

ЛЕКЦИЯ 1

План лекции:

1. Определение понятий «информация» и «информатика». Многоаспектность понятия «информатика».
2. Понятие информационной системы (ИС). Классификация и состав ИС.
3. Структура и состав ИС.

Введение

В основе данного курса лежат два фундаментальных понятия: информация и наука информатика.

Информацию можно определить как совокупность сведений, уменьшающих степень неопределенности знания о конкретных событиях, процессах, явлениях и т.п. В зависимости от сферы использования информация может быть самой разной: экономической, технической, генетической и т.д.

Информатику принято рассматривать и как науку, и как прикладную дисциплину, и как отрасль народного хозяйства. *Наука информатика* включает в себя следующие понятия:

- информация и информационные ресурсы;
- технические средства обработки информации;
- информационные технологии;
- информационные системы.

Задачами *прикладной информатики* являются:

- разработка методов и правил рационального проектирования устройств и систем обработки информации;
- создание технологии применения вышеназванных устройств и систем для решения научных и практических задач;
- разработка методов взаимодействия человека с данными устройствами и системами.

Рассматривая *информатику как отрасль народного хозяйства*, можно выделить в ней три составные части:

- производство технических средств обработки и передачи информации;
- обработку информации;
- создание и внедрение программных средств (ПС) и систем.

В свете идей науки о знаках *семиотики* информацию и ее свойства можно рассматривать в трех аспектах: синтаксическом, семантическом, прагматическом.

Синтаксический аспект связан с рассмотрением формы и среды представления информации: документ, машинный носитель, память компьютера (с оценкой объемов обрабатываемой и хранимой информации, установлением правил преобразования и выбором формата данных и). Информация на синтаксическом уровне традиционно называется *данными*.

На *семантическом* уровне формируются структурные единицы информации – *экономические показатели*, проектируется структура *базы данных* (интегрированной совокупности взаимосвязанных данных), определяется содержание документов и схема документооборота. Семантический аспект требует понимания *содержания* информации.

Прагматический аспект информации связан с оценкой качества и полезности информации для принятия управленческих решений. Качество информации рассматривается на уровне экономического показателя. Оно является совокупностью следующих свойств:

- *Репрезентативность* информации – методическая правильность формирования информации, т.е. выделение наиболее существенных признаков и связей объектов, событий, явлений, а также измерение и выбор правильных алгоритмов формирования расчетных показателей.

- *Содержательность* информации – максимизация отношения количества полезных данных к их общему объему, т.е. максимизация меры устранения неопределенности.

- *Необходимость и достаточность* информации для принятия управленческого решения.

- *Актуальность* информации – сохранение ее полезности во времени.

- *Доступность и своевременность* получения информации.

- *Точность* информации на уровне отдельных экономических показателей.

- *Достоверность* информации – отображение истинного значения в пределах необходимой точности с заданной вероятностью.

- *Ценность* информации – оценка влияния показателя на эффективность функционирования системы.

Информация в системе управления рассматривается как «ресурс управления», имеющий важное стратегическое значение. Информационные ресурсы в значительной степени являются взаимозаменяемыми по отношению к материальным, финансовым или трудовым ресурсам. Организационная форма информационных ресурсов, объем информации, ее качество влияют на эффективность управления. В настоящее время наиболее широко распространены формы организации информационных ресурсов в виде:

- коллекции документов, картотек ручного заполнения и поиска;
- предметных баз данных (БД) на машинных носителях;
- интегрированных баз данных коллективного пользования с применением компьютерных сетей, включая сеть Internet;
- баз знаний, обеспечивающих получение новой информации на основе системы правил вывода.

Технологический процесс обработки данных в информационных системах осуществляется при помощи:

- технических средств сбора и регистрации данных;
- средств телекоммуникаций;

- систем хранения, поиска и выборки данных;
- средств вычислительной обработки данных;
- технических средств оргтехники.

В современных информационных системах технические средства обработки данных используются комплексно на основе технико-экономического расчета целесообразности их применения с учетом соотношения “цена/качество” и надежности работы технических средств.

Информационные технологии можно определить как совокупность *методов* – приемов и алгоритмов обработки данных – и *инструментальных средств* – программных и технических средств обработки данных.

Информационные технологии условно делятся на категории:

- *Базовые* – это универсальные технологические операции обработки данных, как правило, не зависящие от содержания обрабатываемой информации. Они основаны на использовании широко применяемых программных и технических средств обработки данных.

- *Специальные* – комплекс информационно связанных базовых технологий, предназначенных для выполнения специальных операций с учетом содержания и/или формы представления данных.

Информационные технологии являются необходимым базисом для создания информационных систем.

1.1 Информационные системы

Информационная система (ИС) представляет собой коммуникационную систему по сбору, передаче, переработке информации об объекте, снабжающую работников различных рангов информацией для реализации функции управления.

Пользователями ИС являются организационные единицы управления: структурные подразделения, управленческий персонал, исполнители. Содержательную основу ИС составляют функциональные компоненты: модели, методы и алгоритмы формирования управляющей информации. Функциональная структура ИС представляет собой совокупность функциональных компонентов: подсистем, комплексов задач, процедур обработки информации, определяющих последовательность и условия их выполнения.

Внедрение информационных систем производится с целью повышения эффективности производственно-хозяйственной деятельности объекта за счет обработки и хранения информации, автоматизации операций, а также за счет принципиально новых методов управления. Они основаны на моделировании действий специалистов организации в ходе принятия решений, использовании современных средств телекоммуникаций, глобальных и локальных вычислительных сетей. Классификация ИС проводится по следующим признакам:

- характер обработки информации;
- масштаб и интеграция компонентов ИС;

- информационно-технологическая архитектура ИС.

По первому признаку ИС принято делить на два больших класса:

1. *Системы для оперативной обработки данных (СОД)*. Это традиционные ИС для учета и обработки первичных данных большого объема с применением жестко регламентированных алгоритмов, фиксированной структуры базы данных (БД) и т.п.

2. *Системы поддержки и принятия решений*. Они ориентированы на аналитическую обработку больших объемов информации, интеграцию разнородных источников данных, использование методов и средств аналитической обработки.

В настоящее время сложились основные информационно-технологические архитектуры ИС:

- с централизованной обработкой данных;
- архитектура вида «файл-сервер»;
- архитектура вида «клиент-сервер».

Централизованная обработка предполагает объединение на одном компьютере ПС пользовательского интерфейса, приложений и БД.

В *архитектуре «файл-сервер»* многим пользователям сети предоставляются *файлы* главного компьютера, называемого *файл-сервером*. Это могут быть отдельные файлы пользователей, файлы БД и программы приложений. Вся обработка данных выполняется на компьютерах пользователей. Такой компьютер называется *рабочей станцией*. На ней устанавливаются ПС пользовательского интерфейса и приложений, которые могут вводиться как с устройств ввода РС, так и передаваться по сети с файл-сервера. Файл-сервер может использоваться и для централизованного хранения файлов отдельных пользователей, пересылаемых ими по сети с РС. Архитектура *«файл-сервер»* применяется преимущественно в локальных компьютерных сетях.

В *архитектуре «клиент-сервер»* программное обеспечение (ПО) ориентировано не только на коллективное использование ресурсов, но и на их обработку в месте размещения ресурса по запросам пользователей. Программные системы архитектуры «клиент-сервер» состоят из двух частей: ПО сервера и ПО пользователя-клиента. Работа этих систем организуется следующим образом: программы-клиенты выполняются на компьютере пользователя и посылают запросы к программе-серверу, которая работает на компьютере общего доступа.

1.2 Структура и состав информационной системы

Практически все разновидности ИС, независимо от сферы их применения, включают в себя один и тот же набор компонентов:

- функциональные компоненты;
- компоненты системы обработки данных;
- организационные компоненты.

Под функциональными компонентами понимается система функций управления, взаимосвязанных во времени и пространстве и необходимых для достижения поставленных целей. Декомпозиция информационной системы по функциональному признаку (рис. 1.1) включает в себя выделение ее отдельных частей, называемых *функциональными подсистемами*. Функциональный признак определяет назначение подсистемы, ее цели, задачи и функции, которые она выполняет. Функциональные подсистемы в большой степени зависят от предметной области информационных систем. На рис.1.2 приведена функциональная декомпозиция ИС промышленного предприятия.



Рисунок 1.1 – Декомпозиция информационной системы

Специфические особенности каждой функциональной подсистемы содержатся в ее функциональных задачах. Обычно их связывают или с достижением определенных целей функции управления, или определяют их

как работу, которая должна быть выполнена определенным образом в определенный период. Однако с появлением современных информационных технологий, основанных на широком применении средств вычислительной техники, понятие «задача» рассматривается как законченный комплекс обработки информации, обеспечивающий выдачу необходимых наборов информации для принятия решений управленческим персоналом. Поэтому задача постепенно начала рассматриваться как элемент системы управления, а не как элемент системы обработки данных.

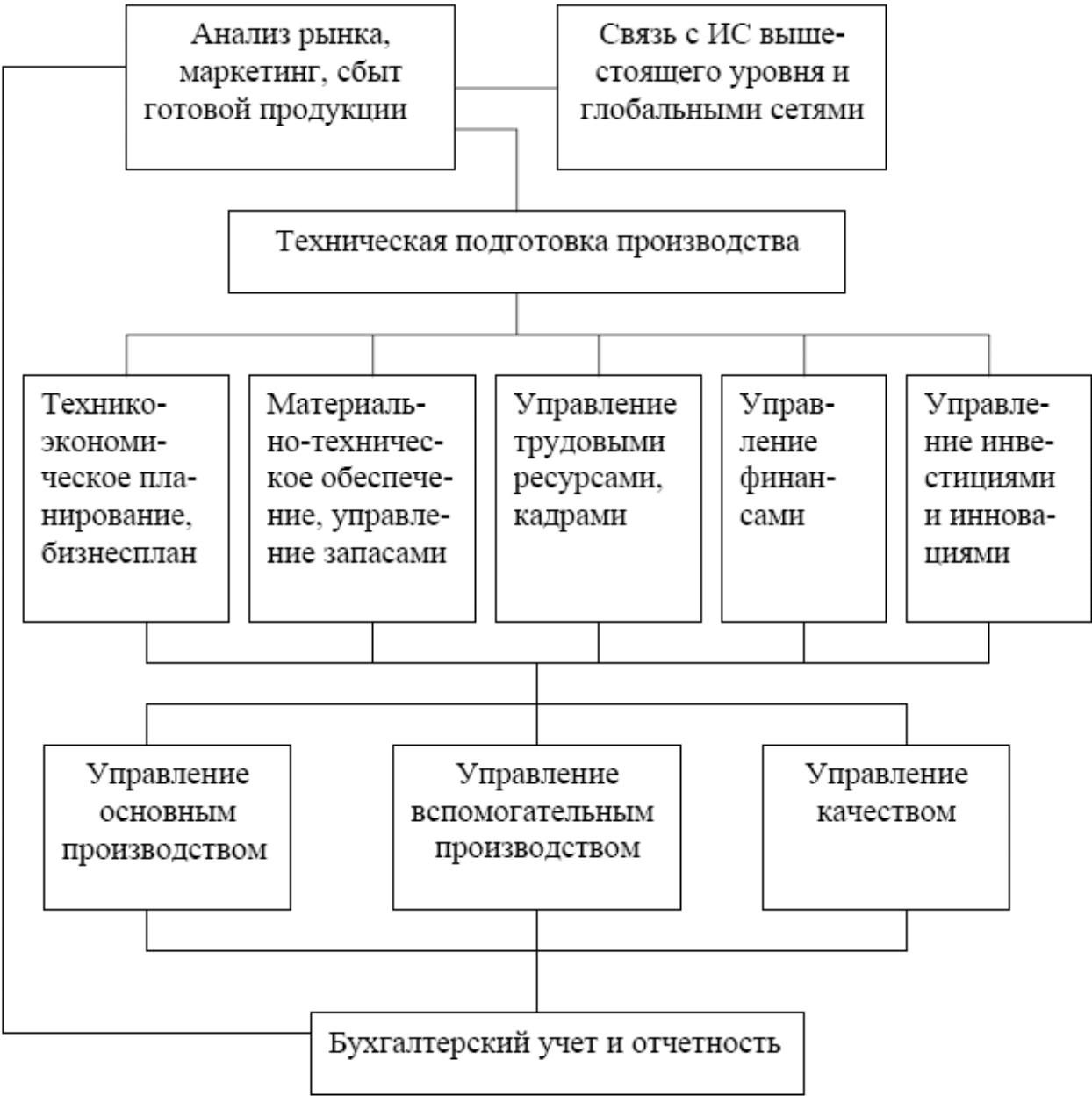


Рисунок 1.2 – Функциональная декомпозиция информационной системы промышленного предприятия

ЛЕКЦИЯ 2

План лекции:

1. Типы обеспечения информационных систем.
2. Тенденции развития информационных систем.
3. Системы классификации данных и их виды.

2.1 Обеспечение, используемое в информационной системе

Выбор состава функциональных задач подсистем управления обычно осуществляется с учетом основных фаз управления: планирования; учета, контроля и анализа; регулирования (исполнения).

Планирование – это управленческая функция, обеспечивающая формирование планов, в соответствии с которыми будет организовано функционирование объекта управления. Выделяют перспективное (5-10 лет), годовое (1 год) и оперативное (сутки, неделя, декада, месяц) планирование.

Учет, контроль и анализ – это функции, обеспечивающие получение данных о состоянии управляемой системы за определенный промежуток времени, определение факта и причины отклонений реального состояния объекта управления от его планируемого состояния, а также нахождение величин такого отклонения. Учет ведется по показателям плана в выбранном диапазоне планирования (оперативный, среднесрочный и т.д.).

Регулирование (исполнение) это функция, обеспечивающая сравнение планируемых и фактических показателей функционирования объекта управления и реализацию необходимых управляющих воздействий при наличии отклонений от запланированных в заданном диапазоне.

Выбор и обоснование состава функциональных задач является одним из важнейших элементов создания информационных систем. Анализ функциональных задач показывает, что их практическая реализация в условиях использования информационных систем многовариантна. Одна и та же задача может быть решена различными математическими методами, моделями и алгоритмами (рис.1.1). Иногда функциональную подсистему называют подсистемой математического обеспечения.

Выделение типовых операций обработки данных позволило создать специализированные программно-аппаратные комплексы, включающие в себя *пакеты прикладных программ (ППП)*, реализующих функциональные задачи ИС. Конфигурация аппаратных комплексов образует *топологию вычислительной системы*.

Практически все системы обработки данных ИС независимо от сферы их применения насчитывают один и тот же набор составных частей (компонентов), называемых *видами обеспечения* (рис.1.1). Принято выделять *информационное, программное, техническое, правовое, лингвистическое обеспечение*.

Информационное обеспечение (ИО) – это совокупность методов и средств по размещению и организации информации, включающих в себя системы классификации и кодирования, унифицированные системы документации,

рационализации документооборота и форм документов, методов создания внутримашинной информационной базы ИС.

Программное обеспечение – совокупность программных средств для создания и эксплуатации СОД средствами вычислительной техники. В состав программного обеспечения входят базовые и прикладные программные средства. *Базовые программные средства* служат для автоматизации взаимодействия человека и компьютера, организации типовых процедур обработки данных и диагностики функционирования технических средств СОД. *Прикладные программные средства* представляют собой совокупность программных продуктов, предназначенных для автоматизации решения функциональных задач информационной системы. Они подразделяются на *универсальные средства* (текстовые редакторы, электронные таблицы, системы управления базами данных) и *специализированные*, реализующие функциональные подсистемы различной природы и разработанные в виде ППП.

Техническое обеспечение представляет собой комплекс технических средств, применяемых для функционирования СОД, и включает в себя устройства, реализующие типовые операции обработки данных как вне компьютера, так и внутри него.

Правовое обеспечение – это совокупность правовых норм, регламентирующих создание и функционирование информационной системы. Правовое обеспечение разработки ИС включает в себя нормативные акты договорных взаимоотношений между заказчиком и разработчиком ИС, правовое регулирование отклонений.

Лингвистическое обеспечение – это совокупность языковых средств, используемых на различных стадиях создания и эксплуатации СОД для повышения эффективности разработки и обеспечения общения человека и компьютера.

2.2 Тенденции развития информационных систем

Логика развития ИС в последние 30 лет наглядно демонстрирует эффект маятника: централизованная модель обработки данных на базе мэйнфреймов, доминировавшая до середины 80-х годов, всего за несколько лет уступила свои позиции распределенной архитектуре одноранговых локальных сетей (ЛС) персональных компьютеров, но затем началось возвратное движение к централизации ресурсов системы. Сегодня в центре внимания оказывается технология “клиент-сервер”, которая эффективно объединяет достоинства своих предшественников. Различают несколько поколений ИС.

Первое поколение ИС (1960-1970 гг.) строилось на базе центральных электронно-вычислительных машин (ЭВМ) по принципу: одно предприятие – один центр обработки.

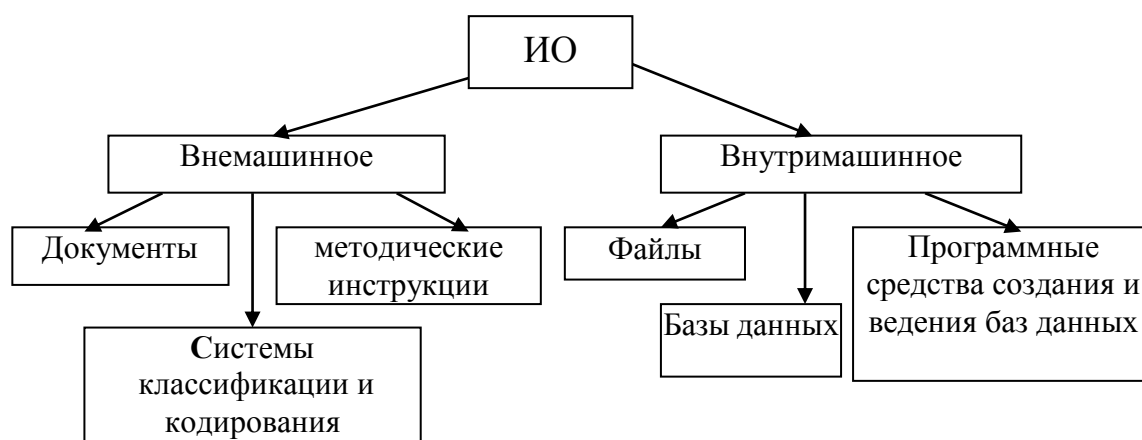
Второе поколение ИС (1970-1980 гг.). Наметились первые шаги к децентрализации ИС. Кардинальным новшеством ИС этого поколения была двух- и трехуровневая модель организации СОД, когда выделялась

центральная ЭВМ в сочетании с информационным фундаментом на основе децентрализованной базы данных и прикладных пакетов.

Третье поколение ИС (1980 - начало 1990-х гг.) – это бум распределенной сетевой обработки, главной движущей силой которого был массовый переход на персональные компьютеры. Логика корпоративного бизнеса потребовала объединения разрозненных рабочих мест в единую ИС, в результате чего появились вычислительные сети и распределенная обработка. При развитии ИС третьего поколения на смену одноранговой распределенной обработке пришла иерархическая модель «клиент-сервер».

Четвертое поколение ИС – это современная модель иерархической организация, в которой централизованная обработка и единое управление ресурсами на верхнем уровне сочетается с распределенной обработкой на нижнем уровне. ИС четвертого поколения аккумулируют следующие особенности:

- полное использование потенциала настольных компьютеров и среды распределенной обработки;
- модульное построение системы, предполагающее существование множества различных типов архитектурных решений в рамках единого комплекса;
- экономия ресурсов системы за счет централизации хранения и обработки данных на верхних уровнях иерархии ИС.



Структура информационного обеспечения

Документы, файлы, базы данных составляют общую информационную базу объекта.

Особенности документов с точки зрения ИО:

1. Основная часть нормативно-справочных документов остается практически неизменной.
2. Оперативная информация быстро растет и постоянно меняется.
3. Плановые документы изменяются достаточно редко.

2.3 Системы классификации данных

Для того чтобы обеспечить эффективную обработку информации на компьютере, а также передачу ее по каналам связи, необходимо представить данные в цифровом виде. С этой целью информацию сначала классифицируют, а затем кодируют с использованием классификатора.

Классификация – это разделение заданного множества на подмножества в соответствии с принятыми методами классификации.

Система классификации – это совокупность методов и средств, с помощью которых осуществляется разбиение исходного множества данных на пересекающиеся подмножества в соответствии с признаками сходства и различия.

Признак, по которому делят множества на подмножества, называют признаком классификации. В разных системах классификационным группировкам присваивают свои названия: классы, подклассы, группы, подгруппы, виды, семейства. Степень классификации – этап разделения заданного множества на подмножества. Число ступеней называют глубиной классификации. После завершения классификации выполняют кодирование – образование и присвоение обозначения объекту классификации, признаку классификации и классификационной группировке. Такое условное кодовое обозначение называют кодом. Количество знаков в кодовом обозначении называют длиной кода.

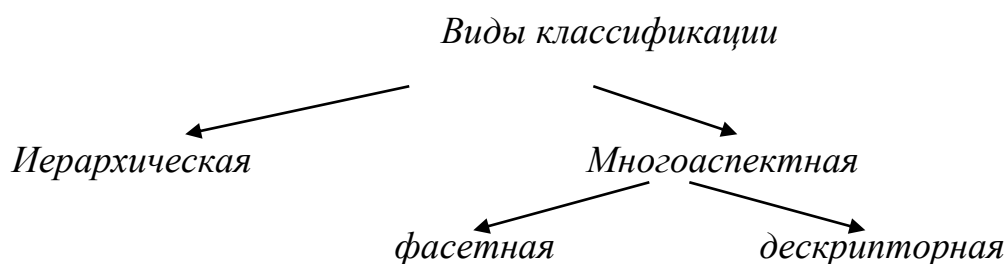
Каждая система классификации характеризуется следующими свойствами: *гибкостью системы*; *емкостью системы*; *степенью заполнения системы*.

Гибкость системы – это способность допускать включение новых признаков, объектов без разрушения структуры классификатора. Необходимая гибкость определяется временем жизни системы.

Емкость системы – это наибольшее количество классификационных группировок, допускаемое в данной системе классификации.

Степень заполнения системы определяется как частное от деления фактического количества группировок на величину емкости системы.

Классификация информации необходима для ее правильной структуризации и составления в дальнейшем корректной модели БД. В процессе классификации вся информация делится по разным признакам. Тот из них, который лежит в основе классификации, называется признаком классификации (основанием). Существуют различные системы классификации:



Иерархическая классификация подразумевает разделение всего множества объектов на группы, которые формируются в виде отдельных уровней или дерева. Дерево – это совокупность вершин и дуг, расположенных в иерархическом порядке. Каждая вершина, кроме корня, может иметь только одну родительскую вершину и несколько дочерних вершин. Количество уровней иерархии – это глубина классификации.

Первоначальный объем классифицируемых объектов разбивается на подмножества по какому-либо признаку и детализируется на каждой следующей ступени классификации. На рис. 2.1 приведена обобщенная схема иерархической классификации.

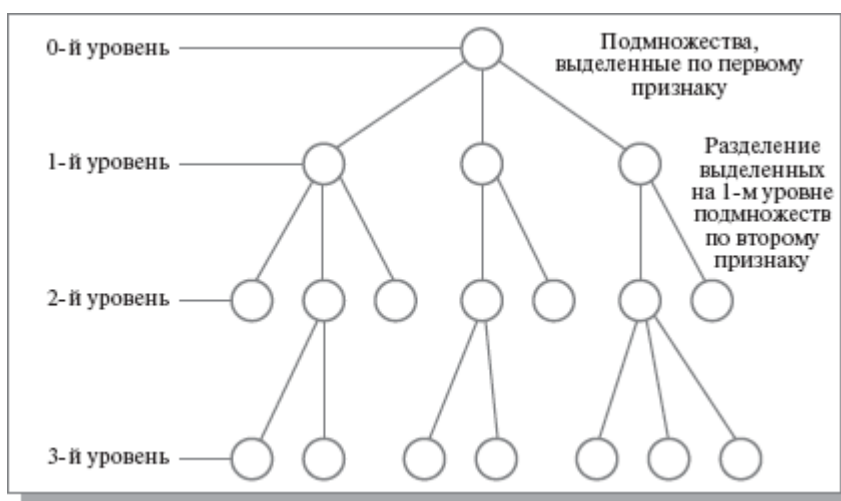


Рис. 2.1 Схема иерархической классификации

Характерными особенностями иерархической системы являются:

- возможность использования неограниченного количества признаков классификации;
- соподчиненность признаков классификации, которая выражается в разбиении каждой классификационной группировки, образованной по одному признаку, на множество группировок по нижестоящему признаку.

Таким образом, классификационные схемы, построенные на основе иерархического принципа, имеют неограниченную емкость, величина которой зависит от глубины классификации и количества объектов классификации, которые можно расположить на каждой ступени. Число объектов на каждой ступени классификации определяется количеством знаков в выбранном алфавите кода. Например, если алфавит – двузначные десятичные цифры, то можно на одном уровне разместить 100 объектов. Выбор необходимой глубины классификации и структуры кода зависит от характера объектов классификации и задач, для решения которых предназначен классификатор. При построении иерархической системы классификации сначала выделяется некоторое множество объектов, подлежащее классифицированию, для которого определяются полное множество признаков и их соподчиненность друг другу. Затем выполняется разбиение исходного множества объектов на классификационные группировки каждой ступени классификации.

К положительным особенностям иерархической системы относят логичность, простоту построения и удобство обработки.

Серьезным недостатком иерархического метода классификации является жесткость классификационной схемы. Она обусловлена заранее установленным выбором признаков классификации и порядком их использования по ступеням классификации. Это ведет к тому, что при изменении состава объектов классификации, их характеристик или характера решаемых при помощи классификатора задач, требуется коренная переработка классификационной схемы. Гибкость данной системы обеспечивается только за счет ввода большой избыточности в ветвях, что приводит к слабому заполнению структуры классификатора. Поэтому при разработке классификаторов следует учитывать, что иерархический метод более предпочтителен для решения неменяющегося комплекса задач, где присутствуют объекты со слабо изменяющимися признаками.

Многоаспектная классификация – это система, которая использует параллельно несколько независимых признаков (аспектов) в качестве основания классификации. Существуют два типа многоаспектных систем: *фасетная* и *дескрипторная*. *Фасет* – это аспект классификации, который используется для образования независимых классификационных группировок. *Дескриптор* – ключевое слово, определяющее некоторое понятие, которое формирует описание объекта и дает принадлежность этого объекта к классу или группе.

При фасетном методе классификации заранее не создаются жесткая классификационная схема и конечные группировки. Разрабатывается только система таблиц признаков объектов классификации, называемых фасетами. При необходимости создания классификационной группировки для решения конкретной задачи осуществляется выборка необходимых признаков из фасетов и их объединение в определенной последовательности. Общий вид фасетной классификационной схемы представлен на рис. 2.2. Внутри фасета значения признаков могут перечисляться в заданном порядке или образовывать иерархическую структуру, если существует соподчиненность выделенных признаков.

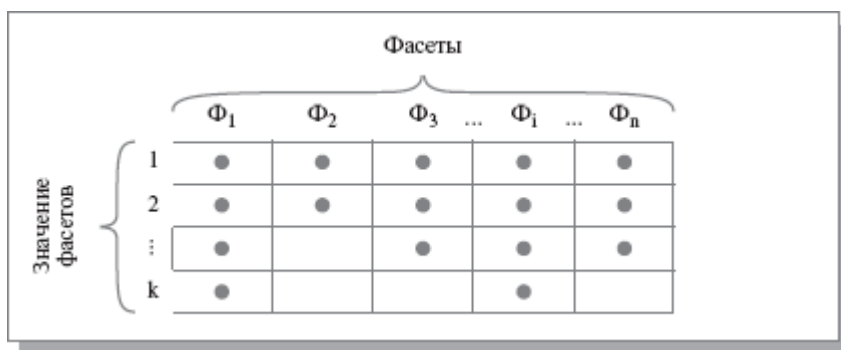


Рис. 2.2 Схема признаков фасетной классификации

Для выполнения многоаспектной дескрипторной классификации выделяется набор ключевых слов. Каждый объект характеризуется некоторой последовательностью ключевых слов и входит в соответствующую группу.

К преимуществам данной системы относят ее большую емкость и высокую степень гибкости, поскольку при необходимости можно вводить дополнительные фасеты и изменять их место в формуле. При изменении характера задач или характеристик объектов классификации разрабатываются новые фасеты или дополняются новыми признаками уже существующие фасеты без коренного изменения структуры всего классификатора.

К недостаткам многоаспектной системы можно отнести сложность ее структуры и низкую степень заполнения системы.

В современных классификационных схемах часто одновременно используются оба метода классификации. Это снижает влияние недостатков методов классификации и расширяет возможность использования классификаторов в информационном обеспечении управления.

ЛЕКЦИЯ 3

План лекции:

1. Системы кодирования и их виды.
2. Основные принципы работы с большими объемами данных.
3. Информационное обеспечение финансовых структур.
4. Виды информационных массивов.

3.1 Системы кодирования

Кодирование информации – это процедура создания специальных кодов и их использования для уменьшения объема информационной базы.

Система кодирования – это совокупность правил для составления кодов. При работе с экономической информацией большую ее часть составляет текст. Поэтому при работе с БД для исключения избыточности информации широко используются системы кодирования данных, в которых текстовая информация заменяется более короткими кодами. Кодирование информации предусматривает следующие действия:

- приведение к единообразию в обозначениях признаков, характеристик и объектов в целом;
 - упорядочение, классификацию и группировку всех номенклатур по определенным сходным признакам;
 - выбор системы кодирования и присвоение кодов;
- приведение информации к форме, удобной для обработки с помощью технических средств.

Ниже представлены виды систем кодирования.



Регистрационные системы могут быть использованы без предварительной классификации. В *порядковых* системах каждый объект получает свой порядковый номер без предварительной классификации. Недостатком является то, что вновь поступающий объект помещается в конец списка. В *серийных* системах для каждой группы объектов выделяется группа кодов, обычно с запасом. Можно обеспечить добавление данных в каждую группу.

Пример:

Объект	Коды
Цеха основного производства	10 - 19
Цеха вспомогательного производства	20 - 30
Отделы и службы	31 - 39

Классификационные системы кодирования основываются на предварительной классификации информации. Для использования *последовательных* систем сначала нужно выполнить иерархическую классификацию. Общий код объекта формируется при помощи включения кодов всех предыдущих уровней иерархии. При этом на каждом уровне иерархии может быть использована либо порядковая, либо серийная системы кодирования, но чаще – порядковая. Для формирования кода объекта берутся коды всех вышестоящих уровней и дописываются от высшего к низшему. По коду всегда можно судить о принадлежности к определенному классу, виду, подвиду и т.д. *Параллельные* системы основаны на фасетной классификации, где все объекты распределены по группам в соответствии с признаками классификации, и каждая группа имеет свой код.

При формировании любой БД обычно все наименования кодируются и составляют классификаторы или справочники. Кроме наименования в него могут быть включены дополнительные сведения об объекте. В дальнейшем вместо наименования используется код, и работа выполняется только с кодом.

Классификатор – это документ, с помощью которого осуществляется формализованное описание информации в информационной системе, содержащей наименования объектов, классификационных группировок и их кодовые обозначения.

Классификаторы могут иметь различную степень локализации (предприятие, отрасль, государство). Если обработка информации ведется только внутри предприятия, то кодирование осуществляется внутри предприятия. Если классификаторы имеют степень локализации выше, чем местная, то используются стандартные отраслевые (общегосударственные) классификаторы. Примеры общегосударственных классификаторов:

- классификатор промышленной и сельхозпродукции;
- классификатор предприятий и организаций;
- классификатор должностей.

Для наиболее распространенных видов документов унифицированы их формы. Например, единая система конструкторской документации (ЕСКД), единая система программной документации (ЕСПД).

Технология подготовки немашинного ИО служит основой для формирования внутримашинного ИО. Наиболее важными вопросами для него являются:

- определение состава документов, содержащих информацию, необходимую для решения задач;
- определение структурных единиц информации и их взаимосвязей;
- классификация и кодирование информации, обрабатываемой в задачах;
- разработка инструкций и методик по ведению документов.

Коды и классификаторы должны удовлетворять двум условиям:

1. Обеспечивать решение большого круга задач финансовых структур, быть общепринятыми, доступными и иметь необходимую резервную емкость на случай увеличения кодируемой информации.

2. Кодовое обозначение должно иметь минимальную длину для снижения затрат машинного времени на передачу и переработку информации. Кроме того, системы классификации и кодирования обычно дополняют системой защиты кодов, обеспечивающей контроль достоверности информации на входе и на выходе. При разработке классификаторов и систем кодирования рекомендуется соблюдать следующие основные требования:

1. *Выбор кодов минимальной длины.* Уменьшение длины кодов, особенно для часто используемых, приводит к уменьшению количества ошибок при переносе информации на машинные носители и сокращению трудоемкости их обработки.

2. *Логичность и запоминаемость кодов.* Выполнение данного требования помогает при освоении кодов, облегчает кодирование и уменьшает число допускаемых ошибок.

3. *Учет особенностей решаемых задач.*

4. *Учет существующей системы кодирования и общепринятых обозначений.*

5. *Учет перспектив развития.* Коды должны составляться таким образом, чтобы обеспечить возможности изменения и резерв на случай появления новых объектов в системе.

6. *Необходимость информационной стыковки.*

3.2 Особенности работы с большими объемами информации

Хранение больших объемов информации практически оправдано только при условии, если ее поиск осуществляется быстро и выдается она в доступной для понимания форме. Предназначенные для этого программные средства должны учитывать структуру информационного фонда и физические свойства запоминающей среды.

Сегодня неоспоримым фактом является то, что наличие большого объема информации не гарантирует оперативного получения необходимых сведений. Возможности электронно-вычислительной техники в последние два десятилетия значительно расширились: при осязательном увеличении ее способности хранить большие объемы информации стоимость хранения

постоянно сокращается. При этом развитие программных средств, связанных с организацией и поиском данных в “электронных хранилищах” происходило не такими быстрыми темпами. Поэтому специалисты, разрабатывающие программы для информационных систем, находятся в положении, когда они вынуждены все время поспевать за новыми возможностями вычислительной техники.

Рассмотрим принципы, лежащие в основе упомянутых ПС. Один из них заключается в том, что лучшая форма организации данных зависит от содержания последних и от того, как должна использоваться информация.

Например, существует коммерческая система типа Soundex, предназначенная для выявления авиапассажиров, зарезервировавших места на конкретный авиарейс. Soundex хранит фамилии в фонетической форме, что снижает количество ошибок, связанных с неправильным написанием, и позволяет отыскать нужную фамилию даже в том случае, если ее правильное написание неизвестно.

Наличие информации, относящейся к определенной области знаний, может повысить эффективность ПС в ходе решения одной задачи, но та же программа становится менее пригодной для других целей.

Вторым существенным фактором является физическая структура запоминающей среды. На магнитном диске данные записываются в виде отдельных блоков. Доступ к данным выполняется наиболее эффективно, если логические разделы также распределяются по отдельным блокам.

Таким образом, возможности ПС, ориентированных на обработку больших объемов информации, ограничиваются с одной стороны структурой используемых технических средств, а с другой – содержанием самой информации. Поэтому в задачи ПС, созданных для работы с информационными системами, входит умение увязать вместе оба вышеназванных ограничивающих фактора.

3.3 Информационное обеспечение финансовых структур

По определению ИО всю имеющуюся информацию систематизируют в специальные массивы, представляющие информационную базу (ИБ). В ее состав входят:

- нормативные и справочные данные – информационный базис;
- текущие сведения о состоянии ИС;
- текущие данные, поступающие извне ИС и требующие ответной реакции системы или влияющие на алгоритмы принятия решений;
- накапливаемые учетные и архивные сведения, необходимые для планирования и развития ИС.

Средства формализованного описания данных предназначены для эффективного поиска и идентификации необходимых данных в массивах, а также для организации доступа к данным внешних абонентов ИС. Эти средства включают в себя системы классификации и кодирования данных и

информационные языки для описания запросов к информационной базе и ответов системы.

Контролируют входные данные и ведение ИБ специальные ПС. Под ведением ИБ понимают обеспечение хранения и накопления данных, своевременное исключение устаревших данных, внесение и контроль изменений.

Основное назначение ИО состоит в создании динамической информационной модели объекта, отражающей его состояние в текущий и предшествующий момент времени. Исходя из назначения можно сформулировать основные требования к ИО:

1. Полнота отображения состояний системы и достоверности информации.

2. Высокая эффективность методов и средств сбора, хранения, накопления, обновления, поиска и выдачи данных.

3. Одноразовая регистрация и однократный ввод информации при ее многократном и многоцелевом использовании.

4. Простота и удобство доступа к данным ИБ.

5. Ввод и накопление в ИБ данных с минимальным дублированием.

6. Организация эффективной системы документооборота.

7. Развитие ИО путем наращивания данных, организации новых связей, а также проектирования более совершенных методов обработки информации.

8. Регламентация доступа к данным с различными условиями доступа и временем хранения документированной информации.

Основные элементы ИО финансовых структур (ФС) – это информационные массивы (ИМ), предназначенные для постоянного или временного хранения информации, которой обмениваются между собой внешние и внутренние по отношению к системе источники и потребители информации.

Необходимость в организации ИМ обусловлена многими факторами:

- несовпадением моментов поступления информации с моментами ее потребления;

- необходимостью хранения исходной информации, промежуточных и окончательных результатов в процессе исполнения программ и других процедур преобразования информации;

- использованием одних и тех же данных различными процедурами, выполняемыми как параллельно, так и последовательно;

- многократным длительным использованием некоторых данных различными процедурами.

Массив представляет собой совокупность данных, объединенных единым смысловым содержанием, которое обычно отражается в его названии. Основными элементами массивов, определяющими их содержание, являются *записи* – наименьшие элементы массива, которыми оперируют пользователи массива при обработке информации. Часто группы отдельных записей объединяют в более крупные образования – *блоки*. Наименьший элемент записи, имеющий единое смысловое значение – *информационное поле*.

3.4 Информационные массивы

Массивы ИО ФС можно классифицировать по различным признакам, основными из которых являются: семантическое содержание, технология использования, носитель информации, технические характеристики.

По семантическому или смысловому содержанию различают массивы:

- информации, необходимой для поддержания заданных эксплуатационных характеристик;
- подпрограмм операционной системы и тестовых программ;
- типовых программ, составляющих библиотеку стандартных программ;
- программ, используемых для решения индивидуальных задач.

Из перечисленных типов массивов три первых составляют ПО, поэтому далее рассматривается только последний тип массивов, содержащий данные, используемые и обрабатываемые в процессе работы ФС.

Технология использования ИМ имеет двоякий характер. С одной стороны, это технология обработки данных, связанная со спецификой формирования массивов, ввода, накопления и обновления данных, корректировки и передачи информации от одной программы обработки к другой и т.д. С другой стороны, существует технология, определяющая общие характеристики ФС. Это такие, как: последовательность и время обработки данных, необходимую точность решения задачи и ряд других ограничений. Все ИМ делят на входные, выходные и внутренние. Внутренние подразделяются на постоянные, вспомогательные, промежуточные, текущие и служебные.

Постоянные массивы формируют до начала функционирования системы обработки данных. Они содержат директивные, нормативные, справочные и другие редко меняющиеся данные. Содержание постоянных массивов может частично или полностью обновляться только в начале цикла обработки данных, а внутри него относительно редкие изменения претерпевают лишь их отдельные элементы. К постоянным массивам можно условно отнести те, в которых за один цикл изменяется не более 2% данных от их полного объема. Информация, хранящаяся в постоянных массивах, составляет ИБ системы. Поскольку эта информация имеет большое значение и время ее хранения велико, необходимо выбирать надежные носители, подходящие условия и методы хранения, контроля над содержанием массивов, обеспечивающие сохранность и достоверность данных, а также обеспечивать резервирование постоянных массивов.

Вспомогательные массивы получают из постоянных путем различных преобразований: сортировкой, объединением, выделением. Вспомогательные массивы позволяют использовать наиболее рациональные методы обработки данных, для реализации которых почему-либо не удобны постоянные массивы.

Промежуточные массивы содержат информацию, которая возникает на стыке решений различных задач или этапов решения одной из них. Эти

массивы являются результатом выполненного этапа и исходными данными для последующего.

Текущие массивы содержат переменную рабочую информацию о состоянии управляемого объекта или процесса во времени, а также о самом процессе функционирования. Характер возникновения, движения, хранения, контроля и подготовки этой информации существенно отличается от характера аналогичных процедур с информацией вышеуказанных массивов. Данные, из которых формируют текущие массивы, поступают или непрерывно в реальном масштабе времени, или пакетами в определенные или случайные моменты времени. Содержание этих массивов в течение цикла решения задачи может меняться. При этом сроки полного или частичного обновления массива случайны. В текущих массивах элементы могут исключаться, исправляться, заменяться, дополняться новыми. Иногда среди текущих массивов выделяют накапливаемые, которые отличаются тем, что содержащиеся в них элементы могут только дополняться новыми. Разнообразие содержания и большое количество тождественных преобразований, которым подвергается текущая информация в процессе ее передачи и обработки, повышают требования к контролю достоверности информации, а также к надежности работы подсистем, которые используются при обработке. Массивы постоянных и текущих данных называют главными массивами.

Служебные массивы содержат информацию, необходимую для работы со всеми вышеназванными массивами. Это могут быть программы, каталоги систем информационных массивов и стандартных программ, трансляторы, всевозможные машинные справочники.

ЛЕКЦИЯ 4

План лекции:

1. Специфика информационных языков.
2. Структура информационных языков.
3. Особенности языков банков данных.
4. Типовые схемы обработки данных.
5. Методы контролирования информации.

4.1 Специальные информационные языки

Проблемы построения и использования специальных информационных языков (ИЯ) получили в настоящее время большое значение в связи с ростом потока обрабатываемой информации. Использование естественного языка позволяет исключить индексирование и кодирование информации, снять ограничения на содержание запросов. Однако его применение для ввода информации и ее обработки увеличивает время обработки и требует исправления ошибок, появляющихся из-за непостоянства форм выражения запросов.

Наиболее важным отличием ИЯ, используемых для поиска и обработки информации, от естественных языков является их однозначность: каждому смысловому значению соответствует только одна последовательность символов и наоборот.

В ИЯ используют методы, позволяющие за счет специальных индексов определить группировки естественного языка, имеющие одинаковый смысл. В отличие от естественных, информационные языки значительно ограничивают возможности перефразирования. Это определяется не только устранением синонимов, но и использованием строгих правил синтаксиса, определяющих расположение знаков в записи. Для ИЯ фонетическое выражение знаков необязательно, так как они ориентированы на письменную форму выражения, состоящую из букв, цифр и некоторых специальных знаков.

В большинстве ИЯ имеют значение имена (понятия), так как они в основном предназначены для передачи существенных сведений о предметах и явлениях. Поэтому в ряде языков отсутствуют связки и весь аппарат выражения отношений между понятиями. Это так называемые языки без грамматики. В языках, поддерживающих грамматику, она выполняет подчиненную словарю роль и имеет иной характер по сравнению с грамматикой естественных языков.

Выделяется ряд требований, предъявляемых к ИЯ при их разработке и использовании. Поисковые образы обычно предназначены для того, чтобы в какой-то момент процедуры поиска их читал человек, выполняющий поиск. Поэтому для их составления нужно использовать словарь и синтаксис, которые были бы понятны человеку. Следовательно, информационный язык должен быть близок к естественному или мог бы быть легко изучен.

Многие массивы предназначены как для ручной системы поиска, так и для автоматизированной. Поэтому форма, словарь и синтаксис

информационного языка должны быть понятны и человеку, и компьютеру. Правильно построенный язык должен обладать возможностями модификации.

4.2 Структура информационных языков

Основные компоненты информационных языков – словарный состав и синтаксис. *Словарный состав* – это совокупность слов, используемых в данном языке. *Синтаксис языка* – совокупность правил построения из элементов словарного состава фразеологических (синтаксических) единиц с такими значениями, которые нельзя выразить с помощью отдельных слов основного словарного состава.

Под *словарем* ИЯ понимают списки, которые определяют вхождение логических элементов в словарный состав языка, а также их написание, употребление и значение. *Фразой* ИЯ является группа терминов, связанных вместе по определенным правилам. Под языковой структурой понимают совокупность языковых средств, обеспечивающих соответствующий информационный обмен между системами. Необходим выбор рациональной языковой структуры, обеспечивающей обмен информацией с минимумом затрат на перекодирование информации. Можно выделить следующие основные варианты языковых структур: с языковыми приоритетами; с независимыми равноправными языками в подсистемах; с языком-посредником; с единым языком.

Структура с языковыми приоритетами. В ней каждая подсистема имеет свой язык, однако при обмене информацией между подсистемами разных уровней всегда используется язык подсистемы более высокого уровня (используется в системах с иерархической организацией подчинения).

Структура с независимыми языками. В ней количество сообщений, кодируемых в каждой отдельной подсистеме, меньше, чем во всей системе. Это позволяет сократить среднюю длину кодовых сообщений, упростить кодирование и поиск информации в отдельной подсистеме, что уменьшает затраты внутри подсистемы на обработку сообщений. Однако каждая подсистема вынуждена хранить большое число трансляторов и словарей, а также затрачивать значительные средства на перевод информации с языка абонента на свой внутренний язык. Эти затраты тем выше, чем интенсивнее обмен между системами. Обычно число языков при такой структуре равно числу подсистем, каждый язык используется внутри одной подсистемы. Возможны два способа обмена информацией между подсистемами: на языке той системы, в которую обращаются с запросом, и на языке обращающейся подсистемы.

Структура с языком-посредником. В такой структуре в каждой системе может использоваться свой внутренний язык, а при обмене информацией между подсистемами используется язык-посредник. Он должен включать в себя все понятия, используемые при обмене информацией и обеспечивать однозначность всех первичных сообщений.

Структура с единым языком. Применение единого языка для обмена информацией между подсистемами удобно, так как при этом не требуется

перевод с одного языка на другой и отпадает необходимость в организации трансляторов. Однако требуются громоздкие словари, что ведет к большой длине кодовых обозначений и увеличению затрат на обработку информации.

В качестве критерия сравнения всех перечисленных структур информационных языков используется сумма полных экономических затрат на работу с информацией. В них входят затраты на обработку, кодирование, хранение и передачу информации.

4.3 Система языков банков данных

Банк данных рассматривают как единую многоцелевую информационную базу, обеспечивающую исходными данными решение различных по назначению и содержанию задач в режиме коллективного доступа к хранимой информации.

Банк данных состоит из совокупности данных большого объема, сложной структуры и комплекса программных средств для организации, ведения, обновления и использования информации, хранимой в банке. Описание данных отделено от программ, их использующих. Хранение данных в банке централизовано, дублирование хранимой информации снижено до минимума, организована выдача результатов в строгом соответствии с поступившими запросами. Банки данных обеспечивают независимость хранения и использования данных, четкую их структуризацию с указанием взаимосвязей между элементами банка, отсутствие избыточности, сохранность, защищенность и ограничение доступа к элементам банка, типизацию структур представления данных, типизацию и унификацию прикладных программ, возможность несложного изменения внутренней структуры данных.

Язык описания данных. Он предназначен для записи схем баз данных на трех уровнях. Эти уровни позволяют определить общую логическую структуру базы данных, независимую от прикладных программ и способов и физической реализации, а также определить ее отображение в структуре данных прикладных программ. Язык описания данных должен включать в себя средство определения механизмов защиты и секретности, вовлечения процедур изменений и вычисления виртуальных элементов данных.

Языки запросов. Служат для связи пользователей с банком данных. Бывает, что эти языки и их интерпретаторы (компиляторы) пишутся специально под приложение, чтобы лучше отразить его специфику. Язык запросов транслируется в общий базисный язык манипулирования данными. Его интерпретатор инициирует непосредственное обращение к банку данных. Кроме того, операторы языка манипулирования данными могут быть включены в языки программирования высокого уровня.

Язык управления. Для общего управления работой банка данных служит язык управления заданиями операционной системы либо специальный язык управления банком данных в зависимости от способа взаимодействия банка с операционной системой. Программное обеспечение банка данных – система

управления банка данных – содержит компиляторы (интерпретаторы) языка описания данных, языка манипулирования данными и языка запросов, а также набор сервисных программ.

4.4 Типовые схемы обработки данных

Большинство задач, решаемых в финансовых структурах, характеризуются огромными объемами обрабатываемой информации и различными видами обработки этой информации. Основными операциями для обработки такого рода информации являются поиск и сортировка. Процесс решения любой задачи можно разбить на этапы:

- сбор и передача исходных данных по каналам связи, а также первичная обработка данных с обеспечением заданной достоверности;
- ввод данных в вычислительную технику;
- решение задачи;
- обработка выходных данных и вывод;
- обработка и проверка полученных документов.

Степень использования каждой процедуры зависит от специфики решаемой задачи.

Исходные данные проходят ряд этапов первичной обработки:

- запись исходных данных на первичный документ;
- передача первичных документов в центр подготовки данных;
- преобразование и перезапись информации и в случае необходимости ее перезапись на промежуточный документ.

На каждом из этапов могут возникнуть ошибки, вызванные такими факторами, как несовершенство структуры обработки данных, недостатки в алгоритмах обработки данных, ошибки в программах, ненадежная работа оборудования, ошибки операторов. Для обработки первичных данных необходимо разработать такую технологию, которая позволила бы получить заданный уровень достоверности при минимальных затратах. Один из возможных путей решения названной задачи – это формализация и оптимизация методов достоверной обработки исходных данных. К ним относятся:

- методы рациональной обработки и контроля достоверности данных;
- методы оптимизации структуры обработки;
- методы поддержки характеристик оборудования в заданных пределах;
- методы контроля качества обработки.

Поскольку создание вечно надежных устройств невозможно, совершенствование технологий – один из путей повышения надежности.

Будем понимать под *контролем* систему мер, которые позволяют установить факт ошибочной работы оборудования. *Диагностика* – это локализация места некорректной работы системы.

Неисправности можно установить несколькими способами: программными средствами, аппаратными средствами, аппаратно-программными средствами.

Программные средства – это система тестов. *Аппаратные средства* – это специальные технические средства, которые не участвуют в вычислительном процессе, их единственная задача – осуществлять контроль.

Достоинством аппаратных средств является постоянный контроль в процессе работы системы. Недостатком считается высокая стоимость такого метода контроля. В системе аппаратных средств, предназначенных для контроля, должно быть намного меньше, чем аппаратных средств, задействованных в вычислительном процессе. В противном случае может возникнуть недоверие к средствам контроля, и появится необходимость контролировать их самих.

Программные средства контроля не требуют дополнительного оборудования, работая на основном, и обходятся гораздо дешевле. Однако ответ, который они выдают, может оказаться устаревшим, так как система могла выйти из строя после того, как завершился тест. Кроме того, к недостаткам программных средств относятся длительность и работы и невозможность выполнения основных программ во время прохождения контрольных тестов.

Реакцией на достоинства и недостатки аппаратных и программных средств контроля стало появление программно-аппаратных средств. В них часть задач возлагается на одну составляющую и часть – на другую. Функции контроля выполняют аппаратные средства, а функции диагностики возложены на программные тесты.

В первую очередь в ходе контролирования проверяется корректность обмена информацией внутри одной системы или между несколькими, во вторую – правильность обработки информации, а затем – все остальное.

Мажоритарный принцип организации контроля предусматривает многократный просчет одной и той же программы. Недостатком такого подхода является избыточность по времени. Он эффективен только при сбоях, так как при отказах постоянный неправильный результат может быть принят как правильный. Способ *аппаратного дублирования* основан на избыточном использовании аппаратуры, когда несколько вычислительных средств решают одну задачу, после чего сравниваются результаты и в качестве окончательного принимается тот, который был получен большее число раз.

Основой изложенных способов контроля является материальная или временная избыточность, что не всегда можно позволить. Альтернативным путем считается построение кодов для передачи информации. Идея кодирования данных заключается в том, что к информационным разрядам добавляются разряды, которые нужны только для контроля информационных.

Рассмотрим несколько способов контроля, применяемых для параллельно передаваемых кодов. Метод *контроля на четность* используется для проверки правильности передачи и хранения данных. К информационным разрядам добавляется один контрольный. С его помощью код, передаваемый

по линии связи, делается или четным, или нечетным, что анализируется на принимающей стороне. Если код оказывается четным числом – передача прошла успешно, в противном случае – ошибка. Правда, могут быть случаи, когда этот метод не обнаруживает ошибку. Контроль на четность определяет, не исправляя, только ошибки нечетной кратности, и не определяет ошибки четной кратности. Недостатки этого метода ликвидированы в *кодировании по Хэммингу*. Код Хэминга позволяет 100% обнаружить однократные и двукратные ошибки, а также 100% исправлять однократные ошибки. Это позволяет увеличить скорость выполнения операций, так как отпадает необходимость в поиске ошибок, как это было в предыдущем методе.

Контроль по Хэммингу состоит в том, что все разряды разбиваются на отдельные группы по определенному правилу и выполняется контроль на четность внутри каждой группы. По результатам внутригруппового контроля определяется номер испорченного разряда, после чего исправляется ошибка. Разбиение на группы осуществляется следующим образом. В первую группу включаются разряды, в двоичном номере которых в младшем разряде стоит единица. Во вторую группу попадают разряды, в двоичном номере которых во втором разряде стоит единица. Такое распределение по группам выполняется для всех разрядов, имеющих в передаваемом коде. Контрольными разрядами назначаются те, номера которых совпадают со значением степени числа 2.

Номер испорченного разряда определяется по результатам контроля на четность внутри группы, т.е. номер разряда будет совпадать с числом, которое выдается в случае ошибки. Если не выполнять требование о назначении номеров контрольных разрядов, то будет сильно затруднено определение испорченных разрядов.

ЛЕКЦИЯ 5

План лекции:

1. Определение понятий «информация» и «информатика». Многоаспектность понятия «информатика».

2. Понятие информационной системы (ИС). Классификация и состав ИС.

3. Структура и состав ИС.

Хранение больших объемов информации практически оправдано только при условии, если ее поиск и обработка осуществляется быстро и выдается она в доступной для понимания форме. Предназначенные для этого программные средства должны учитывать структуру информационного фонда и физические свойства запоминающей среды.

Возможности электронно-вычислительной техники в последние десятилетия значительно расширяются: при осязаемом увеличении ее способности хранить большие объемы информации стоимость хранения постоянно сокращается. При этом развитие программных средств, связанных с организацией и поиском данных в «электронных хранилищах» происходит не такими быстрыми темпами. Поэтому специалисты, разрабатывающие программы для информационных систем, вынуждены все время поспевать за новыми возможностями вычислительной техники.

Большинство существующих подходов к анализу текстов можно разбить на два класса. К первому классу относятся простые, быстрые, но не очень точные механизмы анализа. Чаще всего эти подходы используют формальные статистические методы, основанные на частоте появления в тексте слов различных тематик. Вторым классом формируют достаточно изощренные, дающие хороший результат, но сравнительно медленные подходы, основанные на лингвистических методах. Эффективным же можно считать такой подход, который сочетал бы в себе «простоту» статистических алгоритмов с достаточно высоким качеством обработки лингвистических методов.

В то же время, как показала практика, для достижения приемлемого качества в решении практических задач компьютерного анализа текстовой информации (автоматическое аннотирование, тематическая классификация и т. д.) не требуется полный грамматический анализ фразы. Достаточно выделить наиболее информативные единицы текста: ключевые слова, словосочетания, предложения и фрагменты. При этом в качестве характеристики информативности удобно выбрать частоту повторения слов в тексте.

Для выделения понятий текста, представляющих слова и связанные словосочетания, применяется статистический алгоритм, основанный на анализе частоты встречаемости цепочек слов различной длины и их вхождения друг в друга. Во всех созданных человеком текстах можно выделить статистические закономерности: независимо от текста и языка написания внутренняя структура текста остается неизменной. Она

описывается законами Дж. Зипфа, который предположил, что слова с большим количеством букв встречаются в тексте реже коротких слов. Основываясь на этом постулате, Зипф вывел два универсальных закона.

5.1 Первый закон Зипфа «Ранг - частота»

Если измерить количество вхождений каждого слова в текст и взять только одно значение из каждой группы, имеющей одинаковую частоту, расположить частоты по мере их убывания и пронумеровать (порядковый номер частоты называется рангом частоты – r), то наиболее часто встречающиеся слова будут иметь ранг 1, следующие за ними – 2 и т. д. Вероятность P_i встретить произвольно выбранное слово равна отношению количества вхождений этого слова n_i к общему числу слов n в тексте, т. е.

$$p_i = n_i / n.$$

(5.1)

Зипф обнаружил следующую закономерность: произведение вероятности обнаружения слова в тексте на ранг частоты r – константа (C):

$$C = p_i \cdot r \quad \text{или} \quad C = (n_i \cdot r) / n.$$

(5.2)

Это функция типа $y = k/x$, а её график – равносторонняя гиперболола. Следовательно, по первому закону Зипфа, если самое распространенное слово встречается в тексте, например, 100 раз, то следующее по частоте слово с высокой долей вероятности окажется на уровне 50.

Значение константы C в разных языках различно, но внутри одной языковой группы остается неизменным, в не зависимости от текста. Так, например, для английских текстов константа Зипфа равна приблизительно 0,1.

5.2 Второй закон Зипф «Количество - частота»

В первом законе не учтён тот факт, что разные слова могут входить в текст с одинаковой частотой. Зипф установил, что частота и количество слов, входящих в текст с одинаковой частотой, тоже связаны между собой. Если построить график, отложив по оси X частоту вхождения слова, а по оси Y – количество слов с данной частотой, то получившаяся кривая будет сохранять свои параметры для всех без исключения созданных человеком текстов. Как и в предыдущем случае, это утверждение верно в пределах одного языка. Однако и межъязыковые различия невелики. На каком бы языке текст не был написан, форма кривой Зипфа останется неизменной (рисунок 5.1). Могут немного отличаться лишь коэффициенты, отвечающие за наклон кривой.

Законы Зипфа универсальны. В принципе они применимы не только к текстам. Аналогично определяется, например, зависимость количества городов от числа проживающих в них жителей. Воспользуемся законами Зипфа для извлечения из текста слов, отражающих его смысл, т. е. ключевых слов. На рисунке 5.2 показан график зависимости ранга частоты слов от частоты их вхождения в текст.

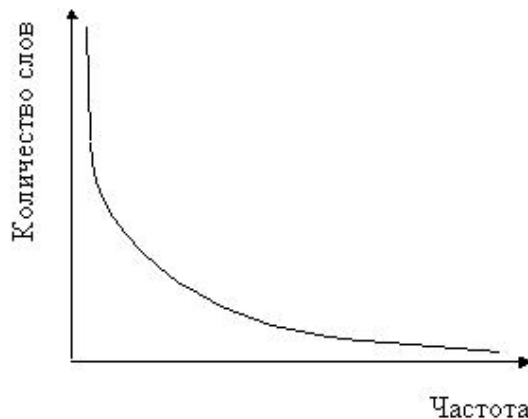


Рисунок 5.1 – График зависимости частоты вхождения слова от количества слов с данной частотой

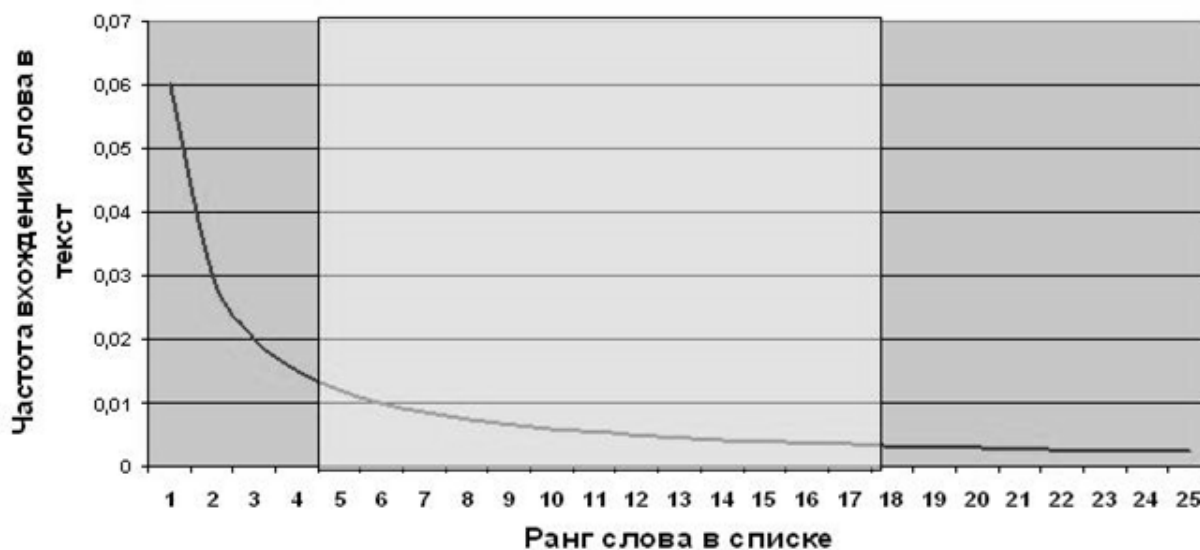


Рисунок 5.2 – График зависимости ранга частоты от частоты вхождения слова

Исследования показали, что наиболее значимые слова лежат в средней части диаграммы (на рисунке 5.2 это слова, имеющие ранг от 4 до 17). Это объясняется тем, что слова, которые встречаются слишком часто, в основном оказываются предлогами и местоимениями, в английском языке – артиклями и т. п. Редко встречающиеся слова, как правило, тоже не имеют решающего смыслового значения.

От того как будет определен диапазон значимых слов, зависит многое. Если диапазон широкий, то ключевыми словами будут вспомогательные слова; если установить узкий диапазон, то можно потерять смысловые термины.

Выделить ключевые слова помогает предварительное исключение из исследуемого текста некоторых слов, которые априори не являются значимыми, а, поэтому являются «шумом». Такие слова называются нейтральными или стоповыми (стоп-словами). Словарь стоп-слов называют стоп-листом. Например, для английского текста стоп-словами считаются термины: the, a, an, in, to, of, and, that... и т. д.

5.3 Весовые коэффициенты

Каждый из рассматриваемых отдельно взятых документов на практике чаще всего входит в базу данных (коллекцию) наряду с множеством других документов. Если представить всю базу данных как единый документ, к ней можно применить те же законы, что и к единичному документу.

В ходе назначения терминов (ключевых слов) коллекции стремятся избавиться от лишних слов (шума) и в то же время поднять рейтинг значимых слов, для чего вводят инверсную частоту термина. *Значение этого параметра тем меньше, чем чаще слово встречается в документах базы данных.* Вычисляют его по формуле

$$\lambda_i = \log(N / N_i),$$

(5.3)

где λ_i – инверсная частота термина i ;
 N – количество документов в базе данных;
 N_i – количество документов с термином i .

Теперь каждому термину можно присвоить весовой коэффициент, отражающий его значимость

$$k_{ij} = n_{ij} \cdot \lambda_i,$$

(5.4)

где k_{ij} – вес термина i в документе j ;
 n_{ij} – частота термина i в документе j ;
 λ_i – инверсная частота термина i .

В качестве ключевых слов будут выбираться слова, имеющие высокий вес.

Современные способы индексирования не ограничиваются анализом перечисленных параметров текста. Поисковая машина может строить весовые коэффициенты с учетом местоположения термина внутри документа, взаимного расположения терминов, частей речи, морфологических особенностей и т. п. В качестве терминов могут выступать не только отдельные слова, но и словосочетания. Без этих законов Зипфа сегодня не обходится ни одна система автоматического поиска информации. Это

обусловлено тем, что математический анализ позволяет машине с хорошей точностью без участия человека распознать суть текста.

5.4 Особенности поиска информации

Накопленные к настоящему времени объемы информации в совокупности с непрерывно увеличивающимися темпами ее роста определяют актуальность и значимость исследований в области информационного поиска. Быстрое развитие сетевых технологий, в том числе и Интернета, способствует значительному увеличению доступных информационных ресурсов и объемов передаваемой информации. Зачастую это разнородная, слабо структурированная и избыточная информация, обладающая высокой динамикой обновления. При сегодняшних объемах доступной информации решение задач информационного поиска становится не только приоритетным, но и элементарно необходимым для обеспечения своевременного доступа к интересующей информации.

Информационный поиск – самостоятельное направление исследований, изучающее вопросы поиска документов, обработки результатов поиска, а также целый ряд смежных вопросов: моделирования, классификации, кластеризации и фильтрации документов, проектирования архитектур поисковых систем и пользовательских интерфейсов, языки запросов, и т. д.

Способы поиска можно разделить на две большие группы.

1. Библиографический поиск или поиск «по каталогу».

Такой вариант поиска обеспечивает нахождение документов по их выходным данным, например: по названию документа, его тематике, именам авторов, датам публикаций и т. д. Эти выходные данные составляют реквизиты документа. Основой каталога является предварительно заданная модель представления реквизитов, реализованная в виде базы данных, в соответствии с которой обеспечивается запись отдельных элементов реквизитов и последующий поиск по ним.

Основная проблема и недостаток такого варианта поиска – это необходимость выполнения значительного объема работ по предварительной организации, наполнению каталога. Как правило, это ручная классификация на основе привлечения экспертов. Подобный подход позволяет организовать лишь самую малую толику доступных информационных ресурсов.

2. Тематический поиск или поиск «по тексту».

Этот вариант поиска ориентирован на нахождение документов по их содержанию. Сюда же относится так называемый полнотекстовый поиск. Общая схема такого поиска заключается в формулировании некоторого запроса пользователем относительно содержания документа и отборе из множества доступных документов тех, которые удовлетворяют запросу. Такой вариант поиска удобен прежде всего, тем, что нет необходимости в предварительном разделении документов по различным категориям. Особенно это актуально при значительном объеме доступных документов, высокой динамике их обновления или отсутствии некоторых реквизитов.

Такая ситуация характерна для Интернета. Основная проблема такого поиска – это сложность однозначной автоматической интерпретации содержания текстов документов и формулировок информационных потребностей пользователей. Сложность интерпретации затрудняет определение соответствия рассматриваемого документа информационным потребностям пользователя. Эти проблемы обусловлены отсутствием какой-либо регулярной структуры у текстовых документов на естественном языке. Такие информационные ресурсы принято называть неструктурированными или слабоструктурированными. Разработка методов анализа слабоструктурированных информационных ресурсов представляется весьма перспективным и многообещающим направлением исследований в области информационного поиска.

В соответствии с вышеприведенной классификацией способов поиска принято выделять два основных класса информационно-поисковых систем:

- поисковые каталоги;
- поисковые системы.

Поисковые каталоги в большей степени ориентированы на структурную организацию тематических коллекций с удобной системой ссылок и иерархией документов по тематическим коллекциям. Это позволяет пользователю самостоятельно находить требуемый документ, просматривая структуру каталога, либо использовать механизмы поиска, ориентированные на данный каталог. Поисковые системы работают со слабоструктурированной информацией. Как правило, они используются для поиска документов в больших и динамичных информационных коллекциях, например в Интернете. Особенностью таких коллекций является отсутствие четко выраженной структурной организации, позволяющей упорядочить и однозначно классифицировать хранящиеся в них документы по тематической направленности.

В рамках данного методического пособия наибольший интерес представляют поисковые системы, а точнее используемые в них методы анализа документов. Процесс поиска текстовой информации, реализуемый типичной поисковой системой, включает в себя следующие этапы:

- формализация пользователем поискового запроса (представление пользователем в том или ином виде своих информационных потребностей);
- предварительный отбор документов по формальным признакам наличия интересующей информации (например, наличие в тексте документа одного из слов запроса, если запрос формулируется на естественном языке);
- анализ отобранных документов (лингвистический, статистический);
- оценка соответствия смыслового содержания найденной информации требованиям поискового запроса (ранжирование).

Одним из ключевых понятий, характеризующих выбор того или иного метода анализа текстовой информации, а также реализацию конкретного варианта поиска, является модель поиска. Модель поиска – это сочетание следующих составляющих:

- способа представления документов;

- способ представления поисковых запросов;
- вида критерия релевантности документов.

Всю совокупность представленных на сегодняшний день методов тематического анализа текста можно разделить на две группы:

- лингвистический анализ;
- статистический анализ.

Первый ориентирован на извлечение смысла текста по его семантической структуре, второй – по частотному распределению слов в тексте. Однако говорить о принадлежности какого-либо из подходов к конкретной группе можно лишь условно, как правило, в реальных задачах обработки текста приходится использовать сочетание методик из обеих групп.

Лингвистический анализ можно разделить на четыре взаимодополняющих анализа: лексический, морфологический, синтаксический, семантический.

Статистический анализ – это, как правило, частотный анализ в тех или иных его вариациях. Суть такого анализа заключается в подсчете количества повторений слов в тексте и использовании результатов подсчета для конкретных целей, например вычисление весовых коэффициентов ключевых слов. Одним из эффективных статистических подходов является способ поиска по документу-образцу.

Документ-образец выступает в качестве одной из форм представления информационных потребностей пользователя. Целью поиска является обнаружение тематически близких документов. Самым простым подходом к решению задачи поиска документов по образцу является использование всех слов документа-образца в качестве запроса. Однако длина такого запроса может оказаться очень большой, что отрицательно скажется на качестве поиска, т. к. результатом поиска будут все документы, в которых присутствовали данные слова, и таких документов может быть очень много. Это отрицательно скажется как на самой поисковой системе – вычислительные ресурсы и трафик не безграничны и система может оказаться перегруженной, так и на человеке – просмотр и анализ найденных документов может занять значительное время, редкий пользователь готов к этому. Приемлемым вариантом в данном случае является выделение тематики документа. Под тематикой понимается множество ключевых слов, описывающих с некоторой степенью адекватности содержание документа. Тематика – это приближенное представление документа. Для повышения точности и адекватности описания содержания документа ключевые слова используются с некоторыми весовыми коэффициентами, которые соотносятся с частотой повторений этих слов в тексте. Вопросы выделения тематики и вычисления тематической близости документов по их тематическому представлению во многом и определяют возможность и эффективность поиска по документу-образцу.

5.5 Алгоритм поиска

Всю коллекцию документов необходимо организовать так, чтобы можно было легко отыскать в ней нужный материал. База данных должна взаимодействовать с пользовательским запросом. Запросы могут быть простыми, состоящими из одного слова, и сложными – из нескольких слов, связанных логическими операторами. Простой запрос оправдывает свое название. Пользователь вводит слово, машина ищет его в списке терминов и выдает все связанные с термином ссылки. Структура такой базы данных проста. Взаимодействие со сложными запросами требует более сложной организации.

Рассмотрим последовательность действий для организации поиска.

1. Подбирается текст-источник. Чем четче описаны проблемы в тексте-источнике, тем качественнее и точнее окажется результат.
2. Удаляются из текста стоп-слова.
3. Без учета морфологии слов вычисляется частота вхождения каждого термина.
4. Ранжируются термины в порядке убывания их частоты вхождения.
5. Выбирается диапазон частот из середины упорядоченного списка, построенного по определенному закону. Достаточно взять 10 – 20 терминов.
6. Составляется запрос из отобранных слов в порядке их следования в списке терминов. Запрос должен восприниматься машиной как слова, связанные логическим оператором ИЛИ.
7. Запрос отправляется поисковой системе.

ЛЕКЦИЯ 6

План лекции:

1. Алгоритм автоматической рубрикации текстов по образцу.
2. Разделение объектов на N классов методом персептрона.

6.1 Алгоритм автоматической рубрикации текстов по образцу

1. Применение законов Зипфа для построения векторов признаков (ключевых слов) текстов-образцов, задающих необходимые рубрики.
2. Обучение системы классификации с помощью текстов-образцов. Применение метода персептрона для построения p линейных функций (по числу классов), отделяющих в r -мерном пространстве признаков каждый класс от всех остальных.
3. Построение по законам Зипфа наборов ключевых слов для текстов, требующих рубрикации.
4. Рубрикация текстов с помощью разделяющих функций, построенных на втором шаге алгоритма.

Рассмотрим более подробно каждый из шагов алгоритма.

1. Для построения векторов с признаками текстов-образцов создадим словарь терминов. Каждая рубрика может быть задана произвольным количеством текстов-образцов. Пусть векторы признаков всех рубрик, чье количество обозначим a , имеют равное количество ключевых слов – b . Тогда словарь ключевых слов можно представить следующей динамической структурой (рисунок 6.1). Где в вертикальной таблице расположены заголовки классов: от 1 до a . Каждый элемент таблицы содержит ссылку на список ключевых слов данного класса. Списки могут быть разной длины в зависимости от количества текстов-образцов, имеющих для каждой рубрики.

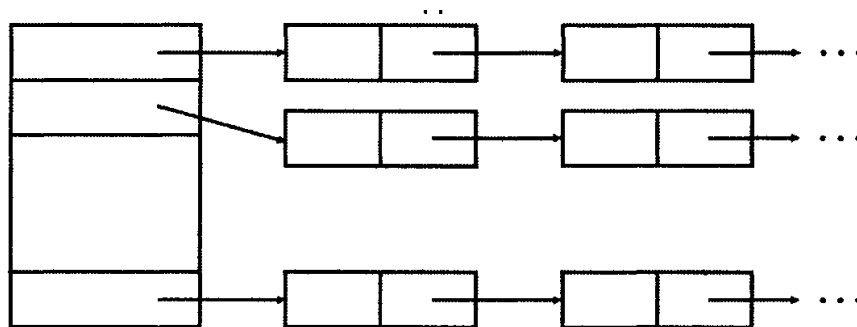


Рисунок 6.1 – Структура словаря терминов рубрик

Вектор признаков каждого класса представим следующей структурой.

x_{ijk}	$x_{i,j,k+1}$...	$x_{i,j,b}$
-----------	---------------	-----	-------------

Здесь x – это ключевые слова j -го текста-образца i -й рубрики. Индекс i изменяется от 1 до a , индекс j зависит от представленного количества образцов для данной рубрики. Индекс k – это номер ключевого слова в векторе, поэтому он изменяется от 1 до b .

На основе векторов признаков всех образцов для всех рубрик определим структуру обобщенного вектора признаков, на базе которого будем строить разделяющие функции. В обобщенный вектор войдут все ключевые слова из всех текстов-образцов. Рассмотрим на примере построение обобщенного вектора ключевых слов.

Пусть имеются следующие исходные данные: $a = 3$, $b = 2$. Рубрики заданы следующими векторами признаков.

$x_{1,1,1}$	$x_{1,1,2}$	$x_{2,1,1}$	$x_{2,1,2}$	$x_{3,1,1}$	$x_{3,1,2}$
$x_{1,2,1}$	$x_{1,2,2}$			$x_{3,2,1}$	$x_{3,2,2}$

Для первого класса предложены два образца, для второго – один и для третьего – два образца. При этом второй признак образца второго класса совпадает с первым признаком второго образца третьего класса, т. е.

$$x_{2,1,2} = x_{3,2,1}$$

Тогда структура обобщенного вектора признаков будет следующей:

$x_{1,1,1}$	$x_{1,1,2}$	$x_{1,2,1}$	$x_{1,2,2}$	$x_{2,1,1}$	$x_{2,1,2}$	$x_{3,1,1}$	$x_{3,1,2}$	$x_{3,2,1}$	$x_{3,2,2}$
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

Теперь каждый из пяти имеющихся текстов-образцов будет также описан вектором из десяти элементов. В позиции вектора, относящиеся к ключевым словам, имеющимся в данном образце, занесем 1 (позже там будем стоять частота встречаемости слова в тексте, определенная по законам Зипфа), в остальные позиции запишем нули. Получим следующие векторы признаков для имеющихся пяти текстов-образцов (рисунок 6.2).

2. После того как для каждого текста-образца составлен вектор признаков (рисунок 6.2), применим метод персептрона для построения разделяющих функций по числу заданных рубрик (метод персептрона описан ниже).

1	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	0	0	0	1	1	0	0	1	0
0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	1	0	0	1	1

Рисунок 6.2 – Векторы признаков для построения разделяющих функций

3. Вектор признаков незнакомого текста имеет такую же структуру и размер, как и вектор признаков текста-образца. Сначала определим вектор признаков, состоящий из b элементов, незнакомого текста. Затем проверим, есть ли среди признаков ключевые слова, входящие в обобщенный вектор. Пусть первоначально вектор признаков незнакомого текста имеет вид

y_1	y_2
-------	-------

При этом $y_1 = x_{1,1,2}$; $y_2 = x_{2,1,2} = x_{3,2,1}$. Тогда вектор признаков незнакомого текста примет следующий вид (рисунок 6.3):

0	1	0	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Рисунок 6.3 – Вектор признаков незнакомого текста

Вектор признаков незнакомого текста подставляется в каждую из разделяющих функций, и текст относится к той рубрике, чья функция показала максимальный результат.

6.2 Разделение объектов на N классов методом персептрона

В настоящее время существует большое множество методов классификации, в том числе и текстовой информации. Однако все методы распознавания можно разделить на две группы. Первая основана на понятии пространства признаков и их обработки в этом пространстве. Вторая – на

исследовании конструкции рассматриваемых образов (синтаксическое распознавание).

Для решения поставленной задачи рассмотрим особенности первой группы методов. Для них в качестве основополагающей принята гипотеза о возможности представления образа в виде вектора, принадлежащего множеству V . Множество образов представляется в виде множества векторов, состоящего из таких N подмножеств, что каждый вектор, отнесенный в результате классификации к j -му классу, принадлежит подмножеству E_j .

Свойства множества V могут быть записаны в виде

$$\bigcup_{i=1}^N E_i = V, E_i \cap E_j = 0 (\forall i \neq j).$$

Задача классификации состоит в отыскании функции f , обеспечивающей разделение пространства V на требуемые классы:

$$f: V \rightarrow \Pi(V).$$

Процедура классификации состоит в том, чтобы для каждой области R_i найти такую решающую функцию $g_i(x)$, что если

$$g_i(\bar{x}) > g_j(\bar{x}), \text{ то } \bar{x} \in R_i \quad \forall j = 1, 2, \dots, N,$$

где N – общее количество областей.

Разделяющую функцию часто представляют в виде линейной суммы

$$g(\bar{x}) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n,$$

где ω_i – весовые коэффициенты, каждый из которых относится к определенной составляющей.

Для удобства записи вводится весовой коэффициент с нулевым индексом ω_0 . Это позволяет записать решающую функцию в более компактной форме:

$$g(\bar{x}_a) = \bar{\omega} \bar{x}_a,$$

где $\bar{x}_a = \{1, x_1, x_2, \dots, x_n\}$ – вектор, в число составляющих которого входит дополнительно одна вещественная константа, чью величину обычно принимают равной единице.

Решающее правило d для случая N сепарабельных классов ($N > 2$) можно записать следующим образом:

$$d = \begin{cases} c_i, & \text{если } g_i(\bar{x}) = \bar{\omega}_i \bar{x}_a \geq 0, \\ \bar{c}_i, & \text{если } g_i(\bar{x}) < 0 \end{cases},$$

где C – множество, состоящее из N классов. $C = \{c_1, c_2, \dots, c_N\}, c_i + \bar{c}_i = C$.

В процессе построения разделяющей функции основная задача заключается в том, чтобы найти весовые коэффициенты вида $\bar{\omega}_i = \{\omega_{0i}, \omega_{1i}, \dots\}$ для каждого конкретного применения.

Рассмотрим один из вариантов применения линейных разделяющих функций для разбиения объектов на N классов.

Существует M решающих функций $d_k(x) = w_k x$, $k = 1, 2, \dots, M$, таких, что если образ x принадлежит классу ω_i , то $d_i(x) > d_j(x)$ для всех $j \neq i$.

Граница между классами ω_i и ω_j определяется теми значениями вектора x , при которых имеет место равенство $d_i(x) = d_j(x)$. Поэтому при выводе уравнения разделяющей границы для классов ω_i и ω_j значения решающих функций $d_i(x)$ и $d_j(x)$ используются совместно.

Пример расположения разделяющих функций приведен на рисунке 6.4.

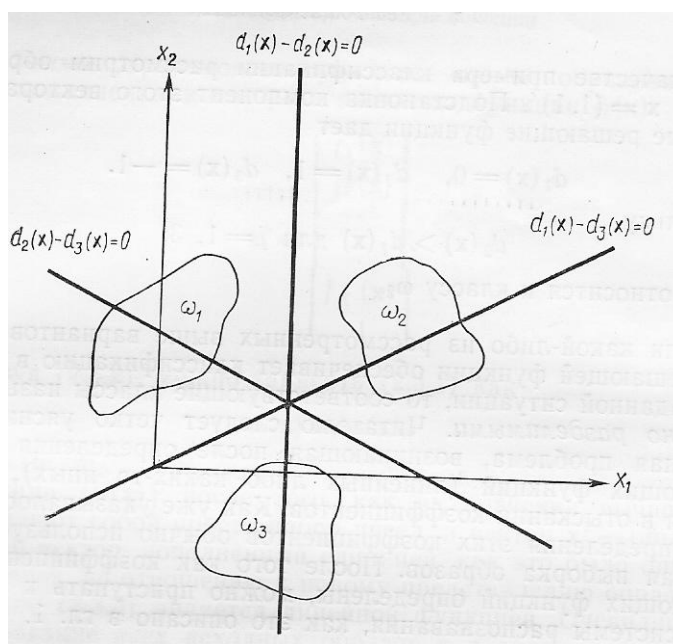


Рисунок 6.4 – Случай разделения образов на три класса

Для образов, принадлежащих классу ω_1 , должны выполняться условия $d_1(x) > d_2(x)$, $d_1(x) > d_3(x)$. В общем случае требуется, чтобы входящие в класс ω_i образы располагались в положительных зонах поверхностей $d_i(x) - d_j(x) = 0$, $j = 1, 2, \dots, M, i \neq j$. Положительная зона границы $d_i(x) - d_j(x) = 0$ совпадает с отрицательной зоной границы $d_j(x) - d_i(x) = 0$.

Пусть в качестве решающих функций выбраны следующие:
 $d_1(x) = -x_1 + x_2$, $d_2(x) = x_1 + x_2 - 1$, $d_3(x) = -x_2$. Разделяющие границы для трех классов выглядят при этом так:

$$\begin{aligned} d_1(x) - d_2(x) &= -2x_1 + 1 = 0, \\ d_1(x) - d_3(x) &= -x_1 + 2x_2 = 0, \\ d_2(x) - d_3(x) &= x_1 + 2x_2 - 1 = 0. \end{aligned}$$

Для того чтобы определить область решений, соответствующую классу ω_1 , необходимо выделить область, в которой выполняются неравенства $d_1(x) > d_2(x)$, $d_1(x) > d_3(x)$. Эта область совпадает с положительными зонами для прямых $-2x_1 + 1 = 0$ и $-x_1 + 2x_2 = 0$.

Область принятия решения о принадлежности образа классу ω_2 совпадает с положительными зонами для прямых $2x_1 - 1 = 0$ и $x_1 + 2x_2 - 1 = 0$. Область, отвечающая классу ω_3 , определяется положительными зонами для прямых $x_1 - 2x_2 = 0$ и $-x_1 - 2x_2 + 1 = 0$ (рисунок 6.5).

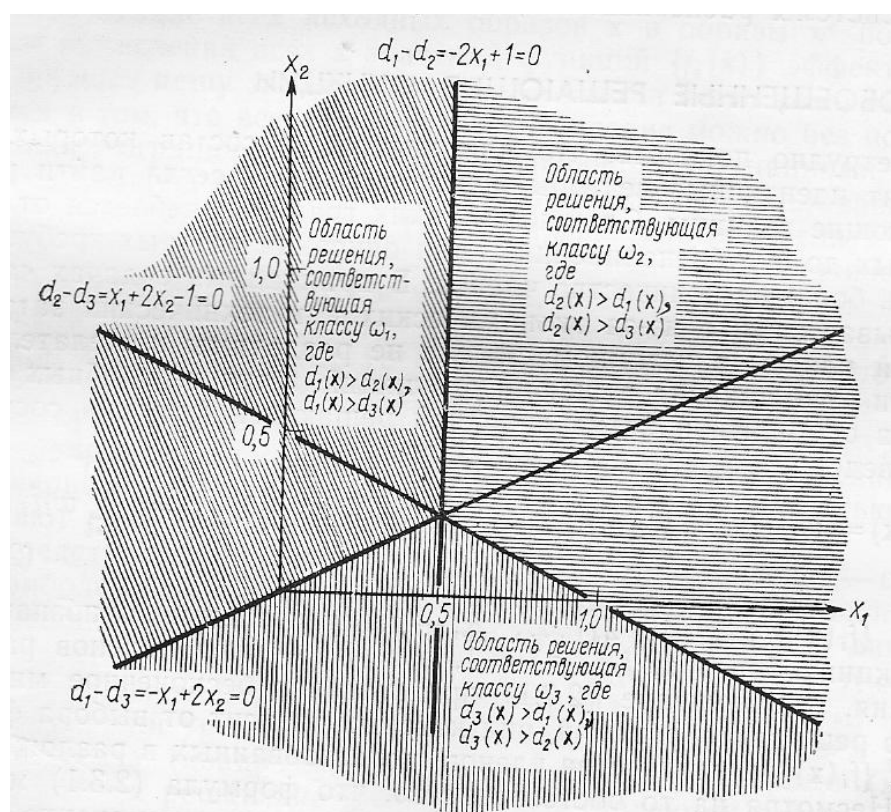


Рисунок 6.5 – Иллюстрация разделения объектов на несколько классов

В качестве примера классификации рассмотрим обработку образа $x = (1, 1)$. Подстановка признаков образа в выбранные решающие функции дает следующие значения:

$$d_1(x) = 0, d_2(x) = 1, d_3(x) = -1.$$

Поскольку $d_2(x) > d_j(x)$, $j = 1, 3$, образ относится к классу ω_2 .

Если какой-либо из рассмотренных вариантов линейной решающей функции обеспечивает классификацию в некоторой заданной ситуации, то соответствующие классы называются *линейно разделимыми*. Основная проблема, возникающая после определения набора решающих функций, заключается в отыскании коэффициентов. Для их определения обычно используется доступная выборка образов. После того как коэффициенты всех

решающих функций определены, можно приступить к построению системы распознавания.

Рассмотрим алгоритм – *метод перцептрона*, который можно применить для определения решающих функций, когда допускается существование M решающих функций, характеризующихся тем свойством, что при $x \in \omega_i$, где x – объект, ω_i – класс $d_i(x) > d_j(x)$ для всех $i \neq j$.

Рассмотрим M классов $\omega_1, \omega_2, \dots, \omega_M$. Пусть на k -м шаге процедуры обучения системе предъявляется образ $x(k)$, принадлежащий классу ω_i . Вычисляются значения M решающих функций

$d_j[x(k)] = w_j(k)x(k)$, $j = 1, 2, \dots, M$. Затем, если выполняются условия $d_i[x(k)] > d_j[x(k)]$, $j = 1, 2, \dots, M; j \neq i$, то векторы весов не изменяются, т.е. $w_j(k+1) = w_j(k)$, $j = 1, 2, \dots, M$.

С другой стороны, допустим, что для некоторого l $d_i[x(k)] \leq d_l[x(k)]$. В этом случае выполняются следующие коррекции весов:

$$\begin{aligned} w_i(k+1) &= w_i(k) + cx(k), \\ w_l(k+1) &= w_l(k) - cx(k), \\ w_j(k+1) &= w_j(k), \quad j = 1, 2, \dots, M; j \neq i, j \neq l, \end{aligned} \quad (6.1)$$

где c – положительная константа.

Если классы разделимы, то доказано, что этот алгоритм сходится за конечное число итераций при произвольных начальных векторах. Рассмотрим это на примере.

Даны классы, причем каждый из них содержит один образ: $\omega_1: \{(0, 0)\}$, $\omega_2: \{(1, 1)\}$, $\omega_3: \{(-1, 1)\}$. Дополним заданные образы: $(0, 0, 1)$, $(1, 1, 1)$, $(-1, 1, 1)$. Выберем в качестве начальных векторов весов $w_1(1) = w_2(1) = w_3(1) = (0, 0, 0)$, положим $c = 1$ и, предъявляя образы в указанном порядке, получим следующее:

$$\begin{aligned} d_1[x(1)] &= w_1(1)x(1) = 0, \\ d_2[x(1)] &= w_2(1)x(1) = 0, \\ d_3[x(1)] &= w_3(1)x(1) = 0. \end{aligned}$$

Поскольку $x(1) \in \omega_1$ и $d_2[x(1)] = d_3[x(1)] = d_1[x(1)]$, первый весовой вектор увеличивается, а два других уменьшаются согласно соотношениям (6.1), т.е.

$$w_1(2) = w_1(1) + x(1) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$w_2(2) = w_2(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},$$

$$w_3(2) = w_3(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

Следующий предъявляемый образ $x(2) = (1, 1, 1)$ принадлежит классу ω_2 . Для него получаем $w_1(2)x(2) = 1, w_2(2)x(2) = -1, w_3(2)x(2) = -1$.

Поскольку все произведения больше либо равны $w_2(2)x(2)$, вводятся коррекции

$$w_1(3) = w_1(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix},$$

$$w_2(3) = w_2(2) + x(2) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

$$w_3(3) = w_3(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ -2 \end{pmatrix}.$$

Следующий предъявленный образ $x(3) = (-1, 1, 1)$ принадлежит классу ω_3 . Для него получаем $w_1(3)x(3) = 0, w_2(3)x(3) = 0, w_3(3)x(3) = -2$. Все эти произведения опять требуют корректировки:

$$w_1(4) = w_1(3) - x(3) = \begin{pmatrix} 0 \\ -2 \\ -1 \end{pmatrix},$$

$$w_2(4) = w_2(3) - x(3) = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix},$$

$$w_3(4) = w_3(3) + x(3) = \begin{pmatrix} -2 \\ 0 \\ -1 \end{pmatrix}.$$

Поскольку в данном цикле итерации присутствовали ошибки, следует провести новый цикл. Положив $x(4)=x(1), x(5)=x(2), x(6)=x(3)$, получим $w_1(4)x(4) = -1, w_2(4)x(4) = -1, w_3(4)x(4) = -1$. Так как образ $x(4)$ принадлежит классу ω_1 , то все произведения «неверны». Поэтому

$$w_1(5) = w_1(4) + x(4) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(5) = w_2(4) - x(4) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(5) = w_3(4) - x(4) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Следующий предъявленный образ $x(5) = (1, 1, 1)$ принадлежит классу ω_2 . Соответствующие скалярные произведения равны $w_1(5)x(5) = -2$, $w_2(5)x(5) = 0$, $w_3(5)x(5) = -4$. Образ $x(5)$ классифицирован правильно. Поэтому

$$w_1(6) = w_1(5) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(6) = w_2(5) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(6) = w_3(5) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Следующий образ $x(6) = (-1, 1, 1)$ принадлежит классу ω_3 , для него получаем $w_1(6)x(6) = -2$, $w_2(6)x(6) = -4$, $w_3(6)x(6) = -0$. Этот образ также классифицирован правильно, так что коррекции не нужны, т. е.

$$w_1(7) = w_1(6) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(7) = w_2(6) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(7) = w_3(6) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Если продолжить процедуру обучения, рассматривая образы $x(7)$, $x(8)$, $x(9)$, можно убедиться, что в следующем полном цикле никакие коррекции не производятся. Поэтому искомые решающие функции имеют следующий вид:

$$d_1(x) = 0 \cdot x_1 - 2x_2 + 0 = -2x_2,$$

$$d_2(x) = 2x_1 - 0 \cdot x_2 - 2 = 2x_1 - 2,$$

$$d_3(x) = -2x_1 + 0 \cdot x_2 - 2 = -2x_1 - 2.$$

Теперь, получив объект, требующий классификации, необходимо его признаки подставить в каждую из решающих функций и в качестве результата выбрать функцию (класс), на которой будет достигнуто максимальное значение.

ЛЕКЦИЯ 7

План лекции:

1. Законы техники в инженерном творчестве.
2. Закон прогрессивной эволюции техники.
3. Закон соответствия между функцией и структурой.
4. Закон стадийного развития техники.

7.1 Законы техники в инженерном творчестве

Наивысший уровень инженерного творчества заключается в выявлении и формировании законов и закономерностей строения и развития техники и сознательном их использовании при поиске более эффективных и рациональных конструкторско-технологических решений.

Сегодня пока еще нет достаточно полных законов развития техники и даже в гипотезах полной замкнутой их системы. Однако уже существуют отдельные теоретические и методические разработки, которые могут эффективно повлиять на инженерное творчество. Такие разработки важны, прежде всего, по двум причинам:

- на основе законов развития техники могут быть разработаны наиболее эффективные методы и методологии инженерного творчества;
- привязка законов к конкретному классу технических объектов позволяет определить наиболее удачные структурные свойства, облик и характеристики подобных объектов в следующих поколениях.

7.2 Закон прогрессивной эволюции техники

Действие этого закона аналогично действию закона о естественном отборе в живой природе. Он отвечает на вопросы, почему происходит переход от одного класса технических объектов к другому, улучшенному; при каких условиях, когда и какие структурные изменения происходят при переходе от поколения к поколению. Гипотеза о законе прогрессивной эволюции техники имеет следующую формулировку.

В технических объектах с одинаковой функцией переход от поколения к поколению вызван устранением выявленного дефекта, связанного с улучшением критериев развития и происходит следующими наиболее вероятными путями иерархического исчерпания возможностей конструкции:

а) при неизменном физическом принципе действия и техническом решении улучшаются параметры технического объекта до приближения к глобальному экстремуму по значениям параметров;

б) после исчерпания возможностей цикла а) происходит переход к более рациональному техническому решению (структуре), после чего развитие опять идет по циклу а); циклы а) и б) повторяются до приближения к глобальному экстремуму по структуре для данного принципа действия;

в) после исчерпания возможностей циклов а) и б) происходит переход к более рациональному физическому принципу действия, после чего развитие

опять идет по циклам а) и б); циклы а), б), в) повторяются до приближения к глобальному экстремуму по принципу действия для множества известных физических эффектов.

При этом из всех возможных конструкций реализуется именно та, которая требует минимум интеллектуальных и производственных затрат. Переход от поколения к поколению происходит при определенном научно-техническом уровне. Это понятие относится к конкретному месту и времени, а также включает используемые технологии, источники энергии, материалы, информацию и т.д.

Социально-экономическая целесообразность создания и использования технических объектов указывает на то, что, *во-первых*, их изготовление и практическое применение экономически возможно и выгодно, *во-вторых*, не ухудшает антропологических критериев развития техники. В результате чего закон развития техники можно представить в виде спирали следующим образом.

На первом витке улучшаются параметры используемого технического решения. Когда изменения параметров перестают давать ожидаемый результат, происходит переход на следующий виток спирали, где осуществляется замена одного технического решения другим. На этом витке опять стараются «выжать» все возможное из технического решения, изменяя параметры, после чего происходит замена данного технического решения. Работа на первом и втором витках спирали продолжается до тех пор, пока в рамках используемого принципа действия удастся находить технические решения. В противном случае происходит переход на третий виток спирали, что расценивается как революционное изменение, связанное с введением нового физического принципа действия.

На законе прогрессивной эволюции основывается методология системного иерархического выбора глобально-оптимальных конструкторско-технологических решений. При этом решается иерархическая последовательность задач поискового проектирования. На первом этапе для заданной функции выбирается наиболее рациональная функциональная структура. На втором этапе для выбранной структуры определяется наиболее эффективный физический принцип действия. На третьем этапе осуществляется поиск наиболее рационального технического решения для выбранного действия. На четвертом – моделирование принятого технического решения и оптимизация его параметров.

Методология системного иерархического выбора «запрещает» останавливаться на частных улучшенных решениях, как это часто делается на практике. Она ориентирована на изучение всех возможных путей улучшения с достаточно полным информационным обеспечением. В законе прогрессивной эволюции иерархическое исчерпание не действует формально, т.е., если переход к новому техническому решению или принципу действия обеспечивает получение дополнительной эффективности, существенно превышающей дополнительные интеллектуальные и производственные затраты, то может произойти скачок к новому решению или принципу

действия без исчерпания возможностей предыдущих. Значение критериев развития технического объекта обычно изменяется в соответствии со специальной функцией. На ее основе выполняется устанавливается, насколько недоиспользованы возможности применяемого физического принципа действия. Если эти возможности имеют значительные резервы, то путем прогнозирования можно сформулировать реальное задание на улучшение интересующих показателей. Если же прогноз покажет, что возможности практически исчерпаны, то будет сделан оправданный переход к новому принципу действия.

7.3 Закон соответствия между функцией и структурой

Суть этого закона заключается в том, что в правильно спроектированном техническом объекте каждый элемент от сложных до простых компонентов имеет вполне определенную функцию по обеспечению работы объекта. И если лишить объект любого компонента или признака, то он либо перестанет работать, либо ухудшит показатели качества. В связи с этим у правильно спроектированного технического объекта нет «лишних» компонентов.

Совокупность всех соответствий между функциями и структурой в техническом объекте представляет собой функциональную структуру в виде ориентированного графа, который отражает системную целостность объекта и соответствие между его функциями и структурой. В данном ориентированном графе вершинами являются элементы объекта, а дугами могут быть функции элементов, обеспечивающие работу других элементов, или потоки данных.

Рассмотрим применение закона к отдельным классам технических систем, первый из которых – это *технологические объекты*. На рисунке 7.1 приведена структурная схема технической системы, относящейся к классу технологических объектов.

Системы этого класса состоят из четырех главных подсистем S_1, S_2, S_3, S_4 , реализующих четыре фундаментальные функции $\Phi_1, \Phi_2, \Phi_3, \Phi_4$. Где Φ_1 – технологическая функция, обеспечивающая превращение сырья A_0 в готовый продукт A_k ; Φ_2 – энергетическая функция, превращающая вещество или извне полученную энергию W_0 в конечный вид энергии W_k , необходимый для реализации Φ_1 ; Φ_3 – функция управления, осуществляющая управляющие воздействия U_1 и U_2 на подсистемы S_1 и S_2 в соответствии с заданной программой Q и полученной информацией U_1^0 и U_2^0 о количестве и качестве выработанных конечного продукта A_k и конечной энергии W_k ; Φ_4 – функция планирования, собирающая информацию Q^0 о произведенном конечном продукте A_k и определяющая качественные и количественные характеристики конечного продукта.

Вторыми рассмотрим технические системы, выполняющие обработку и преобразование информации и энергии. Они, как и предыдущие, состоят из четырех главных подсистем S_1, S_2, S_3, S_4 , реализующих четыре фундаментальные функции $\Phi_1, \Phi_2, \Phi_3, \Phi_4$. Где Φ_1 – функция получения первичной информации (энергии) превращает информацию W_b , полученную

извне, в W_0 , удобную для дальнейшей обработки и преобразования; Φ_2 – функция преобразования, превращающая информацию (энергию) W_0 в конечный вид W_k ; Φ_3 – функция управления, осуществляющая управляющие воздействия U_1 и U_2 на подсистемы S_1 и S_2 в соответствии с заданной программой Q и полученной информацией U_2^0 о количестве и полученной информации (энергии) W_k ; Φ_4 – функция планирования, собирающая информацию Q^0 о W_k и определяющая необходимые для Q качественные и количественные характеристики W_k . На рисунке 7.2 приведена структурная схема описанной технической системы.

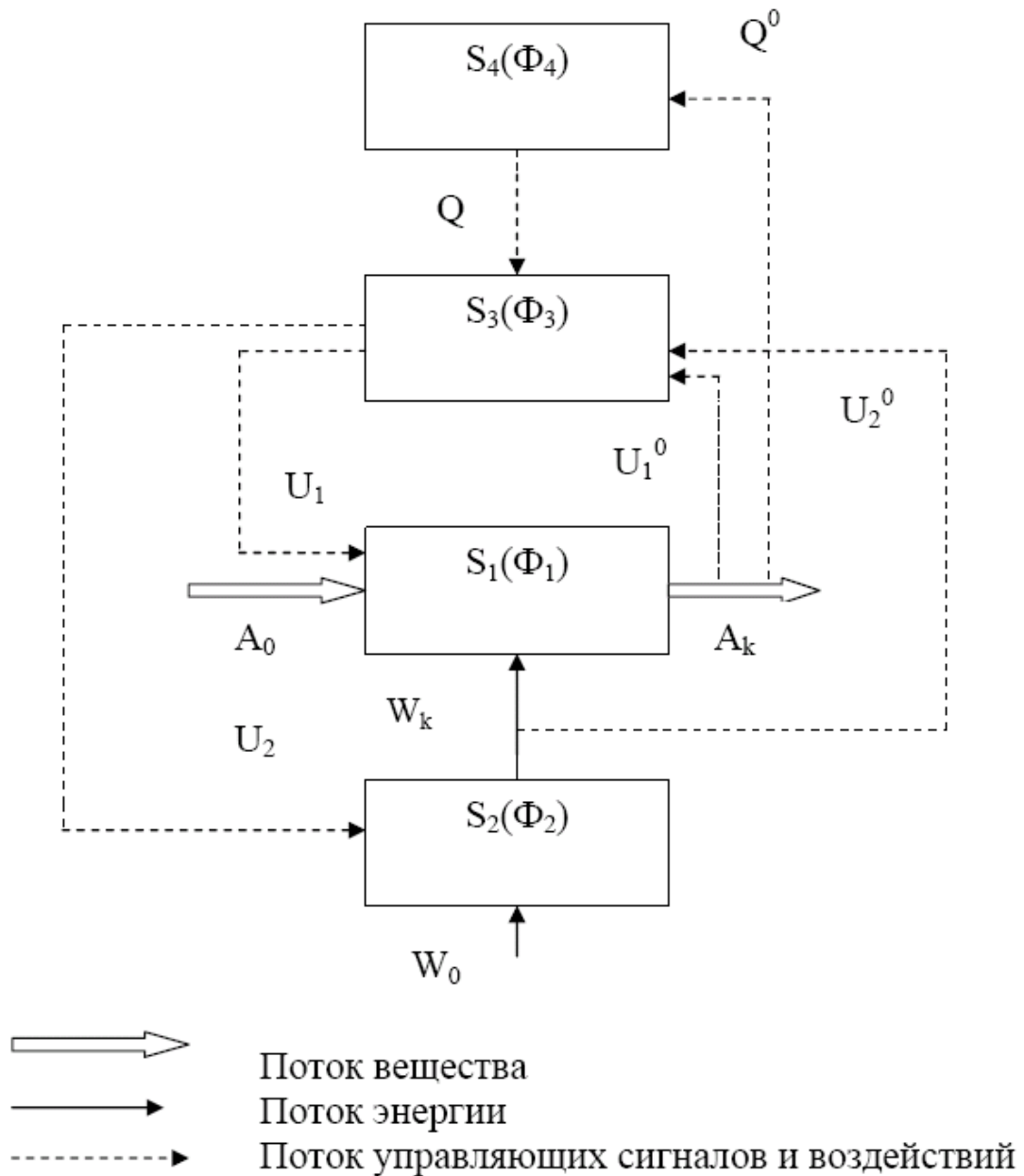


Рисунок 7.1 – Структура технологического объекта

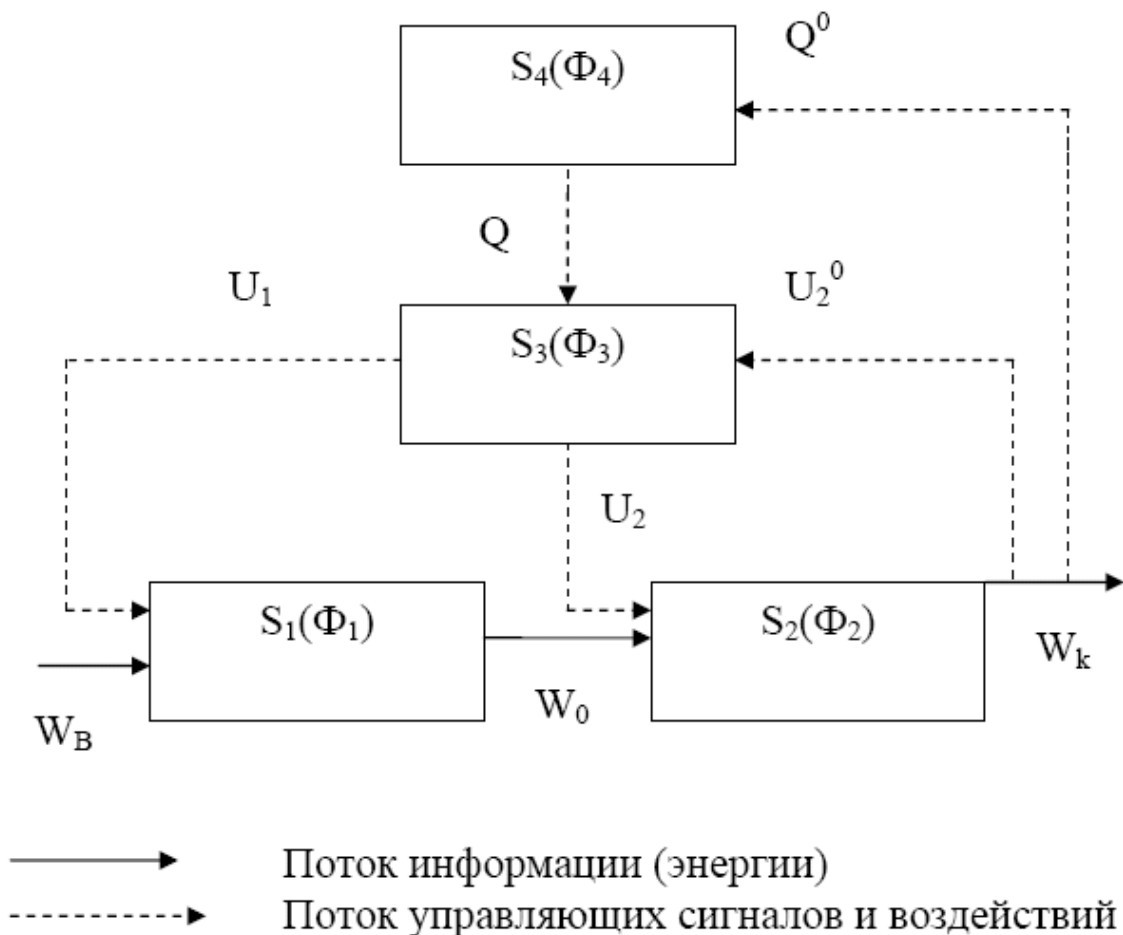


Рисунок 7.2 – Структура системы обработки информации

На практике, как правило, два технических объекта, выполняющих одинаковые функции, имеют различные структуры. Чаще всего структура выбирается так, чтобы система могла выполнять более одной функции. В этом случае в силу вступает закон о многозначном соответствии между функцией и структурой. Поэтому любая функция может быть реализована множеством структур, и наоборот, для технической системы может быть выбрана такая структура, которая предназначена для выполнения более одной функции. Это называется *закономерностью многозначного соответствия между функцией и структурой*. Она лежит в основе многих методов поискового конструирования. Наиболее сложный класс задач решается путем поиска новых, более рациональных функциональных структур. В связи с этим на основе закона о соответствии между функцией и структурой и его закономерности следует проводить анализ функциональных структур по следующему плану:

1. Оценка функциональной ценности каждого элемента с точки зрения его исключения и передачи его функции другому элементу.
2. Выделение в функциональной структуре ряда функций в целях их реализации одной подсистемой.

3. Оценка существующей и выбор более рациональной последовательности функциональных элементов.

4. Оценка целесообразности разделения функций элементов, выполняющих две и более функции.

5. Проверка полноты функциональной структуры в соответствии с закономерностью функционального строения данного класса технических объектов, а также оценка целесообразности введения новых функциональных элементов.

6. Выделение функций, выполняемых человеком, и оценка возможности и целесообразности их выполнения техническими средствами.

7. Оценка возможности использования функциональных структур технических объектов, выполняющих близкие и аналогичные функции и имеющих опережающие темпы развития по сравнению с разрабатываемым классом систем.

7.4 Закон стадийного развития техники

Этот закон отражает революционные изменения, происходящие в процессе развития как отдельных классов технических систем, так и техники в целом. Революционные изменения связаны с передачей техническим средствам функций человека. Закон стадийного развития в основном имеет отношение к задачам, связанным с пионерскими решениями. На инженерном уровне закон имеет следующую формулировку.

Технический объект имеет четыре стадии развития, связанные с последовательной реализацией с помощью технических средств четырех фундаментальных функций и последовательным исключением из технологического процесса функций, выполняемых человеком.

На *первой стадии* технический объект реализует только технологическую функцию.

На *второй стадии* к технологической функции добавляется энергетическая функция.

На *третьей стадии* к первым двум добавляется функция управления.

На *четвертой стадии* к имеющимся добавляется функция планирования.

Переход на каждую следующую стадию происходит при исчерпании природных возможностей человека в улучшении показателей выполнения соответствующих фундаментальных функций в направлении улучшения показателей качества. Следует отметить, что последовательное четырехстадийное развитие технических систем имеет место для классов систем, появившихся еще в XVIII веке. Уже тогда технические системы выполняли сразу же две функции, чему соответствовал существующий научно-технический уровень, и это следовало из требований социально-экономической целесообразности. Появляющиеся сегодня технические системы сразу же выполняют три функции. Поэтому знание закона позволяет ускорять стадийное развитие технических объектов.

ЛЕКЦИЯ 8

План лекции:

1. Функциональные критерии развития техники.
2. Технологические критерии развития техники.
3. Экономические критерии развития техники.
4. Антропологические критерии развития техники.

8.1 Критерии развития технических систем

Среди параметров и показателей, характеризующих любой технический объект (ТО), всегда имеются один или несколько таких, которые на протяжении длительного времени имеют тенденцию монотонного изменения или поддержания на определенном уровне при достижении своего предела. Эти показатели осознаются как мера совершенства и прогрессивности и оказывают сильное влияние на развитие отдельных классов технических объектов и техники в целом. Такие параметры называются *критериями развития* ТО.

Наборы критериев развития для различных классов ТО в большой степени совпадают, поэтому в целом развитие техники подчинено единому набору критериев. Этот набор включает в себя 4 группы:

- функциональные, характеризующие показатели реализации функции ТО;
- технологические, связанные с возможностью и простотой изготовления ТО;
- экономические, определяющие экономическую целесообразность реализации функции с помощью ТО;
- антропологические, связанные с вопросами человеческого фактора, т.е. с воздействием на человека ТО.

Рассмотрим три наиболее распространенных критерия, входящих в группу функциональных критериев.

Критерий *производительности* представляет собой интегральный показатель уровня развития техники, который непосредственно зависит от ряда параметров, влияющих на производительности ТО. Эти параметры представляют собой частные функциональные критерии:

- 1) скорость обработки объекта;
- 2) физические и химические параметры, влияющие на интенсивность обработки;
- 3) степень механизации труда;
- 4) степень автоматизации труда;
- 5) непрерывность процесса обработки.

Критерий механизации определяется как отношение механизированных операций к общему числу операций. Критерий автоматизации определяется как отношение автоматизированных операций к общему числу операций. В основе критерия непрерывности лежит один из главных способов повышения производительности труда.

Критерий *точности* включает в себя следующие частные критерии:

- 1) точность измерений;
- 2) точность попадания в цель;
- 3) точность обработки материала или вещества;
- 4) точность обработки потока энергии;
- 5) точность обработки информации.

Критерий *надежности* содержит следующие частные критерии:

- 1) безотказность;
- 2) долговечность;
- 3) сохраняемость;
- 4) ремонтпригодность.

Под надежностью ТО обычно подразумевают способность без отказов выполнять свою функцию с заданной вероятностью в течение определенного интервала времени. Критерий надежности возрастает с увеличением времени и вероятности безотказной работы. Критерии производительности, точности и надежности представляют собой монотонно возрастающие функции. Актуальность и вес этих критериев всегда были выше других.

Технологические критерии призваны обеспечивать всестороннюю экономию живого труда при создании ТО и подготовке их к эксплуатации. Выделяют 4 частных критерия. Критерий *трудоемкости изготовления* ТО равен отношению суммарной трудоемкости T_c проектирования, изготовления и подготовки к

эксплуатации изделия к его показателю эффективности Q , т.е. представляет собой удельную трудоемкость изготовления на единицу получаемой эффективности: $K_T = T_c/Q$. Главный показатель эффективности Q выбирают так, чтобы K_T объективно отражал прогрессивное развитие рассматриваемых ТО. Критерий трудоемкости представляет собой монотонно убывающую функцию при условии, что сопоставление различных поколений ведется по одному и тому же показателю эффективности Q . Этот критерий считается одним из самых древних.

Критерий *технологических возможностей* связан с выделением в ТО пяти типов элементов:

- 1) A_c – стандартные,
- 2) A_y – унифицированные, заимствованные у существующих ТО,
- 3) A_{k1} – оригинальные, но не сложные,
- 4) A_{k2} – оригинальные и сложные,
- 5) A_{k3} – оригинальные, вызывающие принципиальные трудности.

Критерий технологических возможностей, который должен отражать простоту и принципиальность возможности изготовления ТО, можно определить по формуле:

$$K_{m.в.} = \varepsilon \frac{K_c A_c + K_y A_y + K_{k1} A_{k1} + K_{k2} A_{k2}}{A_c + A_y + A_{k1} + A_{k2}},$$

где $\varepsilon = \begin{cases} 1, \text{если } A_{k3}=0 \\ 0, \text{если } A_{k3} > 0. \end{cases}$

K_c, K_y, K_{k1}, K_{k2} – весовые коэффициенты, причем

$$K_c = 1, K_c > K_y > K_{k1} > K_{k2}.$$

$A_c, A_y, A_{k1}, A_{k2}, A_{k3}$ – число наименований соответствующих элементов.

На практике часто используются частные случаи этого критерия: стандартизации, унификации. Однако в любой форме представления $K_{т.в.}$ находится в диапазоне значений от 0 до 1. Его нельзя отнести к монотонно возрастающей функции, так как часто для улучшения более важных критериев приходится ухудшать $K_{т.в.}$ Основная форма представления этого критерия стимулирует исключение абсолютно нетехнологичных элементов A_{k3} и минимизацию A_y, A_{k1}, A_{k2} в соответствии с их весовыми коэффициентами. $K_{т.в.}$ отражает фактор наследственности в технике. Он заставляет в наибольшей мере сохранять и использовать проверенные практикой функциональные элементы, отработанную технологию изготовления и внедрения.

Критерий *использования материалов* равен отношению массы изделия G к массе израсходованных материалов P , при этом покупные комплектующие элементы не учитываются:

$$K_{и.м.} = G/P.$$

В случае, когда в ТО используются материалы, существенно различающиеся по стоимости, при вычислении данного критерия рекомендуется применять следующие формулы:

$$G_n = \sum_{i=0}^m k_i q_i, P_n = \sum_{i=0}^m k_i p_i,$$

где i – номера материалов, q_i – масса i -ого материала, k_i – весовой коэффициент i -ого материала, p_i – масса израсходованного i -ого материала.

Критерий использования материалов представляет собой монотонно убывающую функцию в диапазоне от 0 до 1. Иногда случаются ступенчатые возрастания данного критерия, связанные с переходом на новые технологические процессы с большей производительностью или на новые более дешевые материалы.

Критерий *разбиения* ТО на элементы. Всегда существует оптимальное разбиение ТО на узлы и детали, которое значительно упрощает технологию разработки, доводки, изготовления, модернизации, являясь основой для

унификации и стандартизации. Данный критерий обеспечивает в каждом новом поколении изделий приближение к оптимальному, с точки зрения декомпозиции, представлению технической системы.

Рассмотрим группу *экономических* критериев, первый из которых критерий *расхода материалов*. Он равен отношению массы технической системы G к ее главному показателю эффективности Q :

$K_{p.m.} = G/Q$, т.е. представляет собой удельную массу материалов на единицу получаемой эффективности. В случае применения в ТО материалов со значительно различающимися стоимостями параметр G следует вычислять как для критерия использования материалов. Критерий расхода материалов обычно является монотонно убывающей функцией при условии, что сопоставление различных поколений ТО ведется по одному показателю эффективности Q .

Критерий *расхода энергии*. При разработке технической системы стараются свести к минимуму энергетические затраты. В связи с этим существует критерий расхода энергии:

$$K_{\text{э}} = \frac{W_n + E}{TQ}, \quad (8.1)$$

где W_n – полная затрата энергии за время эксплуатации ТО, E – затраты энергии при изготовлении ТО, T – время эксплуатации ТО, Q – показатель эффективности. Формулу (1) рекомендуют использовать, когда W_n и E соизмеримы. Если W_n намного больше E , то используется следующая формула:

$$K_{\text{э}} = W/Q, \quad (8.2)$$

где W – затраты энергии при эксплуатации ТО в единицу времени.

В инженерной практике широко используется еще одна модификация этого критерия, которую называют коэффициентом полезного действия:

$$K_{\text{э}} = W_0/W, \quad (8.3)$$

где W_0 – полезная работа, W – вся выполненная работа.

Критерии, представленные формулами (8.1) и (8.2) – монотонно убывающие функции, а формулой (7.3) – монотонно возрастающая функция.

Критерий *затрат на информационное обеспечение*:

$$K_{u.o.} = S/Q,$$

где S – затраты на подготовку и обработку информации, включающие стоимость или эксплуатацию вычислительной техники, разработку программного и информационного обеспечения, Q – показатель эффективности.

Критерий представляет собой монотонно убывающую функцию, однако возможны скачки, связанные с переходом на новую вычислительную технику, которая сразу не дает повышения эффективности ТО.

Критерий *габаритных размеров* ТО выражается следующей формулой:

$$K_2 = V/Q,$$

где V – основные габаритные размеры ТО, Q – показатель эффективности. Обычно этот критерий является монотонно убывающей функцией и влияет на развитие всех объектов, кроме тех, у которых уменьшение габаритов функционально ограничено.

Антропологические критерии. Группа этих критериев обеспечивает по возможности наибольшее соответствие и приспособление ТО к человеку.

Критерий *эргономичности*. Свойство системы человек-машина менять свою эффективность в зависимости от степени использования возможностей человека называют эргономичностью. Эффективность ТО при этом в первую очередь выражается через функциональные критерии развития системы (производительность, надежность, точность). Критерий эргономичности для конкретного ТО равен отношению реализуемой эффективности системы человек-машина к максимально возможной эффективности этой же системы. Он представляет монотонно возрастающую функцию, стремящуюся к 1. Критерий эргономичности можно трактовать как КПД человека в системе. Роль этого критерия возрастает с усложнением ТО. Эргономика стремится создавать наиболее удобные для человека системы.

Критерий *безопасности* ТО. Многие технические системы могут оказывать на человека вредные воздействия разных степеней тяжести. В связи с этим введен критерий безопасности, учитывая который ТО в своем развитии имеет тенденцию понижать или исключать вредные и опасные воздействия на людей.

$$K_6 = \sum_{i=1}^n \beta_i \gamma_i \frac{S_i}{S_i^n},$$

где n – число вредных и опасных факторов, β_i – весовой коэффициент i -ого

фактора при условии, что $\sum_{i=1}^n \beta_i = 1$, γ_i – весовой коэффициент, который равен: 1, если $S_i = S_i^n$, а иначе $1/\min(\beta_i)$, S_i – величина i -ого вредного

фактора, S_i^n – нормативная (предельно допустимая) величина i -ого вредного фактора.

При отсутствии нарушений всех условий данный критерий принимает значение в диапазоне от 0 до 1, при нарушениях он становится больше 1, а в случаях больших нарушений критерий безопасности может намного превышать 1. Данный критерий относится как к новым, так и к существующим ТО.

Критерий *экологичности* регулирует взаимоотношения между ТО и окружающей средой. Он выражается формулой:

$$K_{\text{эк}} = (S_n + S_k) / S_o,$$

где S_n – площадь, на которой по одному или нескольким факторам имеются недопустимые загрязнения или изменения выше нормы, но ниже критических; S_k – площадь, на которой по одному или нескольким факторам имеются критические загрязнения и изменения, при которых жизнь человека становится невозможной; S_o – вся площадь.

К факторам загрязнения относятся инородные примеси, изменения в неживой и живой природе в виде отклонений от нормы. Вредное влияние на природу необходимо свести, по возможности, к минимуму, так как природа очень «напряжена» и любой перегиб может привести к экологической катастрофе.

ЛЕКЦИИ 9-10

План лекции:

1. Эволюция информационных технологий управления.
2. Сравнительный анализ MRP и ERP-систем.
3. Функции ERP-систем.
4. Преимущества внедрения на предприятии ERP-систем.
5. Особенности внедрения MRPII/ERP-систем.
6. Основные принципы выбора ERP-системы.
7. Основные технические требования, предъявляемые к ERP-системе.
8. Рынок ERP-систем.

9.1 Эволюция информационных технологий управления

В 60-е годы прошлого столетия появились так называемые MRP-системы (MRP – material requirement planning: планирование материальных потребностей). Они "умели" решать задачу расчета параметров и планирования потребностей в материальных ресурсах для производственной программы фирмы, как правило, в сборочном производстве. Входными данными MRP-системы являются показатели плана производства готовой продукции на определенный период, спецификации производимой продукции, существующие запасы и т.д. Выходными данными MRP-системы являются графики заказов на закупку материалов и комплектующих и графики заказов на производство компонентов, необходимых для изготовления готовой продукции. Позднее были созданы системы следующего поколения – MRP-II (Manufacturing Resources Planning), в которых с помощью различных математических методов моделирования наряду с планированием материальных ресурсов, через описание в системе технологии производства, осуществляется планирование потребностей в производственных мощностях. В результате чего рассчитываются исходные данные для формирования показателей финансовых планов и бюджетов. Таким образом, в системах MRP-II планируется и учитывается материальные, мощностные и финансовые ресурсы. Следовательно, эти системы ориентированы преимущественно на потребности производства. На смену системам MRP-II пришли ERP-системы (ERP – Enterprise Resources Planning: планирование ресурсов промышленного предприятия). ERP-системы позволяют осуществлять планирование и управление бизнесом по различным направлениям деятельности как отдельного предприятия, так и многопрофильной корпорации. Их не надо смешивать с системами, выполняющими отдельные функции многоаспектного управления некоторой структурой. Внедрение MRP-II и ERP-систем, как правило, сопряжено с изменением и дополнением процессов деятельности, пересмотром и коррекцией принципов управления. Вследствие чего их распространение и внедрение происходят достаточно медленно.

9.2 Различия между MRP-II и ERP системами

Одним из направлений развития систем класса MRP-II стало расширение их функциональности для возможности управления предприятиями с различными типами производства. Кроме того, необходимо учитывать принципиальные отличия дискретного и процессного производств, которые в требуют необходимости разработки специальных подсистем управления именно таким производством. В ERP-системах поддерживается учет различных направлений деятельности потребителей ресурсов.

Планирование ресурсов предприятия неразрывно связано с понятием потока заказов, который, как правило, рассматривается с двух точек зрения: заказы независимого (рыночного) спроса и заказы зависимого спроса. Заказы рыночного спроса поступают извне, определяют формирование программы сбыта готовой продукции или ее компонент и имеют следующие основные категории:

- прогнозируемые, предполагаемые на основе анализа потребностей рынка;

- подтвержденные, существующие в портфеле заказов.

Заказы зависимого спроса порождаются внутри предприятия, определяются программой сбыта заказов независимого спроса и имеют следующие основные категории:

- заказы на производство;

- заказы на закупку;

- заказы на внутреннее перемещение;

- заказы на техобслуживание;

- заказы на транспортное средство и др.

ERP поддерживает функцию учета возможности планирования ресурсов предприятий, входящих в состав корпорации, т.е. многозвенное планирование. Использование многозвенного планирования производства предполагает формирование собственного плана для каждого звена единой организационной структуры с определением источника, потребителя, номенклатуры поставляемой продукции и приоритетов. На этапе создания многозвенного плана ERP-система обобщает потребности в плановых единицах в разных звеньях и создает производственные планы для каждого звена с учетом поставок между звеньями, так называемых внутрикорпоративных поставок. Поэтому прежде, чем утверждать план производства в соответствии с показателями плана сбыта и внутренних потребностей на конкретный период времени, ERP-система предоставляет возможность предварительного моделирования потока заказов для выявления критических состояний с учетом перечисленных (взаимопересекающихся по мощностям) потребностей. На основании этого можно сформулировать главные отличия ERP-систем от развивающихся в этом же направлении MRP-II систем:

- поддержка различных типов производств;

- поддержка планирования ресурсов по различным направлениям деятельности предприятия, а не только производства продукции;

– поддержка планирования ресурсов и управление сложным многопрофильным предприятием (корпорацией).

Кратка характеризуя ERP, можно сказать, что это система планирования ресурсов промышленного предприятия по всем основным направлениям его деятельности. Рассмотрим наиболее важные функции ERP-систем: планирование, учет, анализ, управление, менеджмент.

Планирование. Осуществлять планирование деятельности предприятия на различных уровнях означает следующее:

- формировать программу сбыта;
- выполнять производственное планирование;
- формировать основной производственный план-график;
- формировать планы на закупку;
- проводить финансовое планирование и составление бюджета.

Уточненная и утвержденная программа сбыта является основой плана производства. Интеграция данных этих планов существенно облегчает процесс производственного планирования и обеспечивает их неразрывную связь. На основе детализированного оперативного производственного плана осуществляется планирование и управление заказами на закупку и производство. При наличии необходимых исходных данных система позволяет провести предварительную оценку выполнимости сформированных планов на различных уровнях планирования для внесения необходимых коррекций или принятия решения о привлечении дополнительных ресурсов.

Учет. Планы, получившие свое подтверждение, приобретают статус актуальных планов, после чего начинается их реализация. Ранее смоделированный поток зависимых заказов превращается в реальный, порождающий потребности в материалах, трудовых ресурсах, мощностях и деньгах. Удовлетворение названных потребностей порождает учетные действия, обеспечивающие оперативную регистрацию прямых затрат, относящихся к производимой продукции, и косвенных затрат, распределяемых по центрам финансовой ответственности. Все операции по регистрации прямых затрат вводятся, как правило, в натуральном выражении нормативного потребления. Для отражения соответствующего финансового результата MRP-II, ERP-системы предлагают мощные средства настройки финансовой интеграции, позволяющие обеспечить автоматический перевод потребленных ресурсов в их финансовый эквивалент.

Анализ. За счет оперативного отражения результатов деятельности управленческий персонал получает возможность в режиме реального времени осуществлять сравнительную характеристику планов и результатов.

Управление. Наличие оперативной обратной связи, предоставляющей информацию о состоянии объекта управления, как известно, является основой любой системы управления. MRP-II и ERP-системы создавались для обеспечения такого рода связи и предоставления информации о состоянии проектов, производства, запасов, наличии и движении денежных средств, что в результате позволяет принимать обоснованные управленческие решения.

Менеджмент. Сегодня MRP-II и ERP-системы начинают приобретать популярность у руководителей предприятий в СНГ. Однако рекламируемые потенциальные возможности этих систем не означают, что какой-либо конкретный продукт оптимален для каждой компании. Серьезные знания особенностей бизнеса своего предприятия и стандартов информационных технологий управления, реализующих данные особенности, должны являться обязательными при принятии решения о приобретении того или иного продукта.

9.3 Что может дать предприятию внедрение ERP или MRPII-системы

Системы класса MRPII/ERP представляют собой интегрированные информационные системы управления, а это означает следующее:

- системы не связаны с производственным процессом непосредственно, они не являются автоматизированными системами управления технологическими процессами, но имеют дело с моделью технологического процесса;
- их работа состоит в улучшении деятельности предприятия, оптимизации материальных и финансовых потоков на основе вводимой на рабочих местах информации;
- в одной системе охватывается планирование и управление всей деятельностью производственного предприятия, начиная от закупки сырья и заканчивая отгрузкой товара потребителю;
- информация вводится в систему только один раз в том подразделении, где она возникает, хранится в одном месте, и многократно используется всеми заинтересованными подразделениями.

Как следствие, MRPII/ERP-системы позволяют достичь согласованности работы различных подразделений, снижая при этом административные издержки и устраняя проблему интеграции данных для разных приложений. Эти системы являются инструментом повышения эффективности управления, принятия правильных стратегических и тактических решений на основе своевременной и достоверной информации, выдаваемой компьютером. Использование MRPII/ERP-систем позволяет достичь конкурентных преимуществ за счет оптимизации бизнес-процессов предприятия и снижения издержек. Эти системы создавались именно для управления себестоимостью продукции и достижения за счет этого конкурентных выгод. Поэтому данные программные продукты реализуют методы планирования и управления, позволяющие:

- регулировать количество запасов, устраняя их дефицит и залеживание, и тем самым значительно снижая складские издержки;
- сократить незавершенное производство, поскольку производство планируется только на основе спроса на конечную продукцию, при этом производственные работы инициируются исходя из срока, к которому должен быть исполнен клиентский заказ;

- оценивать выполнимость поступивших заказов с точки зрения имеющихся на предприятии мощностей;
- сократить расходы и время, затрачиваемые на изготовление продукции, за счет оптимизации бизнес-процессов;
- отслеживать фактическую производительность каждой производственной единицы и, сравнивая ее с плановой производительностью, оперативно вносить корректировки в производственные планы;
- в результате уменьшения цикла производства и цикла выполнения заказа более гибко реагировать на спрос;
- улучшить обслуживание клиентов и заказчиков за счет своевременного исполнения поставок.

Реализованные в MRPII/ERP-системах возможности гибкого управления себестоимостью продукции позволяют получать более высокую прибыль. Кроме этого, снижение себестоимости означает возможность варьировать рыночную цену продукции (в сторону понижения), что является мощным преимуществом в конкурентной борьбе. Практика показала, что внедрение MRPII/ERP-системы конкурентом – это сигнал для принятия симметричного решения, поскольку автоматизированные системы управления являются мощным инструментом бизнеса и средством выживания в рыночных условиях.

Внедренная MRPII/ERP-система может помочь компании привлечь инвестиции, так как MRPII/ERP-системы делают бизнес компании более прозрачным, что повышает доверие к нему со стороны инвесторов. Сегодня уже существуют российские разработки MRPII-систем ("Галактика", "Парус", "Босс"), однако часто выбор отечественных предприятий падает на аналогичные названным западные системы. Мотивом такого решения, как правило, является желание привлечь западные инвестиции: обычно иностранные акционеры, инвесторы, консультанты требуют, чтобы на предприятии была установлена знакомая им система.

В последнее время отмечено формированием новой экономики, основным инфраструктурным элементом которой является сеть Интернет. Выводя часть своего бизнеса в Интернет, компании преследуют одновременно несколько целей: от сокращения издержек до улучшения обслуживания клиентов и организации нового онлайн-канала сбыта. Существует много схем электронной коммерции: создание Web-витрины или Интернет-магазина, организация корпоративного портала, участие в работе виртуальной торговой площадки. Однако, если Интернет-решения компании плохо интегрированы с ее бизнес-процессами, эффективность таких решений становится крайне низкой.

Например, компания создает Web-витрину с интерактивной формой для принятия заказов от онлайн-клиентов. Если при этом компания не может оперативно сообщить клиентам, в какой срок будет удовлетворен их заказ, то, скорее всего, она этих клиентов потеряет. И в этом случае Web-витрина компании станет просто дорогостоящей игрушкой.

В качестве основы построения Интернет-решений практически всегда рассматриваются MRPII/ERP-системы, обеспечивающие ресурсное планирование и интегрированное управление всеми бизнес-процессами компании. Следовательно, MRPII/ERP-системы – остов, к которому монтируются Интернет-решения.

9.5 Особенности внедрения MRPII/ERP-систем

Сформулировав основные преимущества MRPII/ERP-систем, следует упомянуть и о проблемах, возникающих в связи с внедрением таких систем. Говоря о внедрении информационных управленческих систем, следует различать:

- производителей систем и их официальных дистрибуторов, поставляющих программное обеспечение;
- консалтинговые компании, которые непосредственно занимаются внедрением систем, консультируют по принципам работы с ними и занимаются техническим сопровождением.

Существует два основных момента, влияющих на решение компаний о внедрении MRPII/ERP-систем:

- высокая стоимость приобретения и установки ПО;
- длительность и трудоемкость внедрения.

В стоимость системы входит несколько составляющих.

1. Цена одной лицензии, т.е. цена одного рабочего места. Для ведущих ERP-систем (SAP R3, Baan, Oracle Applications) она колеблется от 2 до 8 тысяч долларов, для MRPII/ERP-систем среднего класса – от 1,5 до 5 тысяч долларов.

2. Цена консалтинга, внедрения и сопровождения, которая обычно находится в промежутке от 100% до 500% от стоимости системы.

3. Цена обучения пользователей, сильно зависящая от маркетинговой политики фирмы-производителя системы или консалтинговой компании. Например, у SAP R3 стоимость обучения одного человека в неделю составляет 1500 долларов, у Baan – от 1000 долларов.

Некоторые компании снижают стоимость покупки системы за счет уменьшения количества покупаемых лицензий, считая, что не обязательно покупать лицензии для всех предполагаемых пользователей системы, то есть в соотношении 1:1.

Внедрение MRPII/ERP-систем, как правило, требует серьезного пересмотра внутренней логики работы компании и бизнес-процессов. Это зачастую касается отечественных предприятий, которые не имеют большого опыта работы в рыночных условиях, где главными становятся не показатели выпуска, а существующий портфель заказов и прогнозная оценка спроса на продукцию. Именно поэтому внедрению систем обычно предшествует этап системонезависимого обследования предприятия консалтинговой компанией. Происходит оценка предприятия с точки зрения того, можно ли вообще

внедрять на нем какую-либо систему, или сначала необходимо откорректировать бизнес-процессы. И только когда консультанты убеждаются, что предприятие готово к проекту, происходит выработка рекомендаций по внедрению и составление плана внедрения. Если предпроектное обследование предприятия не проводится, то фирма рискует потерять огромные деньги в случае неуспешной или затянувшейся установки системы.

Внедрение подобных систем всегда бывает поэтапным и трудоемким, поскольку требует настройки множества неочевидных параметров и обучения пользователей работе с системой. Для осуществления проекта требуются совместные усилия сотрудников предприятия и консультантов. Обычно процесс внедрения информационных систем происходит в фоновом режиме. Некоторое время на предприятии сосуществуют неавтоматизированное планирование и разработка планов с помощью устанавливаемой MRPII/ERP-системы. И только когда процедура планирования отработана в фоновом режиме, осуществляется запуск MRPII/ERP-системы для промышленной эксплуатации. Внедрение управленческих систем занимает не меньше года. Успех проекта напрямую зависит от готовности предприятия к установке MRPII/ERP-системы. Компания должна ощущать реальную необходимость в интегрированной управленческой системе. В случае успеха внедренная система управления прослужит долгие годы, в случае же неудачи под вопросом может оказаться само существование предприятия.

Одним из перспективных направлений развития новой экономики является предоставление услуг по аренде ASP приложений – Applications Service Providing. Провайдер приложений устанавливает программы на своих серверах и обеспечивает доступ к ним клиента. Клиенту не нужно устанавливать программное обеспечение на свой компьютер, обновлять его, делать резервное копирование, так как все это делает провайдер. Клиент платит провайдеру арендную плату за каждое обращение к системе. В рамках ASP возможно и использование ERP-систем. Обычно ERP-системы строятся на базе трехуровневой архитектуры клиент-сервер, когда между сервером базы данных и клиентскими терминалами появляется сервер приложений, отвечающий за осуществление всей бизнес-логики системы. Сервер приложений и сервер базы данных могут быть установлены на оборудовании провайдера (ASP), тогда пользователи ERP-системы инсталлируют на свои компьютеры лишь клиентские приложения системы и работают с удаленным сервером приложений через Интернет. На рисунке 9.1 приведена схема взаимодействия клиентов с ASP. При этом провайдер ERP-системы может специализироваться на нескольких конкретных промышленных отраслях, и предоставляемая им система будет настроена на бизнес-модели этих же промышленных отраслей. А предприятие получает доступ к системе, смоделированной с учетом специфики конкретной индустрии, поэтому полностью пропускает долгий и не всегда успешный этап внедрения ERP-системы. Преимущества такого подхода очевидны: *возможность доступа к приложениям высокого класса, надежность функционирования приложений.*

Безусловно, хостинг приложений связан с целым рядом проблем, главная из которых – информационная безопасность. В настоящее время они решаются. Более того, уже ведутся пробные проекты. По данным консалтинговой компании IDC, емкость рынка аренды ERP-систем составляет 4,5 млрд. долларов.

Наиболее полезной технология ASP может оказаться для малых и средних компаний, которые не могут надежно прогнозировать будущие обороты и позволить себе купить первоклассную ERP-систему, но и не желают по мере своего развития постоянно менять одну за другой дешевые системы с меньшими возможностями.



Рисунок 9.1 – Схема взаимодействия клиентов с ASP

9.6 Основные принципы выбора ERP-системы

При выборе ERP-системы необходимо обратить особое внимание на следующие основные моменты:

1. Имидж фирмы-разработчика, время ее работы на рынке, репутация самой системы и общее количество успешных внедрений. Однако солидность фирмы не является главным фактором выбора, так как большое число внедрений может быть также заслугой маркетинга, а не действительным качеством системы.

2. Число успешных внедрений в условиях отечественного рынка, в первую очередь, имеются в виду комплексные внедрения. Важно также знать, есть ли внедрения на родственных отраслевых предприятиях и потребовалась ли помощь внешних консультантов. Необходимо также посмотреть, как реально работает система хотя бы на одном-двух объектах, при этом помня о том, что любая ERP-система настраивается под потребности конкретного предприятия, а их полностью одинаковых даже в рамках одной отрасли не существует.

3. Гибкость и открытость являются одним из важнейших принципов выбора ERP-системы. Из практики срок полнофункционального внедрения ERP-системы обычно длится не менее 3 лет, а полноценно работать она должна не менее 10 лет. За это время предприятие значительно меняется: его продукция, структура штатов, организация управления, бизнес-процессы, роли и полномочия должностных. ПО управления предприятием должно меняться вместе с производством, легко настраиваться и интегрироваться с другими программными и инструментальными средствами, установленными на предприятии. Важным моментом при этом является то, что все необходимые доработки системы должна делать фирма-разработчик, юридически отвечающая перед предприятием за качество своей работы.

4. При анализе западной системы необходимо внимательно проанализировать ее терминологию и качество русификации. Документация и система помощи должны быть полными, ясными и понятными, а терминология – привычной.

5. В системы должны быть учтены особенности отечественной экономики, касающиеся юридической, бухгалтерской, налоговой деятельности.

6. Опыт работы и квалификация компании, занимающейся локализацией и внедрением западной системы в условиях отечественной экономики.

7. Географическая близость фирмы-разработчика или компании, выполняющей внедрение системы.

8. Приемлемость цены системы, в которую входят весь цикл установки ERP-системы и приобретение самого ПО.

9. Возможность помодульного приобретения системы и только на необходимое число рабочих мест. Покупка полного комплекта модулей системы сразу может оказаться не лучшим вариантом, так как все модули будут внедрены только через несколько лет, а за это время некоторые из них могут уже устареть.

9.7 Основные технические требования к ERP-системе

Выбираемая ERP-система должна соответствовать следующим общим техническим требованиям или хотя бы большинству из них:

1. Возможность интеграции с большим числом программных продуктов с минимальным уровнем интеграции – на уровне открытых кодов командной строки или поддержкой стандарта OLE Automation.

2. Обеспечение безопасности с помощью различных методов контроля и разграничения доступа к информационным ресурсам. Наличие в составе ERP-системы сертифицированных программно-аппаратных средств защиты информации, позволяющих шифровать данные, поддерживающих электронную цифровую подпись и аутентифицирующих на ее основе пользователей. Эффективность программных средств защиты может быть также существенно повышена за счет применения аппаратных и биометрических средств: аппаратных ключей, смарт-карт, устройств

распознавания отпечатков пальцев, сетчатки глаза, голоса, лица, оцифрованной подписи и др.

3. Масштабируемость.

4. Модульный принцип построения системы из оперативно-независимых функциональных блоков с расширением за счет открытых стандартов (API, COM и др.).

5. Применение 3-звенной архитектуры: сервер базы данных, сервер приложений, клиент.

6. Возможность миграции системы с платформы на платформу. Обязательно должны быть версии для ОС MS Windows NT, Novell NetWare и UNIX.

7. В набор СУБД, поддерживаемых выбираемой ERP-системой, обязательно должно входить распространенное на отечественном рынке ПО, например, Oracle, Sybase, MS SQL Server, Informix и др.

8. Поддержка технологий распределенной обработки информации и интернет технологи, что позволяет использовать стандартные хранилища данных из локальных, корпоративных и глобальных сетей, не требуя существенных затрат на дополнительное администрирование и поддержание целостности, надежности и безопасности хранения данных.

9. Поддержка технологий многоуровневого электронного архивирования информации на различных носителях.

10. Наличие аналитических возможностей и встроенных инструментальных средств, позволяющих самостоятельно наращивать функциональность установленной ERP-системы.

9.8 Рынок ERP-систем

Объединяя в себе лучшие мировые практики в области управления бизнесом, ERP-системы де-факто становятся основой для постановки бизнес-процессов, организации операций, интеграции информационных потоков и построения внешних взаимодействий, в целом способствуя повышению прозрачности и управляемости бизнеса. Призванные оптимизировать все многообразие внутренних и внешних процессов предприятия, ERP-системы позволяют снизить операционные, управленческие и коммерческие затраты, сократить цикл реализации, увеличить оборачиваемость материальных запасов, улучшить утилизацию основных фондов и т.д. Глобальная интеграция всех информационных потоков в рамках единой системы обеспечивает оптимальное использование информации, напрямую влияя на оперативность принятия управленческих решений и быстроту реакции на рыночные изменения.

Поскольку внедрение современной ERP-системы на предприятии – процесс достаточно длительный и затратный, связанный со значительными рисками, существуют компании-консультанты, предлагающие услуги по внедрению ERP-систем лучших производителей. Одна из таких компаний – АйТи. Она следует методологиям внедрения, разработанным

производителями систем и объединяющим в себе лучший опыт внедрения, а также проводит весь комплекс работ, связанных с оптимальным выбором ERP-системы, подготовкой проектной команды к ее внедрению и проектным управлением. Компания-консультант занимается следующим:

- анализирует требования бизнеса, оказывающие существенное влияние на функциональность и развитие информационной системы;
- анализирует существующие на отечественном рынке бизнес-приложения и выбирает системы наиболее адекватные требованиям бизнеса;
- оценивает объем инвестиций с учетом создания или развития необходимой технической архитектуры;
- определяет план работ, связанных с внедрением выбранного решения, периодический аудит выполняемых работ при условии реализации проектов третьими фирмами или силами самого предприятия.

Компания АйТи предлагает услуги по внедрению ERP-систем лучших производителей, каждая из которых является признанным лидером рынка. Из предлагаемого пакета ERP-решений компании с различной отраслевой принадлежностью, разными объемом и структурой бизнеса могут выбрать наиболее соответствующее стратегическим задачам предприятия решение.

mySAP Business Suite, предоставляя мощные инструменты поддержки принятия решений и интегрируя все внутренние и внешние бизнес-процессы и поддерживающие процессы, позволяет значительным образом повысить прозрачность и эффективность управления ресурсами предприятия и выстраивать наиболее адекватную рынку бизнес-стратегию.

Microsoft Business Solutions-Axapta, охватывая абсолютно все аспекты ведения бизнеса, позволяет внедрить современные западные управленческие технологии, оптимизировать ключевые бизнес-процессы и в целом повысить эффективность управления предприятием, обеспечивая при этом оптимальную в этом классе систем стоимость владения.

Oracle E-Business Suite является практически единственным решением для управления предприятием, полностью реализованным в интернет-архитектуре. Благодаря этому Oracle E-Business Suite, обладая всеми преимуществами признанных систем класса ERP, позволяет не только повысить эффективность действующих внутренних и внешних бизнес-процессов предприятия, но и создать основу для решения задач, которые будут актуальны в будущем.

Заказные решения по автоматизации финансово-хозяйственной деятельности предприятия на базе программных платформ *БОСС-Компания* и *БОСС-Корпорация* имеют модульную структуру и большие возможности по реализации индивидуальных требований конкретного предприятия, позволяя создавать системы управления деятельностью, функциональность которых в точности соответствует потребностям бизнеса.

ЛЕКЦИИ 11-12

План лекции:

1. Концепции маркетинга ERP и CRM систем.
2. CRM – управление отношениями с клиентами.
3. Классификация CRM-систем.
4. Технологии CRM-систем.
5. CRM-системы для электронной коммерции.
6. Обзор программных продуктов компании Hyperion.
7. Обзор программных продуктов корпорации Epicor Software.
8. Обзор программных продуктов компании Tranzline.
9. Обзор программных продуктов компании S1 Corporation.
10. Набор приложений компании BroadVision.

11.1 Концепции маркетинга ERP и CRM систем

В теории маркетинга принято выделять четыре основные концепции рыночной ориентации компании. Первые две из них – это производственно-ориентированная и ориентированная на продукт концепции, утверждающие, что компания будет иметь успех, если ее товар обладает наивысшим качеством и наименьшей ценой, т.е. эти концепции направлены на улучшение производственного процесса. Две другие концепции сводятся к тому, что фирма должна вести агрессивную политику продаж, постоянно изучать нужды и потребности целевых рынков и удовлетворять их на более высоком уровне, чем конкуренты. В соответствии с этим в последнее время появился такой подход работы фирмы с клиентами, как кастомизация. Фирма не просто изучает потребности своего целевого рынка, а пытается определить нужды каждого клиента, его индивидуальные предпочтения и предлагает ему уникальный продукт. Такой подход наиболее распространен в сфере услуг, хотя его применение можно найти и в других областях бизнеса.

Можно с уверенностью сказать, что в современном мире ориентация фирмы на одну из четырех концепций не приведет к успеху, хотя раньше компания могла стать лидером на рынке, используя только производственно-ориентированную теорию. Сейчас в идеальном случае компания в своей деятельности должна учитывать все концепции маркетинга, перечисленные выше. Современные технологии позволяют не только существенно уменьшить затраты по ведению данной стратегии, но и повысить эффективность применения каждой концепции маркетинга в деятельности фирмы. В последнее время для оптимизации и автоматизации внутренней деятельности фирмы применяются ERP-системы. При разработке и внедрении ERP-систем в компании клиент рассматривается как “элемент внешнего мира” и определяющего влияния не оказывает. ERP-системы направлены на достижение конкурентных преимуществ за счет оптимизации внутренних бизнес-процессов. В этом отношении противоположными им являются CRM (Customer Relationships Management) системы, в центре внимания которых находится клиент компании. CRM-системы позволяют “интегрировать”

клиента в сферу организации. Фирма получает максимально возможную информацию о своих клиентах и их потребностях и, исходя из этих данных, строит свою организационную стратегию, которая касается всех аспектов ее деятельности: производства, рекламы, продаж, дизайна, обслуживания и пр.

В настоящее время мировой рынок ERP-систем так или иначе стабилизировался, а рынок систем CRM еще только развивается. Уже сейчас многие компании, которые внедрили данные системы, получают от них большую отдачу. Дополнительной эффективности организация может добиться и от их интеграции. Иногда уже сами разработчики предлагают ERP-системы с элементами CRM- систем.

11.2 CRM - управление отношениями с клиентами

Рассмотрим причины возникновения CRM-систем.

Совершенная конкуренция. Современные технологии привели к тому, что покупатель получает доступ к любой части рынка при малых транзакционных издержках, асимметричность информации стала почти равна нулю. Поэтому основной задачей для компании является удержание имеющихся клиентов.

Мультиканальность взаимоотношений. Контакт между клиентом и фирмой может осуществляться разными способами: телефон, факс, web-сайт, почта, личный визит. И клиент ожидает, что вся получаемая по этим каналам информация при следующем взаимоотношении будет рассматриваться компанией во всей совокупности.

Изменение рыночной ориентации компаний. Переход ряда компаний к новым концепциям маркетинга. Многими компаниями пределы качества и минимизации издержек уже достигнуты, в том числе от использования ERP-систем, и клиенты больше обращают внимание на моменты, сопровождающие покупку и обслуживание.

Кроме того, необходимо учитывать, что в становлении CRM-систем важную роль сыграло развитие информационных и коммуникационных технологий, без которых не могли бы существовать приложения, лежащие в основе CRM-систем, и связи между этими приложениями.

Функции CRM-приложения позволяют компании отслеживать историю развития взаимоотношений с заказчиками, координировать многосторонние связи с постоянными клиентами и централизованно управлять продажами и маркетингом, ориентированным на клиента. CRM-системы базируются на довольно давно известных приложениях, которые частично позволяли улучшить отношения с покупателями. Это такие системы, как SFA (Sales Force Automation – система автоматизации работы торговых агентов), SMS (Sales & Marketing System – система информации о продажах и маркетинге), CSS (Customer Support System – система обслуживания клиентов). CRM-системы содержат возможности этих приложений, но предлагают и новые функции. Внедрение CRM-системы сказывается на работе почти всех подразделений фирмы, а не только отдела продаж. Именно через эту систему организована обратная связь клиента компании со всей организацией в целом.

CRM – это система, входными элементами которой, в первую очередь, являются все данные, связанные с клиентом компании, а выходными – информация, которая влияет на поведение компании в целом или на поведение ее отдельных элементов. Другими словами, CRM-система – это набор приложений, которые позволяют, во-первых, собирать информацию о клиенте, во-вторых, ее хранить и обрабатывать, в третьих, делать определенные выводы на базе этой информации, экспортировать ее в другие приложения или просто при необходимости предоставлять эту информацию в удобном виде.

Сбор информации. Система позволяет сотруднику организации удобным способом вводить информацию о клиенте в базу данных, либо же самому клиенту вводить эту информацию (например, при регистрации или покупке товара в интернет-магазине). В CRM-систему вводятся все доступные сведения о клиенте: относящиеся к взаимодействию клиент-компания и личная информация клиента. Все эти данные обновляются при каждом взаимодействии компании с клиентом.

Хранение и обработка. Система позволяет сохранять и ранжировать полученную информацию в соответствии с заданными критериями. Причем все сведения хранятся в стандартной для корпорации форме. Кроме того, CRM-система в соответствии с заданными параметрами может анализировать полученную информацию с целью ее последующего экспорта.

Экспорт информации. Предоставление информации CRM-системой является ее главной функцией. Сведения, хранящиеся в системе, могут быть затребованы различными подразделениями и в разном виде. Например, система CRM на основе экстраполяции исторических данных может определить, какой товар предпочтительнее предложить определенному клиенту. Если клиент является постоянным покупателем, система напомнит, что ему полагается скидка. Наконец, сотруднику компании может понадобиться информация об исторических контактах клиента с фирмой, и система предоставит эти сведения в наглядном виде. Предусмотрена возможность выводить информацию как по отдельному клиенту, так и по целевой группе.

Использование данных. Данными, которые генерирует CRM-система, могут воспользоваться не только сотрудники компании, но и непосредственно сам клиент без помощи сотрудников организации. Иногда CRM-система позволяет клиенту, который впервые обращается к компании, подобрать необходимый ему продукт в режиме реального времени. Клиент вводит данные о продукте, который, по его мнению, может предложить компания, а система, обработав эти данные, выдает ему список товаров, соответствующих введенным параметрам. Для этого в систему автоматически импортируются сведения из той части ERP-системы, которая отвечает за учет произведенной продукции.

11.3 Классификация CRM-систем

В соответствии с целью использования можно выделить три вида стандартных CRM-систем.

1. *Оперативное использование.* Система используется сотрудником компании для оперативного доступа к информации по конкретному клиенту в ходе непосредственного взаимоотношения с клиентом по вопросам процессов продажи и обслуживания. В этом случае основным компонентом системы является приложение, которое в наглядном виде предоставляет сотруднику накопленную информацию по отдельному клиенту. В первую очередь, от системы требуется хорошая интеграция между всеми подсистемами и возможность пополнять базу данных в процессе любого взаимодействия с клиентом. Данный тип CRM-систем является наиболее распространенным в традиционном бизнесе.

2. *Аналитическое использование.* Система применяется для анализа различных данных, относящихся как к самому клиенту/клиентам, так и к деятельности фирмы. Осуществляется поиск статистических закономерностей в этих данных для выработки наиболее эффективной стратегии маркетинга, продаж, обслуживания клиентов и др. Требуется хорошая интеграция подсистем, большой объем наработанных статистических данных, эффективный аналитический инструментарий, интеграция с другими системами, автоматизирующими деятельность предприятия. Данные, генерируемые такими системами, могут быть затребованы отделом маркетинга, представлены клиенту без посредничества сотрудников компании. Этот тип CRM-систем чаще применяется в электронной коммерции, нежели в традиционном бизнесе.

3. *Коллаборационное использование.* CRM-системы предоставляют клиентам возможность достаточно большого влияния на деятельность фирмы в целом, в том числе на процессы разработки дизайна, производства, доставки и обслуживания продукта. Для этого требуются технологии, позволяющие с минимальными затратами подключить клиента к сотрудничеству в рамках внутренних процессов компании. Клиент нередко использует сеть Интернет для доступа к таким системам, и они наиболее распространены в сфере электронной коммерции.

11.4 Технологии CRM-систем

Основная часть современных систем CRM базируется на принципе “клиент-сервер”, т.е. все данные CRM-системы хранятся и обрабатываются в одной централизованной БД, а клиенты имеют к ним доступ через удаленные терминалы. Клиентами таких CRM-систем могут быть как внешние, так и внутренние по отношению к компании пользователи. Взаимодействие между клиентом и сервером может осуществляться посредством сети Интернет. Часть “сервер” обычно состоит из двух приложений – СУБД для хранения, обработки данных и системы OLAP-сервер для анализа этих данных в режиме онлайн. Чаще всего компании используют в качестве СУБД продукты от известных производителей, такие как Oracle, Interbase, Microsoft SQL Server.

Индивидуальным решением каждого разработчика CRM-систем обычно является построение OLAP-приложений.

Большинство CRM-систем построено на Microsoft SQL Server 7.0. Основными преимуществами этого продукта при работе с данными являются симметричная многопроцессорная обработка, встроенные механизмы тиражирования, тесная интеграция с Интернет и электронной почтой. SQL Server ведет статистику, содержит модуль Web Assistant, который помогает публиковать данные в сети. Утилита SQL Trace позволяет в графическом виде фиксировать активность клиентов. Большим достоинством Microsoft SQL Server 7.0 является его тесная интеграция с продуктами семейства Microsoft BackOffice, что делает данную СУБД идеальным инструментом для построения и администрирования корпоративных баз данных. Кроме того, SQL Server предусматривает связь с внешними источниками хранения больших объемов данных (например, с мэйнфреймами при помощи Microsoft SNA Server).

При разработке CRM-систем конкурентом Microsoft SQL Server 7.0 может быть база данных Oracle8i компании Oracle. Система Oracle8i базируется на СУБД Oracle8 и предназначена для вычислений и обработки данных в Интернет. Несомненным преимуществом данной СУБД является то, что она может работать под управлением почти любой современной ОС, а не только под Windows, как Microsoft SQL Server. Oracle8i поддерживает мультимедийные данные и онлайн-обработку транзакций. Это единственная система, в которой есть интегрированные инструментальные средства для управления информацией Web, позволяющие пользователям работать с данными при высокой степени безопасности и надежности. Концепция Oracle8i – применение языка Java на всех уровнях обработки данных, включая VM Java на сервере Oracle8i.

Реализация серверной части CRM-систем в некоторых случаях может быть построена по принципу ASP (application service provider): компания-пользователь арендует все ПО CRM-системы, которое находится на сервере поставщика. Все клиенты, в том числе и администратор, этих систем имеют доступ к приложению при помощи сети Интернет. В общем случае ASP обеспечивает доступ к бизнес-приложению по той или иной схеме на условиях ежемесячной оплаты, взимаемой с компании-покупателя этого приложения. В обязанности фирмы, предлагающей услугу ASP, входит предоставление заказчику программного обеспечения, аппаратных средств и сетевой инфраструктуры, необходимых для работы приложений. ASP обеспечивает также обработку результатов, системную интеграцию и обучение.

11.5 CRM-системы для электронной коммерции

Эти системы обладают всеми функциями обычных CRM-систем, и кроме того, они полностью интегрируются с web-сайтом компании. Вся информация с сайта попадает в систему CRM. Сама система может

определять построение сайта и эффективно обслуживать каждого клиента в процессе интернет-покупки или оказания Интернет-услуги. CRM-системы регистрируют и анализируют все контакты между покупателем и продавцом, осуществленные через Web-сайт компании или по электронной почте. Эти системы могут быть направлены на развитие рекламного Интернет-маркетинга. В его функции входит:

- оценка эффективности web-страниц с точки зрения привлечения внимания посетителей;
- определение того, как объем и интенсивность электронной рекламы отражаются на объеме продаж;
- определение средств связи, дающих больший эффект при проведении рекламных кампаний;
- определение рекламных кампаний, обеспечивающих наибольший коэффициент окупаемости инвестиций.

Подобные CRM-системы активно работают с web-витриной. В этом случае CRM-система является аналитическим приложением, позволяющим прогнозировать и направлять реакцию посетителей web-сайтов. Таким образом, CRM система должна быть тесно интегрирована с web-сайтом магазина. Зачастую компании, предлагающие готовые решения в виде Интернет-сайтов, интегрируют в них модули, имеющие функции, присущие системам CRM.

Наконец, CRM системы используются компаниями, предоставляющими финансовые услуги в Интернет: банкинг, трейдинг, страхование. На современном этапе конкуренции наличие хорошо отлаженной CRM-системы является для компании необходимостью, без которой организация не может эффективно обслуживать своих клиентов. Поэтому в системы, используемые финансовыми компаниями для предоставления услуг через Интернет, изначально интегрированы системы типа CRM, которые рассматриваются как отдельные модули, отвечающие за эффективную работу с клиентом в рамках общей системы. Ниже приведены функции, выделенные самими разработчиками CRM систем для финансовых организаций.

1. CRM-система должна способствовать организации простой и интуитивно понятной для клиента функции самообслуживания. Клиент должен предоставлять минимум информации и затрачивать минимальное время на проведение услуги.

2. Система должна предоставлять возможность подобрать подходящий для клиента продукт, основываясь на исторической и вновь введенной информации, в реальном режиме времени. Желательно, чтобы система так же могла предложить клиенту дополнительный продукт или услугу.

3. Система должна позволять выявлять наиболее привлекательных клиентов для будущей с ними работы, наглядно представляя сотруднику финансовой компании историю каждого клиента, в том числе и кредитную.

4. Система должна позволять определять потенциальные потребности клиента, которые могут быть удовлетворены компанией. Обработываю

информацию о клиенте, CRM сообщает, какой продукт необходим ему в определенный момент.

Рассмотрим некоторые особенности применения CRM-систем в зависимости от вида Интернет-услуги. В банковском обслуживании суть услуги не меняется в зависимости от того, через какие каналы связи осуществляются взаимоотношения между клиентом и компанией. Особенностью применения системы CRM в Интернет-банкинге является то, что система должна способствовать наиболее быстрой и легкой аутентификации клиента в банковской системе, выявлять операции, которые чаще всего совершает этот клиент. К подсистеме банковских CRM относят организацию дополнительных услуг клиентам, например, возможность подбора вида заимствования/депонирования.

CRM-системы для Интернет-страхования схожи с системами CRM для интернет-магазинов, но в данном случае товаром, приобретаемым через Интернет, является страховой полис. Помимо того, что система собирает и обрабатывает всю информацию о клиенте, полученную со страниц web-сайта, из страхового договора, полиса CRM-система должна выполнять следующие функции.

- Подбор индивидуальной страховой программы для каждого нового или постоянного клиента на основе заполнения web-форм.

- Обслуживание клиента: предоставление страхователю возможности посмотреть через Интернет на web-сайте компании состояние оплаты страховки, ее сроки, условия, сообщить о страховом случае. При подборе страховой программы для постоянного клиента CRM-система должна помнить обо всех предыдущих страховках с целью облегчения процесса интернет-покупки.

В рамках услуги интернет-трейдинга CRM-система может использоваться следующим образом.

- Кастомизация представления информации. При заходе трейдера на сайт Интернет-брокера ему предлагается та информация, которую он обычно просматривает при входе в систему трейдинга: новости по сегментам рынка, котировки определенных акций/индексов, аналитические обзоры.

- Предоставление целевых программ на основе анализа потенциальных потребностей. С помощью информации, которую собирает CRM-система, сотрудники компании могут предложить клиенту именно те услуги или товары, которые с наибольшей вероятностью заинтересуют клиента.

11.6 Разработчики CRM-систем

Компания *Hyperion* является одним из лидеров среди производителей аналитического ПО. Системы, разрабатываемые компанией, направлены на анализ как внутренней деятельности организации, так и внешней. Основным продуктом для анализа взаимоотношений компании с клиентами является система *Hyperion CRM Analysis*, которая базируется на *Hyperion Essbase OLAP Server* – открытой и гибкой платформе, поддерживающей все аспекты

деятельности предприятия. Система состоит из нескольких приложений, которые могут работать как по отдельности, так и в тесной интеграции между собой. Приложения построены на платформе клиент-сервер, что обеспечивает масштабируемость и использование аналитических возможностей программ Hyperion Essbase OLAP Server и Hyperion Wired для OLAP. Приложения могут быть реализованы во внутренних сетях и в Интернете. Интерактивный графический интерфейс реализуется двумя способами: на языке Java и на основе web-браузера. Модуль, включающий все приложения, является полноценной CRM-системой. Приложения, входящие в систему Hyperion CRM Analysis:

1. Hyperion Customer Interaction Center – приложение, направленное на улучшение работы центров заказов фирмы. Оно собирает воедино наиболее существенную информацию для оперативного управления центром обработки заказов на предприятии, выполняет разделение и анализ каждого шага процесса взаимодействия с клиентом. Показывает, как изменяются во времени основные рабочие показатели.

2. Hyperion Product Quality – приложение, предназначенное для доработки продукции с учетом пожеланий клиентов. Позволяет анализировать конфигурацию продуктов и оценивать эффективность программ по обеспечению качества, помогает выявить характеристики продукта, наиболее интересующие клиентов.

3. Hyperion Field Services – приложение для анализа эффективности обслуживания на местах. Позволяет отслеживать уровень востребованности специалистов различной квалификации, оценивать степень подготовленности персонала, контролировать прибыльность проектов. Приложение позволяет анализировать услуги, предоставляемые клиентам на местах, по таким критериям, как период времени, вид продукции, класс услуг, тип клиента.

4. Hyperion Customer Profitability – приложение для наглядного предоставления сведений о прямых и косвенных затратах на обеспечение прибыльности. Направлено на выявление наиболее прибыльных клиентов, видов продукции, каналов услуг, торговых точек, подразделений, оценку влияния на прибыльность изменений в ценообразовании, уровне обслуживания и каналах поставки.

5. Hyperion Campaign Management – приложение, позволяющее оценивать и контролировать эффективность проводимых рекламных кампаний. Определяет, в каком случае выше вероятность получения ответа со стороны клиента. Оценивает, как та или иная рекламная политика влияет на вероятность приобретения клиентом, помимо запрашиваемого им товара, дополнительно предложенного ему продукта.

6. Hyperion Customer Segmentation – приложение, позволяющее компаниям разделять своих клиентов по категориям, чтобы посылаемые рекламные сообщения были как можно более информативны и отвечали индивидуальной специфике каждого клиента.

7. Hyperion Billings, Bookings and Backlog (Выставление счетов, размещение и резервирование заказов) – приложение, позволяющее оценить

влияние системы размещения заказов и уровня складских запасов на доходы и прибыль. Проводит анализ того, как разнятся плановые и фактические продажи, какие заказы уже размещены, но еще не поставлены, что дает сопоставление итоговых продаж с данными по каждому отдельному каналу сбыта за определенный период времени. Интегрируется с ERP.

Компанией Hyperion были также разработаны CRM приложения для организаций, работающих в сфере электронной коммерции:

1. Hyperion Marketing Analysis (анализ электронного маркетинга) наглядно предоставляет статистику по эффективности работы программ электронного маркетинга с позиций привлечения клиентов на web-сайт.

2. Hyperion Web Site Analysis (анализ web-сайтов) отражает "поведение" посетителей web-сайта: насколько хорошо они ориентируются на сайте, насколько успешно работает web-сайт с технической точки зрения.

Корпорация *Epicor Software* входит в число десяти крупнейших фирм-производителей ПО для автоматизации предприятий и занимает устойчивое лидирующее положение на рынке программных продуктов, ориентированных на компании среднего масштаба. В настоящее время Epicor является производителем интегрированных систем для автоматизации управления всеми аспектами деятельности предприятия, включая электронную коммерцию. Основные продукты: Bu Epicor, Platinum SQL, Vantage, Clientele, Active Planner, Platinum for Windows.

Продукты корпорации охватывают все виды деятельности, которые компания может осуществлять в Интернете: маркетинг, контакты с поставщиками и клиентами, внутрикорпорационные информационные сети. Система Bu Epicor состоит из 6 приложений. Все они базируются на программных продуктах от Microsoft. Ядро системы – операционная система Windows NT/2000; базы данных – Microsoft SQL Server 6.5 или 7.0; доступ к системе осуществляется с использованием протокола TCP/IP и браузера Microsoft IE 4.01. Управление взаимоотношениями с клиентами организовано на основе двух приложений.

Epicor ePortals – система для разработки интернетпортала компании. Он может использоваться одновременно как обычный интернетмагазин, как web-сайт для взаимодействия с посредниками и как внутрикорпорационный сайт.

Epicor eFrontOffice – приложение, направленное на анализ и улучшение маркетинговых программ, организации продаж и поддержки клиентов. Позволяет управлять отношениями с клиентом на всех этапах деятельности компании. Предназначено для использования в средних по размеру компаниях.

Epicor eFrontOffice тесно интегрируется с ePortals и приложением бэк-офиса компании eBackOffice. Для этого также используются технологии Microsoft - Microsoft Transaction Server и Microsoft Message Queue Server. Система eFrontOffice базируется на трех связанных приложениях.

Epicor eMarketing отслеживает активность клиента, связанную с рекламными акциями компании. Помогает дифференцировать клиентов по

группам, создавать списки рекламных рассылок по электронной почте. Показывает и прогнозирует зависимость доходов от затрат на маркетинг.

Ericor eSales позволяет компании измерять, анализировать и прогнозировать коммерческие результаты продаж. Подсистема иллюстрирует, на какой стадии находится осуществление каждого заказа, хранит всю детальную историю взаимоотношений компании и клиента, позволяет выявить наиболее прибыльных клиентов. Данное приложение является традиционной оперативной CRM-системой.

Ericor eSupport позволяет быстро и эффективно отвечать на любые запросы клиента. При интеграции с eSales и ePortals позволяет клиенту самостоятельно отслеживать выполнение заказа через Интернет. Приложение хранит и наглядно представляет сотруднику контактную информацию по каждому клиенту и интегрируется с Microsoft Outlook для быстрого ответа по e-mail.

Английская компания *Tranzline* представляет на рынке CRM-систему CRMSOft. Она предназначена в первую очередь для крупных компаний традиционного бизнеса, хотя и содержит модуль для поддержки электронной коммерции. CRMSOft – это 32-битная объектно-ориентированная программа, разработанная на Microsoft Visual C++. Доступ к системе CRMSOft построен по традиционному принципу клиент-сервер и может осуществляться через LAN и Internet/Intranet. Основа CRMSOft – это база данных, которая может быть построена с использованием таких СУБД, как Microsoft SQL Server v6.5 & 7.0, Oracle, Informix 7.2, SQLBase.

Система CRMSOft предназначена для использования персоналом компании и условно разделена на четыре подсистемы: маркетинг, продажи, обслуживание, электронный бизнес. Условность заключается в том, что клиент может использовать всю информацию, накопленную в базе данных, независимо от того, какой подсистемой он пользуется. Администратор системы может настроить для каждого пользователя индивидуальную подсистему, которая будет позволять анализировать ту информацию, которая нужна этому клиенту. Основные функции подсистем CRMSOft:

- Маркетинг – приложение предназначено для разработки маркетинговой стратегии по каждой целевой группе. Позволяет автоматически дифференцировать клиентов на определенные группы, подбирать маркетинговую политику и сравнивать затраты на маркетинг и результаты.

- Продажи – наглядным образом представляется иерархия всей дистрибуторской сети компании. Приложение работает с такими данными по клиенту, как способ и место доставки, метод оплаты, предоставление скидок.

- Обслуживание клиента – представление всей информации о контактах клиента с любым отделом компании и целях контакта. Отображается информация о серийном номере и гарантийном обслуживании по каждому проданному товару, контролируется состояние каждого заказа.

- **Business** – модуль предназначен для Интернет-магазинов. Интегрируется с web-сайтом и позволяет автоматически фиксировать все действия посетителя в базе данных и анализировать их.

Компания *S1 Corporation* выпустила программный продукт *S1 Consumer Suite*, который позволяет хранить и обрабатывать информацию по каждому клиенту компании и проведенных им сделках. Приложение может хранить данные по всем услугам компании, сведения о ситуации на рынке, финансовые новости. В рамках этой системы клиент имеет следующие возможности:

- получать полный отчет о своем финансовом положении в компании;
- получать персональную финансовую информацию, финансовые новости, отчеты о рынках, иметь доступ к финансовым инструментам планирования;
- получать предложение об услуге, которая была разработана специально для него.

Одной из составляющих приложения *S1 Consumer Suite* является система *S1 Relationship Management*, предназначенная для использования персоналом компании. Система позволяет собирать и соответствующим образом обрабатывать информацию по каждому клиенту, исторические данные о сделках, данные наблюдений, информацию об услугах компании, отчеты о состоянии рынка. Результатом такого анализа может быть решение о предоставлении той или иной услуги конкретному клиенту, направление ему целевой финансовой информации или советов. Система *S1 Consumer Suite* имеет модуль, с помощью которого персонал может определять содержание web-сайта компании и уровень полномочий каждого клиента в системе *S1 Consumer Suite*.

Разработчик первого российского CRM-приложения – компания "Про-Инвест информационные технологии", созданная в 1999 г. и являющаяся частью корпорации "Про-Инвест консалтинг". Можно выделить следующие CRM-решения, разработанные компанией.

- **Marketing Expert** – оценка реального положения компании на рынке, сравнительный анализ с конкурентами, формирование оптимальной сбытовой структуры и ценовой политики, разработка эффективной стратегии и тактики действия на рынке, оценка риска и расчет планируемых финансовых результатов.

- **Forecast Expert** – прогнозирование цен, объемов сбыта, курсов валют, индексов инфляции и ряда других экономических показателей.

- **Marketing GEO** – программный продукт, представляющий собой информационно-аналитическую систему, предназначенную для хранения, отображения и анализа маркетинговой информации о клиентах, поставщиках, дилерах и конкурентах.

- **Sales Expert** строит систему сбыта продукции, прозрачную для контроля и управления. Сотрудник компании, работая с системой, может получить данные об объемах продаж, представленные в любом необходимом

аспекте: по отделам, менеджерам, группам товаров, регионам, отраслям и т.д. Система Sales Expert позволяет проводить контроль деятельности отдела продаж поэтапно и на основе объективных показателей: эффективности первичных контактов, продолжительности сделки, ее средней величины и т.д.

Набор приложений от *BroadVision*, предназначенный для построения интернетпортала, помогает компании организовать все сферы деятельности, которые она может вести через Интернет: построение информационной части портала, кастомизированное отношение с поставщиками и клиентами, построение платежной компоненты, анализ маркетинговых стратегий и пр. Любое приложение, отвечающее за определенную часть портала, имеет CRM-составляющую. Основные приложения от *BroadVision*.

- *BroadVision One-To-One Enterprise* – основной инструмент, применяемый для кастомизации корпоративного портала, является базовым для остальных приложений.

- *BroadVision Retail Commerce Suite* и *BroadVision MarketMaker* – традиционная eCRM-система, применяемая для построения интернетмагазина, ведения интернетмаркетинг и получения доступ ко всей информации по клиентам.

- *BroadVision Finance* – готовое решение для ведения бизнеса в Интернете финансовыми институтами.

- *BroadVision-Amadeus Travel Commerce* – готовое решение для ведения бизнеса в Интернете для туристических агентств.

- *BroadVision InfoExchange Portal* – готовое интернетрешение для персонализированного представления информационных данных в сети.

- *BroadVision Billing* – организация клиентской части платежной системы.

Корпорация *Oracle* является второй по размерам крупнейшей компанией по производству программных продуктов и лидером по производству программных продуктов для управления информацией в масштабах предприятия. *Oracle* является первой компанией-производителем ПО, реализовавшей модель интернетвычислений для разработки ПО масштаба предприятия. Современная ERP-система *Oracle E-Business Suite 11i* – полностью интегрированный комплекс приложений электронного бизнеса. Он предназначен для автоматизации электронного бизнеса, начиная со сбыта и маркетинга и заканчивая закупками и управлением сетью снабжения через Интернет.

ЛЕКЦИЯ 13

План лекции:

1. Хеширование данных.
2. Внешние деревья поиска.
3. Операторы на B -дереве.
4. Сравнение методов.

13.1 Хеширование данных

На сегодняшний день существует множество способов повышения эффективности работы с большими объемами информации. Например, для ускорения доступа к данным в таблицах можно использовать предварительное упорядочивание таблицы в соответствии со значениями ключей. При этом могут быть использованы методы поиска в упорядоченных структурах данных, что существенно сокращает время поиска данных по значению ключа. Однако при добавлении новой записи требуется переупорядочить таблицу. Потери времени на повторное упорядочивание справочника часто превышают выигрыш от сокращения времени поиска. Рассмотрим способы организации данных, лишенные указанного недостатка. Это различные формы хеширования данных.

На рисунке 13.1 показана базовая структура данных при открытом хешировании. Основная идея метода заключается в том, что множество данных разбивается на конечное число классов. Для B классов, пронумерованных от 0 до $B-1$, строится хеш-функция h , которая для любого элемента x исходного множества функция $h(x)$ принимает целочисленное значение из интервала 0, ..., $B-1$, соответствующее классу, которому принадлежит элемент x . Элемент x называют ключом, $h(x)$ – хеш-значением x , а классы – сегментами. Массив (таблица сегментов), проиндексированный номерами сегментов 0, 1, ... $B-1$, содержит заголовки для B списков. Элемент x i -го списка – это элемент исходного множества, для которого $h(x)=i$.

Если сегменты приблизительно равны по размеру, то в этом случае списки всех сегментов должны быть наиболее короткими при данном числе сегментов. Если исходное множество состоит из N элементов, тогда средняя длина списков будет N/B элементов. Если удастся оценить величину N и выбрать B как можно ближе к этой величине, то в каждом списке будет один-два элемента. Тогда время выполнения операций с данными будет малой постоянной величиной, зависящей от N или от B . Однако не всегда ясно, как выбрать хеш-функцию h так, чтобы она примерно поровну распределяла элементы исходного множества по всем сегментам.

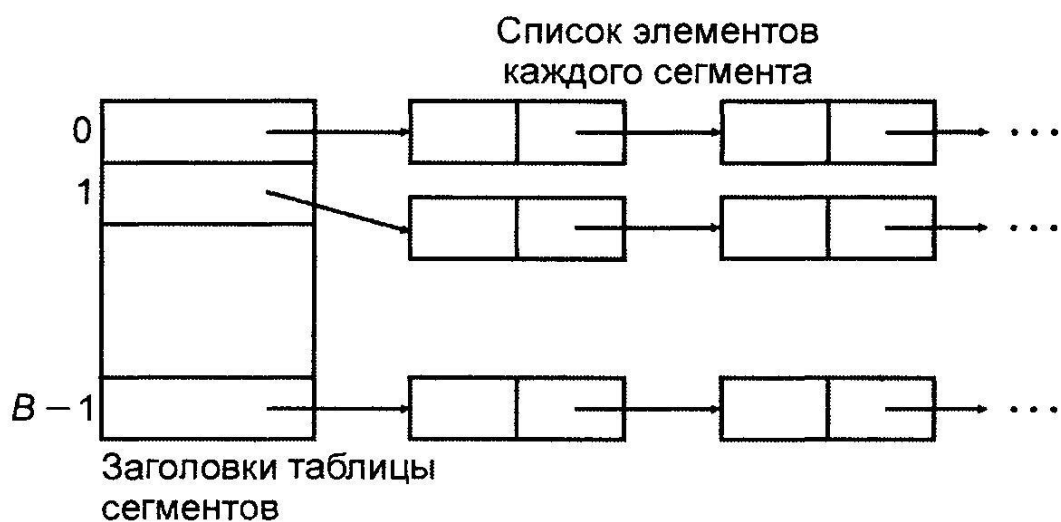


Рисунок 13.1 – Организация данных при открытом хешировании

Идеальной хеш-функцией является такая, которая для любых двух неодинаковых ключей выдает неодинаковые адреса, т. е.

$$(13.1) \quad k_1 \neq k_2 \Rightarrow h(k_1) \neq h(k_2).$$

Однако подобрать такую функцию можно в случае, если все возможные значения ключей известны заранее. Такая организация данных носит название «совершенное хеширование». Если заранее не определено множество значений ключей и длина таблицы ограничена, подбор совершенной функции затруднителен. Поэтому часто используют хеш-функции, которые не гарантируют выполнение условия (13.1).

Хеширование – распространенный метод обеспечения быстрого доступа к информации, хранящейся во вторичной памяти. Записи файла распределяются между *сегментами*, каждый из которых состоит из связанного списка одного или нескольких блоков внешней памяти.

Имеется таблица сегментов, содержащая B указателей, – по одному на каждый сегмент. Каждый указатель в таблице сегментов представляет собой физический адрес первого блока связанного списка блоков для соответствующего сегмента. Сегменты пронумерованы от 1 до B . Хеш-функция h отображает каждое значение ключа в одно из целых чисел от 1 до B . Если x – ключ, то $h(x)$ является номером сегмента, который содержит запись с ключом x , если такая запись вообще существует. Блоки, составляющие каждый сегмент, образуют связанный список. Таким образом, заголовок i -го блока содержит указатель на физический адрес $(i+1)$ -го блока. Последний блок сегмента содержит в своем заголовке *nil*-указатель. Такой способ организации показан на рисунке 13.2. При этом в данном случае элементы, хранящиеся в одном блоке сегмента, не требуется связывать друг с другом с помощью указателей. Связывать между собой нужно только блоки.

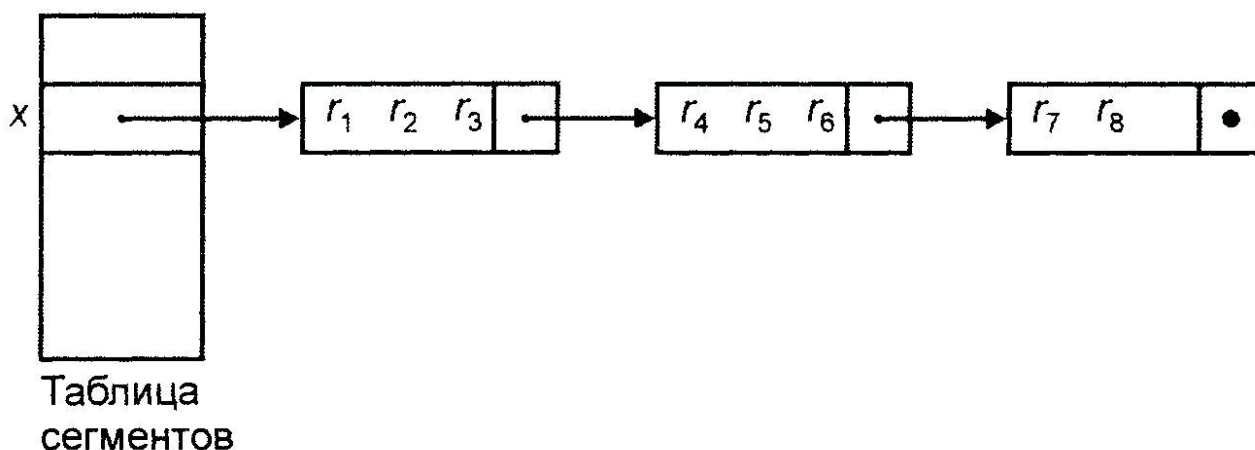


Рисунок 13.2 – Сегменты, состоящие из связанных блоков

Если размер таблицы сегментов невелик, ее можно хранить в основной памяти, иначе ее можно хранить последовательным способом в отдельных блоках внешней памяти. Если требуется найти запись с ключом x , вычисляется $h(x)$ и находится блок таблицы сегментов, содержащий указатель на первый блок сегмента $h(x)$. Затем последовательно считываются блоки сегмента $h(x)$, пока не обнаружится блок, содержащий запись с ключом x . Если исчерпаны все блоки в связном списке для сегмента $h(x)$, делается вывод, что x не является ключом ни одной из записей.

Такая структура оказывается эффективной, если в выполняемом операторе указываются значения ключевых полей. Среднее количество обращений к блокам, требующееся для выполнения оператора, в котором указан ключ записи, приблизительно равняется среднему количеству блоков в сегменте, которое равно n/bk , если n – количество записей, блок содержит b записей, а k соответствует количеству сегментов. В результате при такой организации данных операторы, использующие значения ключей, выполняются в среднем в k раз быстрее, чем в случае неорганизованного файла.

Чтобы вставить запись с ключом, значение которого равняется x , нужно сначала проверить, нет ли в файле записи с таким значением ключа. Если такая запись есть, то выдается сообщение об ошибке, поскольку предполагается, что ключ уникальным образом идентифицирует каждую запись. Если записи с ключом x нет, новая запись вставляется в первый блок цепочки для сегмента $h(x)$, в который ее удастся вставить. Если запись не удастся вставить ни в один из существующих блоков сегмента $h(x)$, файловой системе выдается команда – найти новый блок, в который будет помещена эта запись. Затем новый блок добавляется в конец цепочки блоков сегмента $h(x)$.

Для удаления записи с ключом x требуется сначала найти эту запись, а затем установить ее бит удаления.

Удачная организация файлов с хешированным доступом требует лишь незначительного числа обращений к блокам при выполнении каждой операции с файлами. Если выбрана удачная функция хеширования, а количество сегментов приблизительно равно количеству записей в файле,

деленному на количество записей, которые могут поместиться в одном блоке, тогда средний сегмент состоит из одного блока. Если не учитывать обращения к блокам, которые требуются для просмотра таблицы сегментов, типичная операция поиска данных, основанного на ключах, потребует одного обращения к блоку, а операция вставки, удаления или изменения потребуют двух обращений к блокам. Если среднее количество записей в сегменте намного превосходит количество записей, которые могут поместиться в одном блоке, можно периодически реорганизовывать таблицу сегментов, удваивая количество сегментов и деля каждый сегмент на две части.

13.2 Внешние деревья поиска

Для представления внешних файлов удобно использовать древовидные структуры данных. *B*-дерево, являющееся обобщением бинарных деревьев, удачно подходит для представления внешней памяти. Поэтому оно стало стандартным способом организации индексов в системах баз данных.

Обобщением дерева двоичного поиска является *m*-арное дерево, в котором каждый узел имеет не более *m* сыновей. Так же, как и для деревьев бинарного поиска, для *m*-арного дерева считается, что выполняется следующее условие: если n_1 и n_2 являются двумя сыновьями одного узла и n_1 находится слева от n_2 , тогда все элементы, исходящие вниз от n_1 , оказываются меньше элементов, исходящих вниз от n_2 . Операции поиска, вставки и удаления элементов для *m*-арного дерева поиска реализуются путем обобщения аналогичных операций для деревьев бинарного поиска.

Однако для внешних деревьев поиска важна проблема хранения записей в файлах, когда файлы хранятся в виде блоков внешней памяти. Правильным применением идеи разветвленного дерева является представление об узлах как о физических блоках. Внутренний узел содержит указатели на своих *m* сыновей, а также *m*-1 ключевых значений, которые разделяют потомков этих сыновей. Листья также являются блоками, содержащими записи основного файла.

Если использовать дерево двоичного поиска из *n* узлов для представления файла, хранящегося во внешней памяти, то для поиска записи в таком файле потребуется в среднем $\log_2 n$ обращений к блокам. Если вместо бинарного дерева поиска использовать для представления файла *m*-арное дерево поиска, то для поиска записи в таком файле потребуется $\log_m n$ обращений к блокам. В случае $n = 1\,000\,000$ дерево бинарного поиска потребовало бы примерно 20 обращений к блокам, тогда как 128-арное дерево поиска потребовало бы лишь трех обращений к блокам. Однако нельзя сделать *m* очень большим, поскольку чем больше *m*, тем больше должен быть размер блока. Кроме того, считывание и обработка более крупного блока занимают больше времени, поэтому существует оптимальное значение *m*, которое позволяет

минимизировать время, требующееся для просмотра внешнего m -арного дерева поиска.

B -дерево – это особый вид сбалансированного m -арного дерева, который позволяет выполнять операции поиска, вставки и удаления записей из внешнего файла с гарантированной производительностью для самой неблагоприятной ситуации.

B -дерево порядка m представляет собой m -арное дерево поиска, характеризующееся следующими свойствами.

- 1 Корень либо является листом, либо имеет хотя бы двух сыновей.
- 2 Каждый узел, кроме корня и листьев, имеет от $m/2$ до m сыновей.
- 3 Все пути от корня до любого листа имеют одинаковую длину.

На рисунке 13.3 показано B -дерево порядка 5; предполагается, что в блоке листа помещается не более трех записей.

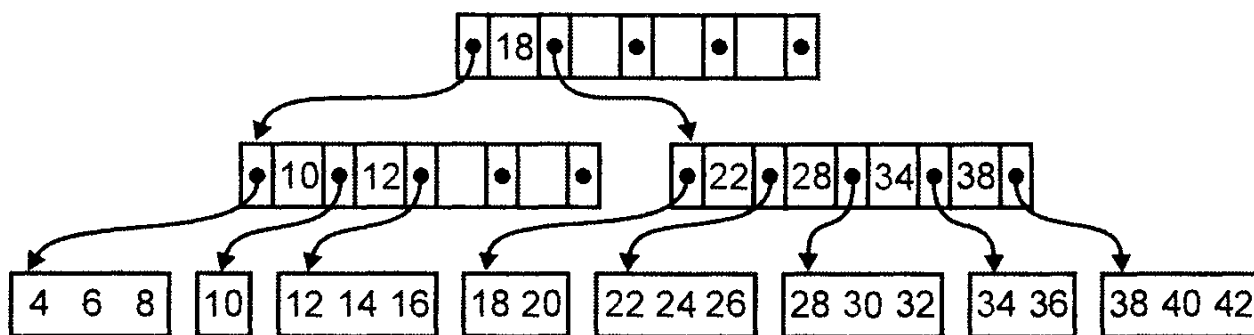


Рисунок 13.3 – B -дерево порядка 5

B -дерево можно рассматривать как иерархический индекс, каждый узел в котором занимает блок во внешней памяти. Корень B -дерева является индексом первого уровня. Каждый нелистовой узел на B -дереве имеет форму $(p_0, k_1, p_1, k_2, p_2, \dots, k_n, p_n)$, где p_i является указателем на i -ого сына, $0 \leq i \leq n$, а k_i – ключ, $1 \leq i \leq n$. Ключи в узле упорядочены, поэтому $k_1 < k_2 < \dots < k_n$. Все ключи в поддереве, на которое указывает p_0 , меньше, чем k_1 . В случае $1 \leq i < n$ все ключи в поддереве, на которое указывает p_i , имеют значения, не меньшие, чем k_i , и меньшие, чем k_{i+1} . Все ключи в поддереве, на которое указывает p_n , имеют значения, не меньшие, чем k_n .

Существует несколько способов организации листьев. В данном случае предполагается, что записи основного файла хранятся только в листьях и каждый лист занимает один блок.

13.3 Операторы на B -дереве

Поиск записей. Если требуется найти запись r со значением ключа x , нужно проследить путь от корня до листа, который содержит r , если эта

запись вообще существует в файле. При прохождении указанного пути последовательно считываются из внешней памяти в основную внутренние узлы $(p_0, k_1, p_1, k_2, p_2, \dots, k_n, p_n)$ и вычисляется положение x относительно ключей k_1, k_2, \dots, k_n . Если $k_i \leq x < k_{i+1}$, тогда в основную память считывается узел, на который указывает p_i , и повторяется описанный процесс. Если $x < k_1$, для считывания в основную память используется указатель p_0 . Если $x \geq k_n$, тогда используется p_n .

Когда в результате изложенного процесса попадают на какой-либо лист, то пытаются найти запись со значением ключа x . Если количество входов в узле невелико, то в этом узле можно использовать линейный поиск, иначе лучше воспользоваться двоичным поиском.

Вставка записей. Если требуется вставить в B -дерево запись r со значением ключа x , нужно сначала воспользоваться процедурой поиска, чтобы найти лист L , которому должна принадлежать запись r . Если в L есть место для новой записи, то она вставляется в требуемом порядке в L . В этом случае не требуется внесения каких-либо изменений в предков листа L .

Если в блоке листа L нет места для записи r , у файловой системы запрашивается новый блок L' и из L в L' перемещается последняя половина записей. При этом r вставляется в требуемом порядке в L или L' . Допустим, узел P является родителем узла L . P известен, поскольку процедура поиска отследила путь от корня к листу L через узел P . Теперь можно рекурсивно применить процедуру вставки, чтобы разместить в P ключ k' и указатель l' на L' . k' и l' вставляются сразу же после ключа и указателя для листа L . Значение k' является наименьшим значением ключа в L' .

Если P уже имеет m указателей, вставка k' и l' в P приведет к расщеплению P и потребует вставки ключа и указателя в узел родителя P . Эта вставка может произвести эффект домино, распространяясь на предков узла L в направлении корня, вдоль пути, который уже был пройден процедурой поиска. Это может привести даже к тому, что понадобится расщепить корень, тогда создается новый корень, причем две половины старого корня выступают в роли двух его сыновей. Это единственная ситуация, когда узел может иметь менее $m/2$ потомков.

Удаление записей. Если требуется удалить запись r со значением ключа x , нужно сначала найти лист L , содержащий запись r . Затем, если такая запись существует, она удаляется из L . Если r является первой записью в L , после этого выполняется переход в узел P – родителя листа L , чтобы установить новое значение первого ключа для L . Если L является первым сыном узла P , то первый ключ L не зафиксирован в P , а появляется в одном из предков P . Таким образом, надо распространить изменение в наименьшем значении ключа L в обратном направлении вдоль пути от L к корню.

Если блок листа L после удаления записи оказывается пустым, он отдается файловой системе. После этого корректируются ключи и указатели в P , чтобы отразить факт удаления листа L . Если количество сыновей узла P

оказывается теперь меньшим, чем $m/2$, проверяется узел P' , расположенный в дереве на том же уровне непосредственно слева или справа от P . Если узел P' имеет хотя бы $m/2+2$ сыновей, ключи и указатели распределяются поровну между P и P' так, чтобы оба эти узла имели хотя бы по $m/2$ потомков, сохраняя упорядоченность записей. Затем необходимо изменить значения ключей для P и P' в родителе P и, если необходимо, рекурсивно распространить воздействие внесенного изменения на всех предков узла P , на которых это изменение отразилось.

Если у P' имеется ровно $m/2$ сыновей, P и P' объединяют в один узел с $2(m/2)-1$ сыновьями. Затем необходимо удалить ключ и указатель на P' из родителя для P' . Это удаление можно выполнить с помощью рекурсивного применения процедуры удаления.

Если обратная волна воздействия удаления докатывается до самого корня, возможно, придется объединить только двух сыновей корня. В этом случае новым корнем становится результирующий объединенный узел, а старый корень можно вернуть файловой системе. Высота B -дерева уменьшается при этом на единицу.

Рассмотрим выполнение описанных операторов на примере B -дерева, изображенного на рисунке 13.4. Вставка записи со значением ключа 23 порождает B -дерево, показанное на рисунке. 13.4. Чтобы вставить эту запись, надо расщепить блок, содержащий записи с ключами 22, 23, 24 и 26, поскольку предполагается, что в один блок помещается не более трех записей. Два меньших остаются в этом блоке, а два больших помещаются в новый блок. Пару указатель-ключ для нового узла нужно вставить в родителя, который в таком случае расщепляется, поскольку не может содержать шесть указателей. Корень принимает пару указатель-ключ для нового узла, однако корень не расщепляется, т. к. он располагает достаточной емкостью.

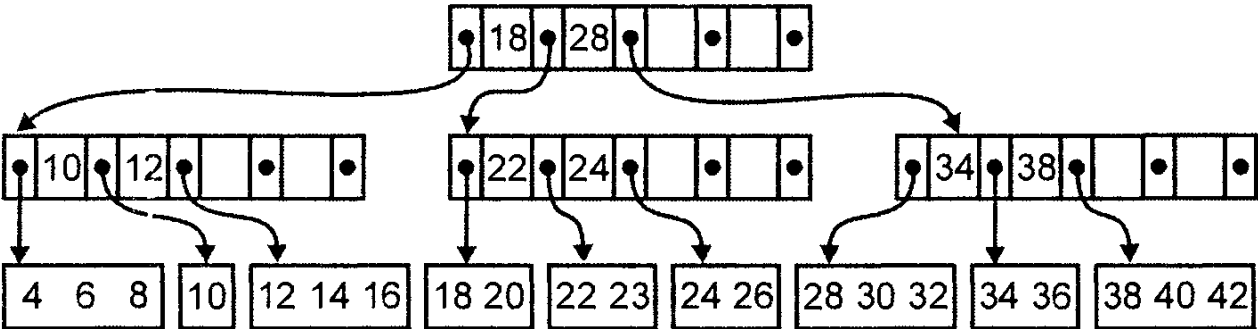


Рисунок 13.4 – B -дерево после вставки в него записи

Удаление записи с ключом 10 из B -дерева, показанного на рисунке 13.4, приводит к B -дереву, изображенному на рисунке 13.5. В этом случае блок, содержащий запись с ключом 10, отбрасывается. У его родителя теперь оказывается только два сына, а у правого брата этого родителя имеется

минимальное количество сыновей – три. Таким образом, в результате объединения родителя и его брата получается один узел с пятью сыновьями.

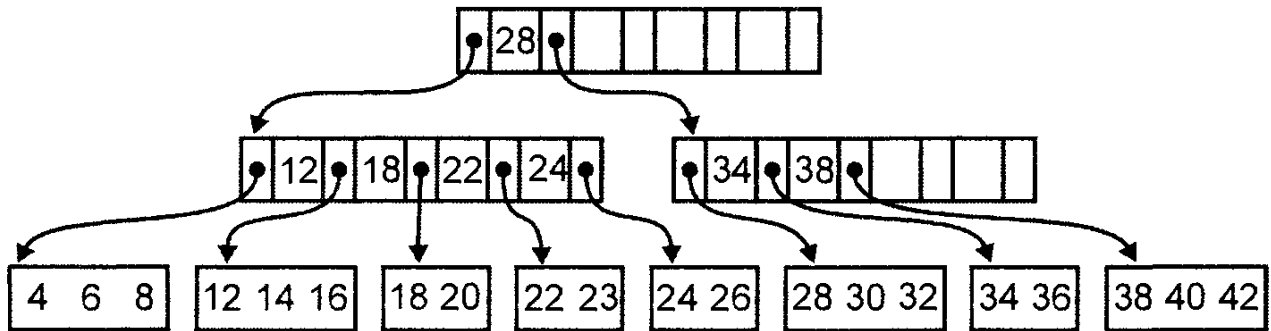


Рисунок 13.5 – B-дерево после удаления записи

13.4 Сравнение методов

На практике сравнивают количество обращений к блокам для разных методов, связанное с выполнением той или иной операции с файлами.

Хеширование зачастую является самым быстрым из методов. Оно требует в среднем двух обращений к блокам по каждой операции, не считая обращений к блокам, которые требуются для просмотра самой таблицы сегментов, если количество сегментов достаточно велико для того, чтобы типичный сегмент использовал только один блок. Однако в случае хеширования сложно обращаться к записям в отсортированной последовательности.

B-деревья получили большую популярность как средство доступа к файлам в системах баз данных. Причина этой популярности частично заключается в их способности обрабатывать запросы, запрашивая записи с ключами, относящимися к определенному диапазону. Кроме того, B-деревья удачно использовать в качестве вторичных указателей, когда ключи в действительности не определяют ту или иную уникальную запись. Все перечисленные методы намного эффективнее обычного последовательного просмотра файла. Временные различия между ними невелики и не поддаются точной аналитической оценке, особенно с учетом того, что соответствующие параметры такие, как ожидаемая длина файла и коэффициенты заполненности блоков, трудно прогнозировать заранее.

ЛЕКЦИЯ 14

План лекции:

1. Определение информационной сети.
2. Компоненты информационной сети.
3. Коммуникационные подсети.
4. Базовая эталонная модель открытой сети.
5. Область взаимодействия открытой сети.

14.1 Информационные сети

Информационные сети (ИС), став базой современной индустрии обработки информации, получили широкое применение. Поэтому актуален вопрос о рациональной структуре, создании технического и программного обеспечения информационных сетей. Реальной сетью называют систему, состоящую из совокупности ЭВМ, периферийного оборудования, терминалов, операторов, физических процессов, средств передачи информации, являющуюся полностью автономной и осуществляющую обработку и передачу информации. В этой совокупности обязательный компонент – комплекс или одна ЭВМ, остальные компонент входят в состав сети по мере необходимости. Любая система создается для выполнения ею прикладных процессов. Каждый прикладной процесс определяется одной или несколькими программами. Вторая часть прикладных процессов взаимодействует с оператором. Специальные прикладные процессы являются вспомогательными и помогают выполнению прикладных процессов. В зависимости от выполняемых задач системы делятся на три класса: *абонентские, административные и ассоциативные*.

Системы, чья основная задачи – выполнение нужд пользователей – *абонентские*. Они являются главными. Административные и ассоциативные системы предназначены для обеспечения взаимодействия абонентских систем.

Административные системы управляют процессами этого взаимодействия, а *ассоциативные* совместно с каналами передачи данных обеспечивают прокладку трактов, соединяющих прикладные процессы абонентских систем. Абонентская система состоит из двух основных частей: прикладных процессов и области взаимодействия. Назначение последней – обеспечение связи прикладных процессов друг с другом, передача информации во внешнюю для системы физическую среду, прием информации из внешней среды. Взаимодействие осуществляется через вход и выход системы.

На рисунке 14.1 изображена информационная сеть. Она представляет собой ассоциацию абонентских систем, взаимодействующих друг с другом через коммуникационную подсеть. Последняя образуется совокупностью физической среды, передающей сигналы, а также программных и технических средств, обеспечивающих передачу блоков информации по адресам их назначения.

14.2 Компоненты информационной сети

По функциональным признакам в ИС выделяют три части, входящие одна в другую. На рисунке 14.2 представлены компоненты ИС. Первая – коммуникационная подсеть. Ее задачи – выполнение функций, которые позволили бы взаимодействовать абонентским системам без учета информации о их местонахождении и о расстояниях их разделяющих. Вторая часть образуется добавлением к коммуникационной подсети области взаимодействия всех абонентских систем.

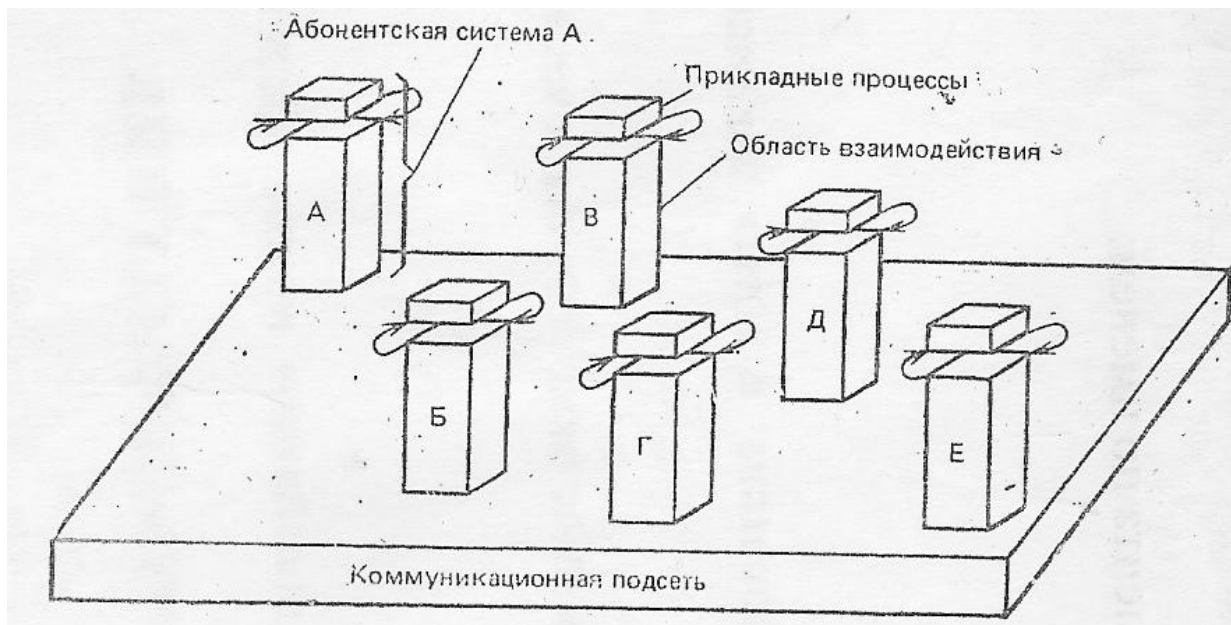


Рисунок 14.1 – Информационная сеть

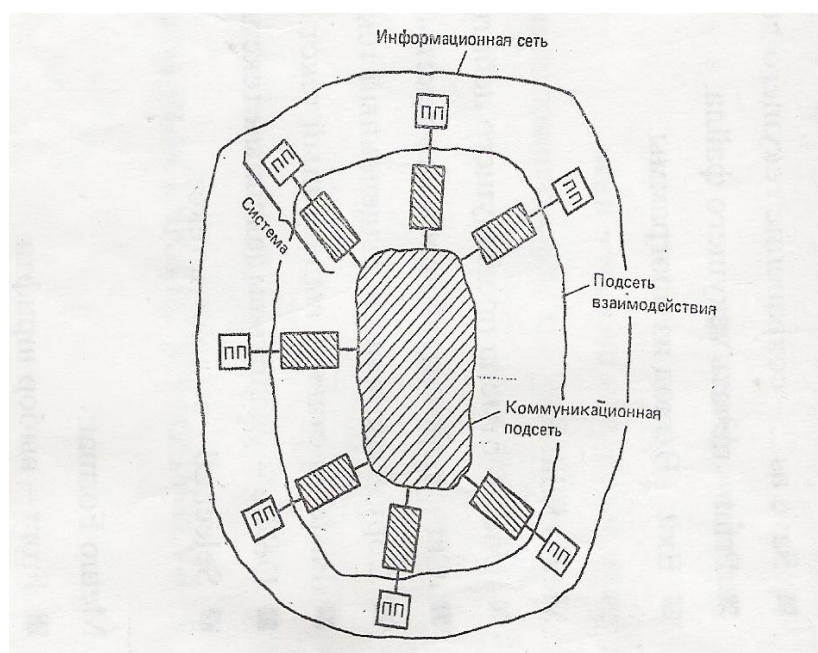


Рисунок 14.2 – Компоненты информационной сети

В результате создается подсеть взаимодействия. Задачей этой подсети является обеспечение взаимодействия любых прикладных процессов, независимо от того, в каких системах они расположены и где эти системы находятся. Третья часть включает в себя как подсеть взаимодействия, так и все прикладные процессы.

Разнообразие и сложность ИС приводят к необходимости разработки различных типов компонентов их образующих. Это связано с необходимостью создания не только отдельных сетей, но и их ассоциаций. Ассоциация состоит из группы ИС, связанных между собой одной или несколькими ассоциативными системами. Каждая из таких систем (рисунок 14.3) соединяет два и более комплексов физических средств соединения. Специальные прикладные процессы, выполняемые ассоциативными системами, обеспечивают те виды преобразования информации, которые нужны при передаче данных из одной сети в другую. Кроме того, эти процессы осуществляют коммутацию информации, связывая друг с другом физические средства соединения. Когда ассоциативная система связывает более двух физических средств соединения, прикладные процессы выполняют еще и маршрутизацию информации, заключающуюся в определении сетей, в которые необходимо передавать приходящие в данную систему блоки данных. Для управления ИС используется одна или несколько административных систем.

В зависимости от характера выполняемых прикладных процессов абонентские системы делятся на три группы. *Рабочей* называют систему, представляющую основные информационные ресурсы широкому кругу пользователей, работающих с ними через коммуникационную подсеть. Терминальная система обеспечивает взаимодействие пользователей через коммуникационную подсеть с информационными ресурсами рабочих систем. Часто роль терминальной системы выполняет персональный компьютер.

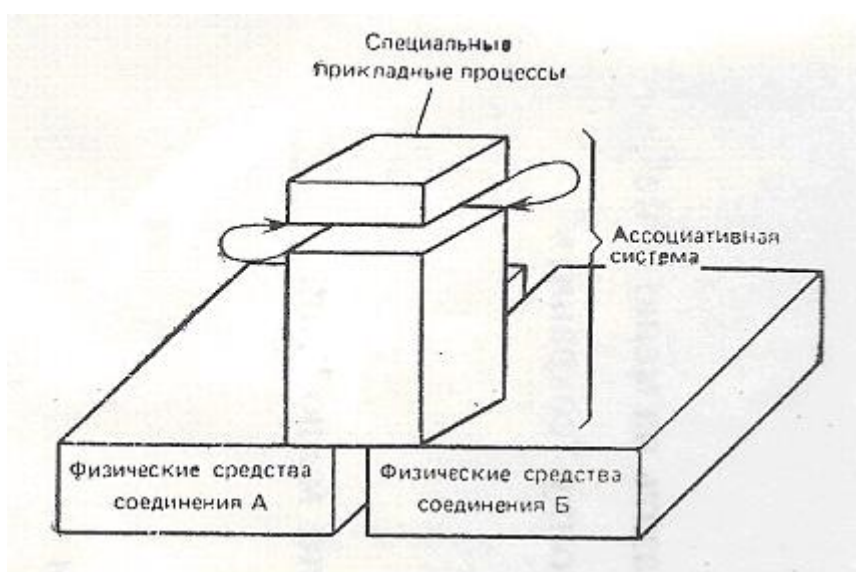


Рисунок 14.3 – Ассоциативная система

Нередко в сети используются комбинированные системы, каждая из которых выполняет функции как рабочей, так и терминальной систем. В комбинированной системе обеспечивается связь как с удаленными информационными ресурсами, так и с близлежащими.

По площади, на которой распространяется ИС, их можно разделить на два класса: территориальные и локальные. Локальными называют сети, охватывающие системы, расположенные ориентировочно в круге диаметром 50 км. Территориальными называют сети, охватывающие регионы, страны и континенты.

14.3 Коммуникационные подсети

Коммуникационная подсеть предназначена для передачи информации между большим числом абонентских систем, является базой, на которой строится вся ИС. В состав подсети входит один либо несколько комплексов физических средств соединения. При использовании нескольких комплексов последние соединяются при помощи ассоциативных систем.

Каждый комплекс *физических средств соединения* состоит из одного либо группы параллельно прокладываемых каналов передачи данных. Физические средства соединения имеют несложные оконечные устройства, образующие интерфейсы для подключения к этим средствам систем. Через один канал может работать две и более систем. Поэтому каждый канал имеет, соответственно, два и более оконечных устройств. Физические средства соединения могут содержать специальные устройства, выполняющие специфические функции.

Таким образом, коммуникационная подсеть состоит из двух видов компонентов: физических средств соединения и ассоциативных систем. В ряде простых подсетей ассоциативные системы могут отсутствовать. Пример коммуникационной подсети показан на рисунке 14.4. Подсеть состоит из семи физических средств соединения (А-Ж) и трех ассоциативных систем. В конечных точках подсети подключаются абонентские системы, создавая ИС.

Любая коммуникационная подсеть передает от абонентской системы-отправителя к абонентской системе-получателю информацию, содержащуюся в блоках данных. Каждый блок содержит определенную порцию информации и дополнительные сведения, необходимые для передачи через коммуникационную подсеть. Подсети характеризуются многими свойствами. Важнейшими из них являются те, которые определяют способы поставки информации конкретным адресатам. По этому признаку подсети делятся на два класса: с *селекцией информации* и с *маршрутизацией информации*.

Подсети с селекцией информации характеризуются тем, что в них любой блок данных передается от одной абонентской системы-отправителя всем абонентским системам. Системы, получив очередной блок данных, проверяют адрес его назначения. Система, которой адресован блок, принимает его, остальные системы отвергают этот блок. В результате происходит селекция информации, которая позволяет посылать блоки данных одной группе, а

также сразу всем абонентским системам, подключенным к коммуникационной подсети.

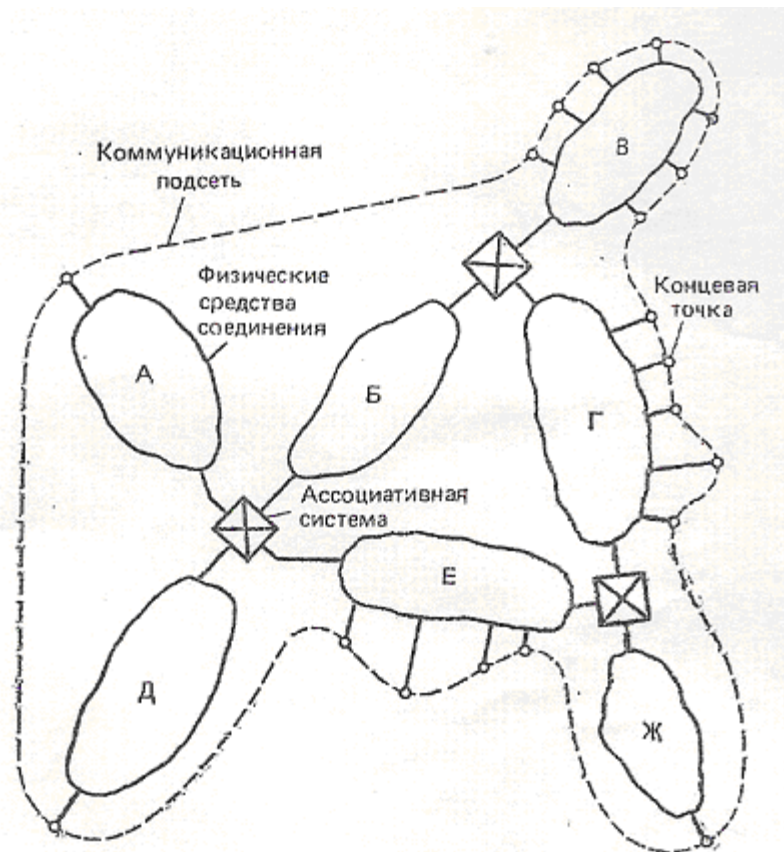


Рисунок 14.4 – Структура коммуникационной подсети

Подсети с селекцией информации делятся на две группы: *моноканальные* и *циклические*. Они различаются тем, что в подсети первой группы каждый посланный блок данных попадает ко всем абонентским системам практически одновременно, а в подсети второй группы каждый передаваемый блок доставляется всем последовательно, проходя мимо каждой из них.

Моноканальная коммуникационная подсеть (моноканал), как следует из рисунка 14.5, строится на основе общего канала, к которому через специальные



Рисунок 14.5 – Структура моноканал

устройства подключаются все абонентские системы сети. Циклическая коммуникационная подсеть, называемая циклическим кольцом, (рисунок 14.6) – это канал, имеющий кольцевую форму. В него врезаются абонентские системы, деля его на сегменты.

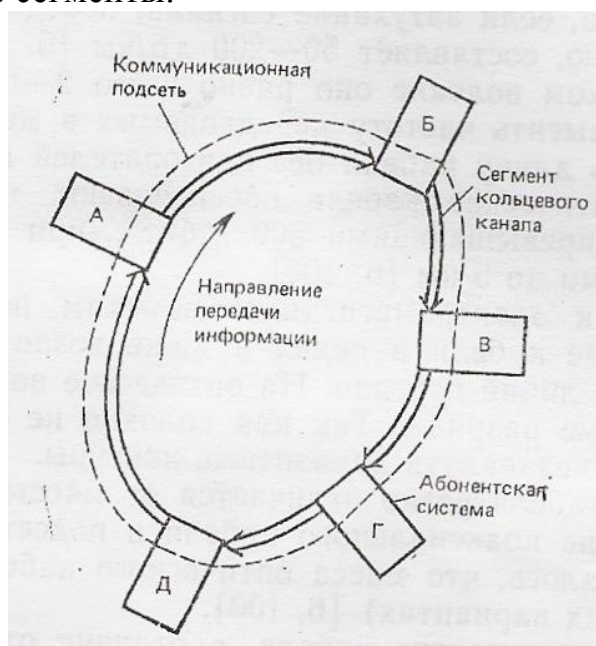


Рисунок 14. 6 – Циклическое кольцо

В коммуникационных подсетях с маршрутизацией информации передача данных осуществляется от одной абонентской системы-отправителя к другой абонентской системе-получателю. Для обеспечения такой доставки информации в подсети используются один либо более узлов коммутации. Поэтому такую подсеть называют узловой. Пример узловой подсети дан на рисунке 14.7.

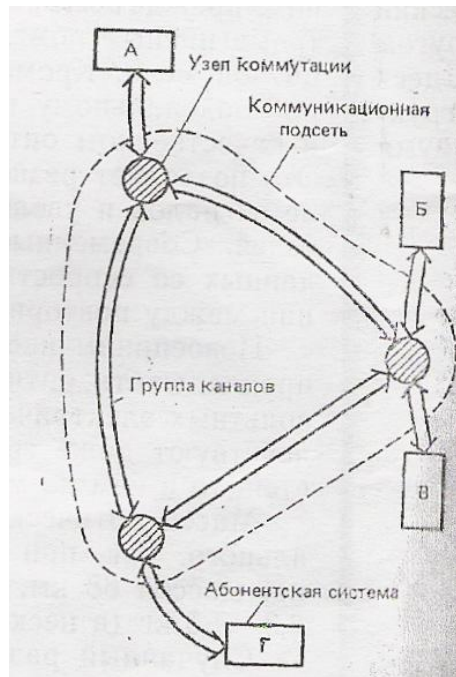


Рисунок 14.7 – Узловая подсеть

Подсеть состоит из трех узлов коммутации, а также каналов, соединяющих их друг с другом и с четырьмя абонентскими системами (А-Г). Каждый узел коммутации принимает блоки данных и передает далее по различным направлениям в зависимости от адресов их назначения. Благодаря этому в подсети осуществляется маршрутизация информации.

Моноканальные, циклические и узловые подсети нередко конкурируют между собой. При этом моноканал имеет перед другими подсетями преимущества в гибкости, расширяемости и надежности.

14.4 Базовая эталонная модель открытой сети

Теоретическую основу современных информационных сетей определяет базовая эталонная модель Международной организации стандартов (ISO). Рассматриваемая модель является основой методологии организации взаимодействия прикладных процессов, расположенных в различных абонентских системах сети. Поэтому базовая эталонная модель ISO является концептуальной и функциональной основой разработки стандартов, которые позволяют объединять в сети системы, создаваемые на компьютерах различных типов и изготовленные разными производителями. Систему, удовлетворяющую требованиям стандартов ISO, называют *открытой*. Соответственно ИС, в которой установлены открытые системы, также называют *открытой*.

Важнейшей проблемой, решаемой на основе базовой эталонной модели, является взаимодействие открытых систем. Под этим понимается абстрактное описание совместного функционирования прикладных процессов, расположенных в различных удаленных друг от друга открытых системах сети.

Сложность области взаимодействия открытых систем привела к тому, что ее разделили на семь расположенных друг над другом уровней. Это рассечение, как показано на рисунке 14.8, проходит через все абонентские системы. Таким образом, каждый из семи уровней является слоем иерархического логического описания области взаимодействия открытых систем. Любой уровень состоит из активных элементов, называемых объектами. Объекты одного и того же уровня связываются друг с другом соединениями. Они создаются расположенными ниже уровнями области взаимодействия и коммуникационной подсетью. Пример соединения объектов уровня i показан на рисунке 14.9. Это соединение проходит через все уровни обеих систем, расположенные под уровнем i , и коммуникационную подсеть.

Каждый из семи уровней обеспечивает сервис для уровня, расположенного над ним. Благодаря этому верхний, прикладной уровень предоставляет прикладным процессам весь сервис, обеспечиваемый семью уровнями, т.е. всей областью взаимодействия. Нижний уровень опирается на физические средства соединения, являющиеся фундаментом коммуникационной подсети. Основой базовой эталонной модели являются ее четыре базовых элемента:

- открытые системы;
- объекты уровня;
- соединения, связывающие объекты и позволяющие им обмениваться информацией;
- физические средства соединения.

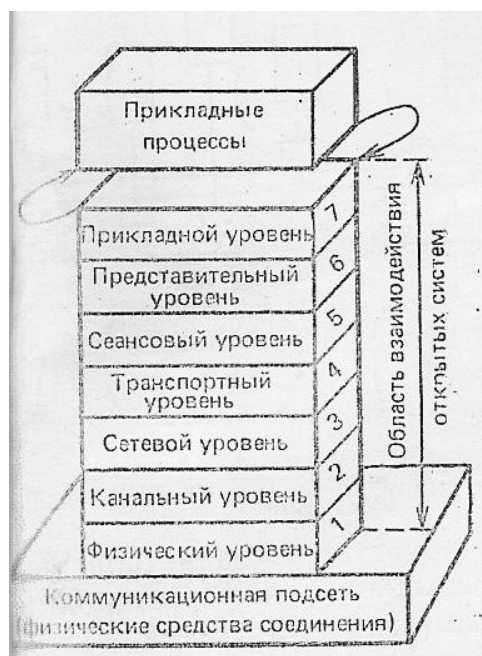


Рисунок 14.8 – Область взаимодействия открытых систем

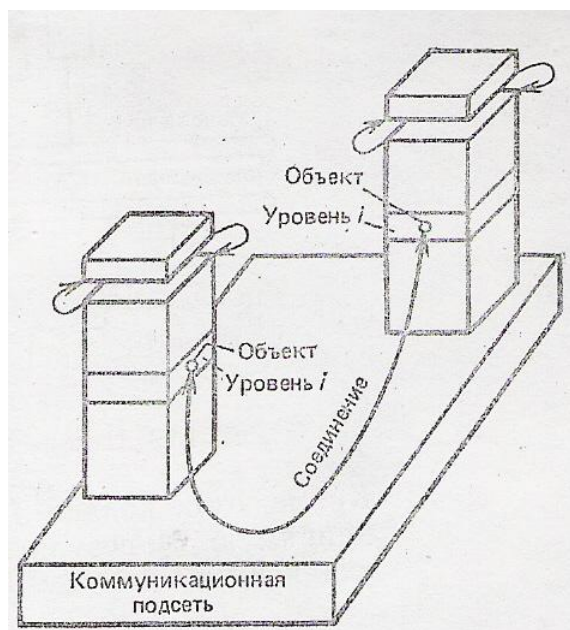


Рисунок 14.9 – Схема взаимодействия объектов

14.5 Область взаимодействия открытых систем

Стандарты ISO определяют структуру и функции области взаимодействия открытых систем, на прикладные процессы и коммуникационную подсеть эти стандарты не распространяются. Однако прикладной уровень находится рядом с прикладными процессами и должен обеспечивать для них необходимый сервис. Аналогично физический уровень предъявляет определенные требования к коммуникационной подсети.

В соответствии с разбиением на уровни открытая система рассматривается как совокупность упорядоченного набора расположенных друг над другом логических подсистем. Подсистемы одного и того же уровня образуют подуровень ИС. Смежные подсистемы взаимосвязаны друг с другом через их общий интерфейс. Сущность уровневой организации функций сводится к тому, что каждый уровень расширяет сервис, обеспечиваемый нижерасположенными уровнями. Благодаря этому наивысший уровень обеспечивает прикладные процессы полным набором всех видов сервиса, необходимых этим процессам для взаимодействия друг с другом. Прикладной процесс получает разнообразные виды сервиса. Наряду с этим уровневая организация обеспечивает относительную независимость подсистем. Это позволяет совершенствовать или заменять подсистемы одного уровня, не затрагивая работу остальных уровней. В дополнение уровневая структура дает возможность одному и тому же уровню работать с несколькими различными нижерасположенными уровнями.

Стандарты ISO для каждого уровня определяют:

- назначение уровня;
- сервис, предоставляемый данным уровнем расположенному над ним уровню;

- функции, выполняемые уровнем, и сервис, получаемый от расположенного под ним уровня.

Прикладной уровень является главным в иерархии, так как он напрямую связан с прикладными процессами и обслуживает взаимодействие этих процессов. Остальные уровни помогают прикладному в выполнении этой задачи.

Представительный уровень обеспечивает прикладному уровню понимание смысла передаваемых по сети данных. Поэтому он описывает эти данные в принятых стандартных формах и предоставляет информацию прикладным процессам. На этом уровне при необходимости происходит преобразование формата и синтаксиса данных для передачи их по сети.

Сеансовый уровень управляет передачей информации между прикладными процессами. Он позволяет обращаться к любому процессу по его имени независимо от того, в какой системе этот процесс находится. Благодаря этому прикладной процесс может перемещаться по сети.

Транспортный уровень определяет адресацию физических устройств в сети. Однако его главной задачей является обеспечение эффективных, удобных и надежных форм передачи информации между любой парой систем, в которых расположены взаимодействующие прикладные процессы. Этот уровень гарантирует доставку блоков информации адресатам и управляет доставкой.

Сетевой уровень обеспечивает прокладку виртуальных каналов между взаимодействующими системами через коммутационную подсеть. *Виртуальный канал* – это такое функционирование компонентов сети, которое создает взаимодействующим объектам иллюзию прокладки между ними нужного тракта. На этом уровне также обеспечивается управление потоками передаваемых блоков данных, называемых *пакетами*.

Канальный уровень определяет процедуры передачи данных по каналу. По каналу передаются блоки информации, называемые *кадрами*. При больших размерах передаваемых блоков данных канальный уровень делит их на кадры и передает кадры в виде последовательностей. Получив кадры, уровень восстанавливает переданные блоки данных.

Физический уровень определяет интерфейсы системы с каналом коммуникационной подсети. Эти интерфейсы включают в себя механические и электрические характеристики соединений. Уровень также описывает процедуры передачи сигналов в канал и получения их из канала.

ЛЕКЦИЯ 15

План лекции:

1. Три группы уровней взаимодействия открытых сетей.
2. Абонентские и ассоциативные системы.
3. Методы коммутации информации.

В соответствии с рассмотренными функциями область взаимодействия открытых систем может быть разделена на три группы уровней. Первую из них, называемую обслуживающей, образуют прикладной, представительный и сеансовый уровни. Задачей этой группы уровней является обслуживание прикладных процессов, в том числе обеспечение надежного и эффективного взаимодействия. Обслуживающая часть области взаимодействия создает среду взаимодействия прикладных процессов. Поэтому она должна обеспечить удобную, надежную и разнообразную по формам связь этих процессов.

Вторую группу, образуемую сетевым, канальным и физическим уровнями, называют физической. Такое название связано с тем, что структура и функции этой группы определяются видом используемой коммуникационной подсети. Данная группа обеспечивает передачу информации по подсети и может взаимодействовать только с определенными видами коммуникационной подсети.

Между обслуживающей и физической располагается третья группа уровней, включающая только один уровень – транспортный. Она освобождает обслуживающую часть от решения вопросов, связанных с передачей информации, обеспечивает независимое от вида коммуникационной подсети взаимодействие любых пар абонентских систем.

Архитектура базовой эталонной модели предусматривает два способа передачи информации: с *установлением* и *без установления соединения*. Первый способ предполагает, что перед передачей данных объекты двух систем выполняют процедуры, связанные с установкой этого соединения. Второй способ основан на том, что объекты систем знают все необходимое друг о друге заранее и направляют блоки информации, не предупреждая партнера. Этот способ проще первого, однако адресат может быть в нерабочем состоянии или не имеет возможности принять блоки в данный момент. В этих случаях передаваемые блоки сеть выбрасывает. Тогда требуются подтверждения о том, что направленная информация получена адресатом.

Взаимодействие с установлением соединения создается проведением процедур установления, поддержания и расторжения соединения. Поэтому соединение объектов существует только во время их взаимодействия, после чего аннулируется. Во время установления соединения о его создании должны договориться два объекта, а также нижний уровень – поставщик необходимого сервиса. По установленному соединению передаются последовательности блоков данных до тех пор, пока один из объектов не

заявит о расторжении соединения. Во время передачи блоков данных осуществляется управление их потоком, чтобы скорости работы соответствовали возможностям партнеров.

Если используется взаимодействие без установления соединения, то передача информации осуществляется двумя способами. Первый из них заключается в наличии двусторонней договоренности. Она состоит в знании адресов объектов, характера обмена данными, видов используемого сервиса. Имея эти сведения, объект без согласия партнера осуществляет передачу ему блоков данных. Второй способ используется тогда, когда нет указанной выше двусторонней договоренности. В этом случае объект, которому нужно передать данные, обращается к партнеру с запросом о передаче ему нужных сведений.

Каждый уровень взаимодействия систем определяется группой стандартизирующих его документов, которые всегда содержат две спецификации: протокол и сервис, обеспечиваемый этим протоколом для вышерасположенного уровня. *Протокол* – свод правил и форматов, определяющих взаимодействие объектов, которые расположены на одном уровне. Протокол любого уровня обеспечивает *сервис* для расположенного над ним уровня. Протокол определяет:

- процедуры передачи управляющей информации и данных между взаимодействующими объектами;
- механизм выбора указанных процедур из списка возможных;
- структуру и способ кодирования протокольных блоков данных.

Благодаря этому осуществляются взаимодействие объектов, использование сервиса нижерасположенного уровня, предоставление сервиса верхнему уровню. В результате системы, включенные в ИС, управляют своими действиями и обеспечивают взаимодействие прикладных процессов.

Процедуры, описываемые каждым протоколом, основаны на операциях теории конечных автоматов. При этом каждый протокол должен обладать свойствами полноты, непротиворечивости, живучести и не приводить к тупиковым ситуациям. Рассмотрим эти требования. Под *полнотой* подразумевается, что в протоколе следует устранить все неопределенные ситуации, в которых протокол «не знает», что ему нужно делать. Требование *непротиворечивости* связано с исключением ситуаций, когда протокол требует действий объекта, которые противоречат одно другому. *Тупиковые ситуации* возникают тогда, когда несколько объектов приостанавливают свои действия, одновременно требуя предоставления одного и того же ресурса. В результате функционирование протокола прекращается, и дальнейшая работа становится невозможной. Протокол должен также обладать высокой *живучестью*, обеспечивающей достижения необходимых целей даже при появлении помех в работе.

Для обеспечения гарантии того, что любой протокол выполняет указанные требования, его подвергают *верификации*. Это процесс проверки того, что объекты рассматриваемого уровня, функционируя, обеспечивают

выполнение поставленных задач и предоставляют вышестоящему уровню требуемые виды сервиса.

15.1 Абонентские системы

Абонентские системы предназначены для обработки прикладных процессов пользователей, поэтому в области взаимодействия они отсекаются на семь уровней и опираются на коммуникационную подсеть, имея структуру, показанную на рисунке 15.1. Параллельно в системе реализуется иерархия протоколов, поддерживающих прикладные процессы управления сетью. Эти протоколы могут быть теми же, что и в иерархии, поддерживающей прикладные процессы пользователей, а могут и отличаться от них. Все уровни в системе связаны с процессом управления системой.

С целью освобождения абонентской системы от выполнения функций области взаимодействия и предоставления ей возможности эффективного выполнения прикладных процессов ее делят на две части: терминальное оборудование и станцию (рисунок 15.2). Терминальное оборудование является основной частью системы, выполняющей прикладные процессы и, возможно, протоколы верхних уровней. Станция – это вспомогательная часть системы, реализующая протоколы нижних либо всех уровней. В зависимости от числа реализуемых протоколов станцию называют канальной, транспортной или абонентской. *Канальная* станция выполняет протоколы уровней 1-2, *транспортная* – протоколы уровней 1-4. *Абонентская* станция реализует все семь уровней области взаимодействия открытых систем. Станция и терминальное оборудование соединяются каналом или шиной. В обоих случаях это соединение должно быть представлено физическим и канальным протоколами.

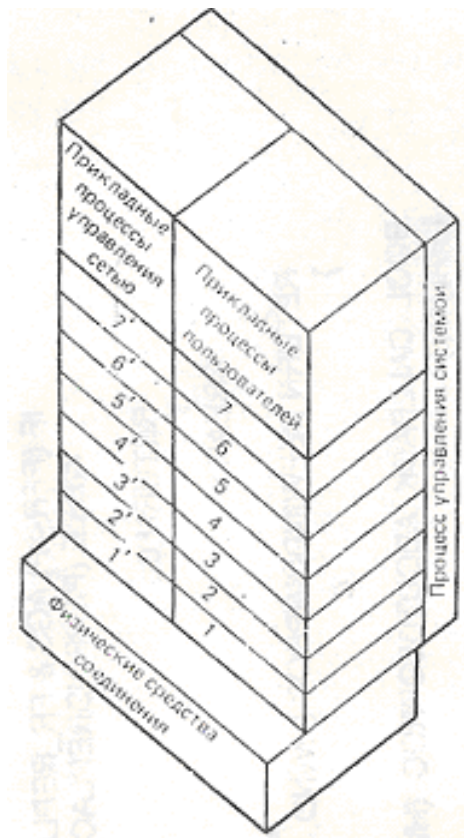


Рисунок 15.1 – Структура абонентской системы

Канальная станция является наиболее простой, однако эта простота требует серьезной загрузки абонента, который должен выполнять функции, описываемые протоколами остальных пяти уровней.

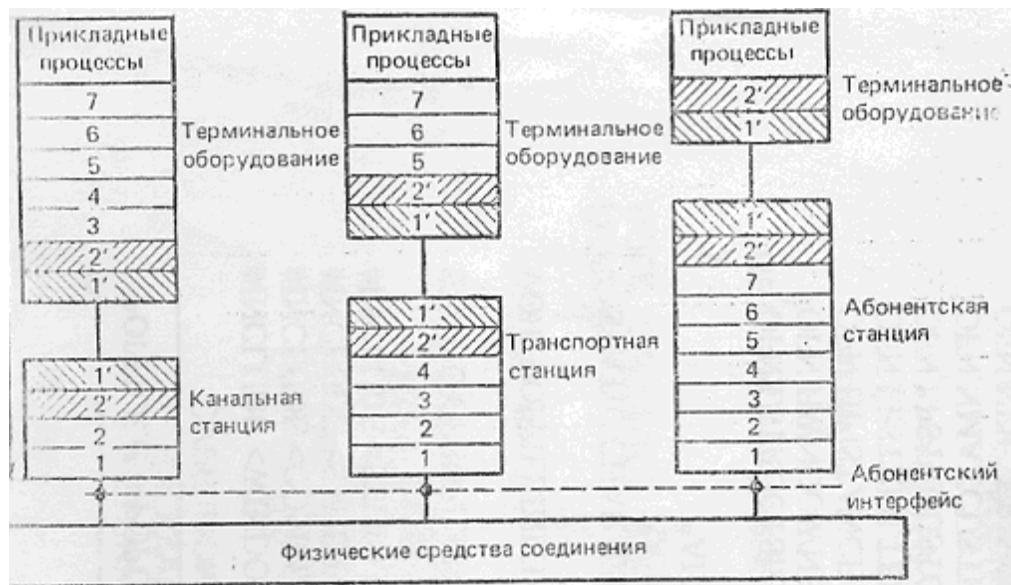


Рисунок 15.2 – Станции и абоненты

Абонентская станция полностью разгружает терминальное оборудование от выполнения задач, обеспечивающих взаимодействие в сети прикладных процессов, однако в сложном терминальном оборудовании часто

работают несколько комплексов прикладных процессов. Обмен информацией между ними происходит через сеансовый уровень. Поэтому в тех случаях, когда пятый уровень находится в станции, работа терминального оборудования оказывается зависящей от надежности, помехоустойчивости и пропускной способности канала (шины) и станции, что не всегда приемлемо.

Поэтому на практике чаще всего используется транспортная станция. Она выполняет все функции, связанные с передачей информации между комплексами терминального оборудования через всю коммутационную подсеть. А терминальное оборудование обеспечивает работу прикладных процессов, которые поддерживаются прикладным, представительным и сеансовым протоколами.

Административные системы имеют ту же структуру, что и абонентские, только в них вместо прикладных процессов пользователей работают прикладные процессы управления сетью или ее частью.

15.2 Ассоциативные системы

Ассоциативная система в отличие от вышеописанных не осуществляет обработку информации для нужд пользователей и управления сетью. Она предназначена для соединения в единое целое частей ИС и обеспечения взаимодействия этих сетей друг с другом.

В зависимости от характеристик объединяемых сетей и их частей выделяют четыре типа ассоциативных систем (рисунок 15.3). Наиболее сложной из них является *шлюз*. Он обеспечивает взаимодействие двух или более ИС с различными наборами протоколов семи уровней. На рисунке 15.3 показаны два набора: 1'-7' и 1''-7'', связанные специальными прикладными процессами. Они преобразуют один семиуровневый набор протоколов в другой, обеспечивая необходимое взаимодействие. Шлюзы чаще всего используются тогда, когда нужно объединить ИС, созданные по различным фирменным стандартам. Когда же проектируется группа сетей в соответствии со стандартами ISO, целесообразен другой подход. В этом случае в соединяемых сетях протоколы уровней 4-7 делаются одинаковыми, что позволяет для соединения сетей использовать более простые ассоциативные системы: маршрутизаторы, мосты.

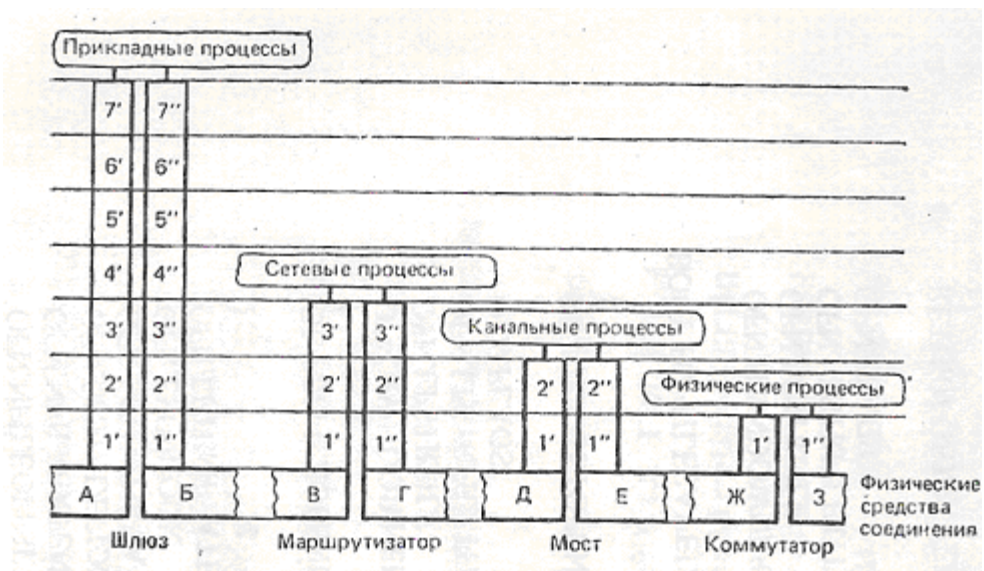


Рисунок 15.3 – Типы ассоциативных систем

Задачей *маршрутизатора* является обеспечение взаимодействия коммуникационных подсетей. Они характеризуются только тремя уровнями протоколов, поэтому логическая структура маршрутизатора имеет вид, показанный на рис. 15.3, откуда следует, что маршрутизатор «не знает» протоколов уровней 4-7 и является прозрачным для них. В его задачу входит преобразование протоколов трех нижних уровней. Иногда в ИС маршрутизаторы связывают части коммуникационной подсети, в которых используются одинаковые протоколы уровней 1-3. В этих случаях в маршрутизаторах не происходит преобразование протоколов, а их самих называют узлами коммутации пакетов. Здесь сетевые процессы осуществляют только коммутацию и маршрутизацию информации. В соединяемых узлами подсетях должна быть осуществлена общая адресация абонентских систем.

Мосты (рисунок 15.3) предназначены для соединения частей сетей с различными типами каналов передачи данных, например, циклического кольца с моноканалом. Любой канал определяется протоколами уровней 1-2, поэтому логическая структура моста имеет двухуровневую структуру. Канальные процессы здесь преобразуют протоколы обоих уровней. При использовании мостов в соединяемых подсетях должны быть согласованы структура адресов и размер кадров.

Более сложные интеллектуальные мосты наряду с указанными задачами выполняют также роль фильтров, не пропускающих сквозь себя пакеты, не адресованные другой части сети. В каждом интеллектуальном мосте содержится небольшая база данных, в которую записываются адреса систем обеих подсетей. Контрольная сумма, предназначенная для проверки кадра, используется не только на входе моста, но и на его выходе. Это позволяет предотвращать появление ошибок внутри моста. Благодаря простоте выполняемых функций мосты имеют относительно несложную структуру и работают с высокой скоростью. Форматы передаваемых блоков данных и размеры этих блоков мост не меняет.

Мосты не имеют механизмов управления потоками. Поэтому, если входной поток кадров больше выходного, то буферы переполняются и «избыточные» кадры выбрасываются. Нередко кадры, которые в течение установленного интервала времени не могут быть переданы, также ликвидируются. Мосты могут соединять друг с другом значительное число сетей.

Наиболее простую структуру имеет *коммутатор* (рисунок 15.3). Это связано с тем, что он соединяет друг с другом только каналы передачи данных, образуя необходимую физическую базу тракта передачи информации между абонентскими системами. В том случае, когда к коммутатору подходит более двух каналов, он выполняет функции, связанные с коммутацией информации. Коммутация выполняется без какой-либо обработки информации. Коммутатор не имеет буферов, поэтому он прозрачен для информации. Коммутатор требует, чтобы скорости передачи данных по соединяемым каналам были одинаковы. Физически процессы, выполняемые коммутатором, реализуются аппаратно.

Объединение сетей осуществляется на основе одного из двух принципов: с установлением соединения, без установления соединения. Первый способ позволяет заранее распределять буферы и другие ресурсы системы. В этом случае обеспечивается простое и надежное управление потоками информации, проходящими из одной сети в другую. При этом обеспечиваются уведомление о потере блоков данных и упорядочение последовательностей этих блоков. Однако организация и поддержание межсетевых соединений требует выполнения сложных протоколов. Второй способ объединения характеризуется простотой протоколов и высокой скоростью работы ассоциативной системы, а все преимущества объединения с установлением соединений становятся здесь недостатками. Для их компенсации абонентские системы обеих сетей должны иметь мощные версии транспортных протоколов.

Таким образом, при объединении сетей используются две модели. Первая из них *протокольная*, характеризуется тем, что в верхней части сетевого уровня располагается межсетевой протокол. Благодаря ему обе сети могут работать практически как одна общая сеть, а ассоциативная система обеспечивает коммутацию проходящей через нее информации. Второй моделью объединения сетей является *эстафетная*. В ней межсетевой протокол отсутствует, и через ассоциативную систему не происходит передача управляющей информации. В результате чего в каждой сети должно выполняться распознавание глобальных адресов, на основе чего выбираются маршруты движения передаваемой информации.

15.3 Методы коммутации информации

В архитектуре открытых сетей большое значение имеет коммутация информации. Ее задачей является прокладка в сетях трактов, необходимых для доставки последовательностей блоков данных абонентским системам-адресатам. Часто в управлении производственными процессами требуется, чтобы каждый передаваемый блок данных был доставлен адресату с временной задержкой, не превышающей определенного максимума. Если такое требование отсутствует, блоки данных могут приходиться со случайным запаздыванием в обозначенном диапазоне значений. В этом случае оборудование и программное обеспечение коммутации информации значительно проще.

В современных сетях используются четыре способа коммутации информации: коммутация каналов, пакетов, смешанная и интегральная коммутация.

Коммутация каналов появилась раньше других в сетях, имеющих большое число каналов. Для этой цели в коммуникационной подсети устанавливается один или более число коммутаторов, называемых узлами коммутации каналов. Каждый из них соединяет нужные пары каналов передачи данных. В результате создаются тракты – последовательности каналов, через которые обеспечивается взаимодействие пар абонентских систем. Пример коммуникационной подсети, в которой работают четыре коммутатора (1-4), показан на рисунке 15.4. Каждый коммутатор выполняет функции, связанные с реализацией протокола уровня 1 и обеспечением физических процессов. Уровень 1 в коммутаторе соединяет рассматриваемый коммутатор с одним или группой подходящих к нему каналов. Физический процесс осуществляет необходимую коммутацию, т.е. передает блоки данных из одного канала в другой.

При работе в режиме коммутации каналов перед началом взаимодействия необходимо выполнить ряд предварительных процедур, связанных с организацией создания тракта. Для этого абонентская система – инициатор взаимодействия, должна сформировать и отправить ближайшему коммутатору запрос на прокладку сквозь коммуникационную подсеть необходимого тракта. После образования тракта две абонентские системы используют его во время сеанса взаимодействия для передачи друг другу последовательностей блоков данных. Как показывает практика, во время сеанса последовательность каналов используется относительно небольшое время, а большую часть времени они простаивают. Другим недостатком коммутации каналов является длительное время создания тракта, так как для этого необходимо дождаться момента, когда в коммуникационной подсети будет свободна вся необходимая последовательность каналов.

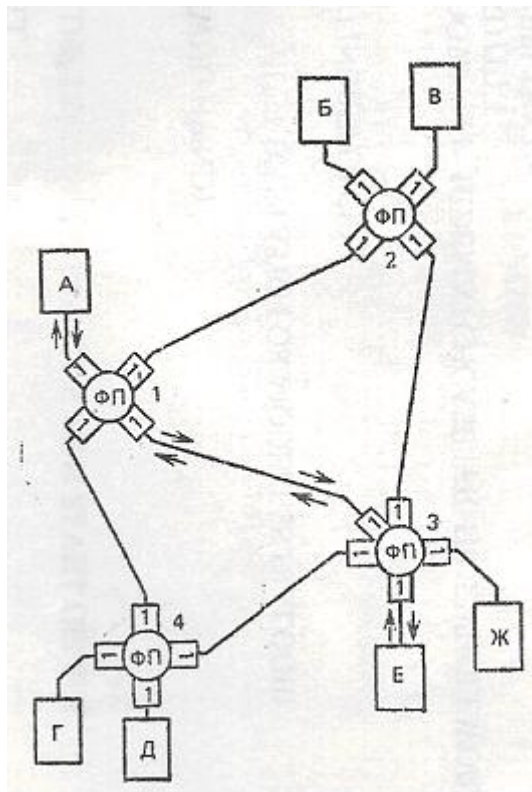


Рисунок 15.4 – Коммутация каналов

Поэтому при коротких сеансах время создания тракта превышает время передачи блоков данных. Все это приводит к тому, что коммутация каналов оказывается самым неэффективным способом коммутации информации. Поэтому он чаще всего используется при передаче больших массивов информации или в тех случаях, когда требуется передача блоков данных с заданным временем их доставки.

Метод *коммутации пакетов* свободен от указанных недостатков. Его важной особенностью является одновременное коллективное использование каналов значительным числом абонентских систем. Здесь физические тракты не создаются, и ни один канал не отдается в монопольное владение паре абонентских систем. По каждому из каналов по мере поступления передаются блоки данных, посылаемые различными абонентскими системами. Однако коммутация пакетов не обеспечивает строго определенное время доставки последовательностей блоков данных. Поэтому использование этого способа в задачах, где это время нужно гарантировать, оказалось невозможным или затрудненным.

Опыт и расчеты показывают, что экономически выгодно по одним и тем же каналам передавать любую информацию. Как ту, которая требует определенного времени доставки, так и ту, которая допускает появление случайных величин запаздывания. В этой связи возникла необходимость создания смешанной коммутации, обеспечивающей на базе одного и того же множества каналов коммутацию как каналов, так и пакетов.

При *смешанной коммутации* используются уровни и процессы, применяемые как в коммутации каналов, так и в коммутации пакетов. Имеющиеся каналы в первую очередь отдаются для создания трактов, соединяющих абонентские системы. Свободные при этом каналы не простаивают и используются для коммутации пакетов. В данном случае в подсети устанавливаются комбинированные узлы, выполняющие роль как коммутаторов каналов, так и коммутаторов пакетов. Смешанная коммутация начинает широко использоваться для одновременной передачи по одним и тем же группам каналов данных и речи.

Четвертым типом коммутации информации является *интегральная коммутация*. Она, как и смешанная, предназначена для обеспечения передачи информации с заданным и случайным временем доставки блоков данных. Однако здесь коммутация каналов и коммутация пакетов осуществляются одновременно в каждом физическом канале. Для обеспечения интегральной коммутации в каждом таком канале прокладывается группа виртуальных каналов. Любой из них работает так, что у пары взаимодействующих абонентских систем создается впечатление, что они, используя виртуальный канал, передают блоки данных по отведенному им физическому каналу.

Один из способов интегральной коммутации называется *асинхронным временным мультиплексированием*. Сущность этого способа иллюстрируется рисунком 15.5, в соответствии с которым, для каждого физического канала сети время делится на повторяющиеся циклы. По каналу передаются разграничители, каждый из которых сообщает о начале очередного цикла. После этого в каждом цикле выделяются n интервалов времени, необходимых для создания n виртуальных каналов.

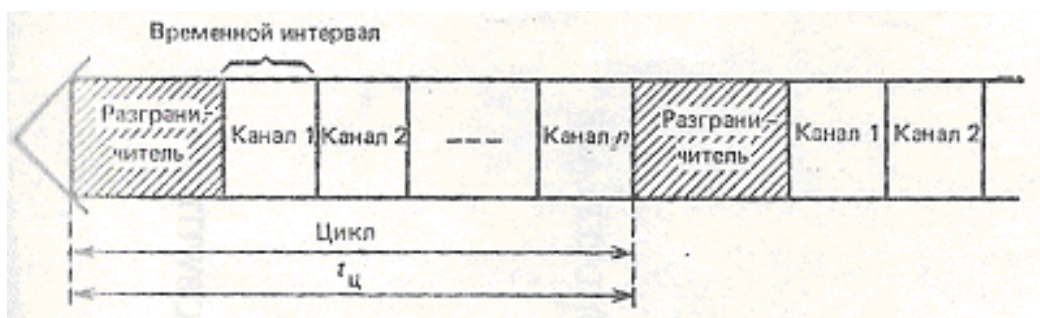


Рисунок 15.5 – Структура асинхронного мультиплексирования

На базе временных интервалов создаются виртуальные каналы. Так, «Канал 1» в последовательности циклов образует виртуальный канал 1, предоставляемый паре абонентских систем. Аналогично, из интервалов i , где $i=1,2,..n$, образуется виртуальный канал i .

В случае интегральной коммутации между системами, расположенными в сети (рисунок 15.6), возможны различные тракты взаимодействия. Через физические каналы, соединяющие абонентские системы с узлами коммутации, также могут передаваться повторяющиеся циклы, показанные на рисунке 15.5. Тогда несколько прикладных процессов одной системы могут

одновременно взаимодействовать с группой процессов других абонентских систем.

После того как временные интервалы распределены, осуществляется вторая часть управления коммутацией информации. Оставшиеся интервалы используются для передачи в порядке очереди блоков данных, направляемых любыми абонентскими системами, т.е. выполняется коммутация пакетов. Поскольку картина запросов на коммутацию каналов все время меняется, изменяется и список интервалов времени, оставляемых для коммутации пакетов. Таким образом, при интегральной коммутации уже фактически нет в классическом понимании ни коммутации каналов, ни коммутации пакетов. Здесь присутствует один способ передачи информации с гарантией или без гарантии доставки блоков данных. В отличие от всех остальных способов коммутации информации интегральная коммутация может осуществляться при наличии только одного физического канала. На базе интегральной коммутации предоставляется возможность передавать любые виды информации.

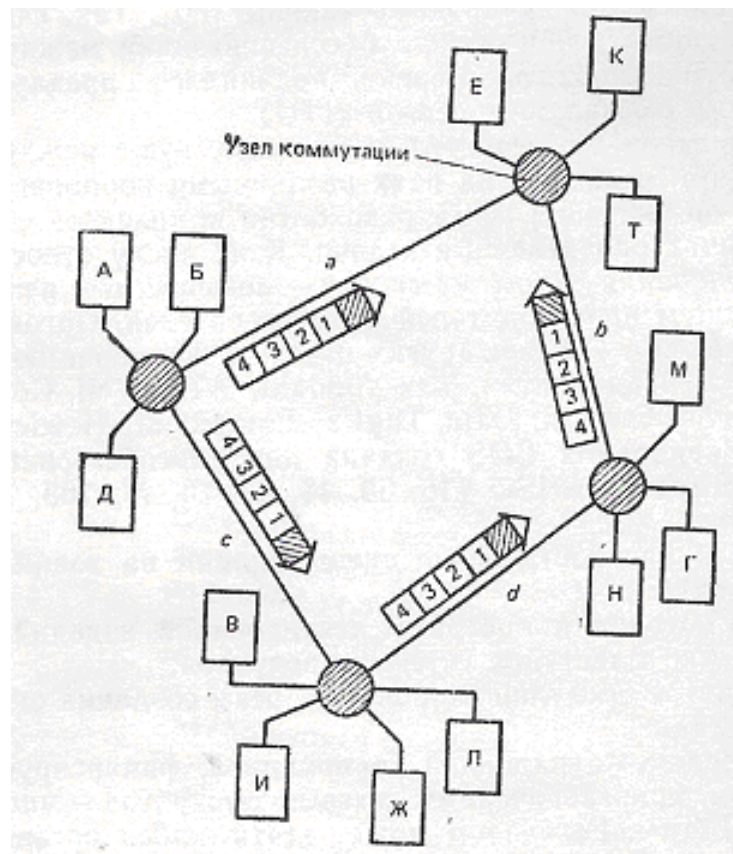


Рисунок 15.6 – Схема организации виртуальных каналов

ЛЕКЦИЯ 16

План лекции:

1. Классификация характеристик качества транспортных услуг вычислительной сети.
2. Характеристики качества обслуживания вычислительной сети.
3. Методы обеспечения качества обслуживания.
4. Классификация трафика вычислительной сети.
5. Алгоритмы управления очередями.

16.1 Классификация характеристик качества транспортных услуг вычислительной сети

Компьютерная сеть представляет собой сложную и дорогую систему, решающую ответственные задачи и обслуживающую большое количество пользователей. Поэтому очень важно, чтобы сеть не просто работала, но работала качественно. Рассмотрим характеристики качества транспортных услуг сети, которые намного проще поддаются формализации, чем характеристики качества информационных услуг. Характеристики качества транспортных услуг отражают такие свойства сети, как *производительности, надежность, безопасность*.

Качество обслуживания связывают с разработкой методов качественной передачи трафика через сеть. Характеристики качества обслуживания объединяет то, что все они отражают отрицательное влияние механизма очередей на передачу трафика, такие как временное снижение скорости передачи трафика, доставку пакетов с переменными задержками и потери пакетов из-за перегрузки буферов коммутаторов.

Все множество характеристик качества транспортных услуг сети можно отнести к одной из следующих групп:

- производительность,
- надежность,
- безопасность,
- характеристики, интересные только поставщику услуг.

Первые три группы соответствуют трем наиболее важным для пользователя характеристикам транспортных услуг – возможности без потерь и перерывов в обслуживании (*надежность*) передавать с заданной скоростью (*производительность*) защищенную от несанкционированного доступа и подмены информацию (*безопасность*).

Поставщик сетевых услуг, стремясь удовлетворить требования пользователей, также уделяет внимание этим характеристикам. В то же время существует ряд важных для поставщика характеристик, которые не интересуют пользователей. Поставщику необходимо найти такой баланс в распределении ресурсов между конкурирующими потоками, чтобы требования всех пользователей были соблюдены. Решение этой задачи включает *планирование* использования ресурсов и *контроль* над их использованием в процессе передачи трафика. Поставщика интересуют те

характеристики ресурсов, с помощью которых он обслуживает пользователей. Например, производительность коммутатора, чтобы знать, какое количество потоков пользователей он может обслужить с помощью данного коммутатора. Пользователя это не интересует. Примером характеристики, интересующей только поставщика, может быть *масштабируемость*, т.е. способность сети работать качественно при значительном увеличении числа пользователей без изменения технологии, применяемой в этой сети.

Рассмотрим еще один способ классификации характеристик – в соответствии с временной шкалой, на которой они определяются.

Долговременные характеристики определяются на промежутках времени от нескольких месяцев до нескольких лет. Их можно назвать характеристиками проектных решений. Примеры таких характеристик: набор моделей и количество коммутаторов в сети, топология и пропускная способность линий связи. Понятно, что полная замена или глубокая модернизация сети связана с большими затратами финансовых средств и времени, поэтому они происходят не слишком часто и влияют на качество сети в течение продолжительного времени.

Среднесрочные характеристики определяются на интервалах от нескольких секунд до нескольких дней. Примеры: средние скорости потоков трафика или средние значения задержек пакетов, определяемые на продолжительном промежутке времени, включающем обслуживание большого количества пакетов. Методы, влияющие на эти характеристики, являются методами определения маршрутов трафика. Маршруты трафика могут быть неизменными в течение часов или дней, если топология сети и параметры трафика остаются постоянными, а каналы и коммутаторы сети не отказывают.

Краткосрочные характеристики относятся к темпу обработки отдельных пакетов и измеряются в микросекундном или миллисекундном диапазонах. Например, время буферизации или время пребывания пакета в очереди коммутатора или маршрутизатора являются характеристиками этой группы. Для анализа и обеспечения требуемого уровня этих характеристик разработано большое количество методов, называемых *методами контроля и предотвращения перегрузок*.

Рассмотрим характеристики качества обслуживания вычислительной сети, представляющие интерес для пользователя.

16.2 Характеристики качества обслуживания вычислительной сети

Производительность компьютерной сети. Будем считать, что сеть работает идеально, если она передает каждый бит информации с постоянной задержкой, равной скорости распространения света в физической среде. Пусть каналы идеальной сети обладают некоторой конечной пропускной способностью, поэтому источник информации передает пакет в сеть не мгновенно, а за некоторое конечное время.

Компьютерная безопасность. Еще одна характеристика сети, относящаяся к характеристикам качества обслуживания – это компьютерная безопасность. При всем своем многообразии средства защиты информации делятся на два больших класса:

– средства компьютерной безопасности предназначены для защиты внутренних информационных ресурсов, находящихся в локальной сети или на отдельном компьютере пользователя;

– средства сетевой безопасности предназначены для защиты информации в процессе ее передачи через сеть.

Такое деление достаточно естественное, т.к. функции по обеспечению безопасности в этих двух случаях существенно различаются. В первом случае нужно защитить все ресурсы, находящиеся внутри локальной сети: аппаратные ресурсы, программные ресурсы, данные. Для этого необходимо контролировать трафик, входящий в сеть из публичной сети, и стараться перекрыть доступ извне для любой информации, которая может навредить. Средства защиты: фильтры, экраны и т.д., которые не пропускают в сеть подозрительную информацию.

В плане обеспечения сетевой безопасности приходится защищать информацию, которая находится вне пределов нашей досягаемости, а в виде пакетов путешествует через сети поставщиков услуг Интернета. В этом случае невозможно отгородиться со своей информацией от мира. На сегодняшний день нет способов 100% защиты. Обеспечение безопасности сводится к тому, чтобы каждое проникновение к информации было контролируемым. Каждый пользователь должен иметь права на доступ к информации, внешним устройствам, на выполнение требуемых действий на каждом из компьютеров сети. Еще один вид опасности – это перехват сообщений и создание ложного трафика в сети. Большая часть средств обеспечения безопасности направлены на борьбу именно с такими нарушениями.

Безопасная информационная система – это система, которая, во-первых, защищает данные от несанкционированного доступа, во-вторых, всегда готова предоставить их своим пользователям, в-третьих, надежно хранит информацию и гарантирует неизменность данных. Таким образом, безопасная система по определению обладает свойствами *конфиденциальности, доступности и целостности.*

Конфиденциальность – гарантия того, что секретные данные будут доступны только тем пользователям, которым этот доступ разрешен.

Доступность – гарантия того, что авторизованные пользователи всегда получат доступ к данным.

Целостность – гарантия сохранности данными правильных значений, которая обеспечивается запретом для неавторизованных пользователей каким-либо образом изменять, разрушать или создавать данные.

К основным средствам сетевой безопасности относятся следующие.

Шифрование – превращение информации из обычного понятного вида в нечитабельный зашифрованный вид. Шифрование должно быть дополнено

дешифрованием, переводящим информацию в первоначальный вид. Шифрование и дешифрование называются *криптосистемой*.

Аутентификация предотвращает доступ к сети нежелательных лиц и разрешает вход для легальных пользователей. Аутентификация данных означает доказательство целостности этих данных, а также факт их поступления именно от того человека, который объявил об этом. Для этого используется механизм *электронной подписи*.

Идентификация заключается в сообщении пользователем систем своего идентификатора, в то время как аутентификация – это процедура доказательства пользователем того, что он является тем, за кого себя выдает, в частности, что ему принадлежит введенный им идентификатор.

Авторизация – процедура контроля доступа легальных пользователей к ресурсам системы с предоставлением каждому из них именно тех прав, которые определены ему администратором.

Аудит – фиксация в системном журнале событий, связанных с доступом к защищенным системным ресурсам.

Технология защищенного канала обеспечивает безопасность передачи данных по открытой транспортной сети за счет:

- взаимной аутентификации абонентов при установлении соединения,
- защиты передаваемых по каналу сообщений от несанкц. Доступа,
- обеспечения целостности поступающих по каналу сообщений.

Основные характеристики качества сети, интересующие поставщика, – расширяемость и масштабируемость.

16.3 Методы обеспечения качества обслуживания

Методы обеспечения качества обслуживания занимают сегодня одно из важнейших мест в арсенале технологий сетей с коммутацией пакетов, т.к. без их применения невозможна работа современных мультимедийных приложений, таких как IP-телефония, видео и радиовещание, интерактивное дистанционное обучение и др. Эти методы оперируют параметрами, характеризующими скорость передачи данных, задержку пакетов и потерю пакетов.

Методы обеспечения качества обслуживания фокусируют внимание на влиянии очередей в коммуникационных устройствах на передачу трафика. В них используются различные алгоритмы управления очередями, резервирования и обратной связи, позволяющие снизить негативное влияние очередей, резервирования и обратной связи.

Очереди являются неотъемлемым атрибутом сетей с коммутацией пакетов. Сам принцип работы таких сетей подразумевает наличие буфера у каждого входного и выходного интерфейсов коммутатора пакетов. Буферизация пакетов во время перегрузок представляет собой основной механизм поддержания пульсирующего трафика, обеспечивающий высокую производительность сетей этого типа. С другой стороны, очереди означают неопределенную задержку при передаче пакетов через сеть, а это главный

источник проблем для чувствительного к задержкам трафика. Так как сегодня операторы пакетных сетей очень заинтересованы в передаче пульсирующего трафика, им необходимы средства обеспечения компромисса между стремлением предельно загрузить свою сеть и выполнением требований качества обслуживания одновременно для всех типов трафика. В этих методах используются различные механизмы, направленные на уменьшение негативных последствий пребывания пакетов в очередях с сохранением в то же время положительной роли очередей. Набор механизмов достаточно широк. Большинство из них учитывает и использует в своей работе факт существования в сети трафика различных типов.

К этим методам примыкают методы инжиниринга трафика. Они служат для управления маршрутами передачи данных, что позволяет обеспечить сбалансированную загрузку всех ресурсов сети и исключить за счет этого переполнение очередей.

16.4 Классификация трафика вычислительной сети

Характеристики качества обслуживания особенно важны, когда сеть передает одновременно трафик разного типа, например, трафик веб-приложений и голосовой трафик. Это связано с тем, что различные типы трафика предъявляют разные требования к характеристикам качества обслуживания. Обычно используют следующий подход. Классифицируют все виды трафика, существующие в сети, относя каждый из них к одному из распространенных типовых видов трафика, а затем добиваются одновременного выполнения определенного подмножества из набора требований для этих типов трафика. В качестве основных критериев классификации были приняты три характеристики трафика:

- относительная предсказуемость скорости передачи данных,
- чувствительность трафика к задержкам пакетов,
- чувствительность трафика к потерям и искажениям пакетов.

С точки зрения предсказуемости скорости передачи данных трафик делится на два больших класса: *поточковый трафик*, *пульсирующий трафик*.

Приложения с поточковым трафиком порождают равномерный поток данных, который поступает в сеть с постоянной битовой скоростью. Трафик представляет собой последовательность пакетов одинакового размера, следующих друг за другом через один и тот же интервал времени.

Приложения с пульсирующим трафиком отличаются высокой степенью непредсказуемости, когда периоды молчания сменяются пульсацией, в течение которой пакеты плотно следуют друг за другом. Трафик характеризуется переменной битовой скоростью. Так интенсивность трафика может падать до нуля, когда файлы не передаются, и повышаться до максимально доступной, ограниченной только возможностями сети, когда передаются файлы.

Можно подсчитать *коэффициент пульсации*, он равен отношению пиковой скорости на каком-либо небольшом периоде времени к средней

скорости трафика, измеренной на длительном промежутке времени. У приложений с пульсирующим трафиком он обычно находится в пределах от 2 до 100, а у потоковых близок к 1. В локальных сетях коэффициент пульсации обычно выше, чем в глобальных, т.к. на магистралях глобальных сетей трафик представляет собой сумму трафиков многих источников, что по закону больших чисел приводит к сглаживанию результирующего трафика.

Чувствительность трафика к задержкам пакетов (по возрастающей):

– асинхронные приложения: практически нет ограничений на время задержки (эластичный трафик), электронная почта;

– интерактивные приложения: задержки могут быть замечены пользователем, но они не сказываются негативно на функциональности приложения, текстовый редактор, работающий с удаленными файлами;

– изохронные приложения: имеется порог чувствительности к вариациям задержек, в случае превышения которого резко снижается функциональность приложения, передача голоса, при превышении порога вариации задержке в 100-150 мс резко снижается качество воспроизводимого голоса;

– сверхчувствительные к задержкам приложения: задержка сводит функциональность приложения к нулю. Приложения, управляющие техническими объектами в реальном времени, при запаздывании управляющего сигнала может произойти авария.

Чувствительность трафика к потерям и искажениям пакетов

Приложения, чувствительные к потерям, – практически все, передающие алфавитно-цифровые данные.

Приложения, устойчивые к потерям, – передающие трафик с информацией об инерционных физических процессах. Небольшое количество потерянных данных можно восстановить на основе принятых. Процент потерянных данных не должен быть выше 1%.

Методы обеспечения качества фокусируют внимание на влиянии очередей в коммуникационных устройствах на передачу трафика. В них используются различные алгоритмы управления очередями, резервирования и обратной связи, позволяющие снизить негативное влияние очередей до приемлемого для пользователей уровня.

Модель теории очередей М/М/1 может использоваться как модель обработки пакетов. 1 означает, что моделируется одно обслуживающее устройство, первая М обозначает тип распределения интервалов поступления заявок (марковское распределение), вторая М – тип распределения значений времени обслуживания (марковское). Модель полезна для понимания основных факторов, влияющих на величину очереди, главный из которых – коэффициент использования ресурса. Поэтому самым простым способом обеспечения требований качества для всех потоков – это работа сети в недогруженном режиме, когда все коммутаторы и каналы работают на 20-30% от своей максимальной производительности. Однако это сводит на нет возможность высокой производительности при передаче пульсирующего трафика.

Для практики единственно приемлемым является вариант работы в нагруженном режиме. Пока все потоки будем делить на два класса: чувствительный к задержкам и эластичный. Постараемся обеспечить для чувствительного трафика коэффициент загрузки каждого ресурса не более 0,2, тогда задержки в каждой очереди будут небольшими и приемлемыми для большинства приложений этого класса. Для эластичного трафика допустим коэффициент загрузки 0,9. Для того чтобы пакеты этого класса не терялись, для них предусматривают буферную память, достаточную для хранения всех пакетов периода пульсации. В сети организуется приоритетное обслуживание различных потоков. Основная идея заключается в следующем: *общая производительности каждого ресурса должна быть разделена между разными классами трафика неравномерно.*

Можно ввести более чем два класса обслуживания и стараться, чтобы каждый класс работал на своем уровне задержек, тем самым повышая характеристики качества обслуживания.

16.5 Алгоритмы управления очередями

1. *Алгоритм FIFO.* В случае перегрузки все пакеты помещаются в одну общую очередь и выбираются из нее в том порядке, в котором поступили. Во всех устройствах с коммутацией пакетов этот алгоритм используется по умолчанию. Достоинства: простота и отсутствие необходимости в конфигурировании. Недостаток: невозможность дифференцированной обработки пакетов различных потоков, все стоят в одной очереди и по порядку обслуживаются.

2. *Алгоритмы приоритетного обслуживания* очень популярны во многих областях вычислительной техники, в частности в ОС, когда одним приложениям нужно отдать предпочтение перед другими при их обработке в мультипрограммной смеси. Весь трафик разбивается на небольшое количество классов, каждому из которых присваивается приоритет. Приоритетное обслуживание обычно применяется для класса трафика, чувствительного к задержкам, имеющего небольшую интенсивность. Тогда обслуживание этого класса не слишком ущемляет остальные классы. Например, голосовой трафик (чувствителен, но его интенсивность обычно не превышает 8-16 Кбит/с).

3. *Взвешенные очереди.* Он гарантирует всем классам трафика определенный минимум пропускной способности. Под весом понимается процент предоставляемой классу трафика пропускной способности от полной пропускной способности выходного интерфейса. С каждой очередью связывается процент пропускной способности ресурса, гарантируемый ему при перегрузках этого ресурса. Для входного потока это – процессор, для выходного – выходной интерфейс. Недостаток: не учитывает требований к задержкам.

4. *Комбинированные алгоритмы.* Обычно в них используется одна приоритетная очередь для чувствительного трафика, а остальные

обслуживаются в соответствии с взвешенным алгоритмом. Им выделяется часть интенсивности ресурса, оставшегося от приоритетной очереди.

16.6 Алгоритмы предотвращения перегрузок

Алгоритмы управления очередями не предотвращают перегрузок, а лишь некоторым образом в условиях дефицита перераспределяют ресурсы между различными потоками. Эти алгоритмы относятся к *механизмам контроля перегрузок*, которые начинают работать, когда сеть уже перегружена. Существует другой класс способов, которые носят название *механизмов предотвращения перегрузок*. Предотвратить перегрузки можно, когда суммарная интенсивность всех потоков меньше пропускной способности этого интерфейса. Этого можно добиться двумя способами: увеличивая пропускную способность интерфейса или уменьшая интенсивность потоков. Первый вариант относится к средствам проектирования и планирования сети.

Рассмотрим второй вариант. Его можно реализовать двумя способами.

1. Используя механизма обратной связи, с помощью которого перегруженный узел сети, реагируя на перегрузку, просит предыдущие узлы, расположенные вдоль маршрута следования потока временно снизить скорость трафика. После того как перегрузка в данном узле исчезнет, он посылает другое сообщение, разрешающее повысить скорость передачи данных.

2. Предварительно резервируя пропускную способность для потоков, протекающих через сеть. Для этого необходима предварительная информация об интенсивностях потоков.

Существует несколько механизмов обратной связи. Они отличаются информацией, которая передается по обратной связи, а также тем, какой тип узла генерирует эту информацию и кто реагирует на эту информацию – конечный или промежуточный узел.

Обратная связь 1. Организована между двумя конечными узлами сети. Радикальный способ, т.к. только конечный узел может снизить скорость поступления информации в сеть. Однако этот способ не относится к методам контроля перегрузок, т.к. он борется с перегрузками узла назначения, а не с перегрузками сетевых устройств.

Обратная связь 2. Организована между двумя соседними коммутаторами. Коммутатор сообщает соседу, находящемуся выше по течению потока, что он испытывает перегрузку, и его буфер заполнился до критической величины. Коммутатор, получивший сообщение, на время снижает интенсивность передачи данных. Для сети в целом это не очень эффективно, т.к. может возникнуть перегрузка в другом ее месте.

Обратная связь 3. Организована между промежуточным коммутатором и узлом-источником. Сообщения обратной связи хотя и передаются несколькими коммутаторами сети в направлении узла-источника, но они на него не реагируют.

Обратная связь 4. Похожа на вариант 1, но все промежуточные узлы реагируют на сообщение, снижают интенсивность передачи данных и могут изменить содержание сообщения.

Обратная связь 5. Передача сообщения о перегрузке узлу назначения, который преобразует его в сообщение обратной связи и отправляет в нужном направлении, т.е. к источнику.

В применяемых методах обратной связи используются следующие основные типы сообщений: *признак перегрузки, максимальная скорость передачи, максимальный объем данных, косвенные признаки.*

Еще одним механизмом предотвращения перегрузок в сети является резервирование ресурсов. Главная идея: ограничить уровень перегрузок некоторой приемлемой величиной. Эта величина должна быть такой, чтобы алгоритмы контроля перегрузки, работающие в коммутаторах сети, справлялись с кратковременными перегрузками и без обратной связи обеспечивали требуемые значения характеристик качества. В случае резервирования с коммутацией пакетов учитывается пульсирующий характер трафика и возможность динамического перераспределения пропускной способности сети между потоками. При резервировании все сетевые устройства вдоль следования потока должны выделить этому потоку некоторую часть пропускной способности своих интерфейсов и производительности процессоров, равную средней требуемой скорости передачи данных потока.

Системы обеспечения качества обслуживания, базирующиеся на резервировании ресурсов, состоят из подсистем нескольких типов: *механизмов обслуживания очередей, протокола резервирования ресурсов, механизмов кондиционирования трафика.*

Механизмов обслуживания очередей используются в периоды временных перегрузок. При этом обычно применяются комбинации приоритетной очереди с очередями с взвешенным обслуживанием.

Протокол резервирования ресурсов нужен для автоматизации процедуры резервирования на всем пути следования некоторого потока.

Механизмы кондиционирования трафика следят за тем, чтобы текущие параметры потоков соответствовали заявленным при резервировании. Это контрольно-пропускные пункты, которые проверяют трафик на входе в коммутатор.

Методы инжиниринга трафика состоят в выборе рациональных маршрутов прохождения потоков через сеть. Выбор маршрутов обеспечивает максимизацию загрузки ресурсов сети при одновременном соблюдении необходимых гарантий в отношении параметров качества обслуживания трафика.

Исходными данными для методов инжиниринга трафика являются:

- характеристики передающей сети, а также производительность ее коммутаторов и линий связи,
- сведения о предложенной нагрузке сети, т.е. о потоках трафика, которые сеть должна передать между своими пограничными коммутаторами.

Задача инжиниринга состоит в определении маршрутов прохождения потоков трафика через сеть, т.е. для каждого потока требуется найти точную последовательность промежуточных коммутаторов и их интерфейсов. При этом маршруты должны быть такими, чтобы все ресурсы сети были нагружены до максимально возможного уровня, а каждый поток получал требуемое качество обслуживания. Наиболее распространенное решение задачи инжиниринга представляется набором маршрутов для заданного множества потоков трафика, для которого все значения коэффициентов использования ресурсов вдоль маршрута следования каждого потока не превышают некоторого заданного порога. Решение можно искать заблаговременно в *фоновом режиме*, зная характеристики сети и потоков. Альтернативное решение – в *оперативном режиме*, поручив коммутаторам сети.