэтому процедуры отображения в информационных технологиях преследуют цель как можно лучше представить информацию для визуального наблюдения. Конечно, в мультимедийных системах сейчас используется и аудио, и видео отображение данных, но при управлении организацией более важным является отображение данных в текстовой или в графической формах. Основными устройствами, воспроизводящими текст или графические фигуры, являются дисплеи и принтеры, на использование которых

(особенно первых) и направлены операции и процедуры отображения.

Для того, чтобы получить на экране дисплея (или на бумаге принтера) изображение, отображающее выводимую из компьютера информацию, данные (т.е. машинное представление этой информации) должны быть соответствующим образом преобразованы, затем, адаптированы (согласованы) с параметрами дисплея и, наконец, воспроизведены. Все эти операции должны выполняться в строгом соответствии с заданной формой воспроизведения и возможностями воспроизводящего устройства. Согласование операции процедуры отображения производится с помощью управляющей процедуры ОВП (организации вычислительного процесса). На рисунке ниже приведена вышеописанная схема взаимодействия процедур при отображении данных.

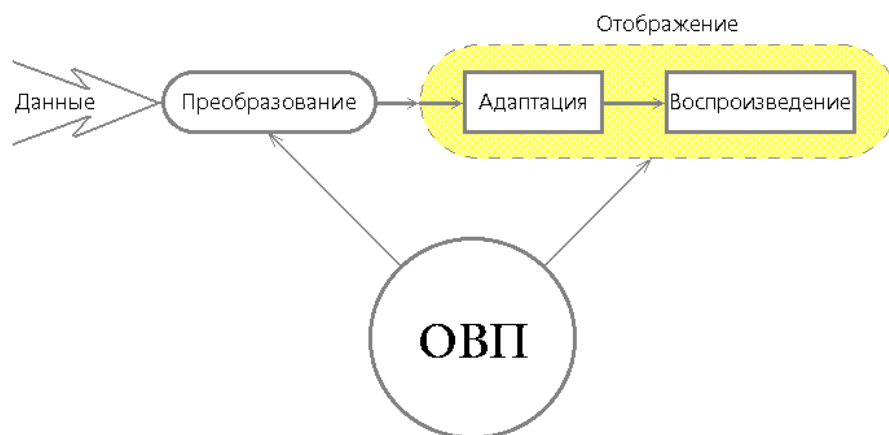


Схема взаимодействия процедур при отображении данных

В современных информационных технологиях при воспроизведении информации предпочтение отдано графическим режимам работы дисплеев (в отличие от исторически более ранних текстовых режимов) как наиболее универсальных. Графический режим позволяет выводить на экран дисплея любую графику (ведь буквы и цифры тоже графические объекты), причем с возможностью изменения масштаба, проекции, цвета и т.д. В последнее время развитие информационных технологий с точки зрения ввода и вывода информации идет по пути создания

объектно-ориентированных систем, в которых настройка систем, программирование функциональных задач, ввод и вывод информации осуществляется с помощью графических объектов, отображаемых на экране дисплея (примером может служить широко распространенный графический интерфейс Windows, объектно-ориентированные языки программирования).

Отображение информации на экране дисплея (на бумаге принтера, графопостроителя) в виде графических объектов (графиков, геометрических фигур, изображений и т. д.), носит название компьютерной (машинной) графики.

На логическом уровне процедура отображения основана на аналитической геометрии, разработанной французским философом и математиком Рене Декартом в 17 веке, согласно которой положение любой точки на плоскости (а экран дисплея - плоскость) задается парой чисел - координатами. Пользуясь декартовой системой координат, любое плоское изображение можно свести к списку координат составляющих его точек. И наоборот, заданные оси координат, масштаб и список координат легко превратить в изображение. Геометрические понятия, формулы и факты, относящиеся, прежде всего, к плоскому и трехмерному случаям, играют в задачах компьютерной графики особую роль. Основой математических моделей компьютерной графики являются аффинные преобразования и сплайн - функции.

Лекция 10.

Информационный процесс накопления данных.

1. Назначение технологического процесса накопления данных.
2. Инфологическая и каноническая модели предметной области. Матрица смежности.
3. Информационный граф матрицы смежности.

1. Назначение технологического процесса накопления данных.

Назначение технологического процесса накопления данных состоит в создании, хранении и поддержании в актуальном состоянии информационного фонда, необходимого для выполнения функциональных задач системы управления, для которой построен контур информационной технологии. Кроме того, хранимые данные по запросу пользователя или программы должны быть быстро (особенно для систем реального времени) и в достаточном объеме извлечены из области хранения и переведены в оперативные запоминающие устройства ЭВМ для последующего либо преобразования по заданным алгоритмам, либо отображения, либо передачи.

Указанные функции, выполняемые в процессе накопления данных, реализуются по алгоритмам, разработанным на основе соответствующих математических моделей.

Процессы и операции, выполняемые при протекании процесса накопления данных, приведены на структурной схеме (рисунок ниже).



Структурная схема жизненного цикла существования данных

Процедура *хранения* состоит в том, чтобы сформировать и поддерживать структуру хранения данных в памяти ЭВМ.

Современные структуры хранения данных должны быть независимы от программ, использующих эти данные и реализовывать вышеуказанные принципы (полнота и минимальная избыточность). Такие структуры получили название *баз данных (БД)*. Осуществление процедур создания структуры хранения (базы данных), актуализация, извлечение и удаление данных производится с помощью специальных программ, называемых системами управления базами данных (СУБД).

Процедура *актуализации данных* позволяет изменить значения данных, записанных в базе, либо дополнить определенный раздел, группу данных. Устаревшие данные могут быть удалены с помощью соответствующей операции.

Процедура *извлечения данных* необходима для пересылки из базы данных требующихся данных либо для преобразования, либо для отображения, либо для передачи по вычислительной сети.

При выполнении процедур актуализации и извлечения обязательно выполняются операции *поиска* данных по заданным признакам и их *сортировки*, состоящие в изменении порядка расположения данных при хранении или извлечении.

На логическом уровне все процедуры процесса накопления должны быть формализованы, что отображается в математических и алгоритмических моделях этих процедур.

2. Инфологическая и каноническая модели предметной области. Матрица смежности.

Информационный фонд системы управления должен обеспечивать получение выходных наборов данных из входных с помощью алгоритмов обработки и корректировки данных. Это возможно, если создана инфологическая модель предметной области, которая вместе с наборами хранимых данных и алгоритмами их обработки позволяет построить каноническую модель (схему) информационной базы, а затем перейти к логической схеме и, далее, к физическому уровню реализации.

Инфологической (концептуальной) моделью предметной области называют описание предметной области без ориентации на используемые в дальнейшем программные и технические средства. Однако, для построения информационной базы инфологической

модели не достаточно. Необходимо провести анализ информационных потоков в системе с целью установления связи между элементами данных, их группировки в наборы входных, промежуточных и выходных элементов данных, исключения избыточных связей и элементов данных. Получаемая в результате такого анализа безыбыточная структура носит название канонической структуры информационной базы и является одной из форм представления инфологической модели предметной области.

Для анализа информационных потоков в управляемой системе исходными являются данные о парных взаимосвязях, или отношениях (т.е. есть отношение или нет отношения) между наборами информационных элементов. Под информационными элементами понимают различные типы входных, промежуточных и выходных данных, которые составляют наборы входных N_1 , промежуточных N_2 и выходных N_3 элементов данных.

Формализовано связи (парные отношения) между наборами информационных элементов отображаются в виде *матрицы смежности* \mathbf{B} , под которой понимают квадратную бинарную матрицу, проиндексированную по обеим осям множеством информационных элементов

$$D = \{d_1, d_2, \dots, d_s\},$$

где s - число этих элементов.

$$B = \begin{array}{c} \\ d_1 \\ d_2 \\ \dots \\ d_i \\ \dots \\ d_s \end{array} \begin{array}{c} \left| \begin{array}{cccccc} d_1 & d_2 & \dots & d_j & \dots & d_s \\ q_{11} & q_{12} & \dots & q_{1j} & \dots & q_{1s} \\ q_{21} & q_{22} & \dots & q_{2j} & \dots & q_{2s} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ q_{i1} & q_{i2} & \dots & q_{ij} & \dots & q_{is} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ q_{s1} & q_{s2} & \dots & q_{sj} & \dots & q_{ss} \end{array} \right. \end{array}$$

$$\text{где } q_{ij} = \begin{cases} 1, & \text{если между } d_i \text{ и } d_j \text{ отношение существует} \\ 0, & \text{в противном случае;} \end{cases}$$

$$i = \overline{1, S}; j = \overline{1, S}$$

В позиции (i, j) матрицы смежности записывают «1» (т.е. $q_{ij} = 1$), если между информационными элементами d_i и d_j существует отношение R_0 , такое, что для получения значения информационного элемента d_j необходимо непосредственное обращение к элементу d_i . Наличие такого отношения между d_i и d_j обозначают в виде, $d_i R_0 d_j$ чему соответствует $q_{ij} = 1$, а отсутствие - $d_i \overline{R_0} d_j$ и запись „0“ в позиции (i, j) , т.е. $q_{ij} = 0$. Для простоты принимают, что каждый информационный элемент недостижим из самого себя:

$$d_i \overline{R_0} d_i; i = \overline{1, S}$$

3. Информационный граф матрицы смежности.

Матрице **B** ставится в соответствие информационный граф $G(D, R_0)$.

Графом называется набор точек (эти точки называются вершинами), некоторые из которых объявляются смежными (или соседними). Считается, что смежные вершины соединены между собой ребрами (или дугами).

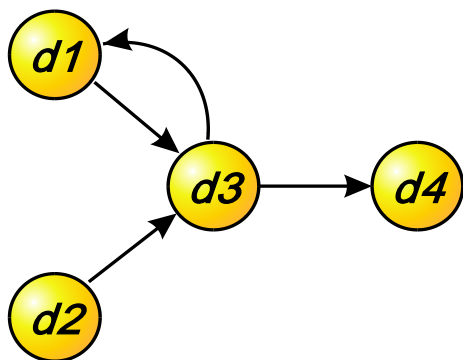
Множеством вершин графа $G(D, R_0)$ является множество **D** информационных элементов, а каждая дуга (d_i, d_j) соответствует условию $d_i R_0 d_j$; т.е. записи „1“ в позиции (ij) матрицы **B**.

Например, задано множество **D** из четырех наборов информационных элементов, т.е. $D = \{d_1, d_2, d_3, d_4\}$. Пусть матрица смежности **B** этих элементов

имеет вид:

$$B = \begin{matrix} & \begin{matrix} d_1 & d_2 & d_3 & d_4 \end{matrix} \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} & \begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \end{matrix} .$$

Из этой матрицы видно, что для вычисления элемента d_3 необходимо обращение к элементам d_1 и d_2 , а для получения элемента d_4 - к элементу d_3 . Чтобы получить элемент d_1 , надо обратиться к d_3 . Элемент d_2 не зависит от других элементов матрицы. Информационный граф в этом простейшем случае будет соответствовать рисунку ниже



Информационный граф $G(D, R_0)$

В общем случае структура графа $G(D, R_0)$ вследствие неупорядоченности сложна для восприятия и анализа. Составленная на основе инфологической модели, она не гарантирована от неточностей, ошибок, избыточности и транзитивности.

Лекция 11.

Модели баз данных информационного процесса накопления данных.

1. Матрица достижимости канонической модели предметной области.
2. Информационный граф канонической модели предметной области.
3. Реляционная модель базы данных.

1. Матрица достижимости канонической модели предметной области.

Для формального выделения входных, промежуточных и выходных наборов информационных элементов, определения последовательности операций их обработки, анализа и уточнения взаимосвязей на основе графа $G(D, R_0)$ строят *матрицу достижимости*.

Матрицей достижимости \mathbf{M} называют квадратную бинарную матрицу, проиндексированную по обеим осям множеством информационных элементов D , аналогично матрице смежности \mathbf{B} . Запись „1“ в каждой позиции (ij) матрицы достижимости соответствует наличию для упорядоченной пары информационных элементов (d_i, d_j) , смыслового отношения достижимости R . Элемент d_j *достижим* из элемента d_i , т.е. выполняется условие $d_i R_0 d_j$, если на графе $G(D, R_0)$ существует направленный путь от вершины d_i к вершине d_j (в процессе получения значения элемента d_j используется значение элемента d_i). Если $d_i \bar{R}_0 d_i$, то отношение достижимости между элементами d_i и d_j отсутствует и в позиции (ij) матрицы \mathbf{M} записывают «0». Отношение достижимости транзитивно, т.е. если $d_i R_0 d_k$ и $d_k R_0 d_j$, то $d_i R_0 d_j$; $i, j, k = \overline{1, S}$.

Записи „1“ в j -м столбце матрицы \mathbf{M} соответствуют информационным элементам d_j , которые необходимы для получения значений элементов d_j , и которые образуют множество элементов предшествования $A(d_j)$ для этого элемента. Записи „1“ в i -ой строке матрицы \mathbf{M} соответствуют всем элементам d_j , достижимым из рассматриваемого элемента d_i и образующим множество достижимости $R(d_i)$ этого элемента. Информационные элементы, строки которых в матрице \mathbf{M} не содержат единиц

(нулевые строки), являются *выходными* информационными элементами, а информационные элементы, соответствующие нулевым столбцам матрицы **M**, являются *входными*. Это условие может служить проверкой правильности заполнения матриц **B** и **M**, если наборы входных и выходных информационных элементов известны. Информационные элементы, не имеющие нулевой строки или столбца, являются *промежуточными*.

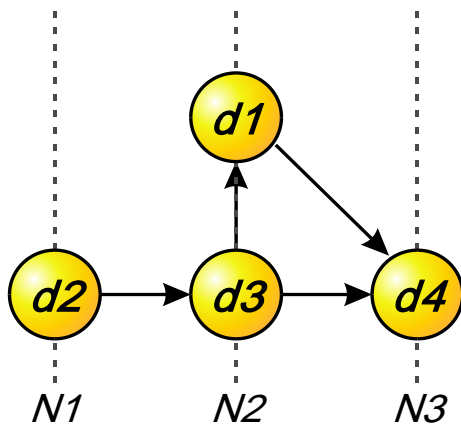
Для полученного в примере предыдущей лекции информационного графа матрицы смежности матрица *M* будет выглядеть так:

$$M = \begin{matrix} & \begin{matrix} d_1 & d_2 & d_3 & d_4 \end{matrix} \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix} & \begin{vmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix} \end{matrix}$$

Отличие столбцов матриц **M** и **B** объясняется тем, что в матрице **M** учитывается смысловое отношение *R* между информационными элементами, а в матрице **B** только непосредственное *R*₀. Например, элемент *d*₄ в матрице **M** достижим из элементов *d*₁, *d*₂ и *d*₃, т.е. *d*₁ *R* *d*₄, *d*₂ *R* *d*₄ и *d*₃ *R* *d*₄, в то время как в матрице **B** для этих элементов *d*₄ достижим только из *d*₃, т.е. только *d*₃ *R*₀ *d*₄, а *d*₁ *R*₀ *d*₄ и *d*₂ *R*₀ *d*₄.

2. Информационный граф канонической модели предметной области.

Из анализа матрицы **M** следует, что элемент *d*₂ является входным, *d*₄ - выходным, остальные - промежуточные. На основе матрицы **M** строится информационный граф *G*_s(*D*,*R*) системы, структурированный по входным (*N*₁), промежуточным (*N*₂) и выходным (*N*₃) наборам информационных элементов, и полученный из анализа множеств элементов предшествования *A*(*d*_{*i*}) и достижимости *R*(*d*_{*j*}). Граф *G*_s(*D*,*R*), полученный из матрицы **M** рассматриваемого примера, приведен на рисунке ниже.



Информационный граф $G_s(D,R)$

В общем случае информационный граф системы, в отличие от вычисленного графа, может иметь контуры и петли, что объясняется необходимостью повторного обращения к отдельным элементам данных.

Информационный граф системы $G_s(D,R)$ структурируется по уровням (N_1, N_2, N_3) с использованием итерационной процедуры, что позволяет определить информационные входы и выходы системы, выделить основные этапы обработки данных, их последовательность и циклы обработки на каждом уровне. Кроме того, удаляются избыточные (лишние) дуги и элементы. Граф, получаемый после структуризации по наборам информационных элементов и удаления избыточных элементов и связей, определяет каноническую структуру информационной базы. Таким образом, каноническая структура задает логически не избыточную информационную базу. Выделение наборов элементов данных по уровням позволяет объединить множество значений конечных элементов в логические записи и тем самым упорядочить их в памяти ЭВМ.

От канонической структуры переходят к логической структуре информационной базы, а затем - к физической организации информационных массивов. Каноническая структура является также основой для автоматизации процессов предпроектного анализа предметных областей систем управления. Процедуры хранения, актуализации и извлечения данных непосредственно связаны с базами данных, поэтому логический уровень этих процедур определяется моделями баз данных.

3. Реляционная модель базы данных.

В *реляционной модели* базы данных взаимосвязи между элементами данных представляются в виде двумерных таблиц, называемых *отношениями*. Отношения обладают следующими свойствами:

каждый элемент таблицы представляет собой один элемент данных (повторяющиеся группы отсутствуют);

элементы столбца имеют одинаковую природу, и столбцам однозначно присвоены имена;

в таблице нет двух одинаковых строк;

строки и столбцы могут просматриваться в любом порядке вне зависимости от их информационного содержания.

Реляционная модель БД обладает следующими преимуществами:

простотой логической модели (таблицы привычны для представления информации);

гибкостью системы защиты (для каждого отношения может быть задана правомерность доступа);

независимостью данных;

возможностью построения простого языка манипулирования данными с помощью математически строгой теории реляционной алгебры (алгебры отношений).

Собственно, наличие строгого математического аппарата для реляционной модели баз данных и обусловило её наибольшее распространение и перспективность в современных информационных технологиях.

Для пояснения логической структуры реляционной модели базы данных рассмотрим такую простую задачу: необходимо разработать логическую структуру БД для хранения данных о трех поставщиках P_1 , P_2 и P_3 , которые могут поставлять товары T_1 , T_2 и T_3 в следующих комбинациях: поставщик P_1 - все три вида товаров, поставщик P_2 - товары T_1 и T_3 , поставщик P_3 - товары T_2 и T_3 .

Для данной задачи логическая структура реляционной БД будет содержать три таблицы (отношения): R_1 и R_2 , состоящие из записей о поставщиках и о товарах соответственно, и R_3 – из записей о поставках товаров поставщиками (рисунок ниже).

П1
П2
П3

T1
T2
T3

П1	T1
П1	T2
П1	T3
П2	T1
П2	T3
П3	T2
П3	T3

Реляционная модель БД

Рассмотрим данный пример более детально.

Таблица R_1 представляет поставщиков. Каждый поставщик имеет номер, уникальный для этого поставщика, фамилию (естественно не уникальную), значение рейтинга и местонахождение (город).

Таблица R_2 представляет виды товаров. Каждый товар имеет уникальный номер, название, вес и цвет.

В таблице R_3 представлена поставка товаров. Она служит для того, чтобы в определенном смысле связать между собой две другие таблицы. Например, первая строка этой таблицы связывает определенного поставщика из таблицы R_1 (поставщика P_1) с определенным товаром из таблицы R_2 (с товаром T_1). Иными словами, она представляет поставку товаров вида T_1 поставщиком по фамилии P_1 и объем поставки, равный 300 штук. Таким образом,

для каждой поставки имеется номер поставщика, номер товара и количество товара.

Детальные таблицы данной реляционной БД представлены на рисунке ниже

R1 (поставщики)

Номер Поставщика	Фамилия	Рейтинг	Город
П1	Иванов	20	Москва
П2	Петров	10	Курск
П3	Сидоров	30	Краснодар

R2 (товары)

Номер товара	Название	Вес	Цвет
T1	Гайка	12	Красный
T2	Болт	17	Зеленый
T3	Шайба	5	Голубой

R3 (поставка товаров)

Номер поставщика	Номер детали	Количество
П1	T1	300
П1	T2	200
П1	T3	400
П2	T1	300
П2	T3	400
П3	T2	200
П3	T3	300

Реляционная БД поставщиков и товаров

Из приведенных на рисунке выше таблиц следует:

а) все значения данных являются *атомарными*, т.е. в каждой таблице на пересечении строки и столбца всегда имеется в точности одно значение данных и никогда не бывает множества значений;

б) полное информационное содержание базы данных представляется в виде *явных значений данных*. Такой метод представления - единственный, имеющийся в распоряжении реляционной базы данных. В частности, не существует каких-либо связей и указателей, соединяющих одну таблицу с другой. Для этой цели служат тоже таблицы. Так таблица R₃ представляет связь таблиц R₁ и R₂.

Как указывалось, математическим термином для обозначения таблицы является „отношение“ (relation) и реляционные системы берут свое начало в математической теории отношений. Основы

реляционной модели данных были первоначально сформулированы доктором Э.Ф. Коддом из фирмы IBM, и опубликованы в 1970 году. С тех пор эти идеи оказали широкое влияние на технологию баз данных во всех её аспектах, а так же и на другие области информационных технологий (например, искусственный интеллект и обработку текстов на естественных языках).

При работе с реляционными моделями используется как математическая терминология, так и терминология исторически принятая в сфере обработки данных. Для того, чтобы не возникало разночтений, ниже приведены основные формальные реляционные термины и соответствующие им неформальные эквиваленты.

Формальный реляционный термин	Неформальный эквивалент
Отношение	Таблица
Кортеж	Запись, строка
Атрибут	Поле, столбец

Лекция 12.

Аспекты данных реляционной модели базы данных.

1. Реляционная структура данных.
2. Целостность реляционных данных.
3. Манипулирование реляционными данными.
Традиционные реляционные операции.

1. Реляционная структура данных.

Реляционная модель БД имеет дело с тремя аспектами данных: со *структурой* данных, с *целостностью* данных и с *манипулированием* данными. Под *структурой* понимается логическая организация данных в БД, под *целостностью* данных понимают безошибочность и точность информации, хранящейся в БД, под *манипулированием* данными - действия, совершаемые над данными в БД. Эти три аспекта отражают и основные процедуры процесса накопления данных (хранение, актуализацию и извлечение).

Наименьшей единицей данных в реляционной модели является отдельное значение данных. Такие значения рассматриваются как атомарные, т.е. неразложимые, когда речь идет о данной модели.

Доменом называют множество подобных значений одного и того же типа. Например, домен номеров поставщиков - это множество допустимых номеров поставщиков, домен объемов поставки - множество целых, больших нуля и меньших, например, 10000. Таким образом, домены представляют собой *пулы значений*, из которых берутся фактические значения, появляющиеся в атрибутах (столбцах). Смысл доменов заключается в следующем. Если значения двух атрибутов берутся из одного домена, то имеют смысл их сравнения, а, следовательно, и соединения, объединения и т.д. Если же значения атрибутов берутся из разных доменов, то всякие их сравнения лишены смысла. Отметим, что домены по природе своей являются в большей степени понятиями концептуальными и могут храниться или не храниться в базе данных как фактическое множество значений. Но они должны специфицироваться как часть определения базы данных, и определение каждого атрибута должно включать ссылку на

соответствующий домен для избежания каких-либо двусмысленностей.

Теперь определим главный элемент реляционной структуры - отношение.

Отношение на доменах D_1, D_2, \dots, D_n . состоит из *заголовка* и *тела*. Заголовок состоит из такого фиксированного множества *атрибутов* A_1, A_2, \dots, A_n , что существует взаимно однозначное соответствие между этими атрибутами A_i и определяющими их доменами D_i ($i = \overline{1, n}$). Тело состоит из меняющегося во времени множества *кортежей*, где каждый кортеж в свою очередь состоит из множества пар атрибут-значение $(A_i:V_i)$ ($i = \overline{1, n}$), по одной такой паре для каждого атрибута A_i в заголовке. Для любой заданной пары атрибут-значение $(A_i:V_i)$, V_i является значением из единственного домена D_i , с которым связан атрибут A_i . Если теперь посмотреть на отношения рисунка ниже, то можно увидеть, что все они соответствуют приведенному определению отношения.

R1 (поставщики)

Номер Поставщика	Фамилия	Рейтинг	Город
П1	Иванов	20	Москва
П2	Петров	10	Курск
П3	Сидоров	30	Краснодар

R2 (товары)

Номер товара	Название	Вес	Цвет
T1	Гайка	12	Красный
T2	Болт	17	Зеленый
T3	Шайба	5	Голубой

R3 (поставка товаров)

Номер поставщика	Номер детали	Количество
П1	T1	300
П1	T2	200
П1	T3	400
П2	T1	300
П2	T3	400
П3	T2	200
П3	T3	300

Строго говоря, когда мы изображаем отношение в виде таблицы, мы просто используем удобный способ представления отношения на бумаге. Таблица и отношение в действительности не одно и то же. Дело в том, что при изображении таблицы мы явно или неявно упорядочиваем расположение столбцов (атрибутов) и

строк (кортежей), хотя отношение – это математическое множество, а множество в математике не обладает каким-либо упорядочением.

Значение « n » - число атрибутов в отношении называется степенью отношения. Отношение степени один называется *унарным*, степени два - *бинарным*, степени три - *тернарным*, степени n – *n-арным*. В приведенной на рисунке выше базе данных степень отношений R_1 и R_2 равна четырем, а отношения R_3 - трем. Число кортежей в отношении называется *кардинальным числом* этого отношения. Кардинальные числа отношений R_1 , R_2 и R_3 на рисунке выше равны соответственно 3, 3 и 7. Кардинальное число отношения изменяется во времени (кортеж может быть добавлен или удален) в отличие от его степени.

2. Целостность реляционных данных.

Важным следствием определений, сделанных выше, является то, что каждое отношение имеет *первичный ключ*, идентифицирующий это отношение. Поскольку отношение это множество, а множества по определению не содержат совпадающих элементов, никакие два кортежа отношения не могут в произвольный заданный момент времени быть дубликатами друг друга. Пусть R - отношение с атрибутами A_1, A_2, \dots, A_n . Говорят, что множество атрибутов $K = (A_1, A_2, \dots, A_k)$ отношения R является возможным ключом R тогда и только тогда, когда удовлетворяются два следующих независимых от времени условия:

- 1) уникальность;
- 2) минимальность.

Первое условие указывает на то, что в произвольный заданный момент времени никакие два различных кортежа отношения R не имеют одного и того же значения (A_1, A_2, \dots, A_k) .

Второе условие говорит о том, что ни один из атрибутов (A_1, A_2, \dots, A_k) не может быть исключен из K без нарушения условий уникальности.

Каждое отношение обладает, по крайней мере, одним возможным ключом, поскольку по меньшей мере комбинация всех его атрибутов удовлетворяет условиям уникальности. Один произвольно выбранный возможный ключ для данного отношения

принимается за его *первичный* ключ, а остальные возможные ключи называются *альтернативными*.

Помимо первичных и альтернативных ключей, идентифицирующих данное отношение, есть еще понятие *внешнего* ключа. В общем случае внешний ключ - это атрибут или комбинация атрибутов одного отношения R'' , значение которого обязательно должно совпадать со значением первичного ключа некоторого другого отношения R' , причем, внешний и первичный ключи должны быть определены на одних и тех же доменах. Внешние ключи в неявном виде связывают отношения. Примером внешнего ключа является атрибут „номер поставщика“ в отношении R_3 на рисунке выше, поскольку этот атрибут может быть первичным ключом отношения R_1 .

Целостность реляционной модели данных определяется двумя общими правилами.

1. Целостность по сущностям. Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе базового отношения, принимал неопределенные значения. *Базовым* отношением называют независимое именованное отношение (для БД поставщиков и товаров - это отношения R_1 и R_2). Мотивировка этого правила определяется тем, что базовые отношения соответствуют сущностям в реальном мире, а следовательно отличимы, т.е. имеют уникальную идентификацию. В реальной же модели функцию уникальной идентификации выполняют первичные ключи, и таким образом, ситуация, когда первичный ключ принимает неопределенное значение, является противоречивой, и говорит о том, что некоторая сущность не обладает индивидуальностью, а значит, не существует. Отсюда название „целостность по сущностям“.

2. Целостность по ссылкам. Если базовое отношение R'' включает некоторый внешний ключ FK, соответствующий некоторому первичному ключу PK какого-либо базового отношения R' , то каждое значение FK в R'' должно либо быть равным значению PK в некотором кортеже R' , либо быть полностью неопределённым. Неопределенность внешнего ключа может возникнуть в ситуации, когда, например, имеется вакансия на должность в некоторый отдел. Для такой должности атрибут „фамилия служащего“, являющийся внешним ключом, имеет

неопределенное значение в кортеже, представляющим эту штатную должность отдела.

3. Манипулирование реляционными данными. Традиционные реляционные операции.

Виды действий (манипуляций) над данными в реляционной модели представляют собой множество операций, получивших в совокупности название *реляционной алгебры*, и реляционной операции присваивания. Последняя производит присваивание значения некоторого производного выражения реляционной алгебры другому отношению.

Каждая операция реляционной алгебры использует одно или два отношения в качестве операндов и создает в результате некоторое новое отношение. Э. Коддом были определены восемь таких операций, объединенные в две группы по четыре операции в каждой.

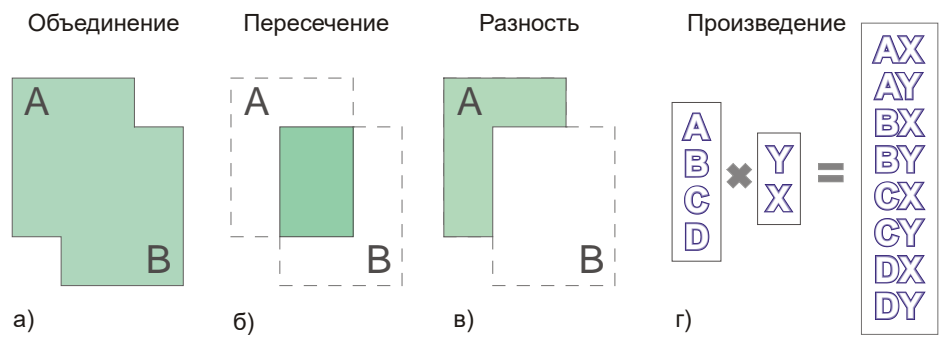
Первая группа - *традиционные теоретико-множественные операции*.

В каждой из этих операций используется два операнда (отношения).

Для всех операций, кроме декартова произведения, эти два операнда должны быть совместимы по объединению, т.е. они должны быть одной степени, и их i -е атрибуты ($i = \overline{1, n}$) должны быть связаны с одним и тем же доменом.

Операция объединения. Объединением двух отношений A и B называется множество всех кортежей t , принадлежащих либо A , либо B , либо им обоим. Символически эта операция показана на рисунке ниже а.

Математически эта операция записывается так:



Диagramмы традиционных теоретико-множественных операций

Математически операция объединения записывается так:

$$A \cup B = \{t : t \in A \text{ или } t \in B\};$$

где \cup -символ объединения, \in - знак принадлежности к определенному отношению (множеству).

Операция пересечения. Пересечением двух отношений A и B называется множество всех кортежей t , каждый из которых принадлежит как A, так и B (рисунок выше б):

$$A \cap B = \{t : t \in A \text{ и } t \in B\},$$

где \cap - символ пересечения.

Операция разность. Разностью между двумя отношениями A и B называется множество всех кортежей t , каждый из которых принадлежит A и не принадлежит B (рисунок выше в):

$$A \setminus B = \{t : t \in A, t \notin B\},$$

где \setminus - символ разности,

\notin - символ отсутствия принадлежности отношению (множеству).

Операция декартово произведение. Декартовым произведением двух отношений A и B называется множество всех кортежей t , таких, что t является конкатенацией некоторого кортежа a , принадлежащего A, и какого-либо кортежа b , принадлежащего B. (конкатенация - это соединение в цепочки). Для рисунка выше г – декартово произведение:

$$A \times B = \{ax, ay, bx, by, cx, cy, dx, dy\}.$$

Лекция 13.

Модели баз данных информационного процесса накопления данных (заключение).

1. Манипулирование реляционными данными.
Специальные реляционные операции.
2. Объектная модель баз данных.
3. Программно-аппаратный уровень процесса накопления данных.

1. Манипулирование реляционными данными. Специальные реляционные операции.

Отформатировано: нумерованный + Уровень: 1 + Стиль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

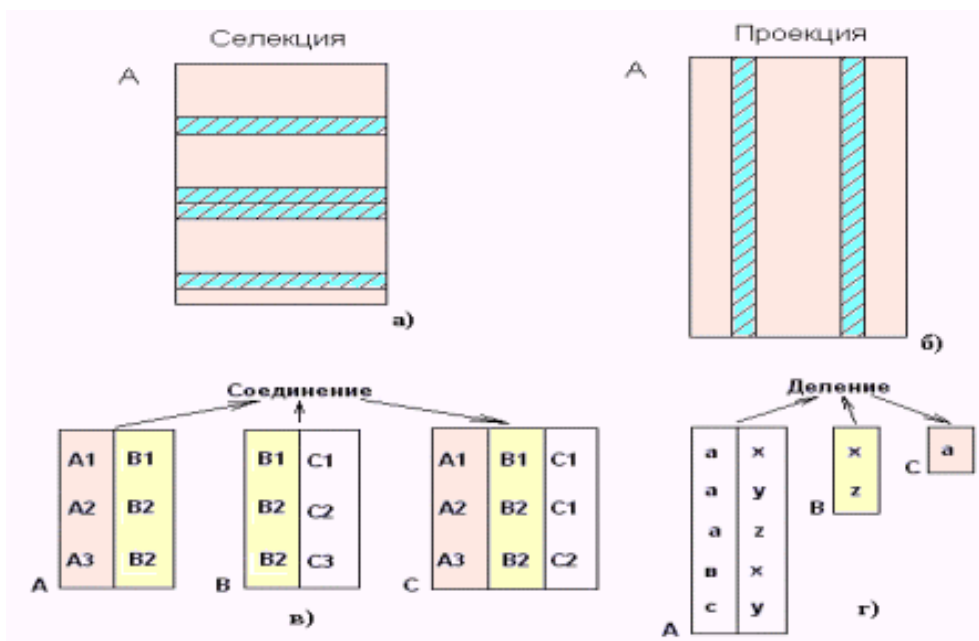
Отформатировано: Отступ: Первая строка: 0 см

Вторая группа операций манипулирования реляционными данными - *специальные реляционные операции*. К ним относятся селекция, проекция, соединение, деление. Диаграммы этих операций приведены на рисунке ниже.

Операция селекция. Пусть θ представляет собой любой достижимый оператор сравнения данных, например, $=, \neq, >, \geq, \leq$ и т.д.

Предикат (n -местный, или n -арный) — это функция с множеством значений $\{0,1\}$ (или «ложь», и «истина»), определённая на множестве $M = M_1 \times M_2 \times \dots \times M_n$. Таким образом, каждый набор элементов множества M характеризуется либо как «истинный», либо как «ложный».

θ - селекцией отношения A по атрибутам x и y называется множество всех кортежей t из A , таких, что истинен предикат « $t.x \theta t.y$ ». Атрибуты x и y должны быть определены на одном и том же домене, и для этого домена оператор θ должен иметь смысл. Вместо атрибута y может быть задана константа (например, выбрать из платёжной ведомости записи о своих сотрудниках имеющих зарплату 500 руб.). Таким образом, оператор θ - селекции позволит получать «горизонтальные» подмножества заданного отношения, т.е. подмножество таких кортежей заданного отношения, для которых выполняются поставленное условие (рисунок ниже а).



Диаграммы специальных реляционных операций

Операция проекция. Операция проекция позволяет получить «вертикальное» подмножество заданного отношения, т.е. такое подмножество, которое получается выбором специфицированных (определенных) атрибутов с последующим исключением, если это необходимо, избыточных дубликатов кортежей, состоящих из значений выбранных атрибутов (рисунок выше б).

Пример проекции:

Пусть задано отношение R вида:

Имя	Возраст	Вес
Анатолий	34	80
Иван	28	64
Петр	29	70
Евгений	54	54
Александр	34	80

Из отношения выбираются неповторяющиеся записи по атрибутам Возраст, Вес, тогда результатом проекции будет отношение R_1 вида:

Возраст	Вес
34	80
28	64
29	70
54	54

Операция соединения. Пусть $theta$ имеет тот же смысл, что и в операции селекции. Тогда $theta$ - соединением отношения A по атрибуту x с отношением B по атрибуту y называется множество всех кортежей t , таких, что t является конкатенацией какого-либо кортежа a , принадлежащего A , и какого-либо кортежа b , принадлежащего B , и предикат « $a.x \theta b.y$ » принимает значение „истина“. При этом атрибуты $A.x$ и $B.y$ должны быть определены на одном и том же домене, а оператор $theta$ должен иметь смысл для этого домена. Если оператор $theta$ - равенство, то соединение называется *экви-соединением*. Из этого определения следует, что результат экви-соединения должен включать два идентичных атрибута. Если один из этих атрибутов исключается, что можно осуществить с помощью проекции, результат называется *естественным соединением*. Под неуточненным термином «соединение» понимают естественное соединение. Операция «соединение» похожа на декартово произведение. Отличие состоит в том, что декартово произведение предполагает сцепление каждого кортежа из отношения A с каждым кортежем из B , а в операции соединения кортеж из отношения A сцепляется только с теми кортежами из B , для которых выполнено условие, например, « $a.x = b.y$ ». Диаграмма операции приведена на рисунке выше в.

Пример соединения:

Пусть заданы два отношения R_1 и R_2 вида:

R_1 (мультфильмы)

Код мультфильма	Название мультфильма	Название канала
0	Симпсоны	2x2
1	Том и Джери	2x2
2	Ну погоди	RenTV

R_2 (каналы)

Код канала	Частота
RenTV	3,14
2x2	783,25

Выполняется соединение этих двух отношений с условием совпадения названия канала с кодом канала (код канала = название канала).

Первый этап — декартово произведение. Результат произведения — отношение R_3 :

Код мультфильма	Название мультфильма	Название канала	Код канала	Частота
0	Симпсоны	2x2	RenTV	3,14
0	Симпсоны	2x2	2x2	783,25
1	Том и Джери	2x2	RenTV	3,14
1	Том и Джери	2x2	2x2	783,25
2	Ну погоди	RenTV	RenTV	3,14
2	Ну погоди	RenTV	2x2	783,25

Второй этап — выборка

Код мультфильма	Название мультфильма	Название канала	Код канала	Частота
0	Симпсоны	2x2	2x2	783,25
1	Том и Джери	2x2	2x2	783,25
2	Ну погоди	RenTV	RenTV	3,14

Операция деление. В простейшей форме операция деления делит отношение степени два (делимое) на отношение степени один (делитель) и создает (производит) результирующее отношение степени один (частное). Пусть делимое A имеет атрибуты x и y , а делитель B атрибут y (рисунок выше г). Атрибуты $A.y$ и $B.y$ должны быть определены на одном домене. Результатом деления A на B является отношение C с единственным атрибутом x , таким, что каждое значение x этого атрибута $C.x$ появляется как значение $A.x$, а пара значений (x,y) входит в A для *всех* значений y , входящих в B . Другими словами, кортеж включается в

Отформатировано: Отступ: Первая строка: 0 см

результатирующее отношение С только в том случае, если его декартово произведение с отношением В содержит отношение А.

Пример деления:

Пусть заданы два соотношения R_1 и R_2 вида:

R_1 (мультфильмы)

Код мультфильма	Название мультфильма	Название канала
0	Симпеоны	RenTV
0	Симпеоны	2x2
0	Симпеоны	СТС
1	Том и Джери	RenTV
1	Том и Джери	2x2
2	Ну погоди	СТС
2	Ну погоди	2x2

R_2 (каналы)

Название канала
RenTV
2x2

Отформатировано: русский

Отформатировано: русский

При делении R_1 на R_2 результатом будет

Код мультфильма	Название мультфильма
0	Симпеоны
1	Том и Джери

Отформатировано: Отступ: Первая строка: 0 см

~~«Симпеоны» и «Том и Джери» — мультфильмы, которые показывались на RenTV и 2x2. «Ну погоди» не показывался по RenTV, поэтому был исключен из результирующей таблицы.~~

Из восьми рассмотренных реляционных операций пять являются базовыми. Это - селекция, проекция, декартово произведение, объединение и разность. Остальные три операции могут быть определены через базовые. Например, естественное

соединение может быть выражено как проекция селекции декартова произведения.

Назначение реляционной операции присваивания состоит в том, чтобы сохранить значение какого-либо алгебраического выражения, например, проекции в виде заданного отношения (таблицы).

Операции реляционной модели данных предоставляют возможность произвольно манипулировать отношениями, позволяя обновлять БД, а также выбирать подмножества хранимых данных и представлять их в нужном виде. Таким образом, особенностями, определившими преимущества реляционной модели, являются:

а) множество объектов реляционной модели БД однородно - структура БД определяется только в терминах отношений;

б) основная единица обработки в операциях реляционной модели не запись (как в сетевых и иерархических моделях), а множество записей – отношение.

2. Объектная модель баз данных.

В последние годы все большее признание и развитие получают объектные базы данных (ОБД), толчок к появлению которых дало объектно-ориентированное программирование и использование компьютера для обработки и представления практически всех форм информации, воспринимаемых человеком.

Объектно-ориентированное программирование (ООП), в отличие от структурного, делает акцент не на программные структуры (циклы, условия и т.д.), а на объекты. Объектом называют почти все, что представляет интерес для решения поставленной задачи на компьютере. Это может быть экранное окно, кнопка в окне поле для ввода данных, пользователь программы, сама программа и т.д. Тогда любые действия можно привязать к такому объекту, а также описать, что произойдет с объектом при выполнении определенных действий (например, при „нажатии“ кнопки). Многократно используемый объект можно сохранить и применять его в различных программах.

Таким образом, при ООП создают необходимые объекты и описывают действия с ними и их реакцию на действия пользователя. Если создан и определен достаточно большой набор объектов, то написание программы будет состоять в том, чтобы

включить в нее и связать с собою те или иные объекты, обеспечивающие выполнение необходимых функций.

Объект - достаточно крупный блок функционально взаимосвязанных данных, при извлечении которого из ОБД включаются процедуры преобразования и отображения данных по программам, входящих в состав объекта. Типы и структуры данных, из которых состоит объект, могут быть различными у разных объектов и создаваться самим программистом на основе стандартных типов данных используемого языка программирования. Создаваемые и описываемые программистом типы данных получили название *абстрактных типов данных*.

Таким образом, *объектом* называется программно связанный набор методов (функций) и свойств, выполняющих одну функциональную задачу. Например, кнопка управления на экране Windows - это объект, который обладает свойствами описывающими его внешний вид и назначение, и набором методов для управления его поведением на экране.

Свойство - это характеристика, с помощью которой описывается внешний вид и работа объекта.

Событие - это действие, которое связано с объектом. Событие может быть вызвано пользователем (щелчок мышью), инициировано прикладной программой или операционной системой.

Метод - это функция или процедура, управляющая работой объекта при его реакции на событие.

Объекты могут быть как визуальными, т.е. которые можно увидеть на экране дисплея (окно, пиктограмма, текст и т.д.), так и не визуальные (например, программа решения какой-либо функциональной задачи).

Если набор объектов имеет описание (концептуальная модель), указаны свойства и логические связи между объектами (логическая модель) и известно их место нахождения в памяти ЭВМ (физическая модель), то это позволяет извлекать объекты и применять в соответствии с их назначением многими пользователями. Следовательно, организуется объектная база данных.

Создание объектов весьма трудоемкая программистская работа. Поэтому для облегчения труда прикладных программистов системными программистами созданы программы и развиваются

системы программирования поддерживающие ООП. В этих системах упорядочены и унифицированы многие процедуры создания объектов, разработаны шаблоны (классы) для описания методов и свойств объектов и т.д.

В настоящее время многие известные фирмы, занимающиеся разработкой программных продуктов, предлагают системы ООП. Например, широко известны такие продукты фирмы Microsoft, как Visual Basic, Visual FoxPro, Access, SQL Server. Такие системы не только упрощают создание объектов, но позволяют организовать ОБД и предоставляют средства работы с нею. Помимо поддержки ООП и ОБД, перечисленные системы дают возможность создания и манипулирования реляционной БД, что впрочем, является их основным назначением.

Объектные модели данных еще не имеют строгой теоретической основы (как, например, реляционные), что затрудняет их создание и использование. Однако развитие средств мультимедиа, вычислительных сетей и передаче по ним аудио- и видеообъектов, заставляют интенсифицировать поиски в направлениях как создания теории, так и практической реализации надежных систем объектных баз данных.

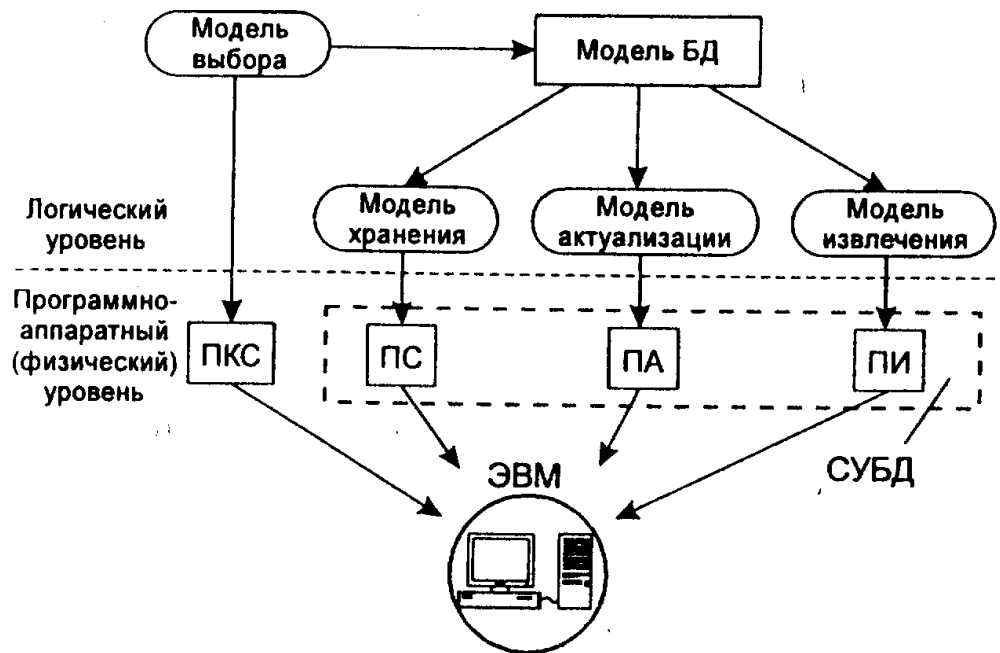
3. Программно-аппаратный уровень процесса накопления данных.

Логический (модельный) уровень процесса накопления связан с физическим через программы, осуществляющие создание канонической структуры БД, схемы её хранения и работу с данными. На рисунке ниже показан состав моделей процесса накопления данных и их связь с программно-аппаратным уровнем.

Каноническая структура БД создается с помощью модели выбора хранимых данных. Формализованное описание БД производится с помощью трех моделей: модели хранения данных (структура БД), модели актуализации данных и модели извлечения данных. На основе этих моделей разрабатываются соответствующие программы: создания канонической структуры БД (ПКС), создания структуры хранения БД (ПС), актуализации (ПА) и извлечения данных (ПИ).

Таким образом, переход к физической модели базы данных, реализуемой и используемой на компьютере, производится с

помощью системы программ, позволяющих создать в памяти ЭВМ (на магнитных и оптических дисках) базу хранимых данных и работать с этими данными, т.е. извлекать, изменять, дополнять, уничтожать. Эти программы называются СУБД (системы управления базами данных). На рисунке ниже программы, входящие в СУБД заключены в пунктирный прямоугольник.



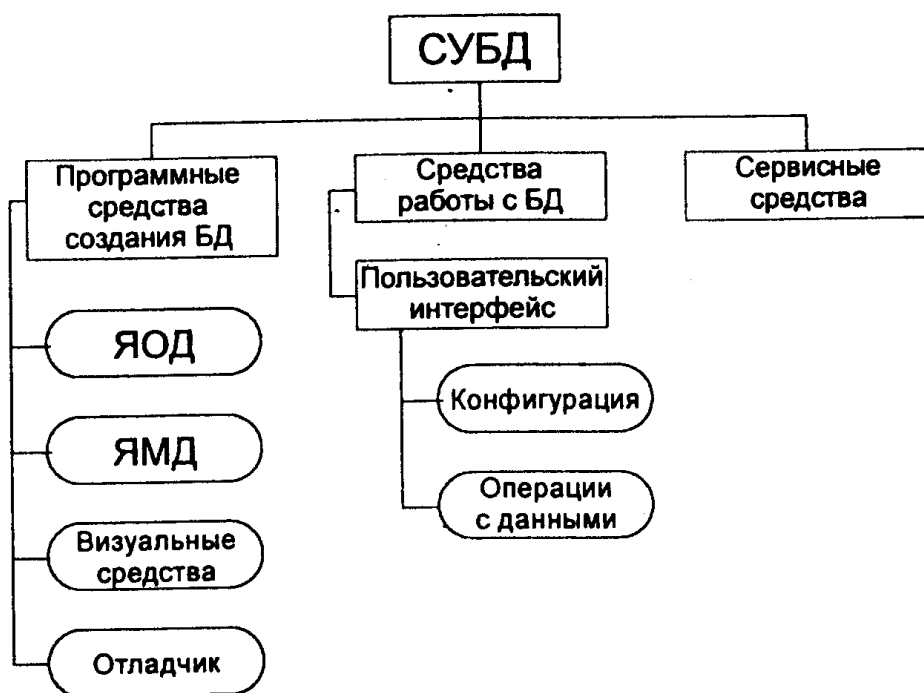
Состав моделей и программ процесса накопления

Современная СУБД содержит в своем составе программные средства создания баз данных, средства работы с данными и дополнительные, сервисные средства (рисунок ниже). С помощью средств создания БД проектировщик, используя язык описания данных (ЯОД), переводит логическую модель БД в физическую структуру, а на языке манипуляции данными (ЯМД) разрабатывает программы, реализующие основные операции с данными (в реляционных БД это реляционные операции). При проектировании привлекаются визуальные средства, т.е. объекты, и программа-отладчик, с помощью которой соединяются и тестируются отдельные блоки разработанной программы управления конкретной БД.

Средства работы с данными предназначены для пользователя БД. Они позволяют установить удобный (как правило, графический многооконный) интерфейс с пользователем, создать необходимую функциональную конфигурацию экранного представления выводимой и вводимой информации (цвет, размер и количество окон, пиктограммы пользователя и т.д.), производить операции с данными БД, манипулируя текстовыми и графическими экранными объектами.

Дополнительные (сервисные) средства позволяют при проектировании и использовании БД привлечь к работе с БД другие системы. Например, воспользоваться текстом из системы редактирования Word или таблицей из табличной системы Excel, или обратиться к сетевому серверу.

СУБД принципиально различаются по моделям БД, с которыми они работают. Если модель БД реляционная, то нужно использовать реляционную СУБД, если сетевая - сетевую СУБД, и т.д.



Состав СУБД

В технологическом, информационном процессе *накопления* данных наибольший вес имеют базы данных как независимые от прикладных программ хранилища данных. Однако, это не единственный способ накопления данных. Напомним, что любой вид представления информации, будь то числа, текст, программа, изображение, графический объект или звук, в ЭВМ превращается в двоичные коды - данные. Одной из форм хранения данных на дисках компьютеров, является файловая форма. Она по-прежнему широко распространена и поддерживается всеми современными операционными системами. Файл - это теоретически неограниченный, статистический набор данных, физически расположенный на магнитном или оптическом диске, имеющий уникальное имя и метки начала и конца. Файлы не имеют между собой функциональной связи, но для облегчения их поиска и проведения необходимых операций, таких запись, копирование, переименование, удаление и т.п., они имеют иерархическую логическую организацию, создаваемую операционной системой компьютера. Современные операционные системы представляют пользователю разнообразный набор графических, экранных средств манипуляции файлами.

Данные, полученные в процессе накопления, используются в информационной технологии для процессов обработки и обмена.

Третий важнейший информационный процесс - процесс обмена данными.

Лекция 14.

Язык структурированных запросов (SQL) в реляционных базах данных.

1. Общие сведения об SQL.

2. Применение SQL-запросов в традиционных реляционных операциях.

3. Применение SQL-запросов в специальных реляционных операциях.

1. Общие сведения об SQL.

SQL (Structured Query Language) – стандартный язык, предназначенный для создания, модификации и управления данными в реляционных БД. Реализация в SQL концепции операций, ориентированных на табличное представление данных, позволила создать компактный язык с небольшим набором предложений.

Основные категории команд языка SQL предназначены для выполнения различных функций, включая построение объектов базы данных и манипулирование ими, начальную загрузку данных в таблицы, обновление и удаление существующей информации, выполнение запросов к базе данных, управление доступом к ней и ее общее администрирование.

Основные категории команд языка SQL:

DDL – язык определения данных;

DML – язык манипулирования данными;

DQL – язык запросов ;

DCL – язык управления данными;

команды администрирования данных;

команды управления транзакциями.

Язык определения данных (DDL) позволяет создавать и изменять структуру объектов базы данных, например, создавать и удалять таблицы. Основными командами языка DDL являются следующие:

CREATE TABLE – создать таблицу,

Отформатировано: русский

Отформатировано: русский

Отформатировано: Обычный, без нумерации

Отформатировано: нумерованный + Уровень: 1 + Стиль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

Отформатировано: Обычный, без нумерации

Отформатировано: Шрифт: не полужирный

Отформатировано: Шрифт: не полужирный, русский

Отформатировано: Шрифт: не полужирный

Отформатировано: Шрифт: не полужирный, русский

Отформатировано: Шрифт: не полужирный

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: не полужирный

Отформатировано: Отступ: Первая строка: 0 см

Отформатировано: без нумерации

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: русский

Отформатировано: русский

Отформатировано: русский

Отформатировано: русский

ALTER TABLE – изменить таблицу, например, добавить или удалить столбцы

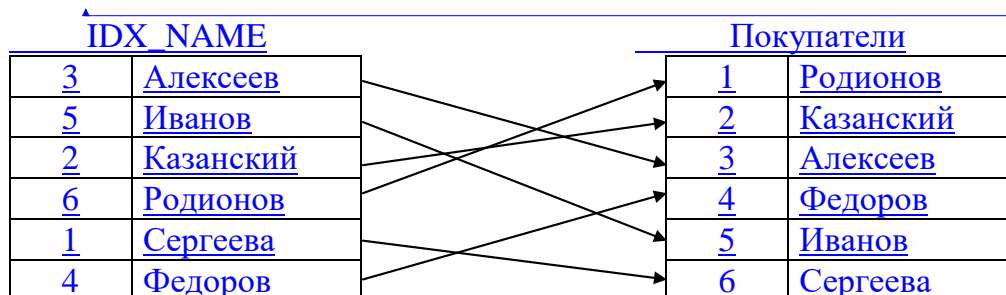
DROP TABLE – удалить таблицу,

CREATE INDEX, ALTER INDEX, DROP INDEX – соответственно команды создания, изменения и удаления индексов.

Данные запоминаются в таблице в том порядке, в котором они вводятся пользователем. Это, так называемый, физический порядок следования записей. Однако, часто требуется представить данные в другом, отличном от физического, порядке. Например, может потребоваться просмотреть данные об авторах книг, упорядоченные по алфавиту. Кроме того, часто необходимо найти в большом объеме информации запись, удовлетворяющую определенному критерию. Простой перебор записей при поиске в большой таблице может потребовать достаточно много времени и поэтому будет неэффективным. Одним из основных требований, предъявляемых к СУБД, является возможность представления данных в определенном, отличном от физического, порядке, и возможность быстрого поиска определенной записи. Эффективным средством решения этих задач является использование индексов.

Индекс представляет собой таблицу, которая содержит ключевые значения для каждой записи в таблице данных и записанные в порядке, требуемом для пользователя. Ключевые значения определяются на основе одного или нескольких полей таблицы. Кроме того, индекс содержит уникальные ссылки на соответствующие записи в таблице.

На рисунке ниже показан пример соответствия индексной таблицы IDX_NAME и фрагмента таблицы, содержащей информацию о покупателях. Индекс IDX_NAME содержит значения ключевого поля Фамилия, упорядоченные в алфавитном порядке, и ссылки на соответствующие записи в таблице Покупатели.



Отформатировано: русский

Отформатировано: русский

Отформатировано: русский

Отформатировано: русский

Отформатировано: русский

Отформатировано: интервал
Перед: 0 пт, После: 0 пт, Узор: Нет

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто, английский (США)

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто, английский (США)

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто, английский (США)

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто, английский (США)

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто, английский (США)

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано: без нумерации

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Отформатировано

Связь между таблицей и индексом

Язык манипулирования данными (DML) используется для манипулирования информацией внутри объектов реляционной базы данных посредством трех основных команд:

INSERT - вставка,

UPDATE – обновление,

DELETE - удаление.

Язык запросов (DQL) наиболее известен пользователям реляционной базы данных, несмотря на то, что он включает всего одну команду SELECT. Эта команда вместе со своими многочисленными опциями и предложениями используется для формирования запросов к реляционной базе данных.

Команды управления данными позволяют управлять доступом к информации, находящейся внутри базы данных. Как правило, они используются для создания объектов, связанных с доступом к данным, а также служат для контроля над распределением привилегий между пользователями. Команды управления данными следующие:

GRANT – предоставляет доступ к защищенному объекту определенному пользователю,

REVOKE – удаляет разрешение, выданное или запрещенное ранее.

С помощью команд администрирования данных пользователь осуществляет контроль за выполняемыми действиями и анализирует операции базы данных. Они также могут оказаться полезными при анализе производительности системы. Не следует путать администрирование данных с администрированием базы данных, которое представляет собой общее управление базой данных и подразумевает использование команд всех уровней.

Транзакция (transaction) — группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными. Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта. Транзакции обрабатываются транзакционными системами.

Пример: необходимо перевести с банковского счёта номер 5 на счёт номер 7 сумму в 10 денежных единиц. Этого можно достичь, к примеру, приведённой последовательностью действий:

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, Цвет шрифта: Авто

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см, интервал После: 0 пт

Отформатировано

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: интервал После: 0 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: По ширине, Отступ: Первая строка: 1,25 см, интервал После: 0 пт

Отформатировано

Отформатировано: По ширине, Отступ: Первая строка: 1,25 см

Отформатировано

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: По ширине

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: По ширине, Отступ: Первая строка: 1,25 см, интервал После: 0 пт

Отформатировано

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, не полужирный, Цвет шрифта: Авто

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см

Отформатировано

Отформатировано

<u>Елена</u>	<u>54</u>	<u>54</u>
<u>Петр</u>	<u>34</u>	<u>80</u>

Персонажи

<u>Имя</u>	<u>Возраст</u>	<u>Вес</u>
<u>Дональд</u>	<u>24</u>	<u>19</u>
<u>Том</u>	<u>25</u>	<u>23</u>
<u>Джерри</u>	<u>81</u>	<u>27</u>
<u>Георгий</u>	<u>29</u>	<u>70</u>
<u>Елена</u>	<u>54</u>	<u>54</u>

Результатом пересечения этих отношений будет отношение:

<u>Имя</u>	<u>Возраст</u>	<u>Вес</u>
<u>Георгий</u>	<u>29</u>	<u>70</u>
<u>Елена</u>	<u>54</u>	<u>54</u>

SQL-запрос для пересечения данных отношений следующий:

SELECT Имя, Возраст, Вес FROM Персоны
INTERSECT
SELECT Имя, Возраст, Вес FROM Персонажи;

Результатом разности отношений А и В будет отношение с тем же заголовком, что и у совместимых по типу отношений А и В, и телом, состоящим из кортежей, принадлежащих отношению А и не принадлежащих отношению В.

Результатом разности отношений Персон и Персонажей предыдущего примера будет отношение вида:

<u>Имя</u>	<u>Возраст</u>	<u>Вес</u>
<u>Александр</u>	<u>34</u>	<u>80</u>
<u>Иван</u>	<u>28</u>	<u>64</u>
<u>Петр</u>	<u>34</u>	<u>80</u>

SQL-запрос для разности отношений следующий:

SELECT Имя, Возраст, Вес FROM Персоны
EXCEPT

Отформатировано: По левому краю

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: По ширине

Отформатировано: русский

Отформатировано: Шрифт: Times New Roman, 16 пт, без подчеркивания

Отформатировано: По ширине, Отступ: Первая строка: 1,25 см, Междустр.интервал: одинарный

Отформатировано: без подчеркивания, русский

Отформатировано: По ширине, Междустр.интервал: одинарный

Отформатировано: Отступ: Слева: 0 см

SELECT Имя, Возраст, Вес FROM Персонажи;

Отформатировано: русский

При выполнении декартового произведения двух отношений получается отношение, кортежи которого являются конкатенацией (сцеплением) кортежей первого и второго операндов. Пусть даны следующие отношения:

Отформатировано: без подчеркивания

Отформатировано: Шрифт: Times New Roman, 16 пт, не полужирный, без подчеркивания

Отформатировано: без подчеркивания

Отформатировано: По ширине, Отступ: Первая строка: 1,25 см, интервал Перед: 0 пт, После: 0 пт, Междустр.интервал: одинарный

Отформатировано: Шрифт: Times New Roman, 16 пт, без подчеркивания

Мультфильмы

Код мультфильма	Название мультфильма
0	Симпсоны
1	Том и Джерри
2	Ну погоди

Каналы

Код канала	Название канала
0	СТС
1	2x2

Отформатировано: Шрифт: Times New Roman, 16 пт

Отформатировано: По ширине, интервал Перед: 0 пт, После: 0 пт, Междустр.интервал: одинарный

Отформатировано: Отступ: Слева: 0 см

Результатом декартового произведения отношений Мультфильмы и Каналы будет отношение вида:

Отформатировано: Отступ: Слева: 0 см

Отформатировано: Отступ: Слева: 0 см, Первая строка: 1,25 см

Код мультфильма	Название мультфильма	Код канала	Название канала
0	Симпсоны	0	СТС
0	Симпсоны	1	2x2
1	Том и Джерри	0	СТС
1	Том и Джерри	1	2x2
2	Ну погоди	0	СТС
2	Ну погоди	1	2x2

Отформатировано: русский

Отформатировано: Отступ: Слева: 0 см

Отформатированная таблица

Отформатировано: английский (США)

Отформатировано: Отступ: Слева: 0 см

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: По ширине

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: русский

SQL-запрос для декартового произведения вышеуказанных отношений следующий:

SELECT * FROM Мультфильмы, Каналы;

Здесь «*» означает, что выбираются все поля из первой и второй таблиц.

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

3. Применение SQL-запросов в специальных реляционных операциях.

Отформатировано: нумерованный + Уровень: 1 + Стиль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

Рассмотрим примеры специальных операций реляционной базы данных.

Отформатировано: Обычный, без нумерации

Результатом селекции (выборки) из некоторого отношения R будет отношение, для которого выполняется заданное данной выборкой условие. Пусть задано отношение:

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Персоны

<u>Имя</u>	<u>Возраст</u>	<u>Вес</u>
<u>Александр</u>	<u>34</u>	<u>80</u>
<u>Иван</u>	<u>28</u>	<u>64</u>
<u>Георгий</u>	<u>29</u>	<u>70</u>
<u>Елена</u>	<u>54</u>	<u>54</u>
<u>Петр</u>	<u>34</u>	<u>80</u>

Отформатировано: Шрифт: не полужирный, без подчеркивания, Цвет шрифта: Авто

Необходимо выбрать те записи отношения, у которых возраст не моложе 34 лет (Возраст \geq 34).

Отформатировано: Обычный, без нумерации

Результатом селекции будет следующее отношение:

<u>Имя</u>	<u>Возраст</u>	<u>Вес</u>
<u>Александр</u>	<u>34</u>	<u>80</u>
<u>Елена</u>	<u>54</u>	<u>54</u>
<u>Петр</u>	<u>34</u>	<u>80</u>

Отформатировано: Обычный, без нумерации

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: русский

Отформатировано: Обычный, без нумерации

Отформатированная таблица

Отформатировано: Обычный, без нумерации

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Обычный, без нумерации

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

Отформатировано: английский (США)

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

SQL-запрос для данной выборки следующий:

SELECT * FROM Персоны WHERE Возраст \geq 34;

Результатом проекции из некоторого отношения R является выбор определенных полей с удалением, если это необходимо, избыточных дубликатов записей, состоящих из одинаковых значений выбранных полей.

Отформатировано: без подчеркивания, Цвет шрифта: Авто, русский

Пусть нужно выбрать из вышеуказанного отношения Персоны поля Возраст, Вес. Тогда результатом данной проекции будет следующее отношение:

<u>Возраст</u>	<u>Вес</u>
<u>28</u>	<u>64</u>
<u>29</u>	<u>70</u>
<u>34</u>	<u>80</u>
<u>54</u>	<u>54</u>

SQL-запрос для данной проекции следующий:

SELECT DISTINCT Возраст, Вес FROM Персоны;

Примечательно, что в SQL для полного соответствия операции проекции необходимо указывать ключевое слово DISTINCT, поскольку без него строка с возрастом 34 и весом 80 отобразится дважды.

Операция соединения есть результат последовательного применения операций декартового произведения и выборки. Если в отношениях и имеются атрибуты с одинаковыми наименованиями, то перед выполнением соединения такие атрибуты необходимо переименовать. Пусть заданы два отношения R_1 и R_2 вида:

R_1 (мультфильмы)

<u>Код мультфильма</u>	<u>Название мультфильма</u>	<u>Название канала</u>
<u>0</u>	<u>Симпсоны</u>	<u>2x2</u>
<u>1</u>	<u>Том и Джери</u>	<u>2x2</u>
<u>2</u>	<u>Ну погоди</u>	<u>RenTV</u>

R_2 (каналы)

<u>Код канала</u>	<u>Частота</u>
<u>RenTV</u>	<u>3,14</u>
<u>2x2</u>	<u>783,25</u>

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: По центру

Отформатировано: По центру

Отформатировано: По центру

Отформатировано: По центру

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Обычный, без нумерации

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

Отформатировано: без подчеркивания, Цвет шрифта: Авто, английский (США)

Отформатировано: английский (США)

Отформатировано: Шрифт: Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Шрифт: 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

Выполняется соединение этих двух отношений с условием совпадения названия канала с кодом канала (код канала = название канала).

Первый этап – декартово произведение. Результат произведения – отношение R_3 :

<u>Код мультфильма</u>	<u>Название мультфильма</u>	<u>Название канала</u>	<u>Код канала</u>	<u>Частота</u>
<u>0</u>	<u>Симпсоны</u>	<u>2x2</u>	<u>RenTV</u>	<u>3,14</u>
<u>0</u>	<u>Симпсоны</u>	<u>2x2</u>	<u>2x2</u>	<u>783,25</u>
<u>1</u>	<u>Том и Джери</u>	<u>2x2</u>	<u>RenTV</u>	<u>3,14</u>
<u>1</u>	<u>Том и Джери</u>	<u>2x2</u>	<u>2x2</u>	<u>783,25</u>
<u>2</u>	<u>Ну погоди</u>	<u>RenTV</u>	<u>RenTV</u>	<u>3,14</u>
<u>2</u>	<u>Ну погоди</u>	<u>RenTV</u>	<u>2x2</u>	<u>783,25</u>

Второй этап – выборка. В итоге результатом соединения вышеуказанных отношений будет отношение:

<u>Код мультфильма</u>	<u>Название мультфильма</u>	<u>Название канала</u>	<u>Код канала</u>	<u>Частота</u>
<u>0</u>	<u>Симпсоны</u>	<u>2x2</u>	<u>2x2</u>	<u>783,25</u>
<u>1</u>	<u>Том и Джери</u>	<u>2x2</u>	<u>2x2</u>	<u>783,25</u>
<u>2</u>	<u>Ну погоди</u>	<u>RenTV</u>	<u>RenTV</u>	<u>3,14</u>

SQL-запрос для данной операции соединения следующий:

```
SELECT *
FROM Мультфильмы, Каналы
WHERE Название_канала=Код_канала;
```

Реляционное деление достаточно нетривиально описать, но на примере его смысл нагляден. В целом, из таблицы А берутся значения строк, для которых присутствуют все комбинации значений из таблицы В. Пример деления:

Пусть заданы два соотношения R_1 и R_2 вида:

Отформатировано: Шрифт: (по умолчанию) Times New Roman, русский

Отформатировано: русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman, русский

Отформатировано: Обычный, без нумерации

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, русский

Отформатировано: русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman

Отформатировано: русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman

Отформатировано: Шрифт: (по умолчанию) Times New Roman, русский

Отформатировано: русский

Отформатировано: Шрифт: не полужирный

Отформатировано: Обычный, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Цвет шрифта: Серый 80%

R₁ (мультфильмы)

<u>Код мультфильма</u>	<u>Название мультфильма</u>	<u>Название канала</u>
<u>0</u>	<u>Симпсоны</u>	<u>RenTV</u>
<u>0</u>	<u>Симпсоны</u>	<u>2x2</u>
<u>0</u>	<u>Симпсоны</u>	<u>СТС</u>
<u>1</u>	<u>Том и Джери</u>	<u>RenTV</u>
<u>1</u>	<u>Том и Джери</u>	<u>2x2</u>
<u>2</u>	<u>Ну погоди</u>	<u>СТС</u>
<u>2</u>	<u>Ну погоди</u>	<u>2x2</u>

R₂ (каналы)

<u>Название канала</u>
<u>RenTV</u>
<u>2x2</u>

При делении R₁ на R₂ результатом будет

<u>Код мультфильма</u>	<u>Название мультфильма</u>
<u>0</u>	<u>Симпсоны</u>
<u>1</u>	<u>Том и Джери</u>

«Симпсоны» и «Том и Джери» - мультфильмы, которые показывались на RenTV и 2x2. «Ну погоди» не показывался по RenTV, поэтому был исключен из результирующей таблицы. Операция деления на практике используется достаточно редко, простейшего SQL-запроса операции деления не существует.

Отформатировано: Обычный, без нумерации

Отформатировано: Шрифт: 16 пт, полужирный

Отформатировано: По ширине

Отформатировано: Шрифт: (по умолчанию) Times New Roman, русский

Отформатировано: Шрифт: не полужирный, русский

Лекция 15.

Информационный процесс обмена данными.

- 1. Назначение технологического процесса обмена данными.**
- 2. Адресация компьютеров в вычислительных сетях.**
- 3. Структуризация сети.**

- 1. Назначение технологического процесса обмена данными.**

Информационный процесс обмена данными происходит в любой вычислительной системе.

Например, в персональном компьютере через системную (магистральную) шину производится обмен данными, их адресами и командами между оперативной памятью и процессором. К этой же шине через контроллеры (согласующие устройства) подключены внешние устройства (дисплей, клавиатура, накопители на гибких и жестких магнитных и оптических дисках, манипуляторы и т.д.), которые обмениваются данными с оперативной памятью. Обмен данными между устройствами ЭВМ обусловлены ограничениями функций, выполняемых этими устройствами, и должен быть запрограммирован. Выполняемая программа хранится в оперативной памяти компьютера и через системную шину передаёт в процессор команды на выполнение определенных операций. Процессор на их основе формирует свои команды управления, которые по системной шине поступают на соответствующие устройства. Для выполнения операций обработки данных процессор передаёт в оперативную память адреса необходимых данных и получает их. Результаты обработки направляются в оперативную память. Данные из оперативной памяти могут быть переданы на хранение во внешние запоминающие устройства, для отображения на дисплее или принтере, для передачи в вычислительную сеть. Напомним, что программа, адреса, команды, собственные данные в компьютере имеют одну и ту же двоичную форму представления и обрабатываются, хранятся и передаются с помощью одних и тех же устройств.

Таким образом, в компьютере все три основных информационных процесса (обработки, накопления и обмена) тесно

связаны на основе общности среды передачи (системная шина) и устройств обработки и накопления.

Процессами обмена данными в компьютере управляет операционная система совместно с прикладными программами (приложениями).

В компьютерах любого класса (ПК, серверы, миникомпьютеры, мэйнфреймы) информационные процессы предельно локализованы и их физическое протекание ограничено размером конструкции ЭВМ. Поэтому процесс обмена, являющийся в ЭВМ связующим между процессами обработки и накопления, реализуется относительно просто через системную шину (шина - это жгут проводов, число которых зависит от разрядности ЭВМ) небольшой протяженности, соединяющую процессор и оперативную память непосредственно. Внешние устройства подключаются к ней через контроллеры, выполняющие функции согласования форматов данных и электрических уровней сигналов. На физическом уровне представления информационных технологий компьютер может быть специализирован для выполнения отдельных технологических информационных процессов. Так, в настоящее время созданы специальные компьютеры, называемые «хранилища данных», главное назначение которых *накапливать* громадные объёмы данных. Многопроцессорные архитектуры, реализующие параллельную и конвейерную обработку данных, предназначены для максимизации производительности процесса *обработки*. Технологическая же природа процесса *обмена* данными в современных информационных технологиях такова, что не может быть реализована на одном специализированном компьютере.

Выделению процесса обмена, как базового в информационной технологии способствует бурное развитие вычислительных сетей, как локальных, так и распределенных, включая глобальную сеть Internet.

Системы, состоящие из двух и более компьютеров, разнесенных в пространстве и объединенных линиями связи, называют распределительными вычислительными системами или *сетями ЭВМ*. Именно в таких системах процесс обмена данными реализуется в наиболее полном виде и составляет основу функционирования *открытых систем*. Под открытыми системами в современном мире понимается концепция объединения с

помощью процессов обмена данными информационного ресурса мирового сообщества. В более узком смысле - это информационно-вычислительные сети, к которым может подключиться через компьютер любой человек Земли, любая организация, корпорация, фирма и т.д., и воспользоваться информационными ресурсами этой системы или предложить ей свой информационный ресурс.

Наиболее ярким представителем такой системы является мировая вычислительная сеть Internet. Её ещё называют «сеть сетей», так как она объединяет многие открытые системы (сети) на всех континентах нашей планеты.

Таким образом, информационный процесс обмена данными включает в себя две основных составляющие: передачу данных и организацию сети.

2. Адресация компьютеров в вычислительных сетях.

При объединении компьютеров в вычислительную сеть необходима уникальная идентификация каждого из них как в локальной сети, так и в глобальной. Современные сети могут включать в себя огромное количество компьютеров, разбиваться на подсети, при этом необходимо сохранить уникальность адресации.

К адресу узла сети и схеме его назначения целесообразно предъявить несколько требований:

- адрес должен *уникально идентифицировать* компьютер в сети любого масштаба;
- схема назначения адресов должна *сводить к минимуму ручной труд* администратора и вероятность дублирования адресов;
- адрес должен *иметь иерархическую структуру*, удобную для построения больших сетей;
- адрес должен быть удобен для пользователей сети, а это значит, что он должен *иметь символьное представление*;
- адрес должен по возможности *иметь компактное представление*, чтобы не перегружать память коммуникационной аппаратуры.

Все перечисленные требования трудно совместить в рамках какой-либо одной схемы адресации, и на практике обычно используется сразу несколько схем, так что компьютер одновременно имеет несколько адресов-имен.

Каждый адрес используется в той ситуации, когда соответствующий вид адресации наиболее удобен.

Различают следующие типы адресов:

- аппаратные (hardware) адреса (MAC-адреса);
- числовые составные адреса (IP-адреса);
- символьные адреса или имена (DNS-имена).

MAC-адреса предназначены для сети небольшого или среднего размера, поэтому они *не имеют иерархической структуры*. Типичным представителем адреса такого типа является *адрес сетевого адаптера локальной сети*.

Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного значения, например 00-1B-38-2C-F0-28.

При задании аппаратных адресов обычно не требуется выполнение ручной работы, так как они *встраиваются в аппаратуру компанией-изготовителем, причем гарантируется уникальность адреса в пределах любой сети*.

Помимо отсутствия иерархии, использование аппаратных адресов связано еще с одним недостатком – при замене аппаратуры, например, сетевого адаптера, изменяется и MAC-адрес компьютера. При установке нескольких сетевых адаптеров у компьютера появляется несколько MAC-адресов.

Для работы в больших сетях в качестве адресов узлов сети используют числовые составные адреса фиксированного и компактного форматов.

Типичными представителями адресов этого типа являются IP-адреса. В них *поддерживается двухуровневая иерархия*, адрес делится на старшую часть – номер сети и младшую – номер узла. Например, 192.168.0.1, где 192.168.1 – номер сети, а 1 – номер узла. Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется только после доставки сообщения в нужную сеть.

Символьные адреса *предназначены для запоминания людьми*, поэтому обычно *несут смысловую нагрузку*. Символьные адреса *легко использовать как в небольших, так и крупных сетях*.

Для работы в больших сетях *символьное имя может иметь сложную иерархическую структуру*, например server.it.net. Этот

адрес говорит о том, что данный компьютер является сервером домена «it» в некоторой сети «net».

При работе в пределах домена it.net (небольшая сеть) длинное символьное имя явно избыточно и вместо него удобно пользоваться кратким символьным именем server.

3. Структуризация сети.

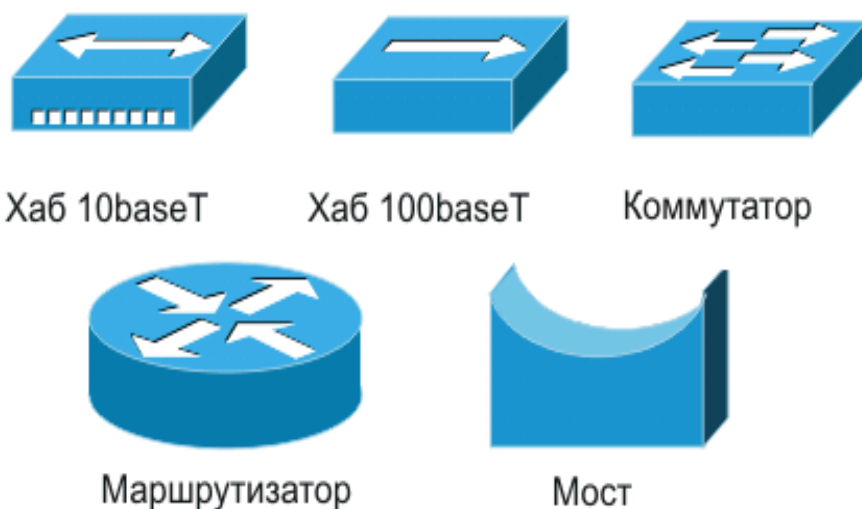
Одной из первых задач, которая стоит перед любой технологией транспортировки данных, является возможность их передачи на максимально большое расстояние. Физическая среда накладывает на этот процесс свое ограничение – рано или поздно мощность сигнала падает, и прием становится невозможным.

В таких ситуациях применяют не усиление, а повторение сигнала. Повторение сигнала обеспечивается активными сетевыми устройствами, которые также выполняют определенные дополнительные задачи.

В зависимости от решаемых задач различают следующие типы активных сетевых устройств:

- концентраторы (хабы);
- мосты;
- коммутаторы;
- маршрутизаторы.

На рисунке ниже представлено общепринятое схематичное изображение активных сетевых устройств.



Активные сетевые устройства

Основное назначение концентраторов (хабов) – это объединение территориально сосредоточенных рабочих мест в рабочую группу. Они используются в локальных вычислительных сетях (ЛВС). Но вполне возможно использование хабов в качестве ретрансляторов между удаленными сетями или связи нескольких рабочих групп.



Использование концентраторов (хабов) в локальных сетях

Принцип работы концентратора с технологией локальной сети Ethernet заключается в том, что принимая сигнал на один из своих портов, он ретранслирует этот сигнал на все остальные порты. В сетях Ethernet концентратор реализует логическую общую шину.

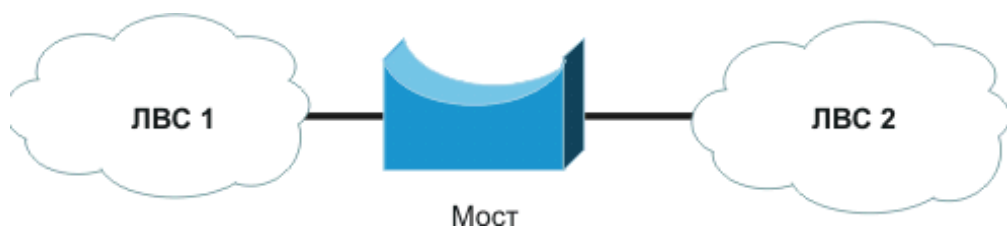
Основной недостаток концентраторов в сетях Ethernet – сам принцип логической общей шины, когда общая среда передачи занимается одним узлом на время передачи данных, при этом другие узлы ожидают освобождения среды.

Коллизия – столкновение сигналов в общей среде при одновременной отправке данных несколькими узлами. Коллизия является обычной ситуацией в сетях Ethernet. Разрешение такой ситуации возлагается на сетевые адаптеры.

Домен коллизий – группа узлов, взаимодействующая по принципу логической общей шины через концентратор. Чем больше узлов подключено в общей среде, тем больше вероятность возникновения коллизии.

Первым решением, призванным снизить частоту коллизий в сетях Ethernet стал мост. Мост разделяет одну ЛВС на 2 домена коллизий (рисунок ниже).

В частности, мост ведет таблицу MAC-адресов, в которой определяется соответствие каждого MAC-адреса одной из подключенных ЛВС. В результате сигнал, сгенерированный узлом ЛВС1 и адресованный другому узлу ЛВС1, не распространяется в ЛВС2, что уменьшает нагрузку на сеть.



Разделение сетей с помощью моста

Коммутатор (switch) – это многопортовый мост (более 2-х портов), к каждому порту которого может быть подключен как отдельный узел, так и ЛВС.

Данные, которые поступают на вход коммутатора (source port), направляется не на все активные порты (как это делает концентратор), а только на тот, к которому подключено устройство с MAC-адресом, совпадающим с адресом назначения кадра (destination port).

Коммутатор, подобно мосту, ведет таблицу MAC-адресов, на основании которой принимает решение о ретрансляции кадра.

На рисунке ниже представлена схема, в которой коммутатор решает задачу взаимодействия двух сегментов ЛВС (доменов коллизий), а также отдельно подключенного сервера.



Коммутаторы и мосты ведут таблицу MAC-адресов, на основании которой принимают решение о ретрансляции данных.

Маршрутизатор – это устройство, обеспечивающее выбор маршрута передачи данных, а также решающее задачу объединения разнородных сетей.

В отличие от концентраторов, мостов и коммутаторов, маршрутизаторы работают с IP-адресами. Классический пример применения маршрутизатора – на границе между ЛВС и Интернет (рисунок ниже).



Использование маршрутизатора для выхода в сеть Интернет

Каждый компьютер ЛВС имеет свой MAC-адрес, а также IP-адрес с номером сети, например, 192.168.0.x (последний символ «x» – номер узла). Если передаваемый пакет данных адресован в другую сеть, например, в 220.95.78.x, то он автоматически отправляется на маршрутизатор (шлюз по умолчанию), который направляет пакет на другой маршрутизатор в сеть Интернет.

Маршрутизаторы обмениваются друг с другом *маршрутной информацией* по определенному *протоколу маршрутизации*. На основании маршрутной информации составляются *таблицы*

маршрутизации, которые используются для определения оптимального маршрута.

Лекция 16.

Взаимодействие компонентов сети в открытых системах.

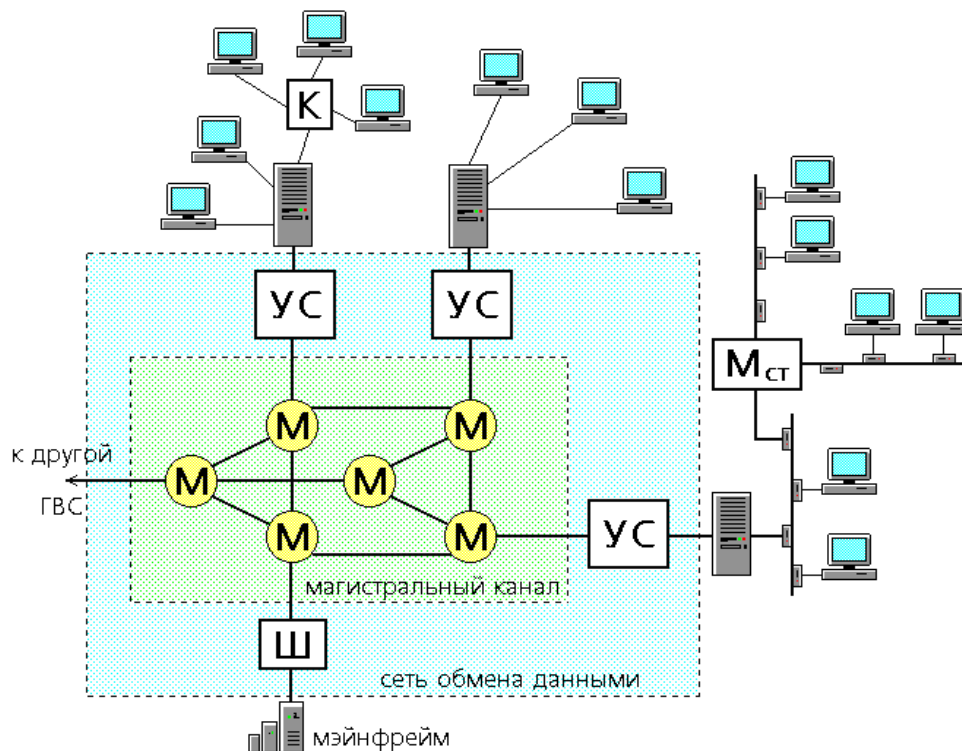
1. Топология глобальной сети.
2. Базовая эталонная модель взаимодействия открытых систем (OSI).
3. Представление заголовков данных в модели OSI Структуризация сети.

1. Топология глобальной сети.

Расширение локальных сетей, как базовых, так и комбинированных топологий, из-за удлинения линий связи приводит к необходимости их расчленения и создания *распределенных сетей*, в которых компонентами служат не отдельные компьютеры, а отдельные локальные сети, иногда называемыми «сегментами». Узлами коммутации таких сетей являются активные коммутаторы (К) или мосты - устройства, коммутирующие линии связи (—в том числе разного типа) и одновременно усиливающие проходящие через них сигналы. Мосты и коммутаторы кроме этого еще и управляют потоками данных между сегментами сети. При соединении компьютеров или сетей (локальных или распределенных), удаленных на большие расстояния, используются каналы связи и устройства коммутации, называемые маршрутизаторами (М) и шлюзами (Ш). Маршрутизаторы взаимодействуют друг с другом и соединяются между собой каналами связи, образуя распределенный магистральный канал связи. Для согласования параметров данных (форматов, уровней сигналов, протоколов и т.п.), передаваемых по магистральному каналу связи, между маршрутизаторами и терминальными компонентами включаются устройства сопряжения (УС). При подключении к магистральному каналу вычислительных сетей (например, мэйнфреймов), которых не возможно согласовать с помощью стандартных устройств сопряжения, используются стандартные средства, называемые шлюзами. Терминальными абонентами называют отдельные компьютеры, локальные или распределенные сети, через маршрутизаторы подключенные к магистральному каналу. Таким образом, возникает глобальная вычислительная сеть. Глобальные сети могут объединяться между собой путем соединения через

маршрутизаторы магистральных каналов, что в конечном итоге приводит к созданию мировой (действительно глобальной) информационно - вычислительной сети.

Типовая топология описанной выше глобальной вычислительной сети приведена на рисунке ниже.



Типовая топология глобальной информационно-вычислительной сети (ГВС)

2. Базовая эталонная модель взаимодействия открытых систем (OSI).

Модель OSI (Open System Interconnection) Reference Model (эталонная модель взаимодействия открытых систем) основана на предложении Международной организации по стандартизации (International Organization for Standardization, ISO) как первый шаг к международной стандартизации протоколов, используемых в различных уровнях. Для краткости эту модель называют OSI.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: Обычный, без нумерации

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Базовая эталонная модель взаимодействия открытых систем является гибкой в том смысле, что допускает эволюцию сетей в зависимости от развития теории и новых технических достижений, а также обеспечивает постепенность перехода от существующих реализаций к новым стандартам.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе участвуют две стороны. При этом требуется организовать связь между ними в виде иерархий (правил), работающих на разных компьютерах.

Протокол – формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

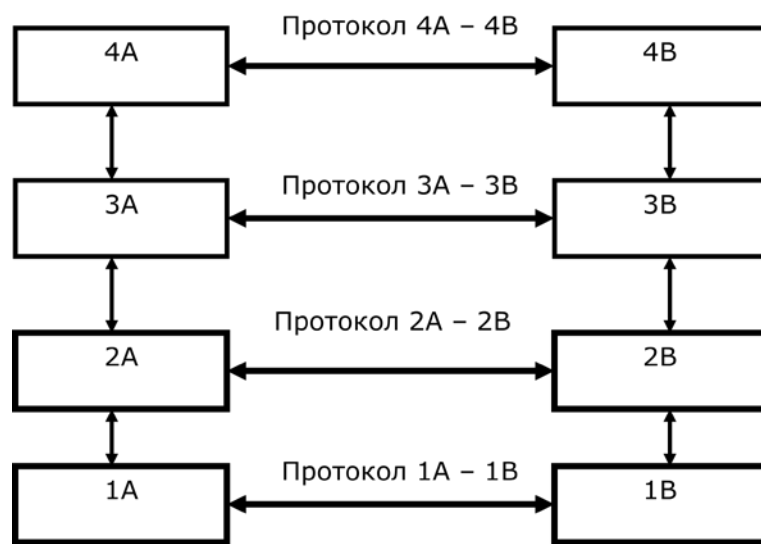
Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть интерфейсом.

Интерфейс определяет набор сервисов, предоставляемый данным уровнем соседнему уровню.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Рисунок ниже иллюстрирует взаимосвязь протоколов одного уровня разных узлов и интерфейсов одного узла.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто



Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Взаимосвязь сетевых протоколов и интерфейсов

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Обычный, без нумерации

Модель OSI имеет семь уровней. Появление именно семи уровней было обусловлено следующими принципами:

- уровень должен создаваться по мере необходимости отдельного уровня абстракции;
- каждый уровень должен выполнять строго определенную функцию;
- выбор функций для каждого уровня должен выбираться с учетом создания стандартизированных международных протоколов;
- границы между уровнями должны выбираться так, чтобы поток данных между интерфейсами был минимальным;
- количество уровней должно быть достаточно большим, чтобы различные функции не объединялись в одном уровне без необходимости, но не слишком высоким, чтобы архитектура не становилась громоздкой.

На рисунке ниже представлена схема эталонной модели взаимодействия открытых систем (OSI).

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

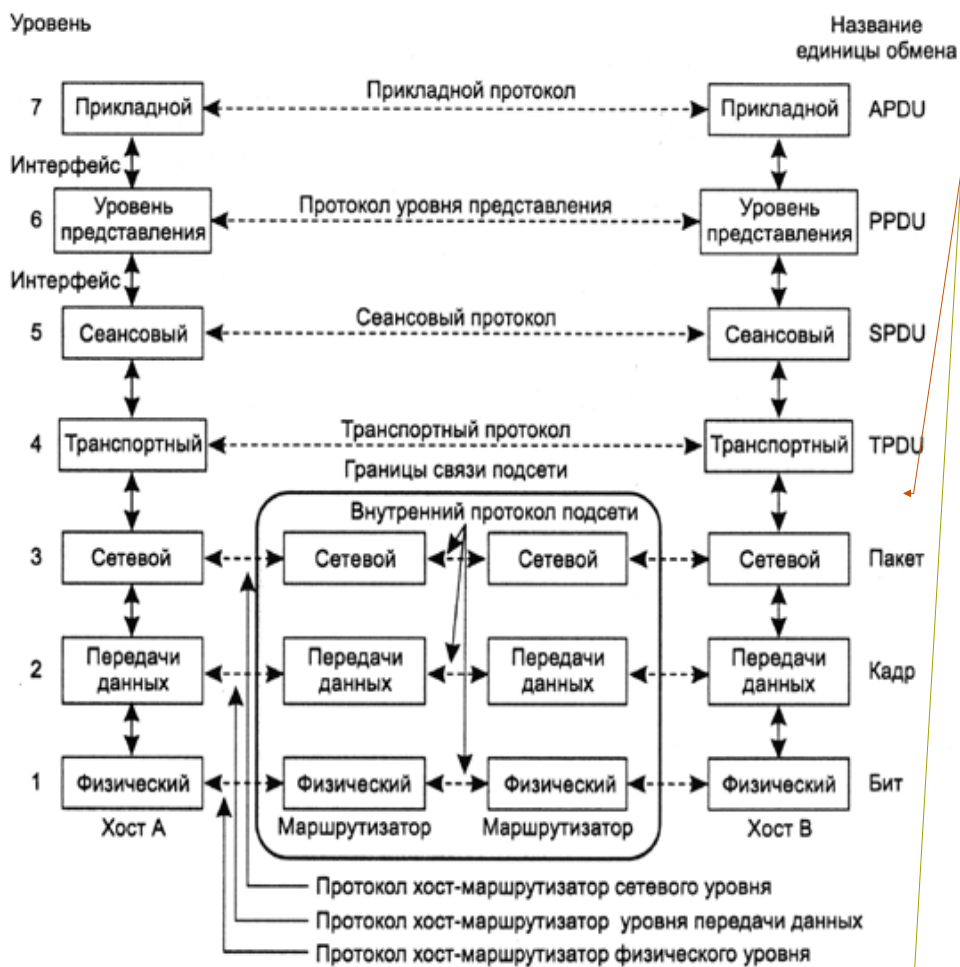
Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто, русский



Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: По левому краю, без нумерации

2.

Отформатировано: Шрифт: 16 пт

Отформатировано: Обычный, без нумерации

Уровни и протоколы базовой эталонной модели взаимодействия открытых систем

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Для передачи данных в сеть наборы битов и байтов объединяются в блоки фиксированного размера. На канальном уровне их называют кадрами, на сетевом – пакетами.

Отформатировано: Отступ: Первая строка: 1,25 см

Физический уровень (Physical layer) обеспечивает передачу двоичных разрядов по каналу связи. При разработке сети необходимо убедиться, что когда одна сторона передает единицу, то принимающая сторона получает также единицу, а не ноль. Здесь важно, какое напряжение должно использоваться для отображения единицы, а какое для нуля; сколько микросекунд длится бит; может

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

ли передача производится одновременно в двух направлениях; как устанавливается начальная связь и как она прекращается, когда обе стороны закончили свои задачи; из какого количества проводов должен состоять кабель и какова функция каждого провода.

Уровень передачи данных (Data Link layer) или канальный уровень решает следующую основную задачу – преобразование данных в формат, приемлемый для передачи по линии связи. Уровень выполняет эту задачу при помощи разбиения входных данных на кадры, обычно размером от нескольких сот до нескольких тысяч байтов. Поскольку физический уровень просто принимает и передает поток бит, невзирая на их значение или структуру, задача создания и распознавания границ кадров возложена на уровень передачи данных. Эта задача может быть выполнена при помощи добавления особых последовательностей битов к началу и концу кадра. Все проблемы, связанные с поврежденными, потерянными и дублированными кадрами, уровень передачи данных должен решать самостоятельно. Уровень передачи данных может предоставлять сетевому уровню услуги различного класса, с разным качеством.

Сетевой уровень (Network layer) обеспечивает единый способ присвоения узлам сети имен, независимо от протоколов физического и канального уровней, а также определение оптимальный путей пересылки сетевых пакетов. Маршруты могут быть жестко заданы в виде таблиц и редко меняться, а также они могут быть в высокой степени динамическими, то есть вычисляемыми заново для каждого пакета, с учетом текущей загруженности сети. Если в подсети одновременно присутствует слишком большое количество пакетов, то они могут закрыть дорогу друг другу, образуя заторы в узких местах. Недопущение подобной закупорки также является задачей сетевого уровня.

Транспортный уровень (Transport layer) обеспечивает надежность доставки пакетов сетевого уровня. Наиболее популярной разновидностью транспортного соединения является защищенный от ошибок канал между двумя точками, поставляющий сообщения или байты в том порядке, в каком они были отправлены. Однако транспортный уровень может предоставлять и другие типы служб, например, пересылку отдельных сообщений без гарантии соблюдения порядка их доставки. Транспортный уровень также обеспечивает управление

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

потоком, синхронизируя скорость передачи между каждой парой взаимодействующих узлов.

Сеансовый уровень (Session layer) позволяет пользователям различных компьютеров устанавливать сеансы связи друг с другом. Одной из служб, предоставляемых сеансовым уровнем, является управление маркерами. Для некоторых протоколов бывает существенно, чтобы две стороны не пытались выполнить одновременно одну и ту же операцию. Для управления подобной активностью сеансовый уровень выпускает маркеры, которыми обмениваются участники сети. Критичную операцию может выполнить только компьютер, владеющий маркером. Еще одной службой, предоставляемой сеансовым уровнем, является синхронизация. Сеансовый уровень сохраняет контрольные точки потока данных, так что после восстановления связи передача возобновляется с момента последней контрольной точки, то есть уже успешно переданные блоки данных повторно не передаются.

Уровень представления (Presentation layer) обеспечивает определенный синтаксис и семантику, т.е. кодирование передаваемой информации. Например, на различных компьютерах могут использоваться различные формы представления строковых данных (ASCII и Unicode). Чтобы общение компьютеров с различным представлением данных было возможным, необходимо преобразовывать различные форматы данных друг в друга, передавая их по сети в некоем стандартизированном виде. Эти преобразования осуществляет уровень представления. К уровню представления также относится криптографическая защита информации.

Прикладной уровень (Application layer) содержит набор протоколов взаимодействия прикладных сетевых программ. Например, web-браузер (программа отображения страниц в формате HTML) использует собственный протокол прикладного уровня, который называется HTTP. Аналогично, файловый сервер может обеспечивать сетевой доступ к файлам на основе протокола FTP. Необходимо понимать, что работа протоколов прикладного уровня базируется на протоколах нижележащих уровней. Например, несколько протоколов прикладного уровня, могут использовать общие протоколы транспортного и сетевого уровней.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

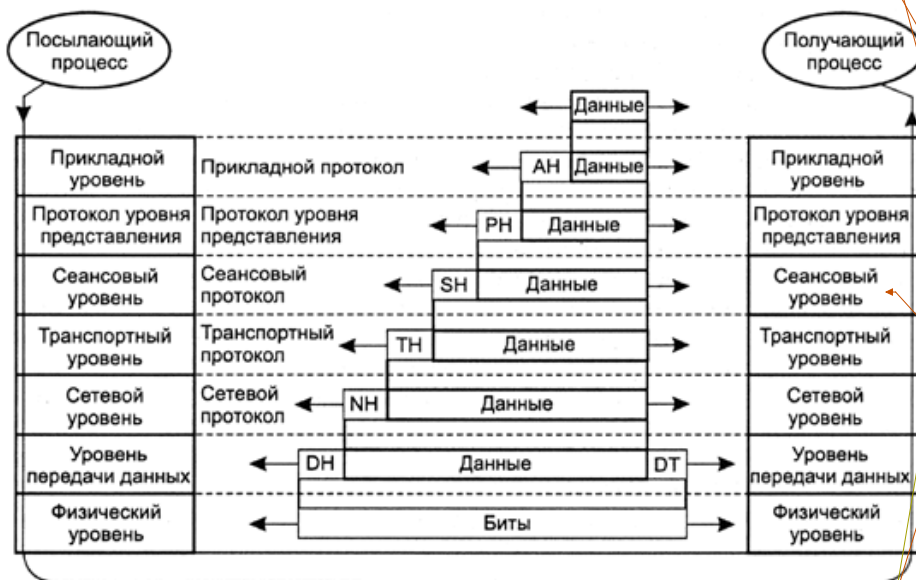
Отформатировано: русский

3. Представление заголовков данных в модели OSI

Схематично процесс обмена данными в эталонной модели взаимодействия открытых систем можно представить следующим образом:

- Некоторый процесс передает данные прикладному уровню.
- Прикладной уровень добавляет в начало данных заголовок прикладного уровня АН (Application Header), и передает результат уровню представлений.
- Уровень представлений не знает, какую часть переданных ему прикладным уровнем данных составляет заголовок АН, а какую – собственно данные, поэтому полученный объект считает данными и добавляет к ним свой заголовок РН (Presentation Header).
- Этот процесс повторяется до тех пор, пока данные не достигнут физического уровня, после чего происходит их передача в канал связи.
- На получающей машине заголовки удаляются один за другим в обратном порядке, по мере продвижения сообщения вверх по уровням, пока сообщение не достигнет получающего процесса.

Рисунок ниже иллюстрирует данную схему



Представление заголовков в модели OSI

Отформатировано: русский

Отформатировано: русский

Отформатировано: Обычный, без нумерации
Отступ: Слева: 0,63 см, без нумерации

Отформатировано: русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: не полужирный

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: не полужирный

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: не полужирный

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: не полужирный

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: не полужирный

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Обычный, Отступ: Слева: 1,27 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Обычный, без нумерации

Отформатировано: Шрифт: не полужирный

Отформатировано: Обычный, Отступ: Слева: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Уровни эталонной модели абстрагированы друг от друга. Например, для протокола транспортного уровня не важно, какова реализация протоколов вышележащего (сеансового) и нижележащего (сетевого) уровней. Подобная независимость уровней друг от друга позволяет создавать различные комбинации взаимодействующих протоколов.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: Отступ: Слева: 0 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: русский

Лекция 17.

Методы коммутации в компьютерных сетях.

1. Сети с коммутацией каналов.
2. Техника частотного мультиплексирования (FDM) в сетях с коммутацией каналов.
3. Техника мультиплексирования с разделением времени (TDM) в сетях с коммутацией каналов.
4. Сети с коммутацией пакетов.
5. Сети с коммутацией сообщений.

1. Сети с коммутацией каналов.

Магистральный канал передачи данных состоит из отдельных линий связи и узлов коммутации, которые обеспечивают соединение территориально удаленных абонентов между собой. Установление соединения (физического или виртуального) осуществляется с помощью того или иного метода коммутации. В зависимости от методов установления соединения и способов передачи данных от одного узла и другому узлу различают сети с коммутацией каналов, коммутацией сообщений и коммутацией пакетов.

Независимо от схемы коммутации абонентов существует классификация сетей:

- с постоянной коммутацией (на основе выделенных (dedicated) или арендуемых (leased) каналов);
- с динамической коммутацией.

В первом случае соединение устанавливается не пользователями, а персоналом, обслуживающим сеть (предварительно соединение заказывается на длительный период времени). Во втором случае сеть разрешает устанавливать соединение по инициативе пользователя сети.

Коммутация каналов подразумевает образование непрерывного составного физического канала из последовательно соединенных отдельных канальных участков для прямой передачи данных между узлами (рисунок ниже).

Отформатировано: русский

Отформатировано: нумерованный + Уровень: 1 + Стиль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто, русский

Отформатировано: нумерованный + Уровень: 1 + Стиль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

Отформатировано: Обычный, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

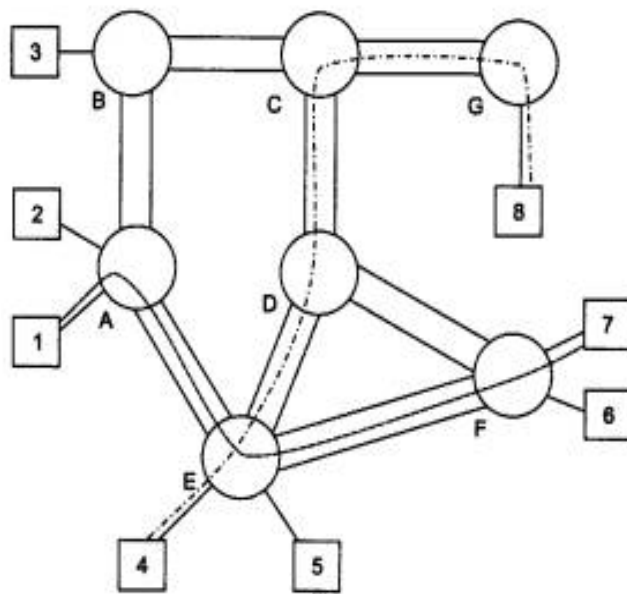
Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто



Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: По центру, Отступ: Первая строка: 1,25 см

Коммутация каналов

Процессом установления соединения управляет источник, который посылает сигнал вызова, получает ответный сигнал (приглашение к набору номера) и вслед за этим передаёт адресную информацию (знаки набора номера). Коммутационный узел обрабатывает эту информацию, занимает один из каналов в пучке, ведущем к следующему коммутационному узлу, и передает последнему знаки набора, необходимые для дальнейшего установления соединения. Таким образом, постепенно, по участкам, вплоть до вызываемого абонента образуется соединительный тракт. После завершения этого процесса, от сети на вызывающую и вызываемую оконечные установки поступают сигналы, извещающие о том, что соединение готово к передаче данных.

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

В течение фазы передачи данных управление осуществляется оконечной установкой. В оконечной установке принимается решение о мерах, которые необходимо принять для обнаружения и исправления ошибок передачи.

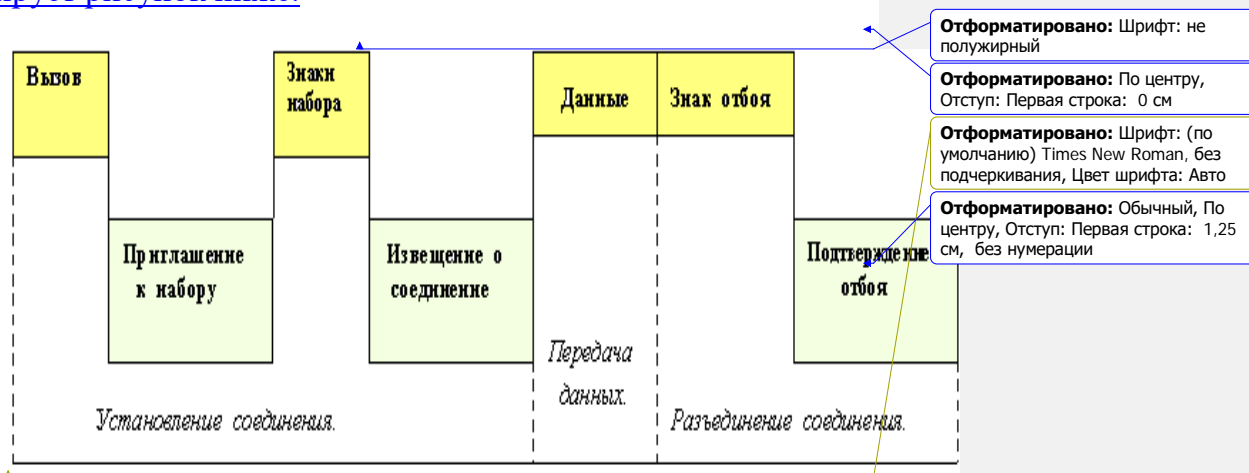
Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Разъединение может быть начато любой из двух связанных между собой оконечных установок с помощью сигнала отбоя. По этому сигналу все коммутационные узлы, участвующие в образовании соединительного тракта, отключают соединения.

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Фазы сеанса связи в сетях с коммутацией каналов демонстрирует рисунок ниже.



Фазы сеанса связи

Коммутаторы, а также соединяющие их каналы должны обеспечивать одновременную передачу данных нескольких абонентских каналов. Для этого они должны быть высокоскоростными и поддерживать какую-либо технику мультиплексирования абонентских каналов.

В настоящее время для мультиплексирования абонентских каналов используются две техники:

- техника частотного мультиплексирования (Frequency Division Multiplexing, FDM);
- техника мультиплексирования с разделением времени (Time Division Multiplexing, TDM).

2. Техника частотного мультиплексирования в сетях с коммутацией каналов.

Техника частотного мультиплексирования каналов (FDM) была разработана для телефонных сетей, но применяется она и для других видов сетей, например сетей кабельного телевидения.

Речевые сигналы имеют спектр шириной примерно в 10 000 Гц, однако основные гармоники укладываются в диапазон от 300 до 3400 Гц. Поэтому для качественной передачи речи достаточно образовать между двумя собеседниками канал с полосой

Отформатировано: Шрифт: не полужирный

Отформатировано: По центру, Отступ: Первая строка: 0 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Обычный, По центру, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Обычный, Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Обычный, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: нумерованный + Уровень: 1 + Стиль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

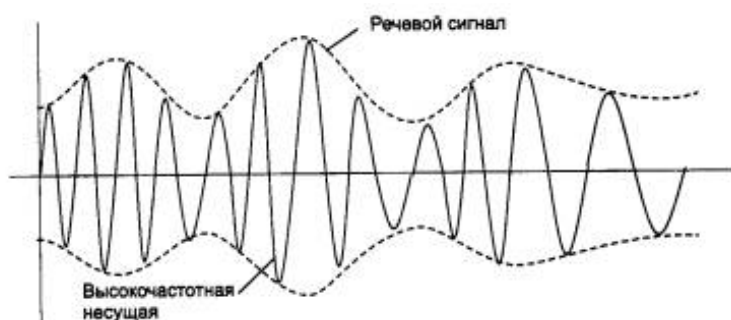
пропускания в 3100 Гц, который и используется в телефонных сетях для соединения двух абонентов. В то же время полоса пропускания кабельных систем обычно составляет сотни килогерц, а иногда и сотни мегагерц.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Для разделения абонентских каналов характерна техника модуляции высокочастотного несущего синусоидального сигнала низкочастотным речевым сигналом (рисунок ниже).

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто



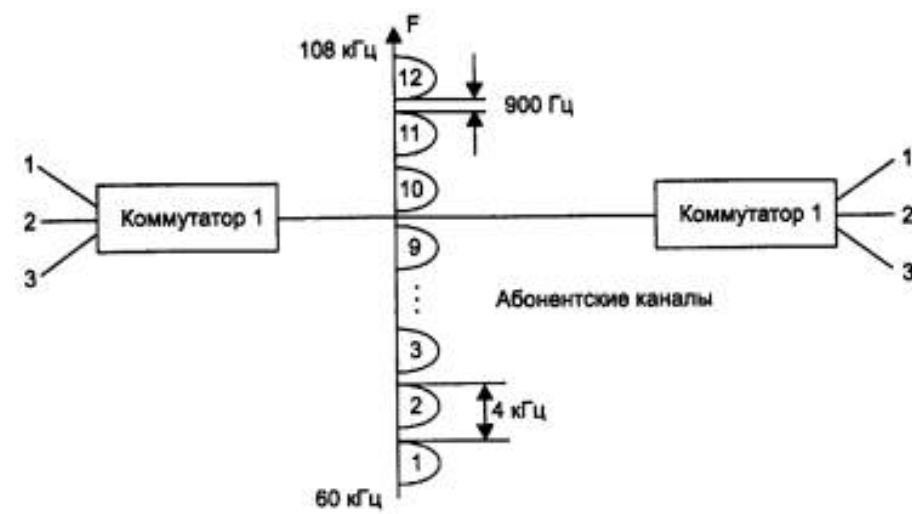
Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 0 см

Модуляция высокочастотного сигнала низкочастотным

В частности, спектр модулированного сигнала переносится в другой диапазон, который симметрично располагается относительно несущей частоты и имеет ширину, приблизительно совпадающую с шириной модулирующего сигнала (рисунок ниже).

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто



Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Пример диапазонов частот FDM коммутатора

На входы FDM-коммутатора поступают исходные сигналы от абонентов телефонной сети. Коммутатор выполняет перенос частоты каждого канала в свой диапазон частот. Обычно высокочастотный диапазон делится на полосы, которые отводятся для передачи данных абонентских каналов.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: русский

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Чтобы низкочастотные составляющие сигналов разных каналов не смешивались между собой, полосы делают шириной в 4 кГц, а не в 3,1 кГц, оставляя между ними страховой промежуток в 900 Гц.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

В канале между двумя FDM-коммутаторами одновременно передаются сигналы всех абонентских каналов, но каждый из них занимает свою полосу частот. Такой канал называют уплотненным.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Выходной FDM-коммутатор выделяет модулированные сигналы каждой несущей частоты и передает их на соответствующий выходной канал, к которому непосредственно подключен абонентский телефон.

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Коммутаторы FDM могут выполнять как динамическую, так и постоянную коммутацию. Техника FDM применяется в модемах (используется 4 частоты, по 2 частоты для кодирования бит в каждом из направлений передачи).

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 0 см

3. Техника мультиплексирования с разделением времени в сетях с коммутацией каналов.

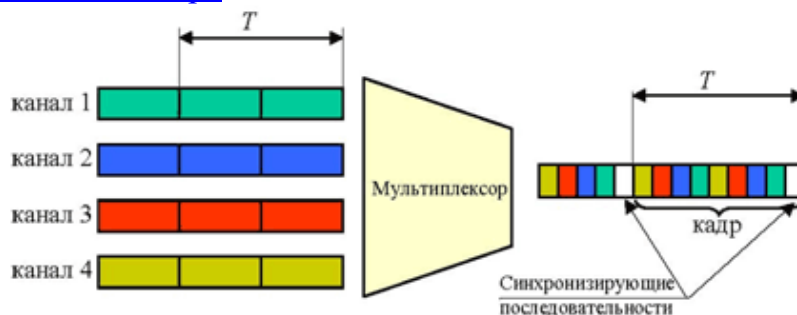
Техника мультиплексирования с разделением времени (Time Division Multiplexing, TDM) разрабатывалась в расчете на дискретный характер передаваемых данных. Другое ее название – техника синхронного режима передачи (Synchronous Transfer Mode, STM).

Аппаратура TDM-сетей – мультиплексоры, коммутаторы и демультимплексоры – работает в режиме разделения времени, поочередно обслуживая в течение цикла своей работы все абонентские каналы.

В каждом цикле мультиплексор выполняет следующие действия:

- прием от каждого канала очередного байта данных;
- составление из принятых байтов уплотненного кадра, называемого также обоймой;
- передача уплотненного кадра на выходной канал с битовой скоростью, равной $N \cdot 64$ Кбит/с.

Рисунок ниже иллюстрирует пример создания уплотненного кадра



Создание уплотненного кадра

Цикл работы оборудования TDM равен 125 мкс, что соответствует периоду следования замеров голоса в цифровом абонентском канале. Это значит, что мультиплексор или коммутатор успевает вовремя обслужить любой абонентский канал и передать его очередной замер далее по сети. Коммутатор принимает уплотненный кадр по скоростному каналу от мультиплексора и записывает каждый байт из него в отдельную

Отформатировано: Отступ: Первая строка: 0 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Слева: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

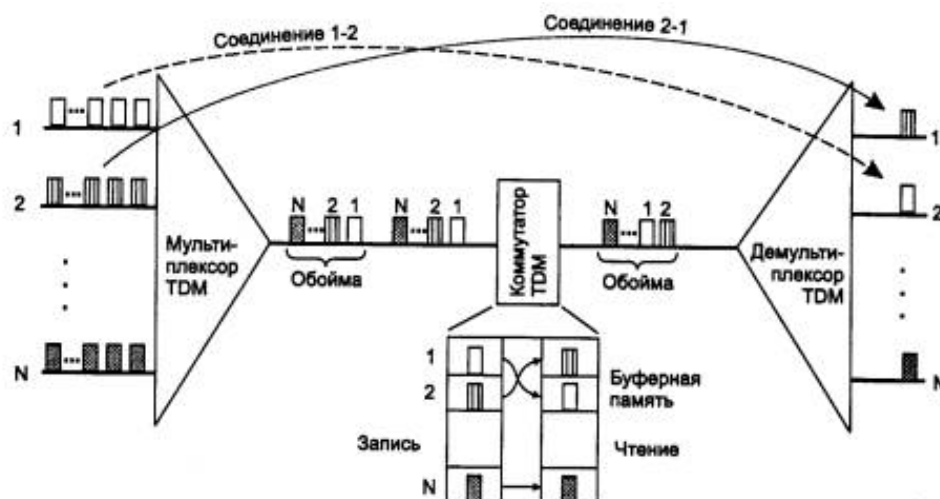
Отформатировано: Отступ: Первая строка: 1,25 см, без нумерации

Отформатировано: Отступ: Слева: 1,25 см, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано

ячейку своей буферной памяти, причем в том порядке, в котором эти байты были упакованы в уплотненный кадр. Для выполнения операции коммутации байты извлекаются из буферной памяти не в порядке поступления, а в таком порядке, который соответствует поддерживаемым в сети соединениям абонентов (рисунок ниже).



Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 0 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Передача данных в сетях TDM

Работа оборудования TDM напоминает работу сетей с коммутацией пакетов, так как каждый байт данных можно считать некоторым элементарным пакетом. Однако, в отличие от пакета компьютерной сети, «пакет» сети TDM не имеет индивидуального адреса. Его адресом является порядковый номер в обойме.

Сети TDM могут поддерживать либо режим динамической коммутации, либо режим постоянной коммутации, а иногда и оба эти режима. Так, например, основным режимом цифровых телефонных сетей, работающих на основе технологии TDM, является динамическая коммутация, но они поддерживают также и постоянную коммутацию, предоставляя своим абонентам службу выделенных каналов.

Отформатировано: Отступ: Слева: 0 см, Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

4. Сети с коммутацией пакетов.

Коммутация пакетов – это техника коммутации абонентов, которая была специально разработана для эффективной передачи компьютерного трафика.

При коммутации пакетов все передаваемые пользователем сети сообщения разбиваются в исходном узле на сравнительно небольшие части, называемые пакетами (рисунок ниже).



Пакетирование сообщений

Сообщением называется логически завершенная порция данных – запрос на передачу файла, ответ на этот запрос, содержащий весь файл, и т.п. Сообщения могут иметь произвольную длину, от нескольких байт до многих мегабайт.

Пакеты могут иметь переменную длину, но в узких пределах, например от 46 до 1500 байт.

Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета узлу назначения, а также номер пакета, который будет использоваться узлом назначения для сборки сообщения.

Пакеты транспортируются в сети как независимые информационные блоки.

Разложение сообщения на пакеты и восстановление его после передачи осуществляется окончательным оборудованием источника и адресата.

В принимающем коммутационном узле каждый пакет проверяется на наличие ошибок. На пакеты, принятые без ошибок, в ответ направляется подтверждение их приёма (положительная

Отформатировано: нумерованный + Уровень: 1 + Стиль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

Отформатировано: Обычный, без нумерации

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

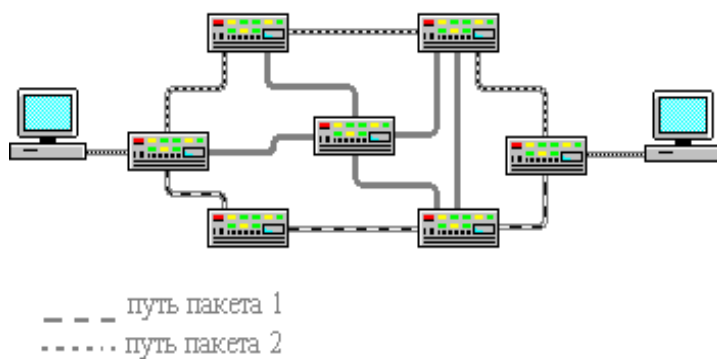
Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

квитанция ДА). Если же в пакете обнаружены ошибки, то посылается запрос на его повторную передачу (отрицательная квитанция НЕТ).

Режим передачи пакетов между двумя конечными узлами сети в основном предполагает независимую маршрутизацию каждого пакета. Такой режим работы сети называется дейтаграммным, и при его использовании коммутатор может изменить маршрут какого-либо пакета в зависимости от состояния сети (рисунок ниже). При этом могут возникнуть ситуации, при которых пакеты могут поступать в адрес получателя в неверной последовательности.



Различные пути передачи пакетов

Метод коммутации пакетов по сравнению с другими методами обеспечивает наименьшую задержку при передаче данных и наибольшую пропускную способность сети.

5. Сети с коммутацией сообщений.

В сети с коммутацией сообщений между оконечными установками, обменивающимися информацией, нет сквозного соединения. В коммутационных узлах сообщения заносятся в память и передаются далее по участкам переприёма от узла к узлу.

Коммутация сообщений – это передача единого блока данных между транзитными компьютерами сети с временной буферизацией этого блока на диске каждого компьютера.

Сообщение, в отличие от пакета, имеет произвольную длину, которая определяется не технологическими соображениями, а

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: русский

Отформатировано: Отступ: Слева: 0 см

Отформатировано: нумерованный + Уровень: 1 + Стилль нумерации: 1, 2, 3, ... + Начать с: 1 + Выравнивание: слева + Выровнять по: 1,25 см + Отступ: 1,88 см

Отформатировано: Отступ: Слева: 0 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Слева: 0 см, Первая строка: 1,25 см

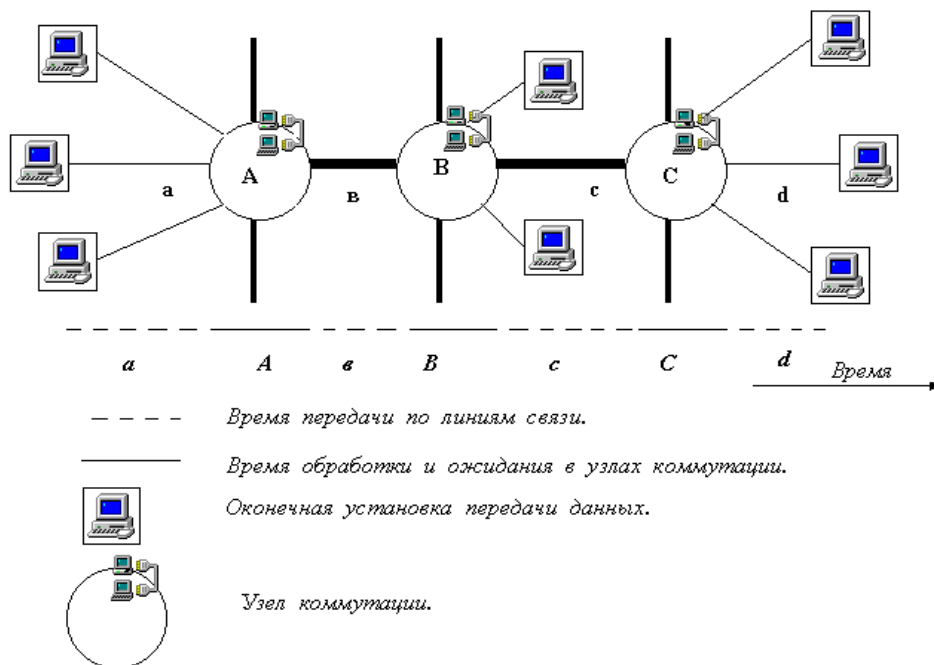
Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

содержанием информации, составляющей сообщение. Например, сообщением может быть текстовый документ, файл с кодом программы, электронное письмо.

Транзитные компьютеры могут соединяться между собой как сеть с коммутацией пакетов, так и сеть с коммутацией каналов. Сообщение хранится в транзитном компьютере на диске, причем время хранения может быть достаточно большим, если компьютер загружен другими работами или сеть временно перегружена. Рисунок ниже демонстрирует сеть с коммутацией сообщений.



Сеть с коммутацией сообщений

На окончательной установке, от которой необходимо передать сообщение, оно снабжается заголовком, содержащим адрес желаемого абонента, и по абонентской линии *a* передается на ближайший коммутационный узел *A*. В нем сообщение запоминается, обрабатывается его заголовок, определяется в какую из исходящих линий далее его нужно направить и, наконец, передается на следующий коммутационный узел *B*. Если линия *b*, по которой нужно передать сообщение далее, занята, то оно

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, не курсив, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Шрифт: (по умолчанию) Times New Roman, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Отступ: Первая строка: 1,25 см

остаётся в запоминающем устройстве (буферном накопителе узла) до тех пор, пока не будут переданы все находящиеся перед ним в очереди другие сообщения. На участке в магистральном канале скорость передачи обычно выше, чем в абонентской линии *a*, следовательно, длительность передачи сообщения будет меньше.

Узел коммутации **B**, в котором сообщение обрабатывается так же, как и в узле **A**, передает сообщение по линии *c* на узел **C**. От последнего оно передается по абонентской линии *d* на принимающий модуль абонента.

Время ожидания, в течение которого сообщение хранится в узле коммутации, зависит от длины очередей на линии связи, поэтому общее время прохождения сообщения между двумя оконечными установками в сети может быть различным. Запись сообщений в память упрощает трансформацию скоростей различного оборудования данных, осуществляемую в коммутационных узлах. Использование на межузловых участках высокоскоростных линий связи позволяет более эффективно, чем в сетях с коммутацией каналов, передавать требуемый объём информации и использовать ресурсы сети. Однако экономию линий связи необходимо сопоставлять с затратами, которых требуют запоминание и обработка сообщений в узлах коммутации.

Сети с коммутацией сообщений обычно используются для передачи сообщений, не требующих немедленного ответа, например, сообщений электронной почты. Режим передачи с промежуточным хранением на диске называется режимом «хранение-и-передача» (store-and-forward).

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: Шрифт: (по умолчанию) Times New Roman, не полужирный, без подчеркивания, Цвет шрифта: Авто

Отформатировано: русский

Отформатировано: Шрифт: (по умолчанию) Times New Roman, русский

|

← **Отформатировано:** Отступ: Слева: 0 см, Первая строка: 1,25 см



Отформатировано: Шрифт: 16 пт, полужирный

Отформатировано: Обычный, Отступ: Слева: 1,25 см, без нумерации

Отформатировано: Отступ: Первая строка: 0 см

Отформатировано: Шрифт: не полужирный

Отформатировано: Отступ: Первая строка: 1,25 см

Отформатировано: Отступ: Первая строка: 0 см

Вопросы к зачету.

1. Возникновение информационных технологий (ИТ) (аспекты технологий, этапы развития ИТ).
2. Автоматизированные информационные технологии.
3. Информатика и информационная технология (моделирование информационного процесса, трехуровневый подход в информатике).
4. Статистический подход оценки информации. Свойства собственной информации.
5. Семантический подход оценки информации.
6. Прагматический подход оценки информации.
7. Структурный подход оценки информации.

Отформатировано: Отступ: Слева: 0 см, Первая строка: 0 см

Отформатировано: По ширине

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: По ширине, Отступ: Слева: 1 см, Выступ: 0,5 см

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: Отступ: Слева: 1 см, Выступ: 0,5 см

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: Шрифт: 16 пт

Отформатировано: Шрифт: (по умолчанию) Times New Roman, 16 пт

Отформатировано: Шрифт: 16 пт

