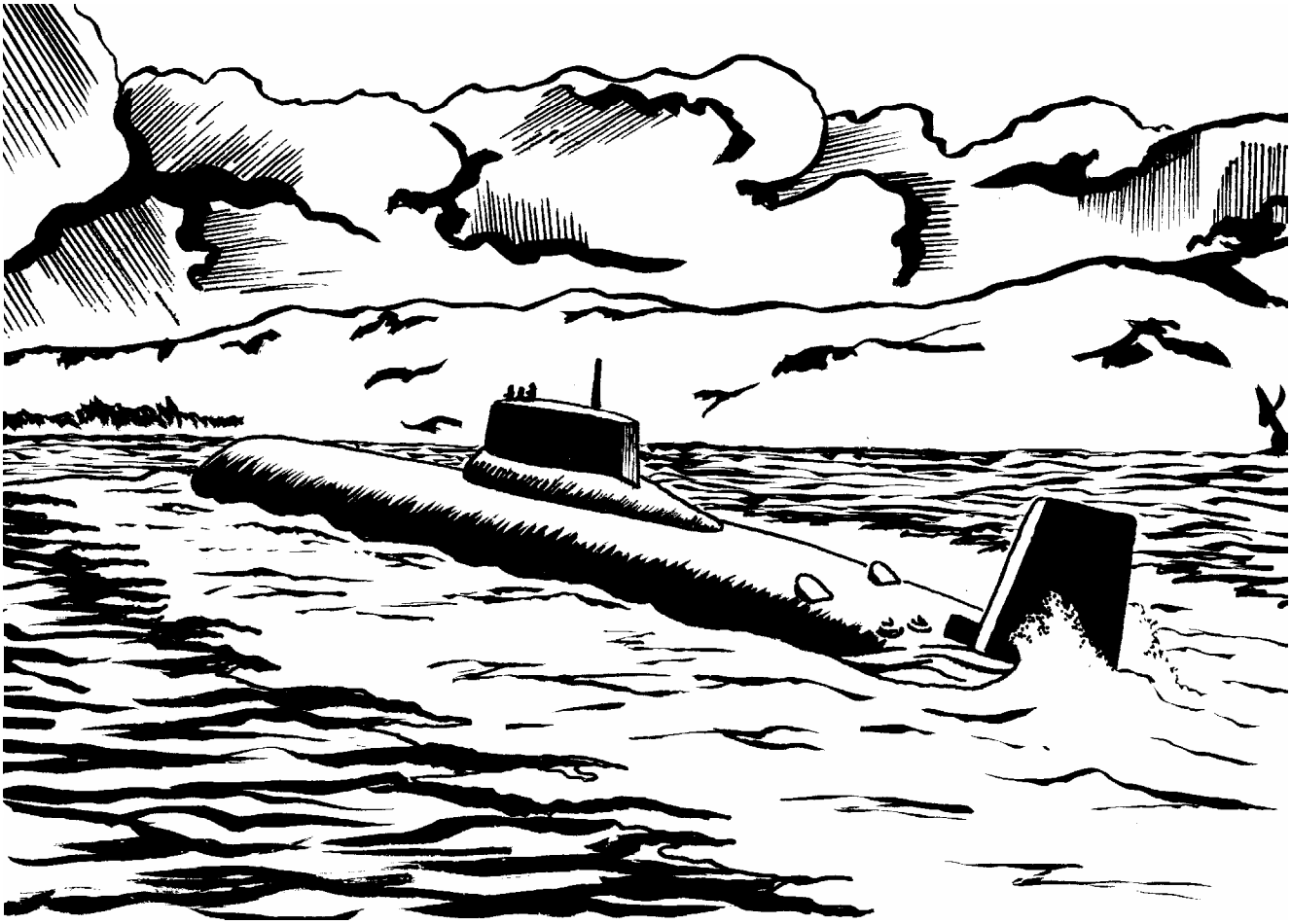
Министерство обороны Российской Федерации
Военно-морской институт радиоэлектроники имени
А.С.Попова

А.В.Лычёв

Распределенные автоматизированные системы



Петродворец
2007 год

Министерство обороны Российской Федерации
Военно-морской институт радиоэлектроники имени А.С.Попова

А.В.Лычёв

Распределенные автоматизированные системы

Учебное пособие

Петродворец
2007 год

УДК 004(075.32)

Лычёв Андрей Владимирович, доцент. «Распределенные автоматизированные системы». Учебное пособие. - Петродворец, изд-во ВМИРЭ, 2007 год

В пособие вошел материал преподаваемых автором в ряде учебных заведений Санкт-Петербурга дисциплин: «Информационное обеспечение, базы и банки данных, защита информации», «Информационные технологии», «Основы построения автоматизированных информационных систем», «Программное обеспечение компьютерных сетей», «Сетевые технологии», «Сети ЭВМ и коммуникаций», «Системы реального времени».

Учебное пособие предназначено для студентов специальностей 230102 «Автоматизированные системы обработки информации и управления», 080507 «Менеджмент организации», 220501 «Управление качеством», 230105 «Программное обеспечение вычислительной техники и автоматизированных систем», 230106 «Техническое обслуживание средств вычислительной техники и компьютерных сетей», а также для всех, кто интересуется проблемами автоматизации управления.

Содержание

Введение 5
Часть I. Автоматизированные системы управления 7
Глава 1. Общие понятия теории систем управления 7
§ 1. Понятие о системах 7
§ 2. Классификация моделей 9
§ 3. Принципы управления 11
Глава 2. Основные понятия автоматизации управления 14
§ 1. Введение в автоматизированные системы управления . 14
§ 2. Основные понятия распределенной обработки информации 18
§ 3. Искусственный интеллект 22
§ 4. Состав и структура автоматизированных систем 25
§ 5. Принципы и стадии разработки АСУ 26
§ 6. Эргономические аспекты проектирования АСУ 29
Глава 3. Программное обеспечение автоматизированных систем 32
§ 1. Основные понятия программного обеспечения АСУ 32
§ 2. Программная документация 36
§ 3. Жизненный цикл программного обеспечения 44
§ 4. Операционная система реального времени QNX 46
Глава 4. Информационное обеспечение АСУ 51
§ 1. Понятие информационного обеспечения АСУ 51
§ 2. Понятие информационной безопасности АСУ 52
§ 3. Защита информации 58
§ 4. Проблемы безопасности автоматизированных систем .. 62
§ 5. Модель угроз безопасности 73
§ 6. Модель противодействия угрозам безопасности 79
§ 7. Архитектура безопасности 82
§ 8. Управление защитой информации 88
§ 9. Использование геоинформационных технологий в АСУ . 91
§ 10. Ситуационный центр 95
Глава 5. Аппаратное обеспечение автоматизированных систем 104
§ 1. Математические модели узлов коммутации 104
§ 2. Сигналы телефонирования 115
§ 3. Сигналы звукового вещания 118
§ 4. Сигналы телеграфирования и передачи данных 118
§ 5. Факсимильные сигналы 119
§ 6. Сигналы телевизионного вещания 121
§ 7. Уровни передачи сигналов в сетях ЭВМ 123
§ 8. Параметры и характеристики сигналов в сетях ЭВМ . 125
Часть II. Распределенные автоматизированные системы 129
Глава 1. Модель «клиент-сервер» 129
Глава 2. Организация связи между процессами 132
§ 1. Удаленный вызов процедур 132

§ 2. Обращение к удаленным объектам 135
§ 3. Связь посредством сообщений 137
§ 4. Связь на основе потоков данных 139
Глава 2. Миграция процессов 140
§ 1. Перенос кода (перенос процессов) 140
§ 2. Программные агенты 142
Глава 3. Именование в распределенных системах 145
§ 1. Понятие сущности 145
§ 2. Пространство имен 146
§ 3. Удаление сущностей, на которые нет ссылок 150
§ 4. Файловые системы ОС РВ QNX 152
Глава 4. Синхронизация в распределенных системах 156
§ 1. Синхронизация с текущим временем 157
§ 2. Синхронизация процессов в распределенных системах 160
§ 3. Взаимное исключение процессов 162
§ 4. Распределенные транзакции 163
§ 5. Механизмы синхронизации процессов в ОС РВ QNX ... 165
Глава 5. Репликация в распределенных системах 167
§ 1. Понятие непротиворечивости 167
§ 2. Непротиворечивость, ориентированная на данные ... 168
§ 3. Непротиворечивость, ориентированная на клиента .. 171
§ 4. Распространение обновлений 172
Глава 6. Надежность распределенной обработки информации 174
§ 1. Основные понятия теории надежности 174
§ 2. Устойчивость вычислительного процесса 181
§ 3. Методы обеспечения надежности 182
§ 4. Физическая избыточность 187
§ 5. Надежная групповая рассылка 193
§ 6. Восстановление после ошибок 197
Приложения: 198
Приложение 1. Классификация сетей ЭВМ 198
§ 1. По принадлежности аппаратных средств. 198
§ 2. По иерархической структуре. 199
§ 3. По топологической структуре. 200
§ 4. По среде передачи данных. 204
§ 5. По виду передаваемых сигналов. 209
Приложение 2. Теоретическая модель сети 216
Приложение 3. Управление сетями ЭВМ 219
Перечень терминов и их определений 225
Литература 246

Введение

Современный рынок учебной и технической литературы в области информационных технологий изобилует практическими руководствами по применению конкретных программных продуктов и оборудования, но, к сожалению, не предлагает никаких фундаментальных источников знаний в этой предметной области. В процессе преподавания дисциплин, связанных с использованием информационных технологий в процессе управления любыми объектами, от организационных до технических, автор столкнулся с полным отсутствием у обучающихся каких- либо элементарных представлений как об основах автоматизации процессов управления, так и о современных технологиях автоматизированного управления, отличных от применяемых в бытовых персональных компьютерах. Изучая книжные рынки Санкт-Петербурга, автору также не удалось найти никакой литературы, которая могла бы оказать помощь студентам в изучении ими преподаваемых автором дисциплин, даже при условии непрофильности этих дисциплин выпускным специальностям студентов. Таким образом, возникла необходимость в создании учебного пособия, не претендующего на глубину исследования заданной предметной области, но дающего достаточное представление об этой области специалистам, призванным использовать автоматизированные системы в своей практической деятельности лишь как прикладной инструмент.

Предлагаемое учебное пособие состоит из двух частей. В первой части рассматриваются общие положения теории систем управления, автоматизации управления и разработки автоматизированных систем. Особое внимание уделено информационному обеспечению АСУ, как основе процесса поддержки принятия решений руководителями любого уровня. Говоря об аппаратном обеспечении автоматизированных систем управления, автор постарался не привязываться к конкретному оборудованию, так как вне зависимости от конкретной модели или поколения аппаратуры, не имея глубоких знаний в области взаимодействия отдельных компонент оборудования между собой, невозможно осознанно грамотно эксплуатировать современные автоматизированные системы. По этой причине в настоящем пособии не будет рассматриваться конкретное аппаратное обеспечение автоматизированных систем управления, а будут даны лишь математические основы функционирования современных распределенных автоматизированных систем управления. Вторая часть пособия целиком посвящена распределенной обработке информации, на основе которой строятся все современные автоматизированные системы управления. Рассматриваемые во второй части технологии нашли свою реализацию в распределенных операционных системах реального масштаба времени, наиболее рас-

пространенным примером которых в промышленности является операционная система QNX. В приложения вынесен материал, напрямую не касающийся тематики настоящего пособия, но без глубокого знания этого материала невозможно адекватно воспринимать рассматриваемые в пособии темы.

Изучая предлагаемое пособие, следует прежде всего отталкиваться от того лекционного материала, который студентам преподносится на аудиторных занятиях. В процессе самостоятельной работы над материалом пособия, необходимо постоянно помнить о наличии терминологического понятийного аппарата, приведенного в перечне терминов и их определений, а также приложений, дающих основу понимания многих механизмов, рассматриваемых в пособии.

Материал пособия соответствует требованиями Государственного образовательного стандарта по специальностям 230102 «Автоматизированные системы обработки информации и управления», 080507 «Менеджмент организации», 220501 «Управление качеством», 230105 «Программное обеспечение вычислительной техники и автоматизированных систем», 230106 «Техническое обслуживание средств вычислительной техники и компьютерных сетей» для преподаваемых дисциплин «Информационное обеспечение, базы и банки данных, защита информации», «Информационные технологии», «Основы построения автоматизированных информационных систем», «Программное обеспечение компьютерных сетей», «Сетевые технологии», «Сети ЭВМ и коммуникаций», «Системы реального времени». Автор надеется, что представленный труд также будет интересен и полезен всем, кто интересуется проблемами автоматизации управления.

Часть I. Автоматизированные системы управления

Глава 1. Общие понятия теории систем управления

§ 1. Понятие о системах

Система – это целенаправленное множество взаимосвязанных элементов любой природы. Общее свойство, объединяющее элементы в систему – направленность элементов на достижение цели.

Внешняя среда – Это множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под её воздействием в условиях рассматриваемой задачи. Воздействия на систему внешней среды называют возмущающими воздействиями. По степени связи с внешней средой системы могут быть замкнутыми (изолированными) и открытыми.

В замкнутой системе любой элемент имеет связи только с элементами самой системы. Замкнутых систем в реальности не существует, так как любая система, находясь в условиях воздействия на нее внешней среды, так или иначе взаимодействует с этой средой. В то же время в исследовательских целях часто бывает необходимо абстрагироваться от воздействий внешней среды и изучить только внутрисистемные связи. Так, например, в школе на уроках физики ставят опыты по термодинамике с использованием калориметра, не учитывая тем самым теплопереноса из внешней среды.

В открытой системе по крайней мере один элемент имеет связь с внешней средой. В зависимости от силы отклонения в функционировании при разрыве или изменении характеристик внешних связей система может быть связана с внешней средой слабо или тесно.

Любая система состоит из каких - либо компонент, называемых подсистемами. Подсистема – это выделенное из системы по определенному правилу целенаправленное подмножество взаимосвязанных элементов любой природы. Если рассматривать подсистемы автономно, без связи с объединяющей их системой, то цели их функционирования часто бывают отличны от целей функционирования самой системы. Так, например, в системе персонального компьютера, целью функционирования которого является обработка информации, имеются такие подсистемы, как:

1. Подсистема блока питания, целью функционирования которой является выработка вторичных напряжений различного номинала;

2. Подсистема дискового накопителя, целью функционирования которой является хранение информации;

3. Другие подсистемы.

Однако будучи собранными вместе в одну систему, в комплексе эти подсистемы реализуют цели обработки информации. По этой причине в теории систем существует правило подсистем, согласно которому подсистемы, непосредственно входящие в одну систему более высокого уровня, действуя совместно, должны выполнять все функции той системы, в которую они входят.

Необходимо заметить, что понятия системы и подсистемы являются условными и зависят от того уровня иерархии, на котором эти понятия рассматриваются. Так, рассматривая в качестве системы персональный компьютер, его подсистемами будут являться блок питания, системная плата, монитор и т.д. Но если в качестве системы рассматривать сеть ЭВМ, в составе которой находятся несколько компьютеров, то рассмотренный нами ранее в качестве системы персональный компьютер будет являться уже не системой, а подсистемой данной сети ЭВМ. В то же время и блок питания компьютера также состоит из каких-то компонент. Опустившись на уровень подсистемы блока питания персонального компьютера, и рассмотрев его в качестве системы, можно также выделить в нем ряд подсистем, являющихся компонентами этого блока питания.

Для того, чтобы понять, как работает система, её надо изучить. Изучение систем производится с помощью системного анализа. Системный анализ – это всестороннее, систематизированное, то есть построенное на основе определенного набора правил, изучение сложного объекта в целом, проводимое для выяснения возможностей улучшения функционирования этого объекта.

Изучать системы можно на макроуровне и на микроуровне. Изучение системы на макроуровне подразумевает изучение взаимодействия системы с внешней средой, причем системы более высокого уровня рассматриваются как часть внешней среды. Изучение системы на микроуровне подразумевает изучение взаимодействия элементов системы между собой.

В результате изучения системы выявляется состав компонент - подсистем, из которых состоит система и связей между этими компонентами. Совокупность связей между элементами системы, отражающих их взаимодействие, называется структурой. Связи между подсистемами могут быть внешними и внутренними. Внутренние связи, в свою очередь, бывают горизонтальными и вертикальными, а внешние связи могут быть входами и выходами.

Горизонтальные связи устанавливаются между одноранговыми подсистемами, не имеющими отношений подчиненности друг

другу. Если подсистемы находятся в подчиненном друг к другу отношении, между ними устанавливаются вертикальные связи. В управлении есть понятие вертикали (например, всем известная вертикаль президентской власти в России). Вертикаль – это множество всех подсистем, вышестоящих и подчиненных по отношению к данной. Все подсистемы, принадлежащие одной вертикали, называются соподчиненными.

При изучении системы необходимо придерживаться определенного плана. Системный анализ включает в себя ряд этапов, придерживаясь которых в качестве плана, можно качественно изучить любую систему:

1. Постановка задачи. Любое действие, в том числе и изучение системы, должно иметь под собой какую-либо цель. Необходимо четко представлять, для чего Вы собираетесь изучать систему.

2. Структуризация системы. Выявление состава подсистем и совокупности связей между ними позволяет наглядно представить структуру системы.

3. Построение модели. Модель – это приближенное, упрощенное представление объекта, процесса или явления, помогающее лучше понять его функционирование и устройство, его характеристики. Таким образом, благодаря модели можно подробно изучить как взаимодействие между элементами системы и внешней средой, так и любой интересующий аспект функционирования системы. Чем более модель соответствует оригиналу, тем она более адекватна оригиналу.

§ 2. Классификация моделей

1. По наличию случайных воздействий в моделируемом процессе:

а) Детерминированные модели. Детерминированные модели предполагают отсутствие случайных воздействий в моделируемом процессе;

б) Стохастические модели. Стохастические модели отображают вероятностные процессы и события в моделируемом процессе;

2. По полноте описания системы:

а) Статические модели. Статические модели описывают поведение объекта в какой либо фиксированный момент времени;

б) Динамические модели. Динамические модели отражают поведение объекта во времени;

3. По сущности исследуемого процесса:

а) Дискретные модели. Дискретные модели описывают дискретные процессы;

б) Непрерывные модели. Непрерывные модели описывают непрерывные процессы;

в) Дискретно-непрерывные модели. Дискретно-непрерывные модели описывают процессы, в которых можно выделить как дискретные, так и непрерывные составляющие;

4. По форме представления объекта:

а) Мысленные модели. Мысленные модели применяются в случае невозможности реализовать объект в заданном интервале времени, или при отсутствии условий, возможных для его физического создания. Мысленные модели бывают трех видов:

- наглядные модели. Наглядные модели создаются на базе представлений человека о реальном объекте и наглядно отображают те или иные свойства объекта-оригинала. При этом не происходит формализации процесса функционирования объекта. Наглядные модели, в свою очередь, также бывают трех видов:

- - гипотетические модели. Гипотетические модели представляют собой гипотезу исследователя о причинно-следственных связях между входом и выходом объекта;

- - аналоговые модели. Аналоговые модели берут в основу существующую наглядную аналогию;

- - макеты. Макеты являются наглядной аналогией объекта;

- символические модели. Символические модели описывают свойства объекта с помощью системы символов. Символические модели бывают двух видов:

- - знаковые модели. Знаковые модели реализуются при использовании условных обозначений отдельных понятий, то есть знаков, а так же определенных операций между этими знаками;

- - языковые модели. Языковые модели реализуются при введении фиксированного набора входящих понятий, лишенных неоднозначности определения;

- математические модели. Математические модели устанавливают соответствия реальному объекту символических высказываний в терминах математической логики с целью исследования полученной математической модели для получения характеристик рассматриваемого объекта. Математические модели, в свою очередь, также бывают трех видов:

- - аналитические модели. В этих моделях процесс функционирования объекта записывается в виде функциональных отношений или логических условий. Аналитическая модель может быть может быть исследована следующими методами:

- - - аналитическими (получение в общем виде явных зависимостей для искомых характеристик);

- - - численными (получение числовых результатов для конкретных начальных условий при неумении решать уравнения в общем виде);

- - - качественными (при отсутствии решения в явном виде находятся некоторые его свойства, например, устойчивость);

- - имитационные модели. Имитационные модели реализуются при помощи средств вычислительной техники;

- - комбинированные модели. Комбинированные модели реализуются при помощи декомпозиции процесса функционирования объекта на подпроцессы. При их изучении по возможности используется аналитическое моделирование, в противном случае имитационное;

б) Реальное моделирование. Реальное моделирование использует возможность исследования различных характеристик либо на реальном объекте целиком, либо на его части;

в) Натурное моделирование. Натурное моделирование проводится полностью на реальном объекте. Натурное моделирование бывают трех видов:

- производственный эксперимент. Производственный эксперимент имеет целью улучшение технологии производства какого-либо изделия;

- комплексные испытания. Комплексные испытания имеют целью выяснение соответствия заявленных производителем и фактических характеристик объекта;

- научный эксперимент. Научный эксперимент предполагает вмешательство человека в исследуемый процесс с целью определения границ его устойчивости;

г) Физическое моделирование. Физическое моделирование проводится на установках, которые сохраняют природу явлений и обладают физическим подобием. Физическое моделирование бывают двух видов:

- моделирование в реальном масштабе времени. Моделирование в реальном масштабе времени производится при соответствии скорости протекания исследуемых процессов возможностям исследователя по наблюдению и регистрации этих процессов;

- моделирование в нереальном масштабе времени. Моделирование в нереальном масштабе времени производится при изучении быстротекущих или вялотекущих процессов (например, при изучении полета артиллерийского снаряда приходится искусственно замедлять течение времени, а при изучении развития какого-либо растения – наоборот, искусственно ускорять течение времени).

§ 3. Принципы управления

Управление – это воздействия, направленные на поддержание или улучшение функционирования управляемого объекта в соответствии с имеющейся целью управления. Управлять можно

по-разному. Одного и того же результата можно достичь многими способами, часть из которых будет сложна и затратна, а другая часть проста и дешева. Поэтому в теории управления существует понятие оптимального управления. Оптимальное управление – это выбор наилучших по некоторому критерию эффективности управляющих воздействий из множества возможных в соответствии с установленной целью управления.

Цель управления определяется внешними по отношению к данной системе факторами. Цели могут быть конечными, частными и промежуточными. Если изучить цели управления с помощью системного анализа, то можно представить некоторую иерархическую структуру, называемую деревом целей. Дерево целей – это результат выделения целей по всем подсистемам с указанием зависимостей между ними.

Для определения оптимальности управления пользуются критериями и показателями эффективности. Критерии эффективности бывают первого и второго рода. Критерий эффективности первого рода представляет собой математическое выражение, позволяющее количественно оценить степень достижения цели системой. Критерий эффективности второго рода представляет собой математическое выражение, позволяющее количественно оценить пути достижения цели.

Сложность и многокритериальность реальных систем часто не позволяют одним математическим выражением оценить оптимальность управления. В таких случаях применяют показатели эффективности. Показатель эффективности – это количественная оценка какого-либо отдельного свойства изучаемого объекта или явления.

Условия определения критерия эффективности:

1. Необходимо определить, с какой точки зрения деятельность системы является эффективной или неэффективной. Один и тот же результат можно интерпретировать как с позитивной точки зрения, так и с негативной;

2. Критерий эффективности должен выражаться числом или вектором (кортежем чисел), то есть быть количественным (по определению);

3. Критерий эффективности должен подчиняться законам статистики. Это необходимо для обеспечения возможности ранжирования системы среди себе подобных;

4. Критерий эффективности должен как можно полнее охватывать деятельность системы. Чем полнее критерий охватывает деятельность системы, тем адекватнее будет оценка этой деятельности;

5. Критерий эффективности должен быть простым (легко вычисляться). Чем проще будет критерий, тем он будет более понятен специалистам и обывателям, часто имеющим пробелы в образовании;

6. Критерий эффективности должен иметь физический смысл. Отвлеченный, не имеющий физического смысла критерий сложно оценивать и позиционировать в структуре системы;

7. Критерий эффективности должен быть нормирован, то есть сравним с идеальным значением. Это позволяет не только ранжировать систему среди себе подобных, но и сравнивать её с идеальной моделью системы;

8. Число компонент векторного критерия должно быть минимально возможным. Увеличение числа компонент векторного критерия приводит к его интеграции и потери физического смысла.

Управление, как целенаправленное воздействие, имеет ряд функций:

1. Планирование, то есть выбор цели системой. Планирование является начальным этапом управления и заканчивается перед началом действий по реализации плана. Планирование бывает стратегическим, определяющим конечные цели системы, и тактическим, определяющим промежуточные цели и траектории движения системы. Планирование включает в себя определение:

а) Конечных и промежуточных целей;

б) Задач, решение которых необходимо для достижения целей;

в) Средств и способов решения этих задач;

г) Требуемых ресурсов, их источников и способов распределения;

2. Организация. Эта функция управления устанавливает постоянные и временные взаимоотношения между всеми элементами системы, определяет порядок и условия их функционирования;

3. Оперативное управление. Данная функция управления обеспечивает функционирование системы в соответствии с намеченным планом. Слово «оперативный» часто используется в управлении (оперативная служба, оперативный дежурный, оперативный уполномоченный и т.д.) и всегда связано с обеспечением функционирования чего-либо в соответствии с намеченным планом;

4. Связь. Это наиболее важная функция управления, объединяющая остальные три функции в единый процесс. Недаром существует поговорка, что потеря связи есть потеря управления. Связь – это передача сведений о состоянии объекта и внешней среды в центры управления системой, взаимообмен информацией между этими центрами, а также между системой и внешней средой.

Глава 2. Основные понятия автоматизации управления

§ 1. Введение в автоматизированные системы управления

Человечество с самого момента своего возникновения пыталось управлять различными процессами своего бытия и окружающего мира. Всем известны великие правители древности, но автоматизировать процессы управления удалось только с появлением соответствующих технических средств. Первые успешные попытки создать средства автоматического управления датируются 18 веком (центробежный регулятор скорости Уатта), но это были только отдельные средства автоматизации управления. Комплексные системы автоматизированного управления появились перед первой мировой войной (начало 20 столетия). Их появление было вызвано развитием морской дальнобойной артиллерии, когда орудийный снаряд улетал за горизонт и наводчик не имел возможности визуально оценивать результаты стрельбы. Потребовался выносной наблюдательный пункт и системы наводки орудия не по полярным координатам относительно самого орудия (пеленг, дистанция), а по географическим координатам цели (широта, долгота) или полярным координатам относительно выносного наблюдательного пункта.

Техническими средствами, позволившими автоматизировать процесс наводки орудия, явились электромеханические счетно-решающие устройства, построенные на основе синусно-косинусных вращающихся трансформаторов и конических передач с переменным передаточным числом. Такие устройства позволяли в реальном масштабе времени решать простейшие дифференциальные уравнения и производить простейшие арифметические вычисления по алгоритму, конструктивно заложенному при проектировании этих устройств.

Таким образом, появление в начале 20 столетия автоматизированных систем на основе электромеханических счетно-решающих устройств ознаменовало начало первого этапа развития автоматизированных систем управления. Автоматизированная система управления (АСУ) – это человеко-машинная система, обеспечивающая автоматизированный сбор и обработку информации, и выработку на её основе рекомендаций для поддержки принятия управленческих решений и (или) управляющих воздействий на физические объекты. Благодаря своей простоте, компактности и надежности, электромеханическим счетно-решающим устройствам была уготована долгая жизнь вплоть до конца 80-х годов прошлого столетия, когда им на смену массово пришли персональные ЭВМ.

В начале 40-х годов прошлого века в Соединенных штатах Америки появились первые электронно-вычислительные машины. В нашей стране они появились позже, после второй мировой

войны, в конце 40-х - начале 50-х годов прошлого столетия. Их появление ознаменовало начало второго этапа развития автоматизированных систем управления, а именно автоматизированных систем управления на основе ЭВМ первого поколения. ЭВМ первого поколения были построены на электронных лампах, имели низкое по нынешним меркам быстродействие (около 300 коротких операций в секунду), потребляли огромное количество электроэнергии, выделяли не менее огромное количество тепла. Стоимость таких ЭВМ была очень высока и позволить себе их иметь могли только крупные федеральные организации и институты. Но главное, что ограничивало их применение – это ручной ввод и вывод информации, а также ручное программирование в машинных кодах, что требовало высокой квалификации обслуживающего персонала, больших трудозатрат и было весьма непроизводительно. По этой причине подобные ЭВМ применялись только в системах организационного управления для поддержки принятия решений руководителями федерального уровня.

Второй этап развития АСУ закончился в конце 50-х - начале 60-х годов прошлого столетия с появлением ЭВМ второго поколения, построенных на основе дискретных полупроводниковых элементов (транзисторов и диодов). Основным отличием ЭВМ второго поколения от ЭВМ первого поколения, кроме элементной базы, стало появление в качестве устройств ввода-вывода терминалов, представлявших собой монитор и клавиатуру, подключенных к центральному процессору. К одной ЭВМ можно было подключить в режиме разделения времени большое количество таких терминалов. Ряд терминалов можно было заменить устройствами для автоматизированного сбора информации и управления физическими объектами. Третий этап развития АСУ, а именно автоматизированных систем управления на основе терминальных устройств ЭВМ второго поколения, начавшись на рубеже 50-х - 60-х годов двадцатого века, продолжался также до конца 1980-х годов, то есть до массового появления персональных ЭВМ.

Если эволюцию первых трех этапов развития автоматизированных систем управления обусловила смена элементной базы, то эволюцию четвертого и пятого этапов развития АСУ обусловило уже развитие информационных технологий. Появление сетей ЭВМ и их коммерциализация в середине 70-х годов прошлого века дало толчок к возникновению и развитию технологий распределенной обработки информации, давшим начало четвертому этапу развития АСУ, а именно распределенных автоматизированных систем управления.

Появление примерно в то же время и развитие средств и методов искусственного интеллекта положило начало пятому этапу развития АСУ – автоматизированных систем управления

на основе средств и методов искусственного интеллекта. Четвертый и пятый этапы развития АСУ, интегрируясь друг в друга, продолжаются и в настоящее время.

Классификация автоматизированных систем управления:

1. По уровню управления:

а) Общегосударственная автоматизированная система управления. Общегосударственная автоматизированная система управления стоит на верхнем уровне иерархии управления государством и вовсе не ограничивается рамками Московского Кремля или Белого Дома, а включает в себя в качестве подсистем нижестоящие уровни управления;

б) Отраслевая автоматизированная система управления. Отраслевая автоматизированная система управления – это система управления уровня федерального министерства или ведомства, крупной отрасли промышленности или сельского хозяйства. Отраслевая автоматизированная система управления в качестве подсистемы входит в общегосударственную автоматизированную систему управления;

в) Территориальная автоматизированная система управления. Это система управления уровня территориального образования – федерального округа, края, области, города и т.п. По уровню иерархии территориальная автоматизированная система управления равнозначна отраслевой автоматизированной системе управления и также является подсистемой общегосударственной автоматизированной системы управления;

г) Автоматизированная система управления производственным объединением (фирмой). Автоматизированная система управления производственным объединением (фирмой) входит в качестве подсистемы в отраслевую автоматизированную систему управления, и, опосредованно через неё, в общегосударственную автоматизированную систему управления;

д) Автоматизированная система управления предприятием. Автоматизированная система управления предприятием входит в качестве подсистемы в автоматизированную систему управления производственным объединением (фирмой), и, опосредованно через неё и отраслевую автоматизированную систему управления, в общегосударственную автоматизированную систему управления;

е) Автоматизированная система управления подразделением. Автоматизированная система управления подразделением (отделом, цехом, производственным участком и т.д.) является подсистемой автоматизированной системы управления предприятием. Таким образом, общегосударственная автоматизированная система управления фактически распространяется на все объекты, от которых зависит экономическое и политическое процветание государства;

2. По объекту управления:

а) Автоматизированная система управления физическими объектами. Как правило, этот вид автоматизированных систем является подсистемой автоматизированной системой управления подразделением и осуществляет непосредственное управление производственными процессами. На производстве этот вид систем получил название АСУ ТП (Автоматизированные Системы Управления Технологическими Процессами);

б) Автоматизированная система организационного управления. Все остальные автоматизированные системы, не управляющие непосредственно физическими объектами, являются автоматизированными системами организационного управления, так как осуществляют поддержку принятия решений руководителями различного ранга;

в) Интегрированная автоматизированная система управления. В интегрированную автоматизированную систему управления в качестве подсистем входят как автоматизированные системы организационного управления, так и автоматизированные системы управления физическими объектами;

3. По назначению:

Любая автоматизированная система управления создается под конкретное применение и по этой причине носит ярко выраженные признаки индивидуальности. По указанному классификационному признаку можно выделить огромное количество систем, столько, сколько существует конкретных применений АСУ (напр.: плановых расчетов, материально-технического снабжения, управления ядерным реактором и т.д. и т.п.);

4. По характеру функционирования:

а) Автоматизированная система управления непрерывного типа. Этот вид автоматизированных систем функционирует, не выключаясь, в течение всего жизненного цикла управляемого объекта (например: система управления ядерным реактором включается в момент его физического пуска, работает непрерывно все время жизни реактора и выключается спустя много лет только при его демонтаже);

б) Автоматизированная система управления дискретного типа. Такие системы управления функционируют эпизодически по мере необходимости (пример: система управления кадрами предприятия – включается в начале рабочего дня, выключается по его окончании, в выходные и праздничные дни не работает);

в) Автоматизированная система управления непрерывно-дискретного типа. В автоматизированную систему управления непрерывно-дискретного типа в качестве подсистем входят как автоматизированные системы управления непрерывного типа, так и автоматизированные системы управления дискретного типа.

Разберем подробнее четвертый и пятый этапы развития автоматизированных систем управления, а именно распределенные автоматизированные системы управления и автоматизированные системы управления на основе средств и методов искусственного интеллекта.

§ 2. Основные понятия распределенной обработки информации

Распределенная система – это сеть ЭВМ, ресурсы которой представляются пользователям рабочих станций сети как виртуальная ЭВМ с неограниченными ресурсами. Сеть ЭВМ – это две или более электронно-вычислительные машины, соединенные между собой для передачи информации.

Первые сети передачи данных появились в шестидесятые годы прошлого столетия и использовались для связи терминалов удаленных рабочих мест с большими ЭВМ первых поколений. Самая популярная из используемых сегодня физических сетевых архитектур Ethernet была разработана в середине шестидесятых годов прошлого века в Гавайском университете и называлась сеть ALOHA. В 1972 году Роберт Меткалф и Дэвид Боффс реализовали в корпорации Xerox сетевую архитектуру на этих принципах, а в 1975 году выпустили первый промышленный продукт Ethernet. В 1977 году в Datapoint Corporation была разработана сетевая технология ARCnet, а в 1983 году в корпорации Macintosh – технология Apple Talk, встраиваемая в каждый компьютер этой фирмы. Дальнейшее развитие сетевых технологий шло по пути увеличения пропускной способности и повышения надежности обмена информацией, глобализации сетевых структур. Результатом стало появление качественно новой категории сетей ЭВМ – цифровых сетей интегрального обслуживания. Цифровая сеть интегрального обслуживания – это совокупность информационно-технологических методов и аппаратно-программных средств доставки информации территориально удаленным пользователям, позволяющая на единой цифровой основе обеспечить различные виды информационных услуг. Такими услугами могут быть не только традиционный обмен данными и программами, но и передача различных видов аудио- и видеоинформации в реальном масштабе времени. В сеть ЭВМ кроме самих компьютеров могут непосредственно подключаться аппаратные ресурсы, называемые общими устройствами или разделяемыми устройствами. Оба эти понятия эквивалентны. ЭВМ и общие (разделяемые) устройства, подключенные в сеть, являются узлами сети.

В распределенной системе от пользователя скрыты:

1. Физическое местоположение ресурсов сети;
2. Способы связи между ресурсами в сети;

3. Организация взаимодействия между ресурсами сети.

Идеальная распределенная система с точки зрения конечного пользователя ведет себя как классическая однопроцессорная локальная ЭВМ. Основная задача распределенной системы заключается в облегчении пользователям доступа к удаленным ресурсам и обеспечении бесконфликтного их совместного использования. Выполнение основной задачи достигается путем реализации следующих свойств распределенной системы:

1. Прозрачность – свойство сокрытия факта того, что процессы и ресурсы физически распределены по различным ЭВМ сети. Исторически свойством прозрачности обладали материалы, предназначенные для использования в окнах зданий в качестве преграды неблагоприятным атмосферным воздействиям. Такие преграды должны обеспечивать проникновение солнечного света в помещения и, одновременно, не мешать беспрепятственному обзору из помещения обстановки на улице. Для выполнения этих требований необходимо сокрытие факта наличия преграды в оконном проеме, что с успехом выполняет такой прозрачный материал, в частности, как стекло.

Виды прозрачности:

а) Прозрачность доступа. Прозрачность доступа призвана скрыть разницу в представлении данных и в способах доступа пользователей к ресурсам;

б) Прозрачность местоположения. Прозрачность местоположения призвана скрыть от пользователей, где именно физически расположен в системе нужный им ресурс;

в) Прозрачность переноса. Прозрачность переноса призвана скрыть от пользователей факт перемещения ресурса в другое место системы;

г) Прозрачность смены местоположения. Прозрачность смены местоположения призвана скрыть от пользователей факт перемещения ресурса в процессе обработки в другое место системы. Основное отличие этого вида прозрачности от прозрачности переноса состоит в том, что прозрачность переноса не требует функционирования ресурса в момент его переноса в другое место системы. Прозрачность смены местоположения, напротив, скрывает факт перемещения ресурса в другое место системы именно в процессе обработки (например: при перемещении мобильной базы данных из одной соты в другую);

д) Прозрачность репликации. Прозрачность репликации призвана скрыть от пользователей тот факт, что в системе существует несколько копий ресурса;

е) Прозрачность параллельного доступа. Прозрачность параллельного доступа призвана скрыть факт возможного совместного использования ресурса несколькими конкурирующими пользователями;

ж) Прозрачность отказа. Прозрачность отказа призвана скрыть от пользователей факт отказа и восстановления системы;

з) Прозрачность сохранности. Прозрачность сохранности призвана скрыть от пользователей месторасположение информационных ресурсов – в оперативной памяти или на долговременных носителях;

2. Открытость – свойство стандартизации доступа к ресурсам системы.

Характеристики открытости системы:

а) Способность к взаимодействию. Способность к взаимодействию характеризует, насколько две реализации систем или их компонент от разных производителей в состоянии совместно работать, полагаясь только на то, что их интерфейс соответствует стандарту;

б) Переносимость. Переносимость характеризует, насколько прикладная программа, разработанная для одной распределенной системы, может без изменения выполняться в другой распределенной системе, реализуя одни и те же интерфейсные средства;

в) Гибкость. Гибкость характеризует, насколько легко конфигурируются системы, состоящие из различных компонент от разных производителей;

3. Масштабируемость. Масштабируемость – это свойство расширения системы.

Виды масштабируемости:

а) Масштабируемость по размеру. Масштабируемость по размеру определяет легкость подключения дополнительных пользователей и ресурсов. Проблема масштабируемости по размеру заключается в централизации служб, данных и алгоритмов;

б) Пространственная масштабируемость. Пространственная масштабируемость определяет легкость разнесения пользователей и ресурсов в пространстве. Проблема пространственной масштабируемости заключается в синхронизации процессов;

в) Административная масштабируемость. Административная масштабируемость определяет легкость в управлении множеством независимых компонент системы. Проблема административной масштабируемости заключается в безопасности доступа к ресурсам.

Технологии масштабирования:

а) Сокрытие времени ожидания связи при пространственном масштабировании;

б) Распределение. Распределение – это разбиение компонентов ресурсов на мелкие части с последующим разнесением этих частей по системе;

в) Репликация – создание копий ресурса его владельцем. Проблема репликации заключается в сохранении непротиворечивости копий ресурса.

Аппаратно распределенная система представляет собой многомашинный вычислительный комплекс, построенный на основе сети ЭВМ. Логическая структура многомашинного вычислительного комплекса (сети ЭВМ) может быть шинная или коммутируемая звездообразная. Сети с кольцевой топологией рассматриваются как частный случай шинной структуры.

Аппаратно распределенные системы могут быть двух видов:

1. Гомогенные распределенные системы – используют одну сетевую технологию на ЭВМ с одинаковыми аппаратными платформами;

2. Гетерогенные распределенные системы – используют одновременно несколько сетевых технологий на ЭВМ с одинаковыми или различными аппаратными платформами.

Программно распределенная система представляет собой операционную систему, выступающую как менеджер ресурсов виртуальной ЭВМ, представляемой пользователю. Основная цель распределенной операционной системы – сокрытие тонкостей управления аппаратным обеспечением, одновременно используемым многими процессами. Рассмотрим различия в функционировании локальной операционной системы, в качестве которой представляется распределенная операционная система, сетевой операционной системы, в качестве которой распределенная операционная система физически является, и собственно самой распределенной операционной системы.

Локальная операционная система осуществляет управление ресурсами одной ЭВМ.

Сетевая операционная система обеспечивает доступ к ресурсам удаленных ЭВМ сети и предоставляет ресурсы своей ЭВМ удаленным пользователям.

Распределенная операционная система работает с точки зрения пользователя как локальная операционная система ЭВМ с неограниченными ресурсами. Для обеспечения такой работы к существующим службам сетевых операционных систем добавляются программные средства, называемые системами промежуточного уровня. Под сетевой службой понимается сетевой компонент, реализующий некоторый набор услуг (напр.: файловая служба, служба печати и т.п.). Добавление промежуточного уровня видоизменяет классическую модель OSI, так как протоколы промежуточного уровня фактически подменяют протоколы сеансового и представительного уровней модели OSI и иерархически располагаются между прикладным и транспортным уровнями.

§ 3. Искусственный интеллект

Искусственный интеллект – это комплексное научно-техническое направление, имеющее целью создание и применение программно-аппаратных средств, позволяющих моделировать процессы человеческого мышления и обеспечить диалог с ЭВМ на языке, естественном для человека.

Искусственный интеллект призван решить следующие задачи:

1. Оказание помощи человеку, вплоть до его подмены, при работе в монотонно-однообразных или особо опасных условиях, например, в условиях длительного полета к неизвестным планетам или радиоактивного заражения;

2. Создание человеко-машинных систем при управлении объектами в быстроменяющейся обстановке. Динамика поведения ряда современных объектов, например, боевого самолета в воздушном бою, столь быстротечна, что превышает возможности человека, даже хорошо тренированного, адекватно управлять таким объектом. В этой ситуации на помощь пилоту приходят средства искусственного интеллекта, являющиеся неотъемлемой частью любой современной системы управления боевым самолетом;

3. Решение сложных многокритериальных задач в автоматизированных системах различного назначения. Для принятия управленческого решения руководитель любого уровня должен проанализировать большое количество взаимоувязанных факторов, влияющих на правильность принятия такого решения. Психологи утверждают, что человек со средними способностями без применения каких-либо средств поддержки принятия управленческих решений способен адекватно взаимоувязать не более трех – пяти факторов. Применение средств искусственного интеллекта позволяет практически любому руководителю принимать взвешенные многокритериальные решения на основе анализа большого количества факторов;

4. Обеспечение процесса выработки политических решений. Политика – это такая же многокритериальная управляющая система, как и любая функция управления. Основное отличие политического управления от управления физическими объектами и организациями состоит в сложности сбора и формализации информации, её неполнотой, неоднозначностью и противоречивостью.

Направления исследований и применения средств и методов искусственного интеллекта:

1. Моделирование процесса функционирования головного мозга человека. Это глобальное направление исследований характеризуется достаточно скромными успехами вследствие большой сложности и разнообразия стилей мышления и поведе-

ния человека в зависимости от возраста, уровня образования, среды обитания и воспитания, а также множества других факторов, делающих людей непохожими друг на друга и непонимающими друг друга, даже если они разговаривают на одном языке;

2. Универсальный решатель задач. В автоматизированных системах управления часто требуется решать не только и не столько алгоритмические задачи, сколько задачи, не имеющие алгоритмического решения. Средства искусственного интеллекта позволяют максимально расширить круг решаемых системой управления задач, вне зависимости от того, решаются эти задачи на алгоритмическом, или на эвристическом уровне;

3. Представление знаний в ЭВМ. Под знаниями в автоматизированных системах управления понимаются формализованные оценки экспертов на конкретные события в своей предметной области. Соответственно, эксперт - это человек, являющийся признанным специалистом в своей предметной области и на основании собственных знаний и практического опыта способный решать сложные задачи, относящиеся к этой предметной области;

4. Экспертные системы. Экспертная система – это комплекс программных средств для выработки рекомендаций по решению трудноформализуемых задач в условиях дефицита времени, противоречивой и недостоверной информации о внешней среде и в непредсказуемых ситуациях на основе обобщенного коллективного опыта экспертов, хранящегося в памяти ЭВМ. Под трудноформализуемой задачей в данном случае понимается такая задача, которая:

а) Не может быть задана в числовой форме;

б) Цели задачи не могут быть выражены в терминах точно определенной целевой функции (расплывчатость целей);

в) Алгоритмического решения задачи не существует;

г) Данные и знания динамически изменяются в процессе решения задачи;

д) Исходные данные и знания о предметной области отличаются неоднозначностью, неполнотой и противоречивостью;

5. Самообучение ЭВМ. Результаты этого направления исследований в настоящее время широко применяются в программных продуктах различного назначения, когда после многократного повторения пользователем каких-либо действий, система запоминает эту последовательность и в дальнейшем предлагает её в качестве одной стандартной операции;

6. Понимание естественных человеку языков. Благодаря бурному развитию мультимедийных вычислительных средств, все большее распространение получают системы распознавания и синтеза человеческой речи, на основании которых создаются системы голосового управления различного назначения;

7. Моделирование органов чувств. Это направление исследований зародилось ещё в пятидесятые годы прошлого столетия, когда появились системы, моделирующие органы чувств, называемые персептронами. Производительность ЭВМ первых поколений не позволила развивать данное направление, оно было признано бесперспективным и закрыто. Появление высокопроизводительных микропроцессоров дало возможность вернуться к моделированию органов чувств – появились электронные «носы», «глаза», «уши» и другие органы чувств, позволяющие автоматически идентифицировать обнаруженные с их помощью объекты;

8. Интеллектуальные роботы. Робот – это программируемая техническая система, способная к автономному самоуправлению и выполнению достаточно сложных операций в пространстве и времени, воспринимающая и приспосабливающаяся к изменениям во внешней среде. Говоря о роботах, не следует их путать с системами дистанционного управления. Часто в средствах массовой информации в сюжетах о разминировании каких-либо объектов рассказывается о так называемых «роботах», которыми из специального укрытия управляет специалист-сапер. Согласно вышеприведенному определению, такая система роботом не является, так как управляется, хотя и дистанционно, человеком. Классическим примером робота может служить межпланетная станция, летящая, например, к Марсу. Конечно, на её борт поступают управляющие воздействия с Земли, но благодаря тому, что радиосигнал с Земли до Марса и обратно распространяется несколько десятков минут, в случае возникновения каких-либо непредвиденных ситуаций центр управления полетом ничем помочь не успеет. По этой причине система управления межпланетной станцией должна обладать признаками искусственного интеллекта и по своей сути являться интеллектуальным роботом, чтобы быть способной самостоятельно принимать адекватные решения в случае возникновения в процессе её функционирования разного рода непредвиденных ситуаций.

Говоря о представлении знаний в автоматизированных системах управления необходимо отметить сложность этого процесса:

1. Большинство экспертов решают сложные задачи в своей предметной области на подсознательном уровне, и будучи вырванными из реальной рабочей обстановки, в которой они находились, решая эти задачи, эксперты не могут воспроизвести те действия, которые они абсолютно адекватно совершали, находясь в этой обстановке;

2. Бинарность работы вычислительного комплекса АСУ. Формализация оценок экспертов заключается в том, что каждая из этих оценок должна иметь бинарное представление (да –

нет, правда – ложь, ноль – единица и т.п.), однако далеко не каждое действие можно представить в бинарном виде.

Вышеназванные проблемы решают специалисты по методам формализованного представления знаний, называемые когнитологами. Когнитологи работают с экспертами, которые часто даже не представляют, что такое экспертная система, и, задавая экспертам определенным образом сформулированные вопросы, представляют их знания в виде, пригодном для обработки с помощью вычислительной техники.

Основой экспертной системы является множество правил, описывающих заданную предметную область. Каждое из этих правил представляет собой так называемую продукцию Поста, то есть правило, порождающее другое правило. Множество таких правил сводится в единую систему, позволяющую выводить новые правила на основе последовательного применения уже имеющихся. Продукция Поста – это принцип взаимооднозначного однонаправленного соответствия между фактами, выраженными каким-либо формализованным способом. Продукция Поста представляет собой высказывание, удовлетворяющее двум условиям:

1. В данном высказывании можно сказать, истинно оно или ложно;
2. Существует механизм проверки истинности или ложности данного высказывания.

Разработка экспертной системы является сложным, длительным и дорогостоящим процессом. Количество правил, содержащихся в экспертных системах различной стадии готовности, и сроки их разработки представлены в таблице:

Стадии готовности экспертной системы	Кол-во правил	Срок разработки
Демонстрационный прототип	до 100	3 мес. – 1 год
Исследовательский прототип	до 500	1 – 2 года
Действующий прототип	до 1000	2 – 3 года
Промышленная система	до 1500	2 – 4 года
Коммерческая система	до 3000	3 – 6 лет

§ 4. Состав и структура автоматизированных систем

Любая автоматизированная система управления состоит из двух частей – функциональной части и обеспечивающей части. Функциональная часть АСУ – это комплекс административных, организационных и математических методов, обеспечивающих решение задач поддержки принятия управленческих решений и управления физическими объектами. Подсистемы, входящие в функциональную часть автоматизированной системы, называют функциональными подсистемами АСУ. Ввиду того, что состав и структуру функциональной части системы определяет её назна-

чение, количество и предназначение функциональных подсистем АСУ может быть весьма разнообразным и соответствовать конкретным целям функционирования автоматизированной системы управления.

Обеспечивающая часть автоматизированной системы управления состоит из пяти видов обеспечения:

1. Информационное обеспечение АСУ. Информационное обеспечение АСУ – это совокупность единой системы классификации и кодирования технико-экономической информации, унифицированных систем документации и массивов информации, используемых в автоматизированных системах управления;

2. Лингвистическое обеспечение АСУ. Лингвистическое обеспечение АСУ – это совокупность научно-технических терминов и других языковых средств, используемых в автоматизированных системах управления, а также правил формализации естественного языка;

3. Техническое (аппаратное) обеспечение АСУ. Техническое (аппаратное) обеспечение АСУ – это комплекс технических средств, предназначенных для обеспечения работы автоматизированной системы управления;

4. Математическое обеспечение АСУ. Математическое обеспечение АСУ – это совокупность математических методов, моделей и алгоритмов для решения задач обработки информации в автоматизированных системах управления;

5. Программное обеспечение АСУ. Программное обеспечение АСУ – это совокупность программ и программных документов для реализации целей и задач автоматизированных систем управления. Необходимо заметить, что до введения ныне действующих стандартов на терминологию в сфере систем обработки информации ГОСТ 15071-90 и ГОСТ 19781-90, программное обеспечение отождествлялось с математическим. По этой причине в литературе издания до 1990 года под математическим обеспечением понимаются как математические методы, модели и алгоритмы, так и совокупность программ для реализации целей и задач АСУ.

Подсистемы, входящие в обеспечивающую часть АСУ, называют обеспечивающими подсистемами автоматизированной системы управления.

§ 5. Принципы и стадии разработки АСУ

Любой специалист в области автоматизированных систем управления в процессе своей профессиональной деятельности так или иначе участвует в разработке АСУ. Для этого совсем не обязательно являться сотрудником какой-либо проектной организации. Сложность и многоаспектность функционирования автоматизированных систем управления вовлекает в процесс их

разработки не только непосредственно разработчиков, но и тех, кто эксплуатирует эти системы, персонал предприятий-заказчиков. По этой причине, любой специалист в сфере автоматизации управления должен представлять, как происходит процесс разработки автоматизированных систем управления.

Принципы разработки автоматизированных систем управления:

1. Научно-технической основой разработки автоматизированной системы управления должен являться системный анализ, охватывающий как саму систему, так и внешнюю среду;

2. Разработка и внедрение АСУ должна находиться в ведении высшего руководителя той организации, для которой она разрабатывается;

3. Эффективность создаваемой автоматизированной системы управления обеспечивается введением новых задач, не использовавшихся в ранее применявшихся технологиях обработки информации;

4. В процессе разработки АСУ необходимо предусмотреть адаптивность системы к изменениям во внешней среде;

5. В разрабатываемой автоматизированной системе управления необходимо обеспечить согласованность пропускной способности отдельных частей системы;

6. При разработке АСУ необходимо использовать опыт предыдущих разработок;

7. Во вновь разрабатываемой автоматизированной системе необходимо автоматизировать все процессы, связанные с движением информации;

8. Во вновь создаваемой АСУ необходимо обеспечить однократный ввод данных для всех решаемых задач;

9. Разрабатываемая автоматизированная система управления должна обладать повышенной надежностью и живучестью;

10. При разработке АСУ необходимо предусмотреть поэтапный ввод системы в эксплуатацию.

Стадии разработки автоматизированных систем управления:

1. Предпроектная стадия:

а) Обследование – это определение в самом общем виде основных целей и ограничений разрабатываемой системы, возможностей повышения эффективности управления при внедрении автоматизированной системы управления;

б) Подготовка технико-экономического обоснования. Технико-экономическое обоснование производится на основе:

- анализа организационной и функциональной структуры автоматизируемого объекта;

- анализа технико-экономических характеристик автоматизируемого объекта;

- исследования материальных потоков автоматизируемого объекта;

- анализа потоков и состава информации между подразделениями и внутри них;

- анализа методов планирования и учета;

в) Подготовка технического задания. Техническое задание – это официальный документ, определяющий требования к создаваемой системе;

2. Стадия разработки проектов:

а) Техническое проектирование. Техническое проектирование начинается с разработки эскизного проекта, представляющего собой документированное описание вариантов предлагаемой системы управления. На основании эскизного проекта на данном этапе разрабатывается технический проект. Технический проект – это комплект технической документации, содержащий общесистемные проектные решения, алгоритмы решения задач, предварительную оценку экономической эффективности АСУ и примерный перечень мероприятий по подготовке объекта к внедрению;

б) Рабочее проектирование. Исходной информацией для рабочего проектирования является технический проект, на основании которого на данном этапе разрабатывается рабочий проект. Рабочий проект – это комплект технической документации, содержащий уточненные и детализированные общесистемные проектные решения, программы и инструкции по решению задач, уточненную оценку экономической эффективности АСУ и утвержденный перечень мероприятий по подготовке объекта к внедрению;

3. Ввод в эксплуатацию:

а) Опытная эксплуатация. Опытная эксплуатация проводится совместно специалистами предприятий разработчика и заказчика и имеет целью выявление ошибок разработки и накопление опыта эксплуатации системы;

б) Промышленная эксплуатация. Промышленная эксплуатация проводится специалистами предприятия-заказчика и имеет целью применение системы по прямому предназначению. В процессе промышленной эксплуатации осуществляется рекламационная работа по выявлению скрытых ошибок разработки, на основании которой поводится следующий этап - модернизация;

в) Модернизация. Модернизация проводится специалистами предприятия-разработчика (изготовителя) на основе предъявленных в процессе промышленной эксплуатации рекламаций и имеет целью совершенствование отдельных характеристик системы.

§ 6. Эргономические аспекты проектирования АСУ

Эргономика – это наука, изучающая любые взаимодействия технических систем с внешней средой. Одной из отраслей эргономики является инженерная психология, которая изучает информационные взаимодействия технических систем с внешней средой, в качестве которой выступает человек.

Говоря о том, что автоматизированная система управления – это человеко-машинная система, необходимо четко понимать роль человека в каждой конкретной системе и его вклад в процесс управления. Для этого существует пять вариантов соотношений элементов «человек – техническая система»:

1. Системотехнический подход. Система состоит только из технических элементов, человек является фактором внешней среды;
2. Равноэлементный подход. Система состоит из равноценных элементов «человек» и «техническая система»;
3. Человеко-системный подход. Основным звеном системы является человек, а техническая – это подчиненное ему средство управления;
4. Узкоантропоцентристский подход. Элементы технических систем не учитываются в целях общих исследований деятельности человека;
5. Узкотехнический подход. Элементы «человек» не учитываются.

Без нарушения принципа системности, в автоматизированных системах управления допустимы первые три подхода в зависимости от фактической роли человека в конкретной системе. В то же время деятельность человека в АСУ отличается от повседневной бытовой деятельности. Рассмотрим особенности деятельности человека в автоматизированных системах управления:

1. Постоянное усложнение и расширение круга решаемых задач;
2. Дистанционное наблюдение за управляемым объектом;
3. Неравномерная нагрузка на различные сенсорные каналы восприятия человека;
4. Работа в условиях жестких ограничений по своевременности и точности действий;
5. Резкие изменения условий деятельности – от расслабляющей монотонности до энергичных и решительных действий;
6. Необычные условия жизнедеятельности.

В зависимости от конкретного предназначения АСУ и роли человека в процессе управления, функции этого человека могут быть самыми разнообразными. Перечислим функции человека в автоматизированной системе управления:

1. Работа с первичными данными – ручной сбор информации, ввод с клавиатуры;

2. Анализ и отбор поступающей информации;

3. Фильтрация устаревшей и избыточной информации;

4. Уточнение и получение недостающих данных;

5. Организация хранения данных на различных носителях;

6. Организация связи и передачи данных;

7. Создание и ведение баз данных и экспертных систем;

8. Управление и контроль за работой аппаратуры;

9. Принятие управленческих решений – наиболее сложная функция.

Рассмотрев варианты соотношений элементов «человек – техническая система» и функции человека в АСУ, необходимо отметить особенности проектирования интерфейса «человек - техническая система» (человеко-машинного интерфейса) в автоматизированной системе управления:

1. Рациональное распределение функций между человеком и техническими средствами;

2. Выбор способов оптимального кодирования входной и выходной информации;

3. Выбор средств отображения, органов управления, конструкций устройств и их рациональная компоновка;

4. Выбор оптимального варианта рабочего места и интерьера помещения;

5. Обучение операторов.

Человек-специалист, выполняющий работу по эксплуатации АСУ и непосредственно участвующий в процессе автоматизации управления, называется оператором. Деятельность операторов подразделяется:

1. Алгоритмизированная, т.е. выполняемая в соответствии с заранее заданными указаниями;

2. Эвристическая, т.е. выполняемая на основе личного опыта, умений и навыков, опирающаяся на интуицию.

Различия между алгоритмизированной и эвристической видами деятельности:

1. Алгоритмизированная деятельность характеризуется результативностью, а эвристическая деятельность не гарантирует получение результата;

2. Основой алгоритмизированной деятельности является плановость, а эвристической деятельности – интуитивность;

3. Алгоритмизированная деятельность использует детерминированные методы, а эвристическая деятельность - правдоподобные методы.

Виды операторов:

1. Оператор-руководитель. Любое лицо, принимающее решение на основе данных и рекомендаций, вырабатываемых АСУ,

можно назвать оператором-руководителем. (например: командир современного пассажирского самолета);

2. Оператор-исследователь. В процессе управления часто необходим ручной мониторинг управляемых процессов с целью анализа обстановки, на основании которого затем будет применен тот или иной алгоритм обработки информации. Такой ручной мониторинг выполняет оператор-исследователь;

3. Оператор-технолог. Оператор-технолог, как правило, эксплуатирует ту или иную автоматизированную систему управления технологическими процессами (АСУ ТП);

4. Оператор-манипулятор. Оператор-манипулятор непосредственно управляет каким-либо процессом;

5. Оператор-наблюдатель (контролер). Оператор-наблюдатель (контролер) наблюдает за процессом автоматизированного управления, вмешиваясь только в случаях возникновения каких-либо нештатных ситуаций;

6. Оператор-проектировщик. Оператор-проектировщик эксплуатирует системы автоматизированного проектирования (САПР).

Существует два варианта распределения функций между человеком и машиной:

1. Человек контролирует машинный процесс решения задачи и утверждает принимаемые решения;

2. Процесс решения задачи осуществляется последовательно с участием машины и человека.

Роль оператора при отказе технических средств заключается в переходе на ручное управление и принятии ответственности за принятое решение.

Говоря об эргономических аспектах проектирования АСУ, нельзя забывать о задачах эргономического проектирования автоматизированных систем управления:

1. Выбор численности персонала автоматизированной системы управления и распределение функций между персоналом – организация;

2. Выбор степени автоматизации и контроля, т.е. распределение функций между человеком и автоматизированной системой управления на каждом рабочем месте;

3. Разработка информационной модели и алгоритмов функционирования АСУ;

4. Проектирование рабочего места;

5. Проектирование условий труда на рабочем месте.

Реализация вышеназванных задач производится на основе следующих эргономических показателей проектирования:

1. Гигиенические показатели. Определяют соответствие конструкций, рабочего места и условий труда санитарно-гигиеническим нормам;

2. Физиолого-биомеханические показатели. Определяют соответствие орудий труда, органов управления, устройств наглядного отображения анатомо-физиологическим особенностям строения и функционирования органов и тела человека;

3. Психологические показатели. Определяют соответствие закрепленных и формируемых навыков возможностям восприятия, памяти и мышления человека;

4. Эстетические показатели. Определяют создание приятного эмоционального воздействия, располагающего к эффективной деятельности.

Глава 3. Программное обеспечение автоматизированных систем

§ 1. Основные понятия программного обеспечения АСУ

Автоматизированные системы управления являются представителями большого класса технических устройств, именуемых системами обработки информации. Согласно действующему в настоящее время стандарту ГОСТ 15971-90 под системой обработки информации понимается совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации. Важной компонентой системы обработки информации является программное обеспечение, то есть совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ. Структура программного обеспечения систем обработки информации в соответствии с ГОСТ 15071-90 и ГОСТ 19781-90 представлена на рисунке.

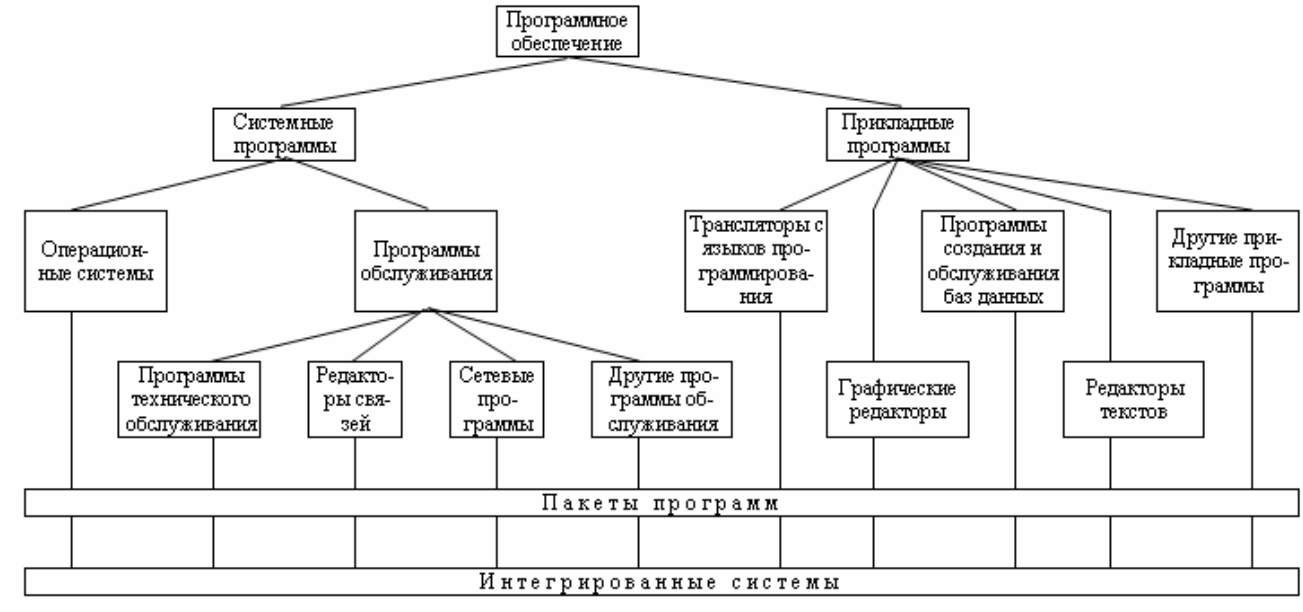


Рис. 1. Структура программного обеспечения.

Все программное обеспечение включает в себя две большие группы программ: системные и прикладные. Системные программы предназначены для поддержания работоспособности системы обработки информации или повышения эффективности ее использования в процессе выполнения прикладных программ.

В приведенном определении заключается основное отличие этих двух больших групп программ: системные программы обеспечивают работу прикладных программ за счет эффективного управления наличными ресурсами электронно-вычислительной машины; прикладные же, в свою очередь, предназначены для решения задачи или класса задач в определенной области применения систем обработки информации.

Системные программы являются неотъемлемой частью вычислительной системы и включают в себя операционные системы и программы обслуживания.

Операционные системы представляют собой совокупность системных программ, предназначенных для обеспечения определенного уровня эффективности систем обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю набора услуг.

Программы обслуживания предназначены для оказания услуг общего характера пользователям и обслуживающему персоналу систем обработки информации.

Согласно вышеприведенным определениям может показаться, что программы операционной системы и программы обслуживания тождественны, так как и те, и другие предоставляют пользователю определенные услуги. Принципиальное отличие этих программ состоит в том, что программы операционной системы являются управляющими, а программы обслуживания непосредственно ресурсами вычислительной системы не управляют. Делают они это посредством обращения к управляющим программам операционной системы.

К числу программ обслуживания относятся: редакторы связей, создающие загрузочные модули программ; различные вспомогательные (сервисные) программы, состав которых может быть весьма различен и зависит от назначения вычислительной системы, потребностей пользователя, возможностей обслуживающего персонала; программы технического обслуживания, предназначенные для настройки и отладки ЭВМ и автоматизирующие процесс ее технического обслуживания; сетевые и другие программы, удовлетворяющие требованию оказания услуг общего характера пользователям и обслуживающему персоналу, но не осуществляющие непосредственно управление ресурсами вычислительной системы.

К прикладным могут быть отнесены самые разнообразные программы, позволяющие проводить научные исследования в различных областях знаний, осуществлять автоматизированное

проектирование, изготавливать различные документы, вплоть до подготовки книг, разрабатывать новое программное обеспечение.

В то же время необходимо заметить, что приведенный выше классификационный состав программного обеспечения систем обработки информации не является застывшим. В процессе развития техники и информационной технологии могут возникать новые группы программ, а существующие – изменять свое функциональное предназначение. Так, например, работа пользователя на ЭВМ первых поколений была немыслима без знания им какого-либо языка программирования. По этой причине при описании программного обеспечения этих машин трансляторы с языков программирования причислялись к системным программам, а некоторые авторы включали их даже в операционную систему. В настоящее время разработкой новых программ занимается лишь достаточно узкий круг профессиональных программистов, а основная масса конечных пользователей ЭВМ языками программирования вообще не пользуется. На этом основании трансляторы потеряли свои системные функции и предназначаются для решения узкоспециальной прикладной задачи разработки новых программ профессиональными программистами.

По той же причине встречающийся в литературе термин «индивидуальная программа» или «программа разового использования» в настоящее время не применяется. Подавляющее большинство конечных пользователей ЭВМ работают с уже готовым программным продуктом, а разработчики программ, как правило, сами плоды своих трудов не эксплуатируют.

Все программы, как системные, так и прикладные, являются программным продуктом, то есть могут продаваться и покупаться, а следовательно, их эксплуатацией занимаются не только авторы. Поэтому все программное обеспечение должно быть хорошо документировано, познаваемо и пригодно к эксплуатации, то есть обладать свойствами промышленного продукта. Для этого программа снабжается программным документом, содержащим, в зависимости от назначения, данные, необходимые для ее эксплуатации и сопровождения. Программные документы и их части могут выполняться любым способом на любом материале, вводиться в ЭВМ и передаваться по каналам связи.

Отдельная программа может существовать самостоятельно или в составе пакета программ. Пакет программ представляет собой систему программ для решения задач определенного класса. В литературе часто встречается понятие «пакета прикладных программ». Термин «пакет программ» имеет более широкое приложение, объединяя в себе любые программы программного обеспечения. Так, например, пакет программ операционной системы представляет собой систему программ для ре-

шения задачи обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователям определенного набора услуг, а пакет программ технического обслуживания – для решения задачи автоматизации процесса настройки, отладки и технического обслуживания систем обработки информации.

Пакеты прикладных программ обязательно включают в себя несколько прикладных программ и решают узкоспециальные задачи в определенной области применения системы обработки информации. Однако в составе пакета программ наряду с прикладными могут находиться и системные программы, расширяющие возможности операционной системы в интересах повышения эффективности выполнения прикладных программ.

Все программы программного обеспечения вычислительных систем для взаимодействия с пользователем предоставляют различные интерфейсные средства. Совокупность средств, предназначенных для управления ходом работы программы и отражения результатов ее работы, представляет собой операционную среду этой программы. Такими средствами могут быть отдельные команды, различного вида меню, текстовые, графические, звуковые сообщения и т. д. Не следует путать понятия операционной среды и операционной системы. «Операционная среда» – это эргономический термин, применимый к любой программе, в том числе и к программам операционной системы. Как системные, так и прикладные программы в процессе своей работы формируют операционную среду, в которую погружается пользователь, решая свои задачи.

Наряду с отдельными программами, пакеты программ также предоставляют пользователю свою операционную среду. При решении сложных задач часто возникает необходимость последовательного использования различных пакетов программ, что без применения специальных средств усложняет и замедляет работу с машиной. Современное программное обеспечение дает пользователю инструмент решения этой проблемы в виде интегрированной системы, представляющей собой пакет системных программ, дающий пользователю одинаковые возможности доступа к различным пакетам программ из одной операционной среды и стыковки программ пакетов по информационным данным.

Таким образом, любая операционная система представляет собой интегрированную систему, так как из ее операционной среды имеется доступ к любой программе любого пакета программ. В то же время, для решения специальных задач операционная среда операционной системы не всегда удобна, поэтому разработчики программного продукта стремятся создать специальные интегрированные системы для решения различных классов задач. Примерами такой специальной интегрированной

системы может служить система программирования, включающая в себя пакеты или отдельные программы редактора текстов, транслятора с языка программирования, отладчика, редактора связей и загрузчика.

§ 2. Программная документация

Когда программист-разработчик в той или иной форме получает задание на программирование, перед ним, перед руководителем проекта и перед всей проектной группой встают следующие вопросы:

1. Что должно быть сделано, кроме собственно программы?
2. Что и как должно быть оформлено в виде документации?
3. Что передавать пользователям, а что — службе сопровождения?
4. Как управлять всем этим процессом?
5. Что должно входить в само задание на программирование?

В настоящее время остается актуальным вопрос о наличии системы, регламентирующей документирование программных средств. Основу отечественной нормативной базы в области документирования программных средств составляет комплекс стандартов единой системы программной документации (ЕСПД). Основная и большая часть комплекса ЕСПД была разработана в 70-е и 80-е годы прошлого века. Сейчас этот комплекс представляет собой систему межгосударственных стандартов стран СНГ (ГОСТ), действующих на территории Российской Федерации на основе межгосударственного соглашения по стандартизации.

Стандарты ЕСПД в основном охватывают ту часть документации, которая создается в процессе разработки программных средств, и связаны, по большей части, с документированием их функциональных характеристик. Следует отметить, что стандарты ЕСПД (ГОСТ 19) носят рекомендательный характер. В соответствии с Законом РФ «О стандартизации» любые стандарты становятся обязательными на контрактной основе — то есть при ссылке на них в договоре на разработку или поставку какого-либо продукта.

Говоря о состоянии ЕСПД в целом, можно констатировать, что большая часть стандартов ЕСПД морально устарела. Однако в настоящее время идет процесс замены устаревших стандартов, но чаще всего он носит лишь косметический характер, не изменяя кардинально всю систему. К числу основных недостатков ЕСПД можно отнести:

1. Ориентацию на единственную, «каскадную» модель жизненного цикла программных средств;

2. Отсутствие четких рекомендаций по документированию характеристик качества программных средств;

3. Отсутствие системной увязки с другими действующими отечественными системами стандартов по жизненному циклу и документированию продукции в целом, например, ЕСКД (Единой системы конструкторской документации);

4. Нечетко выраженный подход к документированию программных средств как товарной продукции;

5. Отсутствие рекомендаций по самодокументированию программных средств, например, в виде экранных меню и средств оперативной помощи пользователю («хелпов»);

6. Отсутствие рекомендаций по составу, содержанию и оформлению перспективных документов на программные средства, согласованных с рекомендациями международных и региональных стандартов.

Стандарты ЕСПД (как и другие ГОСТы) подразделяют на группы:

0. Общие положения;
1. Основополагающие стандарты;
2. Правила выполнения документации разработки;
3. Правила выполнения документации изготовления;
4. Правила выполнения документации сопровождения;
5. Правила выполнения эксплуатационной документации;
6. Правила обращения программной документации;
7, 8. Резервные группы;
9. Прочие стандарты.

Обозначение стандарта ЕСПД строят по классификационному признаку.

Обозначение стандарта ЕСПД должно состоять из:

1. Числа 19 (присвоенных классу стандартов ЕСПД);

2. Одной цифры (после точки), обозначающей код указанной выше классификационной группы стандартов;

3. Двузначного числа (после тире), указывающего год регистрации стандарта.

Из всех стандартов ЕСПД остановимся только на тех, которые могут чаще использоваться на практике.

Первым укажем стандарт, который можно использовать при формировании заданий на программирование.

ГОСТ (СТ СЭВ) 19.201-78 (1626-79). «ЕСПД. Техническое задание. Требование к содержанию и оформлению.»

Техническое задание содержит совокупность требований к программным средствам и может использоваться как критерий проверки и приемки разработанной программы. Поэтому достаточно полно составленное (с учетом возможности внесения дополнительных разделов) и принятое заказчиком и разработчиком, техническое задание является одним из основополагающих документов проекта программных средств.

Техническое задание должно содержать следующие разделы:

1. Введение;
2. Основания для разработки;
3. Назначение разработки;
4. Требования к программе или программному изделию;
5. Требования к программной документации;
6. Технико-экономические показатели;
7. Стадии и этапы разработки;
8. Порядок контроля и приемки;
9. В техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

ГОСТ (СТ СЭВ) 19.101-87 (1626-79). «ЕСПД. Виды программ и программных документов.»

Устанавливает виды программ и программных документов для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

Стандарт определяет следующие виды программ:

1. Программный компонент. Программный компонент - это программа, рассматриваемая как единое целое, выполняющая законченную функцию и применяемая самостоятельно или в составе программного комплекса;
2. Программный комплекс. Программный комплекс – это программа, состоящая из двух или более программных компонент и (или) программных комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе другого программного комплекса

Стандарт определяет следующие виды программных документов:

1. Спецификация. Спецификация содержит перечень состава программы и документации на нее;
2. Ведомость держателей подлинников. Ведомость держателей подлинников содержит перечень предприятий, на которых хранят подлинники программных документов;
3. Текст программы. Текст программы содержит запись программы с необходимыми комментариями;
4. Описание программы. Описание программы содержит сведения о логической структуре и функционировании программы. Определяется ГОСТ 19.402-78 «ЕСПД. Описание программы.»;
5. Программа и методика испытаний. Программа и методика испытаний содержит требования, подлежащие проверке при испытании программы, а также порядок и методы их контроля.

Определяется ГОСТ 19.301-79 «ЕСПД. Программа и методика испытаний.»;

6. Техническое задание. Техническое задание содержит сведения о назначении и области применения программы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний. Определяется ГОСТ (СТ СЭВ) 19.201-78 (1626-79). «ЕСПД. Техническое задание. Требование к содержанию и оформлению.»;

7. Пояснительная записка. Пояснительная записка содержит схему и общее описание алгоритма и (или) функционирования программы, а также обоснование принятых технических и технико-экономических решений. Определяется ГОСТ 19.404-79 «ЕСПД. Пояснительная записка.»;

8. Эксплуатационные документы. Эксплуатационные документы содержат сведения для обеспечения функционирования и эксплуатации программы:

а) Ведомость эксплуатационных документов. Ведомость эксплуатационных документов содержит перечень эксплуатационных документов на программу;

б) Формуляр. Формуляр содержит основные характеристики программы, комплектность и сведения об эксплуатации программы;

в) Описание применения. Описание применения содержит сведения о назначении программы, области применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств. Определяется ГОСТ 19.502-78 «ЕСПД. Описание применения.»;

г) Руководство системного программиста. Руководство системного программиста содержит сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения. Определяется ГОСТ 19.503-79 «ЕСПД. Руководство системного программиста.»;

д) Руководство программиста. Руководство программиста содержит сведения для эксплуатации программы. Определяется ГОСТ 19.504-79 «ЕСПД. Руководство программиста.»;

е) Руководство оператора. Руководство оператора содержит сведения для обеспечения процедуры общения оператора с вычислительной системой в процессе выполнения программы. Определяется ГОСТ 19.505-79 «ЕСПД. Руководство оператора.»;

ж) Описание языка. Описание языка содержит описание синтаксиса и семантики языка;

з) Руководство по техническому обслуживанию. Руководство по техническому обслуживанию содержит сведения для применения тестовых и диагностических программ при обслуживании технических средств.

В зависимости от способа выполнения и характера применения программные документы подразделяются на подлинник, дубликат и копию (ГОСТ 2.102-68), предназначенные для разработки, сопровождения и эксплуатации программы.

Допускается объединять отдельные виды эксплуатационных документов (за исключением ведомости эксплуатационных документов и формуляра). Необходимость объединения этих документов указывается в техническом задании. Объединенному документу присваивают наименование и обозначение одного из объединяемых документов. В объединенных документах должны быть приведены сведения, которые необходимо включать в каждый объединяемый документ.

ГОСТ 19.102-77. «ЕСПД. Стадии разработки.»

Стандарт устанавливает стадии разработки программ и программной документации для вычислительных машин, комплексов и систем независимо от их назначения и области применения:

1. Техническое задание:

а) Обоснование необходимости разработки программы:

- постановка задачи;
- сбор исходных материалов;
- выбор и обоснование критериев эффективности и качества разрабатываемой программы;
- обоснование необходимости проведения научно-исследовательских работ;

б) Научно-исследовательские работы:

- определение структуры входных и выходных данных;
- предварительный выбор методов решения задач;
- обоснование целесообразности применения ранее разработанных программ;
- определение требований к техническим средствам;
- обоснование принципиальной возможности решения поставленной задачи;

в) Разработка и утверждение технического задания:

- определение требований к программе;
- разработка технико-экономического обоснования разработки программы;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- выбор языков программирования;
- определение необходимости проведения научно-исследовательских работ на последующих стадиях;
- согласование и утверждение технического задания;

2. Эскизный проект:

а) Разработка эскизного проекта:

- предварительная разработка структуры входных и выходных данных;

- уточнение методов решения задачи;
- разработка общего описания алгоритма решения задачи;
- разработка технико-экономического обоснования;

б) Утверждение эскизного проекта:

- разработка пояснительной записки;
- согласование и утверждение эскизного проекта;

3. Технический проект:

а) Разработка технического проекта:

- уточнение структуры входных и выходных данных;
- разработка алгоритма решения задачи;
- определение формы представления входных и выходных данных;
- определение семантики и синтаксиса языка;
- разработка структуры программы;
- окончательное определение конфигурации технических средств;

б) Утверждение технического проекта:

- разработка плана мероприятий по разработке и внедрению программ;
- разработка пояснительной записки;
- согласование и утверждение технического проекта;

4. Рабочий проект:

а) Разработка программы:

- программирование и отладка программы;

б) Разработка программной документации:

- разработка программных документов в соответствии с требованиями ГОСТ 19.101-77;

в) Испытания программы:

- разработка, согласование и утверждение программы и методики испытаний;
- проведение предварительных государственных, межведомственных, приемо-сдаточных и других видов испытаний;
- корректировка программы и программной документации по результатам испытаний;

5. Внедрение:

а) Подготовка и передача программы:

- подготовка и передача программы и программной документации для сопровождения и (или) изготовления;
- оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление;
- передача программы в фонд алгоритмов и программ.

Примечания:

1. Допускается исключать вторую стадию разработки, а в технически обоснованных случаях — вторую и третью стадии. Необходимость проведения этих стадий указывается в техническом задании.

2. Допускается объединять, исключать этапы работ и (или) их содержание, а также вводить другие этапы работ по согласованию с заказчиком.

ГОСТ 19.103-77 «ЕСПД. Обозначение программ и программных документов.»

Программа и ее документ "Спецификация" имеют следующую структуру обозначения:

Структура обозначения других программных документов:

1. Код страны-разработчика и код организации-разработчика присваивают в установленном порядке.

2. Регистрационный номер присваивается в порядке возрастания, начиная с 00001 до 99999, для каждой организации-разработчика.

3. Номер издания программы или номер редакции. Номер документа данного вида, номер части документа присваиваются в порядке возрастания с 01 до 99. (Если документ состоит из одной части, то дефис и порядковый номер части не указывают).

Примечание: Номер редакции спецификации и ведомости эксплуатационных документов на программу должны совпадать с номером издания этой же программы.

ГОСТ 19.105-78 «ЕСПД. Общие требования к программным документам.»

Настоящий стандарт устанавливает общие требования к оформлению программных документов для вычислительных машин, комплексов и систем, независимо от их назначения и области применения и предусмотренных стандартами единой системы программной документации (ЕСПД) для любого способа выполнения документов на различных носителях данных. Программный документ может быть представлен на различных типах носителей данных и состоит из следующих условных частей:

1. Титульной;
2. Информационной;
3. Основной.

Правила оформления документа и его частей на каждом носителе данных устанавливаются стандартами ЕСПД на правила оформления документов на соответствующих носителях данных.

ГОСТ 19.106-78 «ЕСПД. Требования к программным документам, выполненным печатным способом.»

В соответствии со стандартом программные документы оформляют:

1. На листах формата А4 при изготовлении документа машинописным или рукописным способом;

2. Допускается оформление на листах формата А3;

3. При машинном способе выполнения документа допускаются отклонения размеров листов, соответствующих форматам А4 и А3, определяемые возможностями применяемых технических

средств; на листах форматов А4 и А3, предусматриваемых выходными характеристиками устройств вывода данных, при изготовлении документа машинным способом;

4. На листах типографических форматов при изготовлении документа типографским способом.

Расположение материалов программного документа осуществляется в следующей последовательности:

1. Титульная часть:

а) Лист утверждения (не входит в общее количество листов документа);

б) Титульный лист (первый лист документа);

2. Информационная часть:

а) Аннотация;

б) Лист содержания;

3. Основная часть:

а) Текст документа (с рисунками, таблицами и т.п.)

б) Перечень терминов и их определений;

в) Перечень сокращений;

г) Приложения;

д) Предметный указатель;

е) Перечень ссылочных документов;

4. Часть регистрации изменений:

а) Лист регистрации изменений.

Перечень терминов и их определений, перечень сокращений, приложения, предметных указатель, перечень ссылочных документов выполняются при необходимости.

Есть также группа стандартов, определяющая требования к фиксации всего набора программ и программных документов, которые оформляются для передачи программных средств. Они порождают лаконичные документы учетного характера и могут быть полезны для упорядочения всего хозяйства программ и программных документов. Есть также и стандарты, определяющие правила ведения документов по учету программных средств.

Для оформления различного рода программной документации, а также для оформления графической части различных курсовых, дипломных работ и рефератов очень полезен ГОСТ 19.701-90 «ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные графические и правила выполнения.». Этот стандарт устанавливает правила выполнения схем, используемых для отображения различных видов задач обработки данных и средств их решения и полностью соответствует стандарту ISO 5807:1985.

Наряду с ЕСПД на межгосударственном уровне действуют еще два стандарта, также относящихся к документированию ПС и принятых не так давно, как большая часть ГОСТ ЕСПД.

1. ГОСТ 19781-90 «Программное Обеспечение систем обработки информации. Термины и определения.». Стандарт устанавливает термины и определения понятий в области программного обеспечения систем обработки данных, применяемые во всех видах документации и литературы, входящих в сферу работ по стандартизации или использующих результаты этих работ;

2. ГОСТ 28388-89 «Системы обработки информации. Документы на магнитных носителях данных. Порядок выполнения и обращения.». Данный стандарт распространяется не только на программные, но и на конструкторские, технологические и другие проектные документы, выполняемые на магнитных носителях.

§ 3. Жизненный цикл программного обеспечения

Жизненный цикл программы – это период времени от начала её разработки до вывода программы из эксплуатации по причине морального устаревания. По длительности жизненного цикла программы бывают:

1. Программы с малой длительностью эксплуатации. Как правило, это программы, которые разрабатываются с учебными или исследовательскими целями, и по окончании цикла обучения или исследований необходимость в использовании таких программ отпадает;

2. Программы с большой длительностью эксплуатации. Как правило, это коммерческие программные продукты, нацеленные на широкую аудиторию пользователей.

Этапы жизненного цикла:

1. Системный анализ:

а) Определение целей программных средств;

б) Выбор методов решения задач;

в) Проектирование алгоритмов;

г) Разработка технического задания на программный комплекс;

2. Проектирование:

а) Структурное проектирование:

- определение структуры программного комплекса;
- определение структуры программных модулей;
- распределение производительности ЭВМ;
- распределение памяти ЭВМ.

Если содержание предыдущих этапов так или иначе уже рассматривалось в настоящем пособии, то о распределении производительности и памяти ЭВМ необходимо сказать подробнее. В ЭВМ общего назначения, как правило, имеются достаточно большие объемы оперативной и внешней долговременной памяти, а также достаточно мощные и производительные про-

цессоры. По этой причине, в таких системах не возникает необходимости в распределении этих ресурсов. Во встраиваемых системах применяются однокристальные микро-ЭВМ, ресурсы которых сильно ограничены особенностями применения. Такими особенностями может быть и ценовая политика, и тяжелые условия эксплуатации (отсутствие охлаждения, вибрация, нестабильность питающих напряжений и т.п.), и массогабаритные ограничения. Благодаря этому, однокристальные микро-ЭВМ, как правило, работают на пониженных по сравнению с ЭВМ общего назначения тактовых частотах и имеют очень маленькие объемы (опять же по сравнению с ЭВМ общего назначения) оперативной и долговременной памяти. Учитывая вышесказанное, для реализации режима реального масштаба времени в таких системах требуется корректное распределение как ресурсов производительности процессора, так и ресурсов памяти;

б) Подготовка технологических средств:

- организация базы данных программного комплекса;
- выбор системы программирования;

в) Разработка программ;

г) Отладка программ в статике:

- планирование отладки программ;
- тестирование программ;
- локализация ошибок и тестирование программ;
- комплексирование программ;

д) Комплексная динамическая отладка:

- выбор средств для имитации абонентов;
- разработка программ имитации;
- создание программ обработки результатов;
- отладка функционирования в реальном масштабе времени.

Если этапы отладки программ в статике хорошо знакомы любому, кто когда-либо занимался практическим программированием, то о комплексной динамической отладке знают только те, кто писал программы для автоматизированных систем управления. Согласно определению, АСУ производит автоматизированный сбор информации и управление физическими объектами. Поэтому программы, реализующие эти функции, должны предусматривать работу с источниками и потребителями информации. Однако не всегда возможно на этапе отладки воспользоваться реальными источниками и потребителями информации, или заставить их работать в контролируемых режимах, поэтому возникает необходимость в их моделировании (имитации).

е) Выпуск машинных носителей и программных документов:

- изготовление машинных носителей;
- изготовление эксплуатационных документов;
- изготовление технологических документов;
- изготовление исследовательских документов;

ж) Испытания программных средств:
- на полноту функционирования;
- на надежность функционирования;
- обработка результатов испытаний;
- разработка акта испытаний;
3. Эксплуатация;
4. Сопровождение.

§ 4. Операционная система реального времени QNX

Операционная система является неотъемлемой компонентой программного обеспечения автоматизированных систем управления. Операционная система жесткого реального масштаба времени QNX явилась миру в 1991 году, когда было объявлено о снятии запрета на её вывоз из Северной Америки как стратегического ресурса США и Канады. Первой публичной версией этой операционной системы стала QNX 2. Однако в настоящее время эта версия повсеместно вышла из употребления. Следующим шагом фирмы-разработчика - канадской компании QSSL (QNX Software Systems Ltd) - стал выпуск операционной системы QNX 4. В настоящее время она является самой распространенной операционной системой реального масштаба времени в промышленности. Несмотря на это, производитель прекратил развивать эту версию, но продолжает выпускать к ней драйверы устройств. Последней наиболее современной версией операционной системы является QNX 6. Эта версия максимально совместима на уровне исходных кодов с операционной системой Linux.

Операционная система реального масштаба времени QNX – это UNIX-подобная масштабируемая отказоустойчивая распределенная операционная система жесткого реального масштаба времени, предназначенная для использования в автоматизированных системах различного назначения. ОС РВ QNX реализует интерфейс стандарта POSIX (Portable Operating System Interface – интерфейс переносимой операционной системы), разработанный в 1988 году в институте инженеров по электротехнике и электронике (IEEE – Institute of Electrical and Electronics Engineers). Этот стандарт был принят международной организацией по стандартизации (ISO – International Standard Organization) в качестве стандарта ISO/IEEE. Стандарт POSIX определяет:

1. Программный интерфейс для систем управления процессами;

2. Программный интерфейс для систем управления вводом-выводом данных в устройствах и файловых системах;

3. Программный интерфейс межпроцессного взаимодействия;

4. Программные средства реализации режима реального масштаба времени (семафоры, сигналы, приоритеты, таймеры и т.д.);

5. Механизмы управления потоками выполнения;

6. Механизмы обработки прерываний.

Стандарт POSIX определяет четыре варианта профилей системы в зависимости от имеющихся ограничений аппаратных ресурсов:

1. Минимальная система. Под минимальной системой понимается встроенная система, в которой отсутствуют механизмы управления памятью, файловая система и система ввода-вывода;

2. Контроллер реального времени. Этот профиль включает в себя минимальную систему, дополненную файловой системой и системой ввода-вывода;

3. Специализированная система. Под специализированной системой понимается встроенная система, в которой отсутствует файловая система;

4. Многоцелевая система. Данный профиль представляет собой систему, обеспечивающую поддержку всей предусмотренной стандартом POSIX функциональности.

Особенности операционной системы реального масштаба времени QNX:

1. Микроядерная архитектура;

2. Взаимодействие между процессами на основе обмена сообщениями.

Микроядерная архитектура ОС РВ QNX основана на использовании миниатюрного программного ядра, обеспечивающего минимальную конфигурацию системы (например: размер ядра QNX 4 всего 10 Кбайт, а в варианте многоцелевой системы ОС РВ QNX 4 требует 10 Мбайт). Все остальные функции операционной системы реализуются дополнительными процессами. Главная цель в проектировании микроядерной операционной системы – сделать её компактной и гибкой. ОС РВ QNX реализуется как группа процессов, взаимодействующих под управлением микроядра. При этом любые процессы, как системные, так и прикладные, имеют одинаковую структуру и равноправны по отношению к микроядру. По этой причине система становится открытой и легко масштабируемой. Микроядро функционирует как логическая программная общая шина, позволяющая динамически, без перезагрузки системы, подключать и отключать по мере необходимости любые программные модули, как системные, так и прикладные. Микроядро ОС РВ QNX выполняет следующие функции:

1. Управление потоками выполнения;

2. Обмен сообщениями;

3. Управление сигналами;

4. Синхронизация процессов;
5. Планирование потоков;
6. Управление таймерами;
7. Управление процессами.

Ввиду того, что ОС РВ QNX изначально ориентирована на многозадачность, то есть на одновременное выполнение нескольких независимых алгоритмов, понятие процесса в операционной системе отличается от общепринятого. Процесс в QNX определяется как работа микроядра распределенной операционной системы по обработке одного или нескольких потоков выполнения. В целях разделения понятий процесса, как любой работы, выполняемой процессором ЭВМ, и процесса, содержащего один или несколько потоков выполнения, последний (QNX-процесс) в рамках настоящего пособия будем называть распределенным процессом.

Все распределенные процессы отделены друг от друга с помощью блока управления памятью (MMU – Memory Management Unit). Задача микроядра при этом состоит в корректном распределении памяти между процессами внутри потоков выполнения и между распределенными процессами, выполняющимися в одном адресном пространстве. При этом адресное пространство может быть как локальным, так и распределенным по нескольким ЭВМ сети. Распределение ресурсов памяти и производительности процессора между потоками выполнения производится с помощью атрибутов потока:

1. Уникальный в системе идентификатор потока;
2. Набор виртуальных регистров потока. Набор виртуальных регистров потока (указатель команд, указатель стека и т.д.) составляет контекст потока;
3. Стек потока;
4. Маска сигналов. Маска сигналов запрещает обработку потоком тех или иных сигналов. Под сигналом в ОС РВ QNX понимается неблокирующее сообщение фиксированного размера (четыре байта данных и один байт кода);
5. Локальная память потока. Локальная память потока представляет собой системную область данных потока, которая используется для хранения информации, относящейся к каждому отдельному потоку;
6. Обработчик завершений. Этот атрибут содержит функции обратного вызова, выполняемые при завершении потока.

Указанные атрибуты фактически являются ресурсами, которые необходимо выделить и инициализировать внутри адресного пространства распределенного процесса для запуска потока выполнения. Запущенный поток может находиться в одном из трех состояний:

1. Готовность;
2. Активность;

3. Блокировка.

В свою очередь, эти три состояния детализируются в двадцать одно состояние потока выполнения. Для связывания данных, относящихся к потоку выполнения и хранящихся в локальной памяти потока, с самим потоком, используется уникальный для каждого потока выполнения глобальный целочисленный ключ процесса. Потоки выполнения создаются и уничтожаются динамически, их количество внутри распределенного процесса может изменяться в значительных пределах. Завершение потока выполнения включает в себя останов всех процессов потока и освобождение ресурсов, занимаемых потоком.

При каждом вызове ядра, исключении или аппаратном прерывании, в результате которых управление передается микроядру, работа активного потока выполнения приостанавливается. Микроядро, получив управление, осуществляет планирование потоков. Действие планирования происходит при изменении состояния любого потока вне зависимости от того, в каком распределенном процессе этот поток расположен. Планирование потоков осуществляется сразу по всем распределенным процессам. Как правило, выполнение приостановленного потока через некоторое время возобновляется. При этом микроядро переключает ресурсы процессора с одного потока на другой, когда активный поток:

1. Блокируется. Активный поток блокируется, если он должен ожидать какого-либо события. Блокированный поток удаляется из очереди готовности, после чего запускается поток с наивысшим приоритетом, находящийся в очереди готовности. Когда блокированный поток разблокируется, он помещается в конец очереди готовности на соответствующий приоритетный уровень;

2. Вытесняется. Активный поток вытесняется, когда поток с более высоким приоритетом в результате снятия блокировки помещается в очередь готовности. Прерванный поток остается на соответствующем приоритетном уровне в начале очереди готовности, а поток с более высоким приоритетом начинает выполняться;

3. Отдает управление. Активный поток в соответствии со своим алгоритмом самостоятельно освобождает процессор и помещается в конец очереди готовности на данном уровне приоритета. После этого запускается поток с наивысшим приоритетом.

Всего в ОС РВ QNX поддерживается до 256 уровней приоритетов. Непривилегированные потоки имеют приоритеты от 1 до 63, причем с ростом номера уровень приоритета повышается. Привилегированные потоки имеют приоритеты от 64 до 255. В ОС РВ QNX имеется специальный поток с нулевым приоритетом по имени «idle» (idle - праздный, ленивый, бесполезный).

Поток «idle» постоянно готов к выполнению и используется для загрузки процессора при отсутствии потоков с более высокими приоритетами. По умолчанию поток всегда наследует приоритет своего родительского потока. Для предотвращения инверсии приоритетов микроядро может временно повышать приоритет потока. Под инверсией приоритетов понимается ситуация, когда поток с низким приоритетом использует ресурс процессора с более высоким, чем у себя, приоритетом. Это происходит, когда поток с более высоким приоритетом работает от имени потока с низким приоритетом.

Потоки, находящиеся в очереди готовности, упорядочиваются по приоритету. Физически очередь готовности состоит из 256 очередей – по одной на каждый уровень приоритета. Потоки выстраиваются в очередь и обслуживаются в соответствии с их приоритетами в порядке поступления (дисциплина FIFO). Для работы с различными прикладными программами в ОС РВ QNX используются три алгоритма планирования:

1. FIFO-планирование. В алгоритме FIFO-планирования потоки выполнения обслуживаются в порядке их поступления в очередь готовности, пока они не блокируются или не прервутся;

2. Циклическое планирование. В этом алгоритме планирования каждому потоку выполнения выделяется промежуток времени, по истечении которого поток прерывается и управление передается следующему потоку, находящемуся в очереди готовности;

3. Спорадическое планирование. При спорадическом планировании (спорадический – появляющийся от случая к случаю, нерегулярный), так же, как и в FIFO-планировании, потоки выполнения обслуживаются в порядке их поступления в очередь готовности, пока они не блокируются или не прервутся. Отличие спорадического планирования от FIFO-планирования заключается в том, что при спорадическом планировании приоритеты потоков выполнения динамически изменяются между нормальным приоритетом потока и пониженным (фоновым) приоритетом потока выполнения. Для управления спорадическим переходом используются следующие параметры:

а) Начальный бюджет потока. Начальный бюджет потока – это промежуток времени, в течение которого поток может выполняться с нормальным приоритетом;

б) Период пополнения. Период пополнения – это промежуток времени, по истечении которого поток выполнения восстанавливает свой нормальный приоритет. Период пополнения в ряде систем иногда еще называют расчетным циклом;

в) Максимальное число текущих пополнений. Этот параметр представляет собой значение, ограничивающее количество

выполняемых операций по восстановлению нормального приоритета потока.

Таким образом, поток выполняется в течение времени начального бюджета потока каждый период пополнения. Если обозначить начальный бюджет потока как *C*, а период пополнения как *T*, то спорадическое планирование позволяет для каждого потока выполнения использовать только $\frac{C}{T}$ часть системных ресурсов.

Каждый поток в системе может планироваться по любому из вышеописанных алгоритмов. Алгоритмы планирования применяются для каждого потока выполнения индивидуально. Во всех случаях, когда поток с более высоким приоритетом переходит в состояние готовности, он вытесняет все другие потоки с более низкими приоритетами. FIFO-планирование и циклическое планирование применяются только в тех случаях, когда два или более потоков с одинаковым приоритетом находятся в состоянии готовности и напрямую конкурируют за использование ресурсов процессора. В процессе работы потока выполнения его приоритет может изменяться либо в результате действий самого потока, либо в результате вмешательства микроядра. Изменяться может не только приоритет, но и алгоритм планирования, реализуемый микроядром.

Глава 4. Информационное обеспечение АСУ

§ 1. Понятие информационного обеспечения АСУ

Информационное обеспечение АСУ – это совокупность единой системы классификации и кодирования технико-экономической информации, унифицированных систем документации и массивов информации, используемых в автоматизированных системах управления.

Состав информационного обеспечения автоматизированной системы управления:

1. Данные;
2. Языковые средства описания данных;
3. Методы организации данных;
4. Методы хранения данных;
5. Методы накопления информации;
6. Методы доступа к информации.

Данные систематизируют в информационной базе автоматизированной системы. Состав информационной базы АСУ:

1. Нормативные документы;
2. Справочные данные;
3. Текущие сведения о состоянии управляемого объекта (оперативная информация);

4. Внешние данные;

5. Накапливаемые архивные данные.

Основное назначение информационного обеспечения АСУ заключается в создании динамической информационной модели управляемого объекта, отражающей его состояние в текущий или предшествующий момент времени.

Требования к информационному обеспечению автоматизированных систем управления:

1. Полнота отражения состояний управляемой системы;

2. Достоверность информации;

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

4. Многократное и многоцелевое использование однократно введенной информации;

5. Простота и удобство доступа к данным;

6. Минимальное дублирование информации;

7. Организация эффективной системы документооборота;

8. Возможность развития информационного обеспечения АСУ;

9. Разграничение доступа и времени хранения информации (информационная безопасность АСУ).

§ 2. Понятие информационной безопасности АСУ

Под информационной безопасностью понимают такое состояние информационного обеспечения АСУ, при котором исключается возможность ознакомления с этим информационным обеспечением, его изменения или уничтожения лицами, не имеющими на это право.

Природа этих воздействий может быть самой разнообразной. Это и попытки проникновения злоумышленников, и ошибки персонала, и выход из строя аппаратных и программных средств, стихийные бедствия (землетрясение, ураган, пожар и т.п.).

Информационная безопасность компьютерных систем и сетей достигается принятием комплекса мер по обеспечению конфиденциальности, целостности, достоверности, юридической значимости информации, оперативности доступа к ней, а также по обеспечению целостности и доступности информационных ресурсов и компонентов системы или сети.

Перечисленные выше базовые свойства информации нуждаются в более полном толковании:

1. Конфиденциальность информации. Конфиденциальность информации - это её свойство быть доступной только ограниченному кругу пользователей информационной системы, в которой циркулирует данная информация. По существу, конфиденциальность информации - это ее свойство быть известной только

допущенным и прошедшим проверку субъектам системы (пользователям, процессам, программам). Для остальных субъектов системы информация должна быть неизвестной. Таким образом, информация, доступ к которой ограничен кругом лиц, которым она была доверена по службе или стала известна в процессе работы, является конфиденциальной. Указ Президента РФ № 188 от 6 марта 1997 года «Перечень сведений конфиденциального характера» определяет шесть видов конфиденциальной информации:

а) Служебная тайна. Служебная тайна – это служебные сведения, доступ к которым ограничен органами государственной власти и федеральными законами;

б) Персональные данные;

в) Тайна следствия и судопроизводства;

г) Коммерческая тайна. Коммерческая тайна – это информация, имеющая действительную или потенциальную коммерческую ценность в силу неизвестности её третьим лицам;

д) Тайна сущности изобретения (Ноу Хау – know how – знаю как);

е) Профессиональная тайна.

Еще один вид конфиденциальной информации устанавливает Конституция Российской Федерации в статье 29 - государственная тайна. Закон РФ «О государственной тайне» определяет существование межведомственной комиссии при Президенте РФ по защите государственной тайны;

2. Целостность информации. Под целостностью информации понимается ее свойство сохранять свою структуру и/или содержание в процессе передачи и хранения. Целостность информации обеспечивается в том случае, если данные в системе не отличаются в семантическом отношении от данных в исходных документах, то есть если не произошло их случайного или преднамеренного искажения или разрушения;

3. Достоверность информации. Достоверность информации – это свойство, выражаемое в строгой принадлежности информации субъекту, который является ее источником, либо тому субъекту, от которого она принята;

4. Юридическая значимость информации. Юридическая значимость информации означает, что документ, являющийся носителем информации, обладает юридической силой;

5. Доступ к информации. Под доступом к информации понимается ознакомление с информацией и ее обработка, в частности копирование, модификация или уничтожение. Различают санкционированный и несанкционированный доступ к информации. Санкционированный доступ к информации не нарушает установленные правила разграничения доступа. Несанкционированный доступ характеризуется нарушением установленных правил разграничения доступа и является одним из видов утечки

информации. Под утечкой информации понимается бесконтрольный выход конфиденциальной информации за пределы организации или круга лиц, которым она была доверена по службе или стала известна в процессе работы. Лицо или процесс, осуществляющие несанкционированный доступ к информации, являются нарушителями таких правил разграничения доступа. Несанкционированный доступ - наиболее распространенный вид компьютерных нарушений, представляет собой совокупность приемов и порядок действий с целью получения конфиденциальной информации незаконным, противоправным путем. Степень опасности несанкционированного доступа определяется принесенным ущербом. Вторым видом утечки информации является непреднамеренная утечка информации, когда субъекты, допустившие такую утечку, изначально не планировали никаких противоправных или преступных действий. Утечка информации в таком случае происходит вследствие неосторожности, низкой бдительности или плохой профессиональной квалификации персонала, допущенного до обработки конфиденциальной информации. Несмотря на непреднамеренность, данный вид утечки также является наказуемым деянием, тяжесть которого определяется степенью конфиденциальности разглашенной информации. Правила разграничения доступа служат для регламентации прав доступа к компонентам системы;

6. Оперативность доступа к информации. Оперативность доступа к информации - это способность информации или некоторого информационного ресурса быть доступными конечному пользователю в соответствии с его оперативными потребностями;

7. Целостность ресурса или компонента системы. Целостность ресурса или компонента системы — это его свойство быть неизменным в семантическом смысле при функционировании системы в условиях случайных или преднамеренных искажений либо разрушающих воздействий;

8. Доступность ресурса или компонента системы. Доступность ресурса или компонента системы - это его свойство быть доступным законным пользователям системы. С допуском к информации и ресурсам системы связана группа таких понятий, как идентификация, аутентификация, авторизация. С каждым объектом системы (сети) связывают некоторую информацию - число, строку символов, - идентифицирующую объект. Эта информация является идентификатором объекта системы (сети). Объект, имеющий зарегистрированный идентификатор, считается законным (легальным);

9. Идентификация объекта. Идентификация объекта - это процедура распознавания объекта. Выполняется при попытке объекта войти в систему. Следующий этап взаимодействия системы с объектом – аутентификация;

10. Аутентификация объекта. Аутентификация объекта - это проверка подлинности объекта. Процедура аутентификации устанавливает, является ли объект именно тем, кем он себя объявил. После идентификации и аутентификации объекта выполняют авторизацию;

11. Авторизация объекта. Авторизация объекта - это процедура предоставления объекту, успешно прошедшему процедуры идентификации и аутентификации, соответствующих полномочий и прав доступа к ресурсам системы;

12. Угроза безопасности. Под угрозой безопасности для системы понимаются возможные воздействия, которые прямо или косвенно могут нанести ущерб ее безопасности;

13. Ущерб безопасности. Ущерб безопасности подразумевает нарушение состояния защищенности информации, содержащейся и обрабатывающейся в системе. С понятием угрозы безопасности тесно связано понятие уязвимости автоматизированной системы;

14. Уязвимость системы. Уязвимость системы — это любая характеристика автоматизированной системы, использование которой может привести к реализации угрозы безопасности;

15. Атака на систему. Атака на автоматизированную систему - это действие, предпринимаемое злоумышленником с целью поиска и использования той или иной уязвимости системы. Таким образом, атака - это реализация угрозы безопасности;

16. Противодействие угрозам безопасности. Противодействие угрозам безопасности - цель, которую призваны выполнить средства защиты автоматизированных систем управления;

17. Безопасная или защищенная система. Безопасная или защищенная система - это система со средствами защиты, которые успешно и эффективно противостоят угрозам безопасности. Комплекс средств защиты представляет собой совокупность программных и технических средств системы. Комплекс средств защиты создается и поддерживается в соответствии с принятой в данной организации политикой обеспечения информационной безопасности системы;

18. Политика безопасности. Политика безопасности - это совокупность норм, правил и практических рекомендаций, регламентирующих работу средств защиты автоматизированной системы от заданного множества угроз безопасности. Политика безопасности представляет собой набор норм, правил и практических рекомендаций, на которых строятся управление, защита и распределение информации в автоматизированной системе управления. Политика безопасности регламентирует эффективную работу средств защиты системы. Она охватывает все особенности процесса обработки информации, определяя поведение системы в различных ситуациях. Политика безопасности реализуется посредством комплексного применения администра-

тивно - организационных мер, физических мер и программно - аппаратных средств и определяет архитектуру системы защиты. Для конкретной организации политика безопасности должна носить индивидуальный характер и зависеть от конкретной технологии обработки информации и используемых программных и технических средств. Политика безопасности зависит от способа управления доступом, определяющего порядок доступа к объектам системы. Различают два основных вида политики безопасности:

а) Избирательная политика безопасности. Избирательная политика безопасности основана на избирательном способе управления доступом. Избирательное, или дискреционное, управление доступом характеризуется задаваемым администратором множеством разрешенных отношений доступа (например, в виде троек <объект, субъект, тип доступа>). Обычно для описания свойств избирательного управления доступом применяют математическую модель на основе матрицы доступа. Матрица доступа представляет собой матрицу, в которой столбец соответствует объекту системы, а строка - субъекту. На пересечении столбца и строки указывается тип разрешенного доступа субъекта к объекту. Обычно выделяют такие типы, как «доступ на чтение», «доступ на запись», «доступ на исполнение» и т.п. Матрица доступа - самый простой подход к моделированию систем управления доступом, однако она является основой сложных моделей, более адекватно описывающих реальные автоматизированные системы управления. Избирательная политика безопасности иногда применяется в автоматизированных системах коммерческого сектора, так как ее реализация соответствует требованиям некоторых коммерческих организаций по разграничению доступа и подотчетности, а также приемлема по стоимости;

б) Полномочная политика безопасности. Полномочная политика безопасности основана на полномочном (мандатном) способе управления доступом. Полномочное, или мандатное, управление доступом характеризуется совокупностью правил предоставления доступа, базирующихся на множестве атрибутов безопасности субъектов и объектов, например в зависимости от метки конфиденциальности информации и уровня допуска пользователя. Полномочное управление доступом подразумевает, что:

- все субъекты и объекты системы однозначно идентифицированы;

- каждому объекту системы присвоена метка конфиденциальности, определяющая ценность содержащейся в нем информации;

- каждому субъекту системы присвоен некий уровень допуска, определяющий максимальное значение метки конфиденци-

альности информации объектов, к которым субъект имеет доступ.

Чем важнее объект, тем выше его метка конфиденциальности. Поэтому самыми защищенными оказываются объекты с наиболее высокими значениями метки конфиденциальности. Основным назначением полномочной политики безопасности являются регулирование доступа субъектов системы к объектам с различными уровнями конфиденциальности, предотвращение утечки информации с верхних уровней должностной иерархии на нижние, а также блокирование возможных проникновений с нижних уровней на верхние. При этом полномочная политика может функционировать на фоне избирательной, придавая ее требованиям иерархически упорядоченный характер в соответствии с уровнями безопасности.

Современные автоматизированные системы управления относятся к распределенным системам, осуществляющим автоматизированную обработку информации. Проблема обеспечения информационной безопасности является центральной для таких систем. Обеспечение безопасности АСУ предполагает организацию противодействия любому несанкционированному вторжению в процесс функционирования системы, а также попыткам модификации, хищения, вывода из строя или разрушения ее компонентов, то есть защиту всех компонентов автоматизированной системы - аппаратных средств, программного обеспечения, данных и персонала. Существуют два подхода к проблеме обеспечения безопасности системы:

1. Фрагментарный подход. Фрагментарный подход направлен на противодействие четко определенным угрозам в заданных условиях. В качестве примеров реализации такого подхода можно указать отдельные средства управления доступом, автономные средства шифрования, специализированные антивирусные программы и т.п. Достоинство этого подхода заключается в высокой избирательности к конкретной угрозе. Существенным недостатком его является отсутствие единой защищенной среды обработки информации. Фрагментарные меры защиты информации обеспечивают защиту конкретных объектов автоматизированной системы только от конкретной угрозы. Даже небольшое видоизменение угрозы ведет к потере эффективности защиты;

2. Комплексный подход. Комплексный подход ориентирован на создание защищенной среды обработки информации в АСУ, сводящей воедино разнородные меры противодействия угрозам. Организация защищенной среды обработки информации позволяет гарантировать определенный уровень безопасности автоматизированной системы, что можно отнести к несомненным достоинствам комплексного подхода. К его недостаткам относятся ограничения на свободу действий пользователей системы, чувствительность к ошибкам установки и настройки средств защиты,

сложность управления. Комплексный подход применяют для защиты автоматизированных систем управления крупных организаций или небольших систем, выполняющих ответственные задачи или обрабатывающих особо важную информацию. Нарушение безопасности информации в АСУ крупных организаций может нанести огромный материальный ущерб как самим организациям, так и их клиентам. Поэтому такие организации вынуждены уделять особое внимание гарантиям безопасности и реализовывать комплексную защиту. Комплексного подхода придерживается большинство государственных и крупных коммерческих предприятий и учреждений. Этот подход нашел свое отражение в различных стандартах. Комплексный подход к проблеме обеспечения безопасности основан на разработанной для конкретной автоматизированной системы политике безопасности.

Помимо управления доступом субъектов к объектам системы проблема защиты информации имеет еще один аспект. Для получения информации о каком-либо объекте системы совсем необязательно искать пути несанкционированного доступа к нему. Необходимые сведения можно собрать, наблюдая за обработкой требуемого объекта, то есть используя каналы утечки информации. В системе всегда существуют информационные потоки. Поэтому администратору необходимо определить, какие информационные потоки в системе являются «легальными» -, то есть не ведут к утечке информации, а какие ведут. Как следствие, возникает необходимость разработки правил, регламентирующих управление информационными потоками в системе. Обычно оно применяется в рамках избирательной или полномочной политики, дополняя её и способствуя повышению надежности системы защиты. Комплексный подход к решению проблемы обеспечения безопасности при рациональном сочетании избирательного и полномочного управления доступом, а также управления информационными потоками служит тем фундаментом, на котором строится вся система защиты.

§ 3. Защита информации

Защита информации – это комплекс мероприятий, направленный на обеспечение целостности и конфиденциальности информационного обеспечения АСУ. Целью защиты информации в автоматизированных системах управления является сведение к минимуму потерь в управлении, вызванных нарушением целостности и конфиденциальности информационного обеспечения АСУ. В информационном обеспечении современных АСУ поддерживается один из двух наиболее общих подходов к вопросу обеспечения безопасности данных: избирательный подход и обязательный подход. В обоих подходах единицей данных или объектом данных для которых должна быть создана система безопасности,

может быть как вся база данных целиком, так и любой объект внутри базы данных.

Эти два подхода отличаются следующими свойствами:

1. В случае избирательного управления некоторый пользователь обладает различными правами (привилегиями или полномочиями) при работе с данными объектами. Разные пользователи могут обладать разными правами доступа к одному и тому же объекту. Избирательные права характеризуются значительной гибкостью;

2. В случае обязательного управления, наоборот, каждому объекту данных присваивается некоторый уровень конфиденциальности, а каждый пользователь обладает некоторым уровнем допуска. При таком подходе доступом к определенному объекту данных обладают только пользователи с соответствующим уровнем допуска.

Для реализации избирательного принципа предусмотрены следующие методы:

1. В базу данных вводится новый тип объектов базы данных — это пользователи. Каждому пользователю в базе данных присваивается уникальный идентификатор. Для дополнительной защиты каждый пользователь кроме уникального идентификатора снабжается уникальным паролем, причем если идентификаторы пользователей в системе доступны системному администратору, то пароли пользователей хранятся чаще всего в специальном кодированном виде и известны только самим пользователям;

2. Пользователи могут быть объединены в специальные группы пользователей. Один пользователь может входить в несколько групп. В базе данных вводится понятие публичной группы, для которой должен быть определен минимальный стандартный набор прав. По умолчанию предполагается, что каждый вновь создаваемый пользователь, если специально не указано иное, относится к публичной группе;

3. Привилегии или полномочия пользователей или групп — это набор действий (операций), которые они могут выполнять над объектами базы данных;

4. В последних версиях ряда коммерческих систем управления базами данных появилось понятие «роли». Роль — это поименованный набор полномочий. Существует ряд стандартных ролей, которые определены в момент установки сервера баз данных. Имеется возможность создавать новые роли, группируя в них произвольные полномочия. Введение ролей позволяет упростить управление привилегиями пользователей, структурировать этот процесс. Кроме того, введение ролей не связано с конкретными пользователями, поэтому роли могут быть определены и сконфигурированы до того, как определены пользователи системы;

5. Пользователю может быть назначена одна или несколько ролей;

6. Объектами базы данных, которые подлежат защите, являются все объекты, хранимые в базе данных: таблицы, представления, хранимые процедуры и триггеры. Для каждого типа объектов есть свои действия, поэтому для каждого типа объектов могут быть определены разные права доступа.

На самом элементарном уровне концепции обеспечения безопасности баз данных в АСУ исключительно просты. Необходимо поддерживать два фундаментальных принципа: проверку полномочий и проверку подлинности (аутентификацию).

Проверка полномочий основана на том, что каждому пользователю или процессу автоматизированной информационной системы соответствует набор действий, которые он может выполнять по отношению к определенным объектам. Проверка подлинности означает достоверное подтверждение того, что пользователь или процесс, пытающийся выполнить санкционированное действие, действительно тот, за кого он себя выдает.

Система назначения полномочий имеет в некотором роде иерархический характер. Самыми высокими правами и полномочиями обладает системный администратор или администратор сервера БД. Традиционно только этот тип пользователей может создавать других пользователей и наделять их определенными полномочиями.

Система управления базами данных в своих системных каталогах хранит как описание самих пользователей, так и описание их привилегий по отношению ко всем объектам.

Далее схема предоставления полномочий строится по следующему принципу. Каждый объект в базе данных имеет владельца — пользователя, который создал данный объект. Владелец объекта обладает всеми правами - полномочиями на данный объект, в том числе он имеет право предоставлять другим пользователям полномочия по работе с данным объектом или забирать у пользователей ранее предоставленные полномочия.

В ряде систем управления базами данных вводится следующий уровень иерархии пользователей — это администратор баз данных. В этих системах управления базами данных один сервер может управлять множеством баз данных (например, MS SQL Server, Sybase).

В автоматизированных системах управления применяются принципы ситуационного управления защищенностью информационного обеспечения: требуемый уровень безопасности информации устанавливается в соответствии с ситуацией, определяющей соотношение между ценностью перерабатываемой информации, затратами, необходимыми для его достижения и возможными суммарными потерями от искажения и несанкционированного использования информации.

Полномочия пользователей хранятся в специальных системных таблицах, и их проверка осуществляется ядром системы управления базами данных при выполнении каждой операции. Логически для каждого пользователя и каждого объекта в базе данных как бы строится некоторая условная матрица, где по одному измерению расположены объекты, а по другому — пользователи. На пересечении каждого столбца и каждой строки расположен перечень разрешенных операций для данного пользователя над данным объектом. С первого взгляда кажется, что эта модель проверки достаточно устойчивая. Но сложность возникает тогда, когда мы используем косвенное обращение к объектам. Например: пользователю не разрешен доступ к какой-либо таблице базы данных, но этому же пользователю разрешен запуск хранимой процедуры, которая делает выборку из этой таблицы. По умолчанию все хранимые процедуры запускаются под именем их владельца.

Такие проблемы должны решаться организационными методами. При разрешении доступа некоторых пользователей необходимо помнить о возможности косвенного доступа. В любом случае проблема защиты никогда не была чисто технической задачей, это комплекс организационно - технических мероприятий, которые должны обеспечить максимальную конфиденциальность информации, хранимой в базе данных. Кроме того, при работе в сети существует еще проблема проверки подлинности полномочий. Эта проблема состоит в следующем. Допустим, одному процессу даны полномочия по работе с базой данных, а другому процессу такие полномочия не даны. Тогда напрямую второй процесс не может обратиться к базе данных, но он может обратиться к первому процессу и через него получить доступ к информации из базы данных. Поэтому в безопасной среде должна присутствовать модель проверки подлинности, которая обеспечивает подтверждение заявленных пользователями или процессами идентификаторов. Проверка полномочий приобрела еще большее значение в условиях массового распространения распределенных вычислений. При существующем высоком уровне связности автоматизированных систем необходимо контролировать все обращения к системе.

В целостной системе информационной безопасности, где четкое выполнение программы защиты информации обеспечивается за счет взаимодействия соответствующих средств в операционных системах, сетях, базах данных, проверка подлинности имеет прямое отношение к безопасности баз данных. Заметим, что модель безопасности, основанная на базовых механизмах проверки полномочий и проверки подлинности, не решает таких проблем, как украденные пользовательские идентификаторы и пароли или злонамеренные действия некоторых пользователей, обладающих полномочиями. Например, программист, работающий

над учетной системой, и имеющей полный доступ к базе данных, встраивает в код программы «троянского коня» с целью хищения или намеренного изменения информации, хранимой в базе данных. Программа обеспечения информационной безопасности должна охватывать не только технические области (такие как защита сетей, баз данных и операционных систем), но и проблемы физической защиты, надежности персонала (скрытые проверки), аудит, различные процедуры поддержки безопасности, выполняемые вручную или частично автоматизированные.

§ 4. Проблемы безопасности автоматизированных систем

Новые информационные технологии активно внедряются во все сферы народного хозяйства. Появление локальных и глобальных сетей передачи данных предоставило пользователям ЭВМ новые возможности оперативного обмена информацией. Если до недавнего времени подобные сети создавались только в специфических и узконаправленных целях (академические сети, сети военных ведомств и т.д.), то развитие Internet и аналогичных систем привело к использованию глобальных сетей передачи данных в повседневной жизни практически каждого человека. По мере развития и усложнения средств, методов и форм автоматизации процессов обработки информации повышается зависимость общества от степени безопасности используемых им информационных технологий. Актуальность и важность проблемы обеспечения информационной безопасности обусловлена следующими факторами:

1. Высокие темпы роста парка персональных компьютеров, применяемых в самых разных сферах деятельности;
2. Резкое расширение круга пользователей, имеющих непосредственный доступ к вычислительным ресурсам и массивам данных;
3. Увеличение объемов информации, накапливаемой, хранимой и обрабатываемой с помощью компьютеров и других средств автоматизации;
4. Сосредоточение в единых базах данных информации различного назначения и различной принадлежности;
5. Бурное развитие программных средств, не удовлетворяющих даже минимальным требованиям безопасности;
6. Повсеместное распространение сетевых технологий и объединение локальных сетей в глобальные;
7. Развитие глобальной сети Internet, практически не препятствующей нарушениям безопасности систем обработки информации во всем мире.

Современные методы накопления, обработки и передачи информации способствовали появлению угроз, связанных с возможностью потери, искажения и раскрытия данных, адресован-

ных или принадлежащих конечным пользователям. Поэтому обеспечение информационной безопасности автоматизированных систем управления является одним из ведущих направлений развития информационных технологий.

Набор протоколов TCP/IP применяется для организации коммуникаций в неоднородной сетевой среде, обеспечивая совместимость между ЭВМ разных типов. Совместимость - одно из основных преимуществ TCP/IP, поэтому большинство локальных сетей ЭВМ поддерживает эти протоколы. Кроме того, протоколы TCP/IP предоставляют доступ к ресурсам глобальной сети Internet. Поскольку TCP/IP поддерживает маршрутизацию пакетов, он обычно используется в качестве межсетевого протокола. Благодаря своей популярности TCP/IP стал стандартом дефакто для межсетевого взаимодействия.

Однако повсеместное распространение стека протоколов TCP/IP обнажило и его слабые стороны. Создавая свое детище, архитекторы стека TCP/IP не видели причин особенно беспокоиться о защите сетей, строящихся на его основе. Они и не предполагали, что когда-нибудь отсутствие эффективных средств защиты станет основным фактором, сдерживающим применение протоколов TCP/IP.

Рассмотрим более подробно проблемы недостаточной информационной безопасности протоколов TCP/IP, IP-сетей и служб Internet. Эти пороки являются «врожденными» практически для всех протоколов стека TCP/IP и служб Internet. Большая часть этих проблем связана с исторической зависимостью Internet от операционной системы UNIX. ARPAnet (прародитель сети Internet) строилась как сеть, связывающая исследовательские центры, научные, военные и правительственные учреждения, крупные университеты США. Эти структуры использовали операционную систему UNIX в качестве платформы для коммуникаций и решения собственных задач. Поэтому особенности методологии программирования в среде UNIX и ее архитектуры наложили отпечаток на реализацию протоколов обмена TCP/IP и политики безопасности в сети. Из-за открытости и распространенности система UNIX оказалась любимой добычей хакеров. Поэтому не удивительно, что набор протоколов TCP/IP имеет «врожденные» недостатки защиты.

На практике IP-сети уязвимы для ряда способов несанкционированного вторжения в процесс обмена данными. По мере развития информационных и сетевых технологий (например, с появлением мобильных Java-приложений и элементов ActiveX) список возможных типов сетевых атак на IP-сети постоянно расширяется. Сетевые атаки столь же разнообразны, как и системы, против которых они направлены. Некоторые атаки отличаются большой сложностью. Другие способен осуществить обычный оператор, даже не предполагающий, к каким последст-

виям может привести его деятельность. На сегодняшний день наиболее распространенными являются следующие варианты атак:

1. Подслушивание. Большинство данных передается по сетям ЭВМ в незащищенном формате (открытым текстом), что позволяет злоумышленнику, получившему доступ к линиям передачи информации сети, подслушивать или считывать трафик. Подслушивание в сетях ЭВМ называется слежением (sniffing, или snooping). Если не использовать служб, обеспечивающих устойчивое шифрование, то передаваемые по сети данные будут доступны для чтения. Для подслушивания в компьютерных сетях могут использоваться так называемые снифферы пакетов. Сниффер пакетов представляет собой прикладную программу, перехватывающую все сетевые пакеты, которые передаются через определенный домен сети. В настоящее время снифферы работают в сетях на вполне законном основании. Они используются для диагностики неисправностей и анализа трафика. Однако ввиду того, что некоторые сетевые приложения передают данные в текстовом формате (Telnet, FTP, SMTP, POP3 и т.д.), с помощью сниффера можно узнать полезную, а иногда и конфиденциальную информацию (например, имена пользователей и пароли);

2. Парольные атаки. Хакеры могут проводить парольные атаки с помощью целого ряда методов, таких как IP-спуфинг и сниффинг пакетов, атака полного перебора (brute force attack), «троянский конь». Перехват паролей и имен пользователей, передаваемых по сети в незашифрованной форме, путем «подслушивания» канала (password sniffing) создает большую опасность, так как пользователи часто применяют один и тот же логин и пароль для множества прикладных программ и систем. Многие пользователи вообще имеют один пароль для доступа ко всем ресурсам и прикладным программам. Если прикладная программа является распределенной, что имеет место в большинстве автоматизированных систем управления, то есть работает в режиме «клиент-сервер», а аутентификационные данные передаются по сети в читаемом текстовом формате, эту информацию с большой вероятностью можно использовать для доступа к другим корпоративным или внешним ресурсам. Хотя логин и пароль часто можно получить при помощи IP-спуфинга и сниффинга пакетов, хакеры часто пытаются подобрать пароль и логин, используя для этого многочисленные попытки доступа. Часто для атаки полного перебора используется специальная программа, которая дает возможность получить доступ к ресурсу общего пользования (например, к серверу). В результате хакер допускается к ресурсам на правах обычного пользователя, пароль которого был подобран. Если этот пользователь имеет значительные привилегии доступа, хакер может

создать себе «проход» для будущего доступа, который будет открыт, даже если пользователь изменит свой пароль и логин;

3. Изменение данных. Злоумышленник, получивший возможность прочитать ваши данные, сможет сделать и следующий шаг - изменить их. Данные в пакете могут быть изменены, даже когда нарушитель ничего не знает ни об отправителе, ни о получателе. Даже если пользователь не нуждается в строгой конфиденциальности передаваемой информации, наверняка он не захочет, чтобы его данные были изменены по пути;

4. «Угаданный ключ защиты информации». Ключ защиты информации представляет собой код или число, необходимое для расшифровки защищенной криптографическим способом информации. Хотя узнать ключ доступа трудно и требуются большие затраты ресурсов, тем не менее это возможно. Ключ, к которому получает доступ атакующий, называется скомпрометированным. Атакующий использует скомпрометированный ключ для получения доступа к защищенным передаваемым данным без ведома отправителя и получателя. Ключ дает право расшифровывать и изменять данные, а также вычислять другие ключи, которые могут дать атакующему доступ и к другим защищенным соединениям;

5. Подмена доверенного субъекта. Большая часть сетей и операционных систем используют IP-адрес компьютера для того, чтобы определять, тот ли это адресат, который нужен. В некоторых случаях возможно некорректное присвоение IP-адреса (подмена IP-адреса отправителя другим адресом) - такой способ атаки называют фальсификацией адреса (IP spoofing). Фальсификация адреса происходит, когда хакер, находящийся внутри корпорации или вне ее, выдает себя за санкционированного пользователя. Это можно сделать двумя способами. Во-первых, нарушитель может воспользоваться IP-адресом, находящимся в пределах диапазона санкционированных IP-адресов, или авторизованным внешним адресом, которому разрешается доступ к определенным сетевым ресурсам. Хакер может применять специальные программы, формирующие IP-пакеты таким образом, чтобы они выглядели как исходящие с разрешенных внутренних адресов корпоративной сети. После получения доступа к вашей сети с разрешенным адресом атакующий может изменять, перенаправлять или удалять ваши данные. Атаки IP-спуфинга часто являются отправной точкой для других атак. Обычно IP-спуфинг ограничивается вставкой ложной информации или вредоносных команд в обычный поток данных, передаваемых между клиентским и серверным приложением или по каналу связи между одноранговыми устройствами. Для двусторонней связи хакер должен изменить все таблицы маршрутизации, чтобы направить трафик на ложный IP-адрес. Если нарушителю это удается, он получает все пакеты и может от-

вечать на них так, будто является санкционированным пользователем;

6. Перехват сеанса (session hijacking). Для осуществления перехвата сеанса по окончании начальной процедуры аутентификации хакер переключает установленное соединение, а исходному серверу выдается команда разорвать соединение. В результате ваш «собеседник» оказывается незаметно подмененным. После получения доступа к сети злоумышленник получает большую свободу действий. Он может:

а) Посылать некорректные данные приложениям и сетевым службам, что приводит к их аварийному завершению или неправильному функционированию;

б) Наводнить компьютер или всю сеть трафиком, пока не произойдет останов системы в связи с перегрузкой;

в) Блокировать трафик, что приведет к потере доступа авторизованных пользователей к сетевым ресурсам;

7. Посредничество. Атака типа «посредничество» подразумевает активное подслушивание, перехват и управление передаваемыми данными невидимым промежуточным узлом. Когда компьютеры взаимодействуют на низких сетевых уровнях, они не всегда могут определить, с кем именно обмениваются данными;

8. Посредничество в обмене незашифрованными (открытыми) ключами защиты информации (man-in-the-middle). Если атаки предыдущих типов увенчались успехом, их автор может вмешаться в процесс передачи ключей между сторонами и подставить им собственный ключ для дальнейшего использования. Вообще для атаки типа man-in-the-middle хакеру нужен доступ к пакетам, передаваемым по сети. Такой доступ ко всем пакетам, передаваемым от провайдера в любую другую сеть, может, к примеру, получить сотрудник этого провайдера. Для атак этого типа часто используются снифферы пакетов, транспортные протоколы и протоколы маршрутизации. Атаки проводятся с целью кражи информации, перехвата текущей сессии и получения доступа к частным сетевым ресурсам, для анализа трафика и получения информации о сети и ее пользователях, для проведения атак типа «отказа в обслуживании», искажения передаваемых данных и ввода несанкционированной информации в сетевые сессии;

9. Отказ в обслуживании. Атаки типа «отказ в обслуживании» являются наиболее известной формой хакерских атак. Эти атаки не нацелены на получение доступа к вашей сети или к какой-либо информации из нее. Атака «отказ в обслуживании» делает вашу сеть недоступной для обычного использования за счет превышения допустимых пределов функционирования сети, операционной системы или прикладной программы. В случае использования некоторых серверных прикладных программ

(таких как Web- или PTP-сервер) атаки «отказ в обслуживании» могут заключаться в том, чтобы занять все соединения, доступные для этих приложений, и держать их в занятом состоянии, не допуская обслуживания пользователей. В ходе атак могут использоваться обычные Internet-протоколы, например TCP и ICMP (Internet Control Message Protocol). Большинство атак «отказ в обслуживании» опирается на общие слабости системной архитектуры. Некоторые атаки «отказ в обслуживании» сводят к нулю производительность сети, переполняя ее нежелательными и ненужными пакетами или сообщая ложную информацию о текущем состоянии сетевых ресурсов. Если атака этого типа проводится одновременно через множество устройств, ее называют распределенной атакой «отказ в обслуживании». Среди хакеров атаки «отказ в обслуживании» считаются тривиальными, потому что для их организации требуется минимум знаний и умений. Тем не менее именно простота реализации и огромный причиняемый вред привлекают к «отказу в обслуживании» пристальное внимание администраторов, отвечающих за сетевую безопасность. Против атак такого типа трудно создать стопроцентную защиту. Их нелегко предотвратить, так как для этого требуется четкая координация действий с провайдером;

10. Атаки на уровне прикладных программ. Такие атаки могут проводиться несколькими способами. Самый распространенный из них состоит в использовании слабостей серверного программного обеспечения (Sendmail, HTTP, FTP). В результате хакеры могут получить доступ к компьютеру от имени пользователя, работающего с прикладной программой (обычно это не простой пользователь, а привилегированный администратор с правами системного доступа). Сведения об атаках на уровне прикладных программ широко публикуются, чтобы дать возможность администраторам предотвратить их с помощью коррекционных модулей (патчей). Однако многие хакеры также имеют доступ к этим сведениям, что позволяет им учиться. Главная проблема с атаками на уровне прикладных программ состоит в том, что они часто пользуются портами, которым разрешен проход через межсетевой экран. К примеру, хакер, эксплуатирующий известную слабость Web-сервера, часто использует в ходе атаки TCP-порт 80. Поскольку Web-сервер открывает пользователям Web-страницы, межсетевой экран должен предоставлять доступ к этому порту. Межсетевой экран рассматривает такую атаку как стандартный трафик для порта 80. Полностью исключить атаки на уровне прикладных программ невозможно. Хакеры постоянно обнаруживают все новые уязвимые места прикладных программ и публикуют в Internet информацию о них. Поэтому очень важно организовать хорошее системное администрирование сети;

11. Злоупотребление доверием. Этот тип действий не является в полном смысле атакой. Такие действия представляют собой злонамеренное использование отношений доверия, существующих в сети. Классическим примером такого злоупотребления является ситуация в периферийной части корпоративной сети. В этом сегменте часто располагаются серверы DNS, SMTP и HTTP. Поскольку все они принадлежат к одному и тому же сегменту, взлом одного из них приводит к взлому и всех остальных, так как эти серверы доверяют другим системам своей сети. В качестве другого примера приведем систему, установленную с внешней стороны межсетевого экрана и имеющую отношения доверия с системой, расположенной с его внутренней стороны. В случае взлома внешней системы хакер может использовать отношения доверия для проникновения в систему, защищенную межсетевым экраном. Риск злоупотребления доверием можно уменьшить за счет более жесткого контроля уровней доверия в пределах сети. Системы, расположенные с внешней стороны межсетевого экрана, не должны пользоваться абсолютным доверием со стороны защищенных экраном систем. Отношения доверия должны ограничиваться определенными протоколами и аутентифицироваться не только по IP-адресам, но и по другим параметрам;

12. Вирусы и прикладные программы типа «троянский конь». Рабочие станции конечных пользователей уязвимы для вирусов и «троянских коней». Вирусами называются вредоносные программы, которые внедряются в другие программы для выполнения определенной нежелательной функции на рабочей станции конечного пользователя. В качестве примера можно привести вирус, который прописывается в файле command.com (главном интерпретаторе систем Windows) и удаляет другие файлы, а также заражает все найденные им версии command.com. «Троянский конь» представляет собой не программную вставку, а настоящую программу, которая выглядит как полезное приложение, на деле выполняя разрушительную акцию. Примером типичного «троянского коня» является программа, которая кажется обычной игрой для рабочей станции пользователя. Однако пока пользователь играет в эту «игру», вредоносная программа отправляет свою копию по электронной почте каждому абоненту, занесенному в адресную книгу пользователя. Все абоненты получают по почте «игру» и невольно способствуют ее дальнейшему распространению. Борьба с вирусами и «троянскими конями» ведется с помощью эффективного антивирусного программного обеспечения, работающего на пользовательском или сетевом уровне. Антивирусные средства способны обнаружить большинство вирусов и «троянских коней» и пресечь их распространение. Регулярное получение и использование самой свежей информации о вирусах помогает эф-

фективно бороться с ними. По мере появления очередных вирусов и «троянских коней» необходимо устанавливать новые версии антивирусных средств и приложений;

13. Сетевая разведка. Сетевой разведкой называется сбор информации о сети с помощью общедоступных данных и прикладных программ. При подготовке атаки против какой-либо сети хакер, как правило, пытается получить о ней как можно больше информации. Сетевая разведка проводится в форме запросов DNS, эхо-тестирования (ping sweep) и сканирования портов. Запросы DNS помогают понять, кто владеет тем или иным доменом и какие адреса этому домену присвоены. Эхо-тестирование адресов, раскрытых с помощью DNS, позволяет увидеть, какие ЭВМ реально работают в данной среде. Получив список машин, хакер использует средства сканирования портов, чтобы составить полный список услуг, поддерживаемых этими компьютерами. И, наконец, «разведчик» анализирует характеристики прикладных программ, работающих на этих машинах. В результате добывается информация, которую можно использовать для реализации атаки.

Нетрудно видеть, что перечисленные выше атаки возможны в силу ряда причин:

1. Аутентификация отправителя осуществляется исключительно по его IP-адресу;

2. Процедура аутентификации выполняется только на стадии установления соединения - в дальнейшем подлинность принимаемых пакетов не проверяется;

3. Важнейшие данные, имеющие отношение к системе, передаются по сети в незашифрованном виде.

Ряд распространенных служб Internet также характеризуется врожденными слабостями. К числу служб Internet относятся:

1. Простой протокол передачи электронной почты SMTP (Simple Mail Transfer Protocol). Простой протокол передачи электронной почты SMTP позволяет осуществлять почтовую транспортную службу Internet. Одна из проблем безопасности, связанная с этим протоколом, заключается в том, что пользователь не может проверить адрес отправителя в заголовке электронного письма. В результате хакер способен направить во внутреннюю сеть большое количество почтовых сообщений, что приведет к перегрузке и блокированию работы почтового сервера;

2. Программа электронной почты Sendmail. Популярная в Internet программа электронной почты Sendmail использует для работы некоторую сетевую информацию - IP-адрес отправителя. Перехватывая сообщения, отправляемые с помощью Sendmail, хакер может употребить эту информацию для нападений, например для спуфинга (подмены адресов);

3. Служба сетевых имен DNS (Domain Name System). Служба сетевых имен DNS представляет собой распределенную базу данных, которая преобразует имена пользователей и ЭВМ в IP-адреса, указываемые в заголовках пакетов, и наоборот. DNS также хранит информацию о структуре сети компании, например о количестве компьютеров с IP-адресами в каждом домене. Одна из проблем DNS заключается в том, что эту базу данных очень трудно скрыть от неавторизированных пользователей. В результате DNS часто используется хакерами как источник информации об именах доверенных узлов сети;

4. Служба эмуляции удаленного терминала Telnet. Служба эмуляции удаленного терминала Telnet употребляется для подключения к удаленным системам, присоединенным к сети. Она применяет базовые возможности эмуляции терминала. При использовании этого сервиса Internet пользователи должны регистрироваться на сервере Telnet, вводя свои имя и пароль. После аутентификации пользователя его рабочая станция функционирует в режиме терминала, подключенного к внешнему компьютеру. С этого терминала пользователь может вводить команды, которые обеспечивают ему доступ к файлам и запуск программ. Подключившись к серверу Telnet, хакер может сконфигурировать его программу таким образом, чтобы она записывала имена и пароли пользователей;

5. Всемирная паутина WWW (World-Wide Web). Всемирная паутина WWW - это система, основанная на сетевых приложениях, которые позволяют пользователям просматривать содержимое различных серверов в Internet или интрасетях. Самое полезное свойство WWW - использование гипертекстовых документов, в которые встроены ссылки на другие документы и Web-узлы, что дает посетителям сайтов возможность легко переходить от одного узла к другому. Однако это же свойство является и наиболее слабым местом системы WWW, поскольку ссылки на Web-узлы, хранящиеся в гипертекстовых документах, содержат информацию о том, как осуществляется доступ к соответствующим узлам. Используя эту информацию, хакеры могут разрушить Web-узел или получить доступ к хранящейся в нем конфиденциальной информации;

6. Протокол передачи файлов FTP (File Transfer Protocol). Протокол передачи файлов FTP обеспечивает передачу текстовых и двоичных файлов, поэтому его часто используют в Internet для организации совместного доступа к информации. Его обычно рассматривают как один из методов работы с удаленными сетями. На FTP-серверах хранятся документы, программы, графики и другие виды информации. К данным этих файлов на FTP-серверах нельзя обратиться напрямую. Это можно сделать, только переписав их целиком с FTP-сервера на локальный сервер. Некоторые FTP-серверы ограничивают доступ

пользователей к своим архивам данных с помощью пароля, другие же предоставляют свободный доступ (так называемый анонимный FTP-сервер). При использовании опции анонимного FTP для своего сервера пользователь должен быть уверен, что на нем хранятся только файлы, предназначенные для свободного распространения;

7. Графическая оконная система X Windows и др.

Рассмотрим основные причины уязвимости сети Internet. Это позволит лучше понять уязвимость сетей и отдельных компьютеров, имеющих доступ к Internet:

1. Сеть Internet разрабатывалась как открытая и децентрализованная сеть с изначальным отсутствием политики безопасности. При этом основные усилия были направлены на достижение удобства обмена информацией в Internet. Кроме того, многие сети спроектированы без механизмов контроля доступа со стороны Internet;

2. Для Internet характерны большая протяженность линий связи и уязвимость основных служб. Сервисные программы базового набора протоколов TCP/IP сети Internet не гарантируют безопасности;

3. Модель «клиент-сервер», на которой основана работа в Internet, не лишена определенных слабостей и лазеек в продуктах отдельных производителей. Данная модель объединяет разнообразное программное и аппаратное обеспечение, в котором могут быть «дыры» для проникновения злоумышленников;

4. При создании Web-страниц ряд компаний использует собственный дизайн, который может не соответствовать требованиям обеспечения определенного класса безопасности для Web-узла компании и связанной с ним локальной или корпоративной сети;

5. Информация о существующих и используемых средствах защиты доступна пользователям. Кроме того, возможна утечка технологий безопасности высокого уровня из закрытых источников при вскрытии представленных в сети Web-узлов и сетей организаций, занимающихся разработкой этих технологий;

6. Существует возможность наблюдения за каналами передачи данных, поскольку значительная часть информации передается через Internet в открытой незащищенной форме. В частности, электронная почта, пароли и вложенные в письма файлы могут быть легко перехвачены злоумышленником при помощи доступных программ;

7. Средства управления доступом зачастую сложно конфигурировать, настраивать и контролировать. Это приводит к неправильной конфигурации средств защиты и, как следствие, к несанкционированному доступу;

8. Существенную роль играет и человеческий фактор. Отдельные пользователи, не отличающиеся высокими моральными принципами, могут за соответствующую плату предоставить злоумышленникам доступ в сеть своей фирмы. Имеются пользователи - дилетанты, которые, не обладая необходимыми знаниями, считают, что средства защиты им вообще не нужны, или неправильно конфигурируют эти средства;

9. Для обслуживания работы в Internet используется большое число сервисов, информационных служб и сетевых протоколов. Знание правильности и тонкостей использования хотя бы большинства этих сервисов, служб и протоколов одному человеку в лице администратора сети практически недоступно;

10. Специалисты по защите информации в Internet готовятся пока в недостаточном объеме; часто в роли администраторов сети работают люди, не имеющие глубокой профессиональной подготовки;

11. Для работы в Internet характерна кажущаяся анонимность. Существует потенциальная возможность обойти средства обнаружения отправителя той или иной информации либо посетителя того или иного Web-узла с помощью использования виртуальных IP-адресов и промежуточных пересыльщиков электронной почты.

Возникает естественный вопрос: сколько потенциально уязвимых мест может быть у сетей, подключенных к Internet? Специалисты компании Internet Security Systems считают, что в любой сети, основанной на протоколе TCP/IP, существует около 135 потенциальных каналов для несанкционированного доступа. Первые средства защиты передаваемых данных появились практически сразу после того, как уязвимость IP-сетей дала о себе знать на практике. Характерными примерами разработок в этой области могут служить PGP/Web-of-Trust для шифрования сообщений электронной почты, SSL (Secure Sockets Layer - протокол безопасных соединений) для передачи через Internet зашифрованных аутентифицированных сообщений, SSH (Secure SHell – оболочка безопасности) для защиты сеансов Telnet и процедур передачи файлов.

Общим недостатком подобных широко распространенных решений является их «привязанность» к определенному типу прикладных программ, а значит, неспособность удовлетворить тем разнообразным требованиям к системам сетевой защиты, которые предъявляют крупные корпорации или Internet-провайдеры.

Самый радикальный способ преодолеть указанное ограничение сводится к тому, чтобы строить систему защиты не для отдельных классов прикладных программ (пусть и весьма популярных), а для сети в целом. Применительно к IP-сетям это означает, что системы защиты должны действовать на сетевом уровне модели OSI. Преимущество такого выбора заключается в

том очевидном факте, что в IP-сетях именно данный уровень отличается наибольшей гомогенностью: независимо от вышележащих протоколов, физической среды передачи и технологии канального уровня транспортировка данных по сети не может быть произведена в обход протокола IP. Поэтому реализация защиты сети на третьем уровне автоматически гарантирует как минимум такую же степень защиты всех сетевых прикладных программ. При этом не требуется какая-либо их модификация. Для пользователей процедуры защиты окажутся столь же прозрачными, как и сам протокол IP.

Слабые стороны стека протоколов TCP/IP первоначально предполагалось восполнить в шестой версии протокола IP. В 1993 году в составе консорциума IETF была создана рабочая группа IP Security Working Group, занявшаяся разработкой архитектуры и протоколов для шифрования данных, передаваемых по сетям IP версии 6. Однако по мере продвижения в этом направлении становилось все очевиднее, что разработки, изначально ориентированные на IP шестой версии, могут пригодиться и в более традиционной среде IP версии 4. В результате на свет появился набор протоколов IP Sec, основанных на современных технологиях шифрования и электронной цифровой подписи данных. Поскольку архитектура протоколов IP Sec совместима с протоколом IP версии 4, её поддержку достаточно обеспечить на обоих концах соединения, а промежуточные сетевые узлы могут вообще ничего не знать о применении IP Sec.

§ 5. Модель угроз безопасности

В литературе, посвященной вопросам защиты информации, можно найти различные варианты моделей угроз ее безопасности. Это объясняется стремлением более точно описать многообразные ситуации воздействия на информацию и определить наиболее адекватные меры парирования. В принципе можно воспользоваться любой подходящей моделью, необходимо только убедиться, что она учитывает максимальное число факторов, влияющих на безопасность информации.

Что же такое угроза безопасности информации? Это возможность осуществления действия, направленного против объекта защиты, проявляемая в опасности искажений и потерь информации. Надо оговориться, что речь идет не обо всей информации, а только о той её части, которая, по мнению ее собственника (пользователя), имеет коммерческую ценность (информация как товар) или подлежит защите в силу закона (конфиденциальная информация).

Необходимо также учитывать, что источники угроз безопасности могут находиться как внутри фирмы - внутренние ис-

точники, так и вне ее - внешние источники. Такое деление оправдано потому, что для одной и той же угрозы (например, в случае кражи) методы парирования для внешних и внутренних источников будут разными.

При составлении модели угроз безопасности корпоративной сети использованы различные варианты моделей, разработанных специалистами в области защиты информации государственных и негосударственных научных учреждений. Исходя из проведенного анализа, все источники угроз безопасности информации, циркулирующей в корпоративной сети, можно разделить на три основные группы:

1. Угрозы, обусловленные действиями субъекта (антропогенные);
2. Угрозы, обусловленные техническими средствами (техногенные);
3. Угрозы, обусловленные стихийными источниками (стихийные).

Первая группа, самая обширная, представляет наибольший интерес с точки зрения организации парирования угрозам данного типа, так как действия субъекта всегда можно оценить, спрогнозировать и принять адекватные меры. Методы противодействия этим угрозам управляемы и напрямую зависят от воли организаторов защиты информации.

Субъекты, действия которых могут привести к нарушению безопасности информации, могут быть:

1. Внешними:
а) Криминальные структуры;
б) Рецидивисты и потенциальные преступники;
в) Недобросовестные партнеры;
г) Конкуренты;
д) Политические противники;
2. Внутренними:
а) Персонал учреждения;
б) Персонал филиалов;
в) Лица с нарушенной психикой;
г) Специально внедренные агенты.

Судя по результатам международного и российского опыта, действия субъектов могут привести к ряду нежелательных последствий, среди которых применительно к корпоративной сети можно выделить следующие:

1. Кража:
а) Технических средств (винчестеров, ноутбуков, системных блоков);
б) Носителей информации (бумажных, магнитных, оптических и пр.);
в) Информации (чтение и несанкционированное копирование);

г) Средств доступа (ключи, пароли, ключевая документация и пр.);

2. Подмена (модификация):

а) Операционных систем;

б) Систем управления базами данных;

в) Прикладных программ;

г) Информации (данных), отрицание факта отправки сообщений;

д) Паролей и правил доступа;

3. Уничтожение (разрушение):

а) Технических средств (винчестеров, ноутбуков, системных блоков);

б) Носителей информации (бумажных, магнитных, оптических и пр.);

в) Программного обеспечения (ОС, СУБД, прикладного ПО);

г) Информации (файлов, данных);

д) Паролей и ключевой информации;

4. Нарушение нормальной работы (прерывание):

а) Снижение скорости обработки информации;

б) Уменьшение пропускной способности каналов связи;

в) Уменьшение объемов свободной оперативной памяти;

г) Уменьшение объемов свободного дискового пространства;

д) Электропитания технических средств;

5. Ошибки:

а) При инсталляции ПО, ОС, СУБД;

б) При написании прикладного ПО;

в) При эксплуатации ПО;

г) При эксплуатации технических средств;

6. Перехват информации (несанкционированный):

а) За счет паразитных электромагнитных излучений от технических средств;

б) За счет наводок по линиям электропитания;

в) За счет наводок по посторонним проводникам;

г) По акустическому каналу от средств вывода;

д) По акустическому каналу при обсуждении вопросов;

е) При подключении к каналам передачи информации;

ж) За счет нарушения установленных правил доступа (взлом).

Вторая группа содержит угрозы менее прогнозируемые, напрямую зависящие от свойств техники и поэтому требующие особого внимания. Технические средства, содержащие потенциальные угрозы безопасности информации, также могут быть:

1. Внутренними:

а) Некачественные технические средства обработки информации;

б) Некачественные программные средства обработки информации;

в) Вспомогательные средства (охраны, сигнализации, телефонии);

г) Другие технические средства, применяемые в учреждении;

2. Внешними:

а) Средства связи;

б) Близко расположенные опасные производства;

г) Сети инженерных коммуникации (энерго-, водоснабжения, канализации);

д) Транспорт.

Последствиями применения таких технических средств, напрямую влияющими на безопасность информации, могут быть:

1. Нарушение нормальной работы:

а) Нарушение работоспособности системы обработки информации;

б) Нарушение работоспособности связи и телекоммуникаций;

в) Старение носителей информации и средств ее обработки;

г) Нарушение установленных правил доступа;

д) Электромагнитное воздействие на технические средства.

2. Уничтожение (разрушение):

а) Программного обеспечения, ОС, СУБД;

б) Средств обработки информации (броски напряжений, протечки);

в) Помещений;

г) Информации (размагничивание, радиация, протечки и пр.);

д) Персонала.

3. Модификация (изменение):

а) Программного обеспечения, ОС, СУБД;

б) Информации при передаче по каналам связи и телекоммуникациям.

Третью группу составляют угрозы, которые совершенно не поддаются прогнозированию, и поэтому меры их парирования должны применяться всегда. Стихийные источники, составляющие потенциальные угрозы информационной безопасности, как правило, являются внешними по отношению к рассматриваемому объекту, и под ними понимаются, прежде всего, природные катаклизмы:

1. Пожары;

2. Землетрясения;

3. Наводнения;

4. Ураганы;

5. Другие форс-мажорные обстоятельства;

6. Различные непредвиденные обстоятельства;

7. Необъяснимые явления.

Эти природные и необъяснимые явления также влияют на информационную безопасность, опасны для всех элементов корпоративной сети и могут привести к последствиям, перечисленным ниже:

1. Уничтожение (разрушение):

а) Технических средств обработки информации;

б) Носителей информации;

в) Программного обеспечения (ОС, СУБД, прикладного ПО);

г) Информации (файлов, данных);

д) Помещений;

е) Персонала;

2. Исчезновение (пропажа):

а) Информации в средствах обработки;

б) Информации при передаче по телекоммуникационным каналам;

в) Носителей информации;

г) Персонала.

Даже первичный анализ приведенного перечня угроз безопасности информации показывает, что для обеспечения комплексной безопасности необходимо принятие как организационных, так и технических решений парирования. Такой подход позволяет дифференцированно подойти к распределению материальных ресурсов, выделенных на обеспечение информационной безопасности.

Необходимо отметить, что оценить весовые коэффициенты каждой угрозы достаточно трудно из-за высокой латентности их проявлений и отсутствия вразумительной статистики по этому вопросу. Поэтому в современной литературе приводятся различные шкалы оценок. Вместе с тем на основе анализа, проводимого различными специалистами в области компьютерных преступлений, можно расставить угрозы безопасности по частоте проявления следующим образом:

1. Кража (копирование) программного обеспечения;

2. Подмена (несанкционированный ввод) информации;

3. Уничтожение (разрушение) данных на носителях информации;

4. Нарушение нормальной работы (прерывание) в результате вирусных атак;

5. Модификация (изменение) данных на носителях информации;

6. Перехват (несанкционированный съем) информации;

7. Кража (несанкционированное копирование) ресурсов;

8. Нарушение нормальной работы (перегрузка) каналов связи;

9. Непредсказуемые потери.

Несмотря на предложенное ранжирование (примем его только к сведению) для простоты будем считать, что каждая угроза может себя рано или поздно проявить, и поэтому все они равны, то есть при построении модели принято, что весовые коэффициенты каждой угрозы равны единице.

Однако, описание состава угроз безопасности информации, не решает проблемы моделирования их воздействия. Все эти угрозы по-разному проявляются в каждой точке корпоративной сети.

Наложение угроз безопасности информации на модель автоматизированной информационной системы позволяет в первом приближении оценить их опасность и методом исключения выделить наиболее актуальные для конкретного объекта защиты. Кроме того, можно в первом приближении оценить объемы необходимых работ и выбрать магистральное направление по обеспечению защиты информации.

Следствием реализации выявленных угроз безопасности информации в конечном счете может стать ущемление прав собственника (пользователя) информации или нанесение ему материального ущерба, наступившее в результате:

1. Уничтожения информации из-за нарушения программных, аппаратных или программно-аппаратных средств её обработки либо систем защиты, а также из-за форс-мажорных обстоятельств, применения специальных технических (например, размагничивающих генераторов), программных (например, логических бомб) средств воздействия, осуществляемого конкурентами, персоналом учреждения или его филиалов, преступными элементами либо поставщиками средств обработки информации в интересах третьих лиц;

2. Модификации или искажения информации вследствие нарушения программных, аппаратных или программно - аппаратных средств её обработки либо систем защиты, а также форс-мажорных обстоятельств, применения специальных программных (например, лазеек) средств воздействия, осуществляемого конкурентами, персоналом учреждения, поставщиками средств обработки информации в интересах третьих лиц;

3. Хищения информации путем подключения к линиям связи или техническим средствам, за счет снятия и расшифровки сигналов паразитных электромагнитных излучений, фотографирования, кражи носителей информации, подкупа или шантажа персонала учреждения или его филиалов, прослушивания конфиденциальных переговоров, осуществляемых конкурентами, персоналом учреждения или преступными элементами, несанкционированного копирования информации, считывания данных других

пользователей, мистификации (маскировки под запросы системы), маскировки под зарегистрированного пользователя, проводимых обслуживающим персоналом автоматизированной системы, хищение информации с помощью программных ловушек;

4. Махинаций с информацией (путем применения программных, программно-аппаратных или аппаратных средств), осуществляемых в интересах третьих лиц поставщиками средств обработки информации или проводимых персоналом учреждения. Также возможны подделка электронной подписи или отказ от нее.

Итак, теперь мы знаем, что и от кого (чего) надо защищать. Попробуем разобраться с тем, как организовать защиту.

§ 6. Модель противодействия угрозам безопасности

Уменьшить отрицательное воздействие угроз безопасности информации можно различными методами. Среди таких методов выделяются четыре основные группы:

1. Организационные методы. Организационные методы в основном ориентированы на:

а) Работу с персоналом;

б) Выбор местоположения и размещения объектов корпоративной сети;

в) Организацию систем физической и противопожарной защиты;

г) Осуществление контроля выполнения принятых мер;

д) Возложение персональной ответственности за выполнение мер защиты.

Эти методы применяются не только для защиты информации и, как правило, уже частично реализованы на объектах корпоративной сети. Однако их применение дает значительный эффект и сокращает общее число угроз;

2. Инженерно-технические методы. Инженерно-технические методы связаны с построением оптимальных сетей инженерных коммуникаций при учете требований безопасности информации. Это довольно дорогостоящие решения, но они, как правило, реализуются еще на этапе строительства или реконструкции объекта, способствуют повышению его общей живучести и дают высокий эффект при устранении некоторых угроз безопасности информации. Некоторые источники угроз, например обусловленные стихийными бедствиями или техногенными факторами, вообще не устранимы другими способами;

3. Технические методы. Технические методы основаны на применении специальных технических средств защиты информации и контроля обстановки и дают значительный эффект при устранении угроз безопасности информации, связанных с действиями криминогенных элементов по добыванию информации не-

законными техническими средствами. Кроме того, некоторые методы, например резервирование средств и каналов связи, оказывают эффект при определенных техногенных факторах;

4. Программно-аппаратные методы. Программно-аппаратные методы главным образом нацелены на устранение угроз, непосредственно связанных с процессом обработки и передачи информации. Без этих методов невозможно построение целостной комплексной системы информационной безопасности.

Сопоставление описанных выше угроз безопасности информации и групп методов их парирования позволяет решить, какими способами какие угрозы наиболее целесообразно парировать, а также определить рациональное соотношение групп методов при распределении средств, выделенных на обеспечение безопасности информации.

Анализ результатов моделирования с учетом принятых в модели ограничений и допущений позволяет считать, что все группы методов парирования угроз безопасности информации имеют примерно равную долю в организации комплексной защиты информации. Однако необходимо учесть, что некоторые варианты могут быть использованы только для решения ограниченного круга задач защиты. Это особенно характерно для устранения угроз техногенного и стихийного характера.

Наибольший эффект достигается при применении совокупности организационных и программно-аппаратных методов парирования. Анализ программно-аппаратных методов позволяет сделать вывод, что гипотетическое средство защиты корпоративной сети, прежде всего, должно обеспечивать разграничение доступа субъектов к объектам (мандатный и дискреционный принципы), управлять внешними потоками информации (фильтрация, ограничение, исключение) и, как минимум, обеспечивать управление внутренними потоками информации с одновременным контролем целостности программного обеспечения, конфигурации сети и возможности атак разрушающих воздействий. Рассмотрим основные принципы построения системы комплексной защиты информации автоматизированной системы управления:

1. Принцип максимальной дружественности. Парирование угроз безопасности информации всегда носит недружественный характер по отношению к пользователям и обслуживающему персоналу АСУ. Это происходит из-за того, что любая система защиты по определению налагает ограничения на работу организационного и технического характера. Поэтому одним из основных принципов создания системы комплексной защиты информации должен стать принцип максимальной дружественности. Иными словами, не надо вводить запреты там, где без них можно обойтись (на всякий случай), а если уж и налагать ограничения, то предварительно нужно продумать, как это сделать с минимальными неудобствами для пользователя. Притом

следует учесть совместимость создаваемой системы комплексной защиты с используемой операционной системой и программно-аппаратной структурой автоматизированной системы, а также сложившимися традициями фирмы;

2. Принцип прозрачности. Автоматизированной системой управления пользуются не только высококлассные программисты. Кроме того, основное назначение АСУ состоит в обеспечении производственных потребностей пользователей, то есть работы с информацией. Поэтому система защиты информации должна работать в «фоновом» режиме, быть незаметной и не мешать пользователям в основной работе, но при этом выполнять все возложенные на нее функции;

3. Принцип превентивности. Надо всегда помнить, что последствия реализации угроз безопасности информации могут потребовать значительно больших финансовых, временных и материальных затрат по сравнению с затратами на создание системы комплексной защиты информации;

4. Принцип оптимальности. Оптимальный выбор соотношения различных методов и способов парирования угроз безопасности информации при принятии решения позволит в значительной степени сократить расходы на создание системы защиты информации;

5. Принцип адекватности. Принимаемые решения должны быть дифференцированы в зависимости от важности, частоты и вероятности возникновения угроз безопасности информации, степени конфиденциальности самой информации и её коммерческой стоимости;

6. Принцип системного подхода к построению системы защиты позволяет заложить комплекс мероприятий по парированию угроз безопасности информации уже на стадии проектирования автоматизированной системы, обеспечив оптимальное сочетание организационных и инженерно-технических мер защиты информации. Важность реализации этого принципа основана на том, что оборудование действующей незащищенной системы средствами защиты информации сложнее и дороже, чем изначальное проектирование и построение её в защищенном варианте;

7. Принцип адаптивности. Система защиты информации должна строиться с учетом возможного изменения конфигурации системы, числа пользователей, степени конфиденциальности и ценности информации. При этом введение каждого нового элемента АСУ или изменение действующих условий не должно снижать достигнутого уровня защищенности автоматизированной системы в целом;

8. Принцип доказательности. При создании системы защиты информации необходимы соблюдение организационных мер внутри автоматизированной системы управления, включая привязку логического и физического рабочих мест друг к другу,

а также применение специальных аппаратно-программных средств идентификации, аутентификации и подтверждения подлинности информации. Реализация данного принципа позволяет сократить расходы на усложнение системы, например применять цифровую электронную подпись только при работе с удаленными и внешними рабочими местами и терминалами, соединенными с корпоративной сетью по каналам связи.

Рассмотренные принципы должны быть положены в основу при выборе направлений обеспечения безопасности корпоративной сети, функций и мер защиты информации.

При выборе средств защиты информации обязательно встает вопрос о необходимости подтверждения выполнения тех или иных функций конкретным средством защиты. Это немаловажный процесс, который в определенных случаях (например: при организации защиты информации, содержащей государственную тайну или сведения о личности - персональные данные) строго регламентирован. Свидетельством того, что те или иные функции защиты реализованы конкретным средством защиты, является сертификат соответствия - документ, которым независимые эксперты подтверждают готовность средства выполнить возложенные на него задачи.

§ 7. Архитектура безопасности

Международная организация по стандартизации ISO разработала стандарт ISO 7498-2 «Архитектура безопасности». В соответствии с этим стандартом любая система безопасности должна обеспечивать следующие восемь услуг безопасности:

1. Аутентификация объекта;
2. Аутентификация источника данных;
3. Контроль доступа;
4. Конфиденциальность и целостность соединения;
5. Конфиденциальность и целостность полей памяти;
6. Конфиденциальность и целостность трафика;
7. Целостность блока данных;
8. Доказательство источника;
9. Доказательство доставки.

Указанные услуги безопасности в соответствии со стандартом реализуются с помощью механизмов безопасности, размещенных в распределенных системах на промежуточном уровне модели OSI. Комплекс всех механизмов защиты называется доверенной вычислительной базой и включает в себя механизмы защиты локальных операционных систем ЭВМ, входящих в распределенную систему. Рассмотрим вышеупомянутые механизмы безопасности:

1. Шифрование (криптография). Шифрование (криптография) – это замена исходного сообщения его нечитабельным ва-

риантом с целью сохранения конфиденциальности содержащейся в сообщении информации. Существует два вида криптосистем:

а) Симметричные криптосистемы. В симметричных криптосистемах для шифрования сообщения и его дешифровки используется один и тот же ключ. В качестве симметричной криптосистемы в настоящее время чаще всего применяется стандарт шифрования данных DES (Data Encryption Standard). Стандарт преобразует 64-битовый блок данных за 16 шагов, на каждом из которых используется собственный 48-битовый ключ. Каждый из этих шестнадцати ключей порождается из единого 56-битового ключа. Для дешифровки используются преобразования, обратные применяемым для шифрования. Достоинства симметричных криптосистем:

- простота реализации;
- высокая стойкость шифра.

Недостатки симметричных криптосистем:

- необходимость использования закрытого (секретного) ключа;
- возможность подбора ключа.

В настоящее время наблюдается тенденция увеличения размера обрабатываемых блоков данных до 128 бит и более, а также длины ключа;

б) Несимметричные криптосистемы. В несимметричных криптосистемах используется уникальная пара различных ключей – для шифрования и для дешифровки. Несимметричные криптосистемы основаны на использовании односторонних функций, обладающих следующими свойствами:

- при заданном значении аргумента X можно вычислить значение функции $Y = f(X)$;
- по известному значению функции $Y = f(X)$ вычислить значение X крайне сложно или в идеальном случае невозможно.

Например, в несимметричном алгоритме RSA (аббревиатура по начальным буквам фамилий авторов алгоритма Ривеста, Шамира, Адлемана) в качестве односторонней функции используется функция $f(X) = a^X \bmod p$, где X – целое число, лежащее в диапазоне $1 < X < p - 1$, p – очень большое простое число, a – простое число, степени которого равны 1, 2, …, $p - 1$. Один из ключей несимметричной криптосистемы закрытый (секретный), а другой – открытый. Какой из ключей – шифрующий или дешифрующий – будет открытым, зависит от конкретного применения криптосистемы. Достоинства несимметричных криптосистем:

- наличие открытых ключей, возможность их пересылки по незащищенным каналам связи;
- высокая стойкость шифра. Эффективного метода нахождения простых множителей больших чисел не существует.

Недостаток несимметричных криптосистем заключается в высокой сложности вычислений. Так, в частности, расшифровка сообщения занимает от 10^2 до 10^3 большее время по сравнению с симметричными криптосистемами. По этой причине несимметричные криптосистемы чаще используют лишь для обмена закрытыми ключами симметричных криптосистем, основной же закрытый трафик производится с использованием симметричных криптосистем.

Федеральная служба безопасности России устанавливает для программных продуктов, требующих сертификации ФСБ РФ, пять классов криптографической защиты информации, которым должны соответствовать сертифицируемые программные продукты:

- класс КС1. Программные продукты, сертифицируемые по этому классу криптографической защиты информации, должны обеспечивать теоретическую стойкость используемого криптографического алгоритма и правильную реализацию самого шифровального средства;
- класс КС2. Программные продукты, сертифицируемые по этому классу криптографической защиты информации, должны обеспечивать практическую стойкость шифра в условиях возможного доступа к системе неквалифицированного персонала;
- класс КС3. Программные продукты, сертифицируемые по этому классу криптографической защиты информации, должны обеспечивать практическую стойкость шифра в условиях осуществления атак со стороны авторизованного пользователя системы;
- класс КВ1. Программные продукты, сертифицируемые по этому классу криптографической защиты информации, должны обеспечивать практическую стойкость шифра в условиях осуществления нарушителем перехвата паразитных электромагнитных излучений и наводок, сопровождающих работу автоматизированной системы;
- класс КВ2. Программные продукты, сертифицируемые по этому классу криптографической защиты информации, должны обеспечивать практическую стойкость шифра в условиях использования нарушителем недокументируемых возможностей в прикладном программном обеспечении;

2. Аутентификация. Аутентификация – это проверка подлинности объекта. Существуют три алгоритма аутентификации:

а) Аутентификация на основе симметричных криптосистем. Алгоритм такой аутентификации содержит пять этапов:

- отправитель посылает получателю открытый запрос на установление закрытого соединения;
- получатель посылает отправителю открытую квитанцию о готовности установить закрытое соединение;

- отправитель возвращает получателю его квитанцию, зашифрованную сгенерированным отправителем сеансовым симметричным ключом;

- отправитель посылает получателю открытое контрольное сообщение;

- получатель возвращает отправителю его контрольное сообщение, зашифрованное сеансовым симметричным ключом;

б) Аутентификация на основе несимметричных криптосистем. Алгоритм такой аутентификации проще и содержит всего три этапа:

- отправитель посылает получателю запрос на установление закрытого соединения, зашифрованный открытым несимметричным ключом получателя;

- получатель посылает отправителю квитанцию о готовности установить закрытое соединение и сгенерированный получателем сеансовый симметричный ключ, зашифрованные открытым несимметричным ключом отправителя;

- отправитель возвращает получателю его квитанцию, зашифрованную сеансовым симметричным ключом;

в) Аутентификация с использованием центра распространения ключей KDC (Key Distribution Center – центр распространения ключей). Центр распространения ключей осуществляет генерацию как симметричных, так и несимметричных ключей защиты информации. Это самый сложный алгоритм аутентификации, содержащий шесть этапов:

- отправитель посылает в центр распространения ключей открытый запрос на установление закрытого соединения с получателем;

- центр распространения ключей посылает отправителю сгенерированный центром сеансовый симметричный ключ, зашифрованный закрытым несимметричным ключом отправителя;

- центр распространения ключей посылает получателю запрос отправителя на установление закрытого соединения с получателем и тот же самый сеансовый симметричный ключ, который был послан отправителю, все зашифрованное закрытым несимметричным ключом получателя;

- отправитель посылает получателю сгенерированное отправителем случайное число, называемое талон, зашифрованное сеансовым симметричным ключом;

- получатель посылает полученный от отправителя талон в центр распространения ключей, зашифровав его своим открытым несимметричным ключом;

- центр распространения ключей возвращает талон отправителю, зашифровав его закрытым несимметричным ключом отправителя.

При организации защищенных каналов связи закрытые сеансовые симметричные ключи уничтожаются по завершении каж-

дого очередного сеанса связи. Для каждого нового сеанса связи генерируется новый сеансовый ключ;

3. Целостность данных. 4. Цифровая подпись. Целостность данных – это механизм безопасности, обеспечивающий защиту сообщения от изменений. Одним из способов обеспечения целостности данных служит цифровая подпись, которая, в свою очередь, также является одним из механизмов безопасности. Для создания цифровой подписи используются криптографические хэш-функции (hash - рандомизация, перемешивание), с помощью которых генерируется так называемый дайджест сообщения. Криптографическая хэш-функция представляет собой ряд последовательных алгебраических преобразований исходного сообщения, обеспечивающих наличие в получаемом дайджесте сообщения информации как о содержании исходного сообщения, так и о его длине. Дайджест сообщения шифруется любым способом и пересылается вместе с исходным сообщением. Исходное сообщение в зависимости от требуемой степени конфиденциальности может пересылаться как в открытом, так и в закрытом виде. Если исходное сообщение передается в закрытом виде, то оно шифруется другим ключом, чем пересылаемый совместно с этим сообщением дайджест (например: симметричный и несимметричный ключи, несимметричные ключи отправителя и получателя и т.п.).

В распределенных системах большую роль играет взаимодействие процессов в группах, которое так же должно быть защищено. Использование единого симметричного ключа для обмена информацией между процессами группы недопустимо, так как при дискредитации ключа одним из процессов, вся группа лишается конфиденциальности. Для группового взаимодействия используются криптосистемы с открытым ключом, когда каждый процесс группы имеет свою пару ключей, из которых открытые ключи используются всеми членами группы для посылки сообщений конкретному процессу.

Для повышения стойкости системы защиты информации используется репликация защищенных хранилищ данных и серверов, генерация для каждой из реплик своих уникальных ключей и последующее сравнение закрытых откликов по мажоритарному принципу. Несовпадение одного или нескольких откликов с большинством остальных может означать взлом сервера (серверов) или хранилища (хранилищ) данных.

В случае совместной работы двух или более процессов с одним хранилищем данных или сервером, используется технология разделения секрета, когда ключ делится на несколько частей по числу совместно работающих процессов, и доступ к хранилищу данных или серверу становится возможным только при совместном обращении всех процессов группы к хранилищу данных или серверу. В качестве варианта разделения секрета

применяется деление на несколько частей дайджеста цифровой подписи;

5. Контроль доступа. Контроль доступа заключается в формальном подтверждении прав доступа к ресурсам. Понятие контроля доступа тесно связано с понятием авторизации. Авторизация – это предоставление процессу, успешно прошедшему процедуры идентификации и аутентификации, соответствующих полномочий и прав доступа к ресурсам системы. В простейшем случае контроль доступа реализуется программой под названием монитор ссылок, которая запускается при любом обращении к защищаемому ресурсу. В локальных системах монитор ссылок формирует единую для всей системы матрицу контроля доступа, строки которой представлены авторизуемыми процессами, а столбцы – защищаемыми ресурсами. Элементы матрицы определяют, выполнение каких операций над ресурсом доступно процессу.

В распределенных системах, где процессы и запрашиваемые ими ресурсы находятся на разных ЭВМ сети, создание матрицы контроля доступа не решает задачи авторизации. В распределенных системах каждый ресурс имеет свой собственный монитор ссылок и список контроля доступа ACL (Access Control List – список контроля доступа), в котором указываются процессы и их права доступа к ресурсу. Таким образом, список контроля доступа представляет собой разбиение матрицы контроля доступа по столбцам. В распределенных системах, построенных на базе локальных сетей ЭВМ с низкой вероятностью несанкционированного доступа к ресурсам, могут применяться мониторы ссылок, привязанные к авторизуемым процессам. При этом в качестве исходной информации для работы монитора ссылок используется список мандатов, в котором указываются ресурсы и права доступа к ним данного процесса. Таким образом, список мандатов представляет собой разбиение матрицы контроля доступа по строкам.

При масштабировании системы списки контроля доступа или списки мандатов могут стать излишне большими. Для уменьшения размера списка контроля доступа или списка мандатов создают защищенные домены. Защищенные домены представляют собой группы процессов или ресурсов с одинаковыми правами доступа, объединенные таким образом, что в соответствующие списки заносятся не отдельные процессы или ресурсы, а группы процессов или ресурсов. Защищенные домены могут структурироваться по одноранговому или иерархическому принципам.

Для защиты ресурсов локальных распределенных систем, имеющих выход в глобальные сети, применяется монитор ссылок особого типа, называемый брандмауэр. Брандмауэр – это монитор ссылок, работающий как шлюз прикладного уровня между

локальным сегментом сети и остальной глобальной сетью, и предназначенный для тотальной авторизации всех пакетов, поступающих в локальный сегмент сети, и выходящих из него. Различают три вида брандмауэров:

а) Шлюз фильтрации пакетов – это тип брандмауэра, анализирующий только заголовки пакетов и принимающий решение об авторизации пакета только на основании адресов отправителя и получателя;

б) Шлюз прикладного уровня – это тип брандмауэра, анализирующий всю информацию пакетов и на основе этого анализа принимающий решение об авторизации пакета;

в) Прокси-шлюз – это тип брандмауэра, скрывающий работу ряда прикладных программ (например, апплетов рекламного характера).

§ 8. Управление защитой информации

Когда мы рассматривали методы криптографической защиты информации, мы заранее предполагали наличие ключей защиты информации. Однако в реальности генерация и распространение ключей защиты является сложным процессом, также нуждающимся в защите. Под управлением защитой информации подразумевается:

1. Создание ключей криптографической защиты информации. Симметричные ключи, как правило, создаются двумя способами:

а) Генерация ключа одной из сторон сеанса с последующим доведением этого ключа до другой стороны при помощи несимметричных криптосистем;

б) Использование метода Диффи-Хеллмана:

- обе стороны сеанса генерируют каждая свое большие случайные числа X и Y;
- отправитель посылает получателю открытым текстом три числа: $a = g^X \bmod n$, g и n;
- получатель посылает отправителю открытым текстом число $b = g^Y \bmod n$;
- обе стороны сеанса вычисляют закрытый симметричный ключ, равный $g^{XY} \bmod n = a^Y = b^X$.

Создание ключей несимметричной криптосистемы происходит в четыре этапа:

а) Выбираются два больших простых числа a и b;

б) Вычисляются произведения

$$m = a \cdot b \text{ и } n = (a - 1)(b - 1);$$

в) Выбирается нечетное не простое число c, меньшее m и взаимно простое с n. Два или несколько чисел называются взаимно простыми, если наибольший общий делитель этих чисел равен единице (например: 15 и 22; 7, 19, 32, и 84);

г) Вычисляется число d такое, чтобы произведение $c \cdot d$ нацело делилось на n.

Числа c и m образуют открытый ключ, а число d - закрытый, или наоборот. Для шифрования необходимы числа d и m, а для дешифровки – c и m. Рассмотренный нами метод Диффи-Хеллмана является примером несимметричной криптосистемы, где X и Y – закрытые ключи, а $a = g^X \bmod n$ и $b = g^Y \bmod n$ – открытые ключи;

2. Распространение ключей криптографической защиты информации. Для того, чтобы получатели были уверены, что распространяемый открытый несимметричный ключ действительно является парой к заявленному закрытому ключу, необходимо, чтобы канал, по которому он пересылается, обеспечивал аутентификацию. Аутентифицированное распространение открытых ключей осуществляется при помощи сертификатов открытого ключа. Сертификат открытого ключа представляет собой сообщение, содержащее сам открытый ключ и идентификатор сущности, с которой ассоциирован этот ключ, и то, и другое вместе защищенное цифровой подписью сертифицирующей организации.

Цифровая подпись производится закрытым несимметричным ключом сертифицирующей организации. Открытые ключи различных сертифицирующих организаций свободно распространяются вместе с файлами коммуникационных программ (например: почтовый клиент, web-браузер и т.п.). При желании установить, принадлежит ли на самом деле открытый ключ, содержащийся в сертификате, указанной в нем же сущности, клиент с помощью открытого ключа соответствующей сертифицирующей организации проверяет цифровую подпись сертификата. В случае сомнения подлинности сертификата, клиент может проверить правильность открытого ключа через другой сертификат, полученный от вышестоящей сертифицирующей организации. Примером может служить конфиденциальная почта Интернета PEM (Privacy Enhanced Mail - [электронная] почта с повышенной защитой), которая использует трехуровневую иерархическую службу сертификации, предусматривающую безусловное доверие к сертифицирующей организации верхнего уровня.

Для обеспечения защиты аутентифицированных каналов сертификаты должны периодически меняться. Для этого существует механизм отзыва сертификата, позволяющий сделать известным тот факт, что сертификат более не действителен. Имеется три способа отзыва сертификатов:

а) Список отозванных сертификатов CRL (Certificate Revocation List – список отозванных сертификатов). Сертифицирующая организация регулярно публикует списки отозванных сертификатов. При аутентификации ключа клиент просматривает списки отозванных сертификатов, и при обнаружении там имею-

щегося у него сертификата, связывается с сертифицирующей организацией для получения нового сертификата;

б) Ограничение срока жизни сертификатов. Каждому сертификату устанавливается предельный срок его действия (обычно один год), по истечении которого сертификат автоматически прекращает свое действие. В этом случае клиент должен связаться с сертифицирующей организацией для получения нового сертификата;

в) Отказ от использования сертификатов. Для реализации этого способа отзыва сертификатов клиент для проверки годности открытого ключа всякий раз должен связываться с сертифицирующей организацией;

3. Управление защищенными группами. Управление защищенными группами заключается в принятии решения на членство в группе нового процесса. Для этого новый процесс должен подтвердить, что он не в состоянии нарушить целостность группы. Аутентификация нового процесса происходит стандартным образом с помощью сертификата открытого ключа. По окончании аутентификации новый процесс получает открытые ключи всех членов группы и рассылает свой открытый ключ всем членам группы. Обмен ключами может производиться как непосредственно между процессами, так и с использованием центра распространения ключей;

4. Управление авторизацией. Управление авторизацией (правами доступа) в распределенных системах усложнено тем, что ресурсы системы распределены по нескольким ЭВМ. Управление авторизацией заключается в назначении процессам прав доступа к ресурсам системы и дальнейшей поддержке этих прав, исключающей их фальсификацию. В распределенных системах для управления авторизацией используются мандаты. Мандат – это нефальсифицируемая структура данных, относящаяся к некоторому ресурсу и точно определяющая права доступа владельца мандата к этому ресурсу. Мандат содержит:

а) Идентификатор сервера, на котором размещен защищенный ресурс;

б) Идентификатор защищенного ресурса;

в) Права доступа владельца мандата к защищенному ресурсу;

г) Поле защиты мандата, предназначенное для невозможности его подделки.

Мандат создается сервером, который записывает данные прав доступа и защиты мандата в свою копию списка мандатов и высылает мандат в закрытом виде клиенту, где он заносится в список мандатов клиента. Когда мандат в качестве запроса клиента на доступ к ресурсу пересылается обратно на сервер, он сравнивается с данными, хранящимися в серверной копии

списка мандатов, и, при их совпадении, клиенту предоставляются соответствующие права доступа.

В распределенных системах возможна ситуация, когда клиент, обладая некоторыми правами доступа к ресурсу, пытается осуществить удаленную обработку информации, предоставляемой этим ресурсом, с помощью сервера, не имеющего прав доступа к этому ресурсу. При этом клиент должен делегировать серверу свои права доступа к ресурсу. Делегирование прав доступа осуществляется с помощью маркера-заместителя, позволяющего своему владельцу работать с теми же правами доступа, что и процесс, выдавший этот маркер. Маркер защищается с помощью сертификата или шифруется стандартным образом.

§ 9. Использование геоинформационных технологий в АСУ

Одной из возможных компонент информационного обеспечения автоматизированных систем управления являются геоинформационные технологии. Геоинформационная технология – это совокупность средств и методов сбора, обработки, накопления и использования пространственно-временной информации. Геоинформационные технологии широко используются при управлении земельными ресурсами, городским хозяйством, организации транспорта, розничной торговли, использовании океанов и других пространственных объектов. На основе использования геоинформационных технологий строятся геоинформационные системы – интегрированные информационные автоматизированные системы управления территориального уровня, предназначенные для поддержки принятия решений, основанных на анализе пространственно-временной информации. Поддержка принятия решений определяется как совокупность технологий:

1. Сбор, обобщение и анализ данных обстановки;
2. Выявление изменяемых и неизменяемых факторов внешней среды;
3. Оценка неопределенности ситуации внешней среды;
4. Уменьшение неопределенности путем моделирования состояний внешней среды;
5. Использование базы типовых решений для регламентных ситуаций;
6. Разработка набора альтернативных вариантов решений;
7. Разработка прогнозных оценок по альтернативам;
8. Построение множества решающих правил выбора альтернатив.

Как правило, геоинформационные системы используются для управления территориальными объектами, то есть организационно-техническими системами, функционирующими на определенном участке земной поверхности или в пространстве от-

носительно земной поверхности. Особенности геоинформационных систем:

1. Основой интеграции данных в геоинформационных системах является географическая информация, но большинство решаемых геоинформационными системами задач далеки от географических;

2. Основой интеграции технологий в геоинформационных системах являются технологии систем автоматизированного проектирования (САПР), но решаемые геоинформационными системами задачи далеки от проектных;

3. Двойственность функционального предназначения геоинформационных систем заключается в одновременной реализации функций хранения и обработки информации и управления;

4. Многоаспектность геоинформационных систем:

а) Как система управления, геоинформационная система предназначена для поддержки принятия решений по управлению пространственными объектами;

б) Как автоматизированная информационная система, геоинформационная система являет собой пример современной интегрированной системы, включающей интеграцию технологий на основе технологий систем автоматизированного проектирования, интеграцию данных на основе географической информации, объединение различных методов и технологий в единый комплекс;

в) Как системы, использующие базы данных, геоинформационные системы объединяют в себе как базы данных обычной цифровой информации, так и графические базы данных;

г) Как системы моделирования, геоинформационные системы используют максимальное количество методов и процессов моделирования, применяемых в автоматизированных системах управления;

д) Как системы проектирования, геоинформационные системы используют методы автоматизированного проектирования, но решают ряд специальных проектных задач, не встречающихся в типовом автоматизированном проектировании;

е) Как системы представления информации, геоинформационные системы обеспечивают большую наглядность представления данных с использованием современных мультимедийных технологий;

ж) Как прикладные системы, геоинформационные системы не имеют себе равных по широте применения.

Основное назначение геоинформационных систем заключается в поддержке принятия решений. Этапы геоинформационной технологии принятия решений:

1. Постановка задачи с учетом её территориального выражения;

2. Обобщенная формализация задачи. Формализация заключается в построении и идентификации моделей географических объектов и среды, в которой эти объекты функционируют;

3. Декомпозиция задачи. Декомпозиция заключается в разбиении исходной общей задачи на подзадачи с учетом выявленных параметров географических объектов и среды;

4. Пространственная декомпозиция задачи с помощью геоинформационных технологий преобразования и анализа географической информации. Данный этап составляет существенную специфику геоинформационной поддержки управления и осуществляется параллельно предыдущему этапу и независимо от него. Содержанием данного этапа является распределение параметров по пространственной схеме и выявление связей между пространственными переменными и содержательными данными;

5. Геоинформационное сегментирование. Под геоинформационным сегментированием понимается оценка территории с точки зрения её влияния на решение поставленной задачи и деление территории на районы и зоны целесообразной содержательной деятельности. Данный этап является объединением результатов вышерассмотренных третьего и четвертого этапов;

6. Выбор информационной системы поддержки управления. В качестве такой системы могут использоваться геоинформационные и экспертные системы, базы данных и знаний, системы деловой графики и т.п.;

7. Сбор априорной информации об обстановке (геоинформационные исследования):

а) Сбор исходных пространственных данных (сбор информации об обстановке). Данный процесс обязателен и специфичен именно для территориального управления. Он обеспечивает не только сбор пространственных данных об объектах управления и географической среды, но и выбор картографической основы (базовой карты, её проекции, масштаба и т.п.), которая будет служить для визуализации и анализа статистической и экономической информации;

б) Сбор всей дополнительной информации к пространственным данным. Этот процесс выполняется параллельно предыдущему процессу;

в) Сбор оперативной информации, информационное доопределение своих объектов и противодействующих систем. Процесс осуществляется для получения информации, необходимой при построении и идентификации моделей среды, своих объектов и недружественного (конкурирующего) социума, а также для последующего принятия решений:

- сбор первичных данных о своих и конкурирующих объектах и внешней среде;

- сбор косвенных (вторичных) данных для получения на их основе необходимой информации. Выполняется при отсутст-

вии первичных данных о конкурирующих объектах или о внешней среде;

8. Обобщение данных. Результаты информационной разведки анализируются, обобщаются, проверяются на непротиворечивость, актуальность и достоверность;

9. Интеграция данных в единую систему на единой информационной основе. Это дает возможность их комплексного анализа и сопоставления разных видов информации по единым критериям;

10. Анализ данных с применением разнообразных методов и систем. Исследование географической ситуации и выделение факторов:

а) Способствующих решению задачи;

б) Препятствующих решению задачи;

в) Независимых факторов, на которые невозможно воздействовать в рамках системы управления;

г) Изменяемых факторов, которые можно изменить, применяя управляющие решения;

11. Формирование наборов пространственных альтернатив для поддержки принятия решения. Данный этап осуществляется по результатам выполнения всех предыдущих этапов;

12. Визуализация сформированных альтернатив для их представления и последующего анализа;

13. Принятие решения. Выбор пространственно-содержательной альтернативы по решающему правилу;

14. Разработка пространственно-содержательного плана реализации решения;

15. Контроль исполнения плана.

Наличие большого количества независимых факторов и неопределенности в состоянии внешней среды исключает возможность однозначного принятия решения. Уменьшение неопределенности осуществляется путем моделирования и получения наборов условных состояний внешней среды. Среди этих состояний выделяют существенные (значимые) и несущественные. Для каждого из значимых состояний внешней среды разрабатывают набор проектов решений. В отличие от набора альтернатив, предоставляемых классическими автоматизированными системами управления, геоинформационные системы дополнительно обеспечивают набор визуальных средств поддержки принятия решений (карты и планы). Метрические характеристики представляемых карт и планов служат основой для точных расчетов и экономических оценок решений, основанных на анализе пространственно-временной информации. Различают два уровня управления территориальным объектом:

1. Макроуровень. Макроуровень рассматривает управление территориальным объектом как единой неделимой точкой про-

странства, имеющей собственные интегральные характеристики состояния (например: координаты, курс, скорость и т.п.);

2. Микроуровень. Микроуровень управляет состоянием функциональных подсистем территориального объекта.

Выделение двух уровней управления обуславливается необходимостью учета влияния состояния подсистем территориального объекта на формирование окружающей территориальной обстановки и, наоборот, учета состояния географической среды и территориальной обстановки на планирование функций подсистем территориального объекта.

§ 10. Ситуационный центр

В настоящее время становится все более очевидным, что ни высокий экономический, человеческий, технологический, военный, культурный и другие потенциалы не гарантируют безусловную жизнеспособность и стабильное развитие страны. Только организационный потенциал совместно с информационно-аналитическими ресурсами способен в полной мере актуализировать все ресурсы государства и общества и успешно решить стоящие перед ними проблемы. Примером не использования организационного потенциала и информационно-аналитических ресурсов в кризисной ситуации при наличии огромных экономических, военных, материальных и других потенциалов может служить распад СССР.

Важнейшим инструментом, обеспечивающим консолидацию и эффективное использование организационного потенциала, являются система ситуационно-аналитических центров органов государственного управления.

Основное назначение ситуационно-аналитического центра - это обеспечение эффективной консолидации, целенаправленного использования и развития организационных возможностей государства, общества и личности на основе широкого применения новейших информационно-аналитических методов и технологий как для оперативного управления крупными географическими объектами (страна, регион, область), так и для их организационного строительства и развития, включая как внешнюю, так и внутреннюю организационные среду.

Ситуационно-аналитичекий центр состоит из четырех основных подсистем:

1. Аппаратно-программная среда общего назначения;
2. Подсистема методического обеспечения;
3. Комплекс средств специального обеспечения;
4. Объединенная подсистема баз данных и знаний.

Интеллектуальным ядром ситуационно-аналитического центра является комплекс взаимосвязанных моделей, основными из которых являются:

1. Динамическая четырехуровневая модель социально-технического природного образования, решающего задачу собственного выживания и развития в окружающей его социальной и природной среде и способного как к адаптации к требованиям внешней среды, так и целенаправленному воздействию на нее. При функционировании модели учитываются не только схемотехнические прагматические аспекты объектов, средств и субъектов управления, но также их духовно-нравственные идеи, менталитет, культурные, генетические и психофизические особенности;

2. Индикаторные модели критериального пространства, с которым взаимодействуют через свои входные и выходные информационные потоки все остальные модели ситуационного центра;

3. Модели выявления проблемных ситуаций, раннего предупреждения и разработки мероприятий по их парированию и ликвидации негативных последствий;

4. Информационные модели объектов управления на основе интерактивных документов, которые представляются в ориентированном на руководителя и проблему виде и позволяют:

а) Моделировать ситуацию непосредственно в рамках самого документа;

б) Вырабатывать и оформлять варианты оперативных решений и протоколов их разработки;

в) Осуществлять доведение принятых документов до руководителей и исполнителей, участвующих в решении конкретной проблемы;

г) Осуществлять текущий контроль за ходом выполнения программ;

5. Система искусственного интеллекта для поддержки принятия долгосрочных крупных социально-экономических и политических решений государственного, регионального и областного уровней на основе многосторонней деловой стратегической компьютерной игры как средства поиска компромисса интересов всех заинтересованных сторон, исходя из имеющихся у них ресурсов и складывающихся условий;

6. Модель гибкого социально-экономического и политического мониторинга, способного адаптироваться к динамике развития проблемной (кризисной) ситуации;

7. Модель управления психической и физической активностью населения.

В ситуационно-аналитическом центре циркулируют следующие виды информации:

1. Текущая справочная информация по всем внутренним и внешним субъектам и объектам сферы интересов государственных органов управления: хронология (даты и сроки), экономика, финансы, экология, социальная сфера, организации и пер-

соналии, транспорт, связь, расписания, цены, тарифы и т.п.;

2. Картографическая информация с различными слоями (земля, демография, населенные пункты, предприятия, дороги, коммуникации, технические сети и т.д.);

3. Фундаментальная справочная информация в сфере внутренних и внешних интересов (структурная, технологическая, организационная, географическая, экономическая, демографическая, историческая, культурологическая, политическая, правовая, о средствах массовой информации, о влиятельных персонах, о ключевых проблемах, о предпочтениях населения, о конфликтных фоне и потенциале и др.);

4. Оценочная информация о состоянии и развитии государства и его административно-территориальных образований с учетом внешней среды (обобщенная и по сферам деятельности, по территориям, кадровому составу и функциям, по технологии и финансам, по нормативным и директивным документам, текущая и прогнозная, по организациям и влиятельным политическим фигурам и т.д.);

5. Многовариантная информация о легальных и недобросовестных схемах ведения бизнеса, управления финансовыми ресурсами и обязательствами, адаптации к таможенным и фискальным нормам, правилам и действиям, варианты действий по защите национальных интересов в сфере бизнеса;

6. Оперативные доклады ведомств, подразделений, служб и должностных лиц о состоянии объектов управления, событиях и процессах в соответствии с разделением компетенции;

7. Доклады о контроле хода и результатах выполнения программ, планов, решений, распоряжений и т.п.;

8. Доклады о выявленных опасностях, угрозах, негативных процессах, тенденциях и проблемных ситуациях;

9. Доклады о симптомах неявных (предполагаемых) проблемных ситуаций.

При создании ситуационного центра производятся настройки поставляемых процедур и моделей, а также частичное заполнение базы данных и знаний под конкретные задачи заказчика. После приемки ситуационного центра в эксплуатацию (в полной или выборочной конфигурации) на первом этапе его функционирования предприятие-разработчик готово осуществлять методическое и техническое сопровождения работ по модернизации и текущим доработкам всех видов обеспечения информационно-аналитического центра.

Основное отличие ситуационного центра от традиционных систем автоматизации управления состоит в том, что в процессе проведения производственно-управленческого совещания в режиме реального времени можно просчитывать и анализировать последствия любых управленческих решений.

Успех работы предприятия прежде всего зависит от того, как им управляют. По оценкам специалистов, средняя выработка на одного человека по 200 крупнейшим предприятиям России составила около 35 000 долларов. В США данный показатель равен 264 000 долларов. Выходит, что наши предприятия работают примерно в семь раз менее эффективно.

В 1947 году для моделирования сложных экономических ситуаций стали использовать методы причинного нелогического вывода - позже они легли в основу методов системной динамики. С середины 50-х годов прошлого века, когда впервые был введен термин «искусственный интеллект», и вплоть до ранних 1970-х создание систем анализа рассматривалось в рамках логического решения задач. Сейчас на их основе развиваются методы моделирования мыслительно-познавательной (когнитивной) активности человека при принятии деловых решений. В 1980-х появились информационные технологии для улучшения эффективности профессиональной деятельности лиц, принимающих решения. Практическое применение получили подходы, основанные на использовании немонотонных логик и нечетких систем. В конце 80-х годов прошлого века внимание разработчиков все больше акцентируется на исследовании адаптивных свойств информационных систем, учитывающих умственную активность человека при принятии решений.

Опыт, полученный при создании и внедрении аналитических систем разного класса, в конце 1990-х лег в основу концепции ситуационного центра.

Мода прошлых лет, когда непременным атрибутом стола руководителя был компьютер, проходит. У высокопоставленного руководителя нет времени для анализа объемных электронных таблиц и многостраничных документов. Сегодня инструмент анализа должен находиться на качественно новом уровне, и таким инструментом стал ситуационный центр, который представляет собой специально организованный комплекс рабочих мест для высших должностных лиц. Начало создания «командных пунктов» специалисты объясняют еще и необходимостью отхода от традиционных способов обработки данных, когда при лице, принимающем решение, создается группа экспертов, ответственных за отбор и анализ информации. Как убедились на Западе, при таком подходе в большинстве случаев результирующее решение фактически принимается упомянутой группой экспертов, соответствующим образом подготавливающих входную информацию для формального принятия решения первым лицом компании. В случае некомпетентности, тенденциозности, лоббистских интересов экспертов процесс принятия решения и его результат будут очевидным образом искажены.

Основное отличие ситуационного центра от традиционных систем автоматизации управления состоит в том, что в про-

цессе проведения производственно-управленческого совещания в режиме реального времени можно просчитывать и анализировать последствия любых управленческих решений. К традиционным задачам, стоящим перед ситуационным центром, относятся:

1. Прогноз состояния объекта управления;
2. Моделирование последствий управленческих решений;
3. Решение управленческих задач с учетом постоянного изменения типов взаимодействия с внешней средой;
4. Решение управленческих задач при изменяющихся целевых функциях и критериях объекта.

В зависимости от стоящих перед руководящим составом задач и с учетом имеющихся в компании ресурсов (информационных, интеллектуальных, материальных) ситуационный центр может быть скомпонован с различным уровнем сложности:

1. Стратегический ситуационный центр ориентирован на сложные, масштабные, ответственные задачи, направленные на структурную и функциональную перестройку системы или на стратегический анализ ее развития и прогноз жизнедеятельности. Стратегические ситуационные центры настроены на объекты класса: отрасль, регион, крупное предприятие (холдинг), ведомство, сложный распределенный в пространстве процесс;
2. Оперативный ситуационный центр решает задачи автоматического перевода оперативной информации в ситуационную модель, дающую первому лицу возможность оперировать "модулями" системы в реальном времени. Оперативные ситуационные центры настроены на объекты класса: предприятие (компания), задача, процесс, кампания, проект, крупная акция, однородная функция значительных масштабов. К данному классу относятся и ситуационные центры для анализа и управления кризисными ситуациями. При этом основное назначение подобных центров заключается в предотвращении кризиса за счет своевременного предоставления лицам, принимающим решения, не только исчерпывающей информации по текущему состоянию контролируемых объектов, но и прогнозов возможных сценариев развития событий. Если же кризиса избежать не удалось, такие ситуационные центры становятся по сути оперативными штабами по управлению процессами ликвидации кризиса. К этому же классу относятся «центры виртуальной реальности» (Realyty Centre), служащие для воссоздания разрабатываемых (и еще не существующих в реальности) объектов;
3. Персональный ситуационный центр решает задачу экспресс-оценки ситуации, оперативного доступа к управляемому объекту и поддерживающий возможность первого руководителя всегда "быть в курсе" независимо от времени и места нахождения управляющего субъекта. Персональный ситуационный центр представляет собой, как правило, мобильное компьютеризированное рабочее место руководителя с необходимым аппа-

ратным, программным и информационным обеспечением. Руководитель получает возможность не только оперативно ознакомиться с текущей экономической обстановкой, но и проанализировать ее и на основании этого наметить необходимые решения.

Эффективность ситуационных центров основана на том, что они позволяют подключить к активной работе по принятию решения резервы образного, ассоциативного мышления. Представление ситуации в виде образов как бы сжимает информацию.

В общем случае для ситуационного центра необходимо специальное помещение, где на несколько мониторов выводятся сведения об основных подразделениях, функциях или производственных процессах в соответствии с поставленной задачей. Зал оснащается также одним - двумя обобщающими мониторами (экранами), на которые по каналам связи непрерывно поступает информация о положении в интересующих секторах.

Современные системы работают в реальном масштабе времени со сложнейшей графической информацией и поэтому требуют огромных ресурсов от компьютерной техники. Для адекватного восприятия визуальной информации используются специальные экраны. Панорамный экран как бы помещает пользователя внутрь компьютерного кинотеатра, проекционные экраны более компактны и не нуждаются в специальном зале, многоплоскостные экраны позволяют создать систему наблюдения, окружающую человека с шести сторон.

Руководители, располагающие стратегическим ситуационным центром, получают преимущество перед конкурентами при планировании продвижения на новые сектора рынка, долгосрочной ценовой и товарной политики и т. д. В результате руководство компании переходит от принятия отдельных решений к выработке сценариев (системных решений), когда каждое отдельное решение подчинено целям поддержания долгосрочной стабильности компании. Для обеспечения работы стратегического ситуационного центра действует соответствующая служба, в состав которой должны входить системный аналитик (консультант), руководитель систем внешних и внутренних информационных сетей и коммуникаций, разработчик программного обеспечения и дежурный оператор.

Решения принимаются с учетом не только мнения руководителей подразделений и представлений первого лица, но и на основе объективной внутренней и внешней информации, поскольку ситуационный центр синхронизирует подсистему внутреннего управления с подсистемой, отвечающей за взаимодействие с внешней средой. Текущая информация используется на совещании в виде электронных сводок, подготовленных на основе данных бухгалтерской, финансовой, планово-

экономической, маркетинговой, инженерной и других служб в виде наглядных визуальных объектов. Таким образом, производственные совещания приобретают принципиально новое качество в процессе управления.

Важнейшим дополнительным элементом, существенно повышающим качество управления компанией, является возможность оперативно обрабатывать внешнюю информацию. Основным источником информации при этом являются общедоступные базы данных сети Internet, специализированная и региональная статистическая информация. Внешняя информация автоматически оперативно вводится в систему внутреннего информационного оборота в компании через специальные программы-конверторы. Тем самым руководитель получает возможность принимать в расчет внешние точки измерения: место компании в отраслевом разрезе, место компании в регионе, анализ конкурентной среды, оценка перспектив рынка.

Вместе с тем основой диагностики предприятия и сценарного просчитывания является внутренняя система сбора и обработки данных. Без всеобъемлющей информационной системы компании любой, даже самый современный ситуационный центр останется «слепым». Недостаточное развитие или ограниченная архитектура локальных сетей компании отмечается специалистами как одна из существенных причин, влияющих на скорость внедрения и эффективность работы ситуационных центров.

Сегодня в мире насчитывается несколько сот ситуационных центров и их количество с каждым годом неуклонно увеличивается. Называются они по-разному: центры стратегического управления, визионариумы, центры мультимедиа, ситуационные комнаты. В администрации президента США сейчас действует более пяти ситуационных центров. В Европе их количество приближается к сотне. В одной только Норвегии их больше десяти. Подобные системы имеются у всех крупных фирм и корпораций. Это направление сейчас бурно развивается, а уже существующие системы активно переоснащаются.

Проект по созданию визионариума для президента США обошелся американским налогоплательщикам в 50 млн долларов. В составе центра три команды из специально подготовленных офицеров (Duty Oficers), аналитиков разведки и специалистов по коммуникациям, которые ведут непрерывный мониторинг и анализ всех доступных информационных каналов, включая (в это даже трудно поверить) все каналы телевидения и радио (их ведь в США несколько сот). Одной из многочисленных задач центра стала оценка доходных и расходных частей бюджета, в которой задействованы средства виртуальной реальности. Электронная карта страны помещена в трехмерное виртуальное пространство и над каждым штатом США построены башенки-гистограммы, демонстрирующие те или иные показатели.

Взглянув на карту, можно очень быстро определить, где эти показатели достигают очень высоких или, наоборот, очень низких значений.

Есть свой ситуационный центр и у правительства Германии, он используется для анализа социальных, экономических и политических проблем.

В феврале 1996 года был введен в строй ситуационный центр в резиденции Президента Российской Федерации. Это достаточно сложный программно-мультимедийный комплекс: три экрана размером 1,5 х 2 м, более десятка рабочих станций (студий нелинейного монтажа, графических станций, компьютеров для подготовки презентаций), мощный сервер, который хранит огромные объемы информации, а также набор различных инструментальных средств, позволяющих обрабатывать информацию и представлять ее Президенту. На её основе вырабатываются решения, которые доводятся до исполнителей средствами того же ситуационного центра. По существу в этой системе реализован полный цикл управления любым проектом, программой, регионом или страной. С помощью средств центра происходит планирование мероприятий Администрации Президента. Создан специальный программный модуль, позволяющий оценивать приоритеты, выбирать наиболее эффективную и рациональную последовательность проведения этих мероприятий.

Другой пример - ситуационный центр Министерства по чрезвычайным ситуациям РФ. Визуализируется картинка со схематическим изображением текущего и прогнозируемого состояния, на которой видно, какие силы и средства есть в наличии, какие предлагаются рекомендации. На основе всей этой информации принимаются решения, которые с помощью имеющихся в ситуационном центре средств доводятся до спасателей.

Еще один ситуационный центр был создан в 1994 году в Совете безопасности при Президенте РФ. Он достаточно успешно функционирует и по сей день. Этот центр позволяет осуществлять мониторинг, моделирование последствий, анализ событий, которые происходят в экономике, социальной сфере, в области национальной безопасности, помогая таким образом вырабатывать решения.

Также создан и действует ситуационный центр при Правительстве РФ и ряд ситуационных комнат для губернаторов в Ленинградской, Орловской, Белгородской, Тюменской и ряде других областях. В общей сложности уже стартовало более 20 таких региональных и отраслевых проектов, в том числе ситуационный центр для руководства "Газпрома".

Одним из мировых лидеров среди разработчиков ситуационных центров на Западе традиционно считается компания Silicon Graphics Incorporation (SGI), которая приступила к их созданию в начале 90-х годов прошлого столетия. За по-

следние пятнадцать лет специалистами SGI создано более ста таких центров.

Если несколько лет назад существенной причиной остановки внедрения ситуационных центров была цена проекта, то сегодня данная проблема отошла на второй план. Во-первых, потому что появились комплексные решения, сопоставимые с бюджетом предприятия, во-вторых, необходимость изменений в технологии управления и преимущества ситуационных комнат стали очевидны настолько, что руководящему составу компаний от них уже трудно отказаться.

Сегодня цена ситуационных центров сильно зависит от заданных масштабов, но может быть значительно снижена при наличии готовой информационно-коммуникационной структуры, корпоративной сети, базы данных. При переходе от одного масштаба центра к другому цена может меняться примерно на порядок, причем это утверждение относится в большей мере к стоимости инсталляции, и в меньшей - к стоимости сопровождения. При выборе решения существенно также то, что в составе суммарных затрат на ситуационный центр относительно высока (но неуклонно снижается) доля стоимости каналов связи и стоимости доступа к информации.

Другая проблема лежит в области проведения определенных изменений в культуре корпоративного управления. Дело в том, что для проведения эффективных совещаний в ситуационной комнате высший менеджмент должен работать, как одна команда, для которой ситуационный центр является основным рабочим инструментом. К сожалению, не во всех случаях российские корпоративные заказчики готовы к внедрению подобных технологий и работе в таком режиме.

При подготовке обсуждения значительную сложность представляет работа над сценариями демонстрации (предварительная режиссура), поскольку всегда сложно спрогнозировать ход обсуждения проблемы. Это обсуждение может отклониться от намеченного сценария и сделанных заранее заготовок. Для повышения устойчивости управления сценарием обсуждения необходима максимальная информационная открытость системы для получения требуемых сведений и аналитических материалов из внешних источников. Вместе с тем следует иметь в виду, что увеличение открытости некоторой информационной системы снижает целенаправленность обсуждения.

Приведенный выше список проблем далеко не полон. Часть из них - просто болезни роста, и они, безусловно, - временное явление. По общему мнению экспертов, внедрение и активное использование ситуационных центров в самых разных областях - дело ближайшего будущего.

Глава 5. Аппаратное обеспечение автоматизированных систем

Про аппаратное обеспечение персональных компьютеров написано столько книг, что большинству конечных пользователей кажется, будто в данной сфере им известно всё или почти всё. Однако, присутствующая на рынке литература, как правило, описывает конкретное «железо», не давая никаких знаний или конкретных рекомендаций по основам взаимодействия отдельных компонент автоматизированных систем. В то же время, в эпоху стремительного развития всех видов техники и технологии, аппаратное обеспечение вычислительных систем морально устаревает и меняется практически ежегодно. Следом за оборудованием, такие книги также стремительно морально устаревают, не давая своим читателям времени даже как следует осознать, что же там написано. Но, к сожалению, без глубоких знаний в области взаимодействия отдельных компонент оборудования между собой невозможно осознанно грамотно эксплуатировать современные автоматизированные системы. По этой причине в настоящем пособии не будет рассматриваться конкретное аппаратное обеспечение автоматизированных систем управления, а будут даны лишь математические основы функционирования современных распределенных автоматизированных систем управления.

§ 1. Математические модели узлов коммутации

Для того, чтобы маршрутизатор, как узел коммутации, мог эффективно выполнять свои функции определения оптимального маршрута пересылки данных, управляющая ЭВМ маршрутизатора должна работать с некоторой математической моделью сети. Такую модель можно разбить на несколько подмоделей:

1. Подмодель внешних воздействий (запросов на использование ресурсов сети). Введем основные характеристики условий работы для описания подмодели внешних воздействий:

а) Поток сообщений пользователей. Как известно, сообщение пользователя длиной L_C разбивается на сетевом уровне модели OSI на пакеты длиной $L_П$, а на канальном уровне – на кадры стандартной длины L_K. Поток сообщений пользователей – это случайный процесс, для которого задается закон распределения на основе опыта многолетних наблюдений за действующими сетями ЭВМ. Как правило, в качестве потока сообщений пользователей принимается пуассоновский поток с плотностью распределения во времени $P_K(t) = \frac{(\lambda t)^K}{K!} e^{-\lambda t}$, где K – номер очередного сообщения в момент времени t, λ – интенсивность поступления сообщений;

б) Второй характеристикой внешних воздействий является продолжительность (объем) сообщения, которая часто задается не временем, а количеством кадров, поступивших по одному входящему каналу за один сеанс передачи.

Ввиду того, что кроме информации пользователей, по сети передается служебная информация (например: изменение направляющих таблиц), можно выделить три вида потока сообщений пользователей:

а) Простой поток сообщений пользователей. В простом потоке длина кадра равна длине сообщения пользователя, параметр потока $\xi = \frac{L_K}{L_C} = 1$;

б) Сложный поток сообщений пользователей. В сложном потоке длина кадра меньше длины сообщения пользователя, параметр потока $\xi = \frac{L_K}{L_C} < 1$;

в) Прореженный поток сообщений пользователей. В прореженном потоке длина кадра больше длины сообщения пользователя, параметр потока $\xi = \frac{L_K}{L_C} > 1$.

Для простоты примем, что начало передачи каждого кадра возможно только в фиксированные моменты времени, кратные $\nu = \frac{n}{c}$, где n – количество бит в кадре, c – скорость передачи кадра в бод (бит в сек). При этом можно считать, что квант времени ν равен одной единице времени. Обозначим через X число кадров, поступающих в канал передачи в течение единицы времени ν. Тогда для случая $\xi = 1$ распределение вероятности числа поступающих кадров определяется выражением: $P(X) = \frac{\lambda^X}{X!} e^{-\lambda}$. Математическое ожидание числа поступающих кадров равно дисперсии и равно интенсивности поступления сообщений $M[X] = D[X] = \lambda$.

Для случая $\xi < 1$ необходимо задаться законом распределения параметра потока. Из анализа статистики распределения длин сообщений пользователей известно, что параметр потока с учетом дискретности процесса передачи имеет геометрическое распределение вероятности $P(\xi) = \xi(1 - \xi)^{j-1}$, где $j = 1, 2, \ldots$. Тогда общее число кадров, поступающих в канал, является случайной величиной и определяется по формуле $U(X) = \sum_{i=0}^{X} y_i$, где y_i – число кадров, содержащихся в i-м сообщении пользователя. Вероятность того, что в течение кванта времени ν в канал поступят i кадров, равна

$$P(U) = \begin{cases} e^{-\lambda}, \text{ при } i = 0 \\ e^{-\lambda}\sum_{K=1}^{i} C_{i-1}^{K-1}\frac{(\lambda\xi)^K(1-\xi)^{i-K}}{K!}, \text{ при } i = 1, 2, \ldots. \end{cases}$$

Математическое ожидание числа поступающих кадров равно $M[X] = \frac{\lambda}{\xi}$, а дисперсия равна $D[X] = \frac{\lambda(2-\xi)}{\xi^2}$.

Для случая $\xi > 1$ поток кадров будет эрланговским потоком ξ-го порядка, плотность распределения вероятностей которого будет иметь вид: $f(t) = \frac{\lambda(\lambda t)^{\xi-1}}{(\xi-1)!} e^{-\lambda t}$. Математическое ожидание числа поступающих кадров равно $M[X] = \frac{\lambda}{\xi}$, а дисперсия равна $D[X] = \frac{\lambda}{\xi^2}$.

Достаточно высокие требования к верности и скорости передачи информации, предъявляемые со стороны пользователей к сети, приводят к необходимости применения специальных мер по борьбе с ошибками, возникающими в канале передачи данных. Наиболее простой из математических моделей является модель независимых ошибок, которая исходит из предположения, что ошибки в дискретном канале связи возникают независимо друг от друга, то есть между ними отсутствует корреляция. Модель независимых ошибок может быть использована для описания потока ошибок в оптоволоконных линиях связи и линиях связи локальных сетей, построенных на основе медного кабеля. Функция распределения интервалов между независимыми ошибками может быть выражена формулой: $\Phi(u) = \sum_{i=1}^{K} A_i e^{-\varphi u}$, где K – количество причин, вызывающих ошибки, A – коэффициент воздействия K-й ошибки на передаваемый полезный сигнал, φ – интенсивность возникновения ошибок в канале передачи данных, u – длительность интервала между ошибками. Исследования каналов связи общего пользования, построенных на основе других сред передачи данных, отличных от волоконнооптических, не подтвердили гипотезу о независимости ошибок в этой группе каналов связи, а, напротив, выявили тенденцию к группированию (пакетированию) ошибок. Под пакетом ошибок понимается непрерывная последовательность искаженных элементов в информационном пакете. Функция распределения интервалов между пакетированными ошибками имеет вид: $\Phi(u) = Ae^{-\varphi u} + (1 - A)e^{-\varphi u}$.

2. Подмодель процесса коммутации. В технике связи различают три метода коммутации:

а) Коммутация каналов. Коммутация каналов – это метод статической коммутации, когда между отправителем и получателем устанавливается выделенное физическое соединение, поддерживаемое на все время передачи информации. Передающая ЭВМ запрашивает соединение с адресатом, ЭВМ-получатель дает подтверждение о готовности к приему данных. После передачи всей информации ЭВМ-получатель дает подтверждение о приеме информации, и канал разрывается. Преимущество коммутации каналов состоит в обеспечении гарантированной производительности линии передачи данных, что бывает важно для передачи больших объемов мультимедийной информации в реальном масштабе времени. В то же время, коммутация каналов имеет ряд существенных недостатков, ограничивающих её применение:

- неэффективное использование среды передачи данных;
- требование большой полосы пропускания;
- наличие задержек соединения;

б) Коммутация сообщений. Коммутация сообщений – это метод динамической коммутации, когда каждое сообщение интерпретируется как независимая единица, передающаяся от одного узла сети к другому. Каждый промежуточный узел сети получает и хранит сообщение, пока следующий узел не будет готов его принять. По этой причине сеть с коммутацией сообщений называют сетью с промежуточным хранением. Время пересылки по такой сети может достигать нескольких минут. Преимущества коммутации сообщений:

- эффективное управление сетевым обменом. Присваивая приоритеты коммутируемым сообщениям, можно достичь оптимального времени их доставки адресатам;
- уменьшение нагрузки на сеть. Маршрутизаторы могут хранить сообщения, пока не станет доступным коммуникационный канал;
- возможность передачи сообщений с той скоростью, какую позволяет пропускная способность коммуникационного канала;
- обеспечение коммуникаций между различными временными зонами.

Недостатки коммутации сообщений:

- невозможность передачи сообщений в реальном масштабе времени;
- высокая стоимость промежуточного оборудования (маршрутизаторов), обусловленная большими объемами дисковой и оперативной памяти, необходимой для промежуточного хранения сообщений;

в) Коммутация пакетов. Коммутация пакетов – это метод динамической коммутации, когда сообщение разбивается на короткие пакеты, которые передаются адресату одновременно по разным маршрутам. Это позволяет совместить преимущества

коммутации каналов и коммутации сообщений и, в то же время, избежать их недостатков. Маршрутизация одного сообщения к одному адресату по разным маршрутам называется независимой. Преимущества независимой маршрутизации:

- эффективное управление загрузкой коммуникационных каналов;
- высокая живучесть коммуникационного процесса.

Размер пакета выбирается таким, чтобы при промежуточном хранении его не надо было записывать на диск, то есть чтобы он полностью помещался в оперативную память. Различают два вида коммутации пакетов:

- коммутация датаграмм пакетов. Датаграмма – это пакет, передаваемый через сеть независимо от других пакетов без установления логического соединения и подтверждения приема. Датаграмма передается без всякого предупреждения и подготовки. Маршрутизаторы направляют потоки датаграмм в коммуникационные каналы таким образом, чтобы обеспечить полную и равномерную без перегрузки загрузку канала. К адресату датаграммы попадают в произвольном порядке, в зависимости от длины пути, пройденного каждой конкретной датаграммой. На сетевом уровне ЭВМ-получателя датаграммы упорядочиваются в готовое сообщение;
- коммутация пакетов в виртуальных каналах. Коммутация пакетов в виртуальных каналах устанавливает логическое соединение между передающей и принимающей ЭВМ, называемое виртуальным каналом. Физического канала при этом не создается, а благодаря большой пропускной способности линии связи, создается впечатление, что этот канал имеется и поддерживается в течение всего сеанса передачи. Коммутация пакетов в виртуальных каналах используется, как правило, для передачи мультимедийной информации.

Преимущества коммутации пакетов:

- возможность работы нескольких пар абонентов по одному коммуникационному каналу в реальном масштабе времени;
- малые задержки при передаче сообщений.

Недостатки коммутации пакетов:

- высокая стоимость промежуточного оборудования (маршрутизаторов), обусловленная:
- - большими объемами быстродействующей оперативной памяти, необходимой для промежуточного хранения пакетов;
- - высоким быстродействием специализированных управляющих ЭВМ маршрутизаторов;
- большой процент потерянных пакетов.

Процесс коммутации удобно представлять в матричной форме с использованием квадратных матриц размера N x N, где N – число входов и выходов коммутационной структуры, входя-

щей в состав маршрутизатора. Такие матрицы называются матрицами связи. Матрица связи имеет следующий вид:

$$|M| = \begin{vmatrix} m_{11} & m_{12} & \dots & m_{1N} \\ m_{21} & m_{22} & \dots & m_{2N} \\ \dots & \dots & \dots & \dots \\ m_{N1} & m_{N2} & \dots & m_{NN} \end{vmatrix}$$

Каждый элемент матрицы связи m_{ij} ассоциирован с конкретной точкой коммутации коммутационной структуры маршрутизатора и равен нулю, если отсутствует связь между входом i и выходом j, и, напротив, равен единице, если есть связь между входом i и выходом j. Ввиду того, что элементы матрицы связи могут принимать только два значения – ноль и единица, к ним применим аппарат булевой алгебры логики. Для оптимизации процесса принятия решения о том, какую точку коммутации выбрать при наличии нескольких возможных вариантов направлений передачи информации, анализируются критерии эффективности коммутируемых каналов связи:

а) Коэффициент использования канала связи (коэффициент загрузки) $\rho = \frac{I_П}{I_П + I_С}$, где $I_П$ – количество информации пользователя в битах, передаваемое по каналу связи за одну секунду, $I_С$ – количество служебной информации в битах, передаваемое по каналу связи за одну секунду, $I_П + I_С$ – общее количество информации в битах, передаваемое по каналу связи за одну секунду;

б) Допустимое время задержки прохождения сообщения пользователя по каналу связи $T_{ДОП}^{ЗАД} = t_0$;

в) Вероятность отказа канала $P_{ОТК}$.

Общее время задержки прохождения сообщения пользователя между абонентами будет складываться из сумм задержек на прямых соединениях между узлами сети, времени обработки на каждом из узлов сети и времени обработки вне сети (на конечных участках маршрута): $T_{ОБЩ}^{ЗАД} = \sum_{i=1}^{N} T_i^{ЗАД}$, где N – число элементов маршрута. Каждое из слагаемых времени задержки может оказаться решающим при выборе наиболее эффективного элемента матрицы связи (точки коммутации). Плотность распределения времени задержки прохождения сообщения пользователя определяется формулой:

$$f(T_i^{ЗАД}) = \begin{cases} \frac{\alpha}{t_0}\left(\frac{t_0}{T_i^{ЗАД}}\right)^{\lambda+1}, & \text{при } T_i^{ЗАД} > t_0 \\ 0, & \text{при } T_i^{ЗАД} \le t_0 \end{cases}$$

где λ – интенсивность поступления сообщений в канал связи. Математическое ожидание времени задержки прохождения сообщения пользователя равно $M[T_i^{ЗАД}] = \frac{\lambda}{\lambda-1} t_0$, а дисперсия равна $D[T_i^{ЗАД}] = \frac{\lambda}{(\lambda-1)(\lambda-2)} t_0^2$. Численный анализ вышеприведенных формул дает результаты, из которых можно сделать следующие выводы:

а) Резервирование при методе коммутации каналов вызывает снижение общей производительности сети за счет неполного использования пропускной способности каналов связи;

б) Для больших длин сообщений пользователей метод коммутации каналов дает лучшие результаты по сравнению с методами коммутации пакетов и коммутации сообщений, так как обеспечивает ме́ньшую задержку и примерно такое же время установления соединения;

в) Использование метода коммутации пакетов позволяет получить ме́ньшую среднюю задержку, чем при методе коммутации сообщений для малых и средних интенсивностей поступления сообщений пользователей;

г) При росте интенсивностей поступления сообщений пользователей и малых значениях количества узлов сети, задержка при коммутации пакетов может оказаться больше, чем при коммутации сообщений. Также возрастает вероятность перегрузки (снижается коэффициент использования канала связи ρ).

Ввиду того, что параметры информационных потоков в цифровых сетях, учитываемые алгоритмами управления коммутацией в маршрутизаторах, оцениваются с запаздыванием, они в определенной степени не соответствуют текущему состоянию обмена и коммутационной системы. Для повышения эффективности этих алгоритмов используются адаптивные методы краткосрочного прогнозирования состояния коммутаторов. Прогнозирование позволяет заранее предвидеть изменение состояния коммутаторов и вовремя произвести начальный расчет по прогнозируемым данным для нахождения оптимального распределения ресурсов системы. Для правильной оценки состояния коммутаторов необходимо иметь точные значения параметров нагрузки по видам обмена, коэффициентов использования ресурсов сети и каналов связи. Цели адаптивного прогноза:

а) Предоставить устройству управления маршрутизатора возможность использовать более достоверную информацию о состоянии занятости входящих каналов связи;

б) Обеспечить заблаговременную загрузку и запуск соответствующих управляющих программ, обеспечивающих коммутацию в маршрутизаторах.

Из-за жестких ограничений, накладываемых на время работы программ прогнозирования, на практике применяются в основном только простейшие прогнозные модели экспоненциального типа. Краткосрочный прогноз обеспечивается при выполнении следующих условий:

а) Период времени, в течение которого изучается прогнозируемый процесс, должен быть достаточным, чтобы можно было проследить его закономерности;

б) Прогнозируемый процесс должен развиваться эволюционно;

в) Прогнозируемый процесс должен обладать инерционностью;

г) Прогнозируемый процесс должен описываться затухающей функцией.

Сущность метода адаптивной коммутации состоит в:

а) Замене алгоритма коммутации в маршрутизаторе при изменении состояния устройств сети;

б) Автоматической корректировке значений параметров ограничений на работу маршрутизатора;

в) Прогнозе состояния сети для обеспечения эффективной работы алгоритмов коммутации и маршрутизации.

Для предотвращения замедления работы маршрутизатора при переходе с одной управляющей программы на другую вследствие необходимости загрузки программных модулей и производства начальных расчетов, осуществляется параллельная работа программ, реализующих различные алгоритмы коммутации, в виде потоков выполнения. При этом активным является только один алгоритм, результаты работы других алгоритмов не используются, но в любой момент времени могут быть востребованы. В целях дальнейшего повышения эффективности коммутационного процесса, алгоритмы адаптивной коммутации распространяются в том числе на сетевой и канальный уровни модели OSI. На сетевом уровне алгоритмы коммутации выполняют следующие функции:

а) Изменение метода коммутации передаваемой информации;

б) Изменение длины передаваемых кадров;

в) Перераспределение потоков передаваемой информации между коммуникационными каналами в заданном направлении передачи;

г) Реорганизация логических каналов в соответствии с приоритетами передаваемой информации;

д) Осуществление управления потоками данных.

Изменение алгоритма коммутации производится в следующих случаях:

а) При отсутствии свободной пропускной способности канала связи для передачи относительно коротких несвязанных между собой пакетов;

б) Если в режиме коммутации каналов передаются данные;

в) При увеличении задержек в передаче пакетов;

г) При перегрузке каналов в режиме коммутации каналов.

3. Подмодель маршрутизации. Определение оптимальных маршрутов для потоков информации в цифровых сетях является решением задачи распределения потоков. Основным критерием эффективности распределения потоков является средняя вероятность блокировки кадров: $P = \sum_{i=1}^{M}(\frac{\lambda_i}{\gamma} P_i)$, где M – число каналов связи в сети, λ_i – интенсивность потока кадров по i-му каналу, γ – коэффициент избыточности в сети, P_i - вероятность блокировки кадра в i-м канале связи при интенсивности потока кадров λ_i. Эффективное управление процессом маршрутизации информацией в цифровой сети возможно лишь при использовании адаптивных методов маршрутизации и ограничения интенсивности потоков. Сравнивать различные методы управления маршрутизацией необходимо с точки зрения конечных результатов их функционирования. Критерии эффективности различных методов управления маршрутизацией:

а) Производительность сети ЭВМ (R_N). Под производительностью сети ЭВМ понимают количество информации пользователей, содержащейся во всех кадрах, обслуженных сетью полностью и с заданным качеством, за единичный интервал времени её функционирования. Кадр считается обслуженным полностью и с заданным качеством, если:

- в режиме коммутации каналов установлено соединение, обеспечивающее передачу всей необходимой пользователю информации с заданным качеством;

- в режимах коммутации сообщений и коммутации пакетов сообщение пользователя передано отправителем и принято получателем за определенный интервал времени и с заданным качеством.

Если пренебречь повторными поступлениями кадров, вызванными их потерями в сети, либо доставкой сообщений пользователя с качеством ниже заданного и нарушающим пуассоновский характер входного потока, то производительность сети равна: $R_N = J_\Sigma(1 - P_{HO}(J_\Sigma))$, где J_Σ – сумма частных производительностей сети по каждому виду передаваемой информации, P_{HO} – вероятность необслуживания кадра, то есть вероятность недоставки, неполной доставки, либо доставки с качеством ниже заданного, информации, содержащейся в кадре. При отсутствии перегрузок в сети $P_{HO} \ll 1$;

б) Среднее время задержки сообщения пользователя в сети ($\overline{T_c}$). Среднее время задержки сообщения пользователя в сети – это среднее по всему множеству сообщений время от момента приема первого бита сообщения от пользователя в рабочей станции-отправителе до передачи последнего бита сообщения рабочей станцией-получателем пользователю. Критерий среднего времени задержки сообщения пользователя в сети необходимо рассматривать отдельно для каждого вида передаваемой информации и каждого реализованного метода коммутации;

в) Средняя дисперсия времени задержки сообщения пользователя в сети ($\overline{D_c}$). Средняя дисперсия времени задержки сообщения пользователя в сети – это среднее по сети среднеквадратическое отклонение времени задержки сообщения пользователя для осуществления запроса на передачу информации определенного вида и приоритета при заданном методе коммутации;

г) Среднее время задержки пакета в сети ($\overline{T_п}$). Среднее время задержки пакета в сети – это среднее по всем пакетам время от момента передачи пакета отправителем до момента его успешного приема получателем. С точки зрения реализации алгоритмов управления маршрутизацией и ограничением потоков, этот критерий является более удобным для использования, чем среднее время задержки сообщения пользователя в сети, так как имеет чёткую связь с длинами очередей пакетов по исходящим направлениям маршрутизатора и достаточно легко физически оценивается;

д) Среднее время установления соединения ($\overline{\tau_c}$). Среднее время установления соединения – это среднее по всем устанавливаемым соединениям время от момента передачи запроса на соединение отправителем до момента получения им подтверждения о том, что соединение установлено;

е) Средняя вероятность искажения информации ($\overline{P_e}$). Средняя вероятность искажения информации – это средняя вероятность искажения информационных бит кодовой комбинации кадра. Для программных файлов средняя вероятность искажения информации допускается не хуже $10^{-12} \div 10^{-10}$, а для речи – $10^{-2} \div 10^{-1}$.

С точки зрения администрации сети наиболее важным представляется критерий производительности сети, так как в конечном счете именно он определяет её экономическую эффективность. В сетях ЭВМ одновременно могут передаваться потоки самой разнообразной информации, причем бóльшая доля этой информации обрабатывается не всеми, а только лишь частью маршрутизаторов сети. Взаимное влияние этих потоков друг на друга, наличие запросов различных приоритетов, а также раз-

личные требования к показателям качества передачи для сообщений пользователей различных типов, вызывают необходимость одновременного выполнения в маршрутизаторах нескольких алгоритмов управления в форме потоков выполнения. По этой причине модель маршрутизатора с точки зрения реализации алгоритмов маршрутизации и ограничения интенсивности потоков должна предусматривать:

а) Возможность одновременного выполнения в маршрутизаторах нескольких алгоритмов управления;

б) Параллельную работу в маршрутизаторах нескольких алгоритмов управления без взаимных блокировок;

в) Переход от одного алгоритма управления к другому в зависимости от условий работы цифровой сети;

г) Добавление новых алгоритмов управления по мере расширения возможностей сети без изменения старых.

Рассмотрим логическую структуру маршрутизатора.

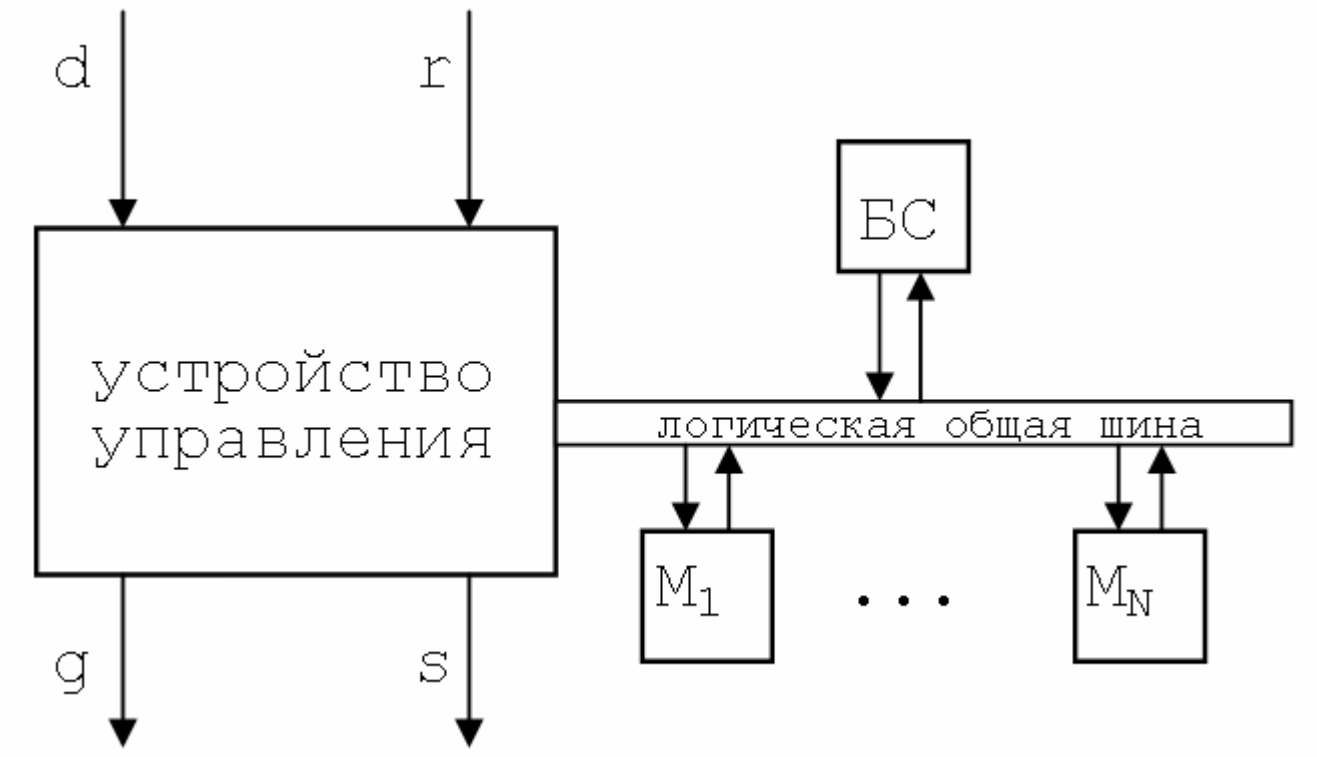


Рис. 2. Логическая структура маршрутизатора.

БС – программный блок состояния

M_i – программные модули маршрутизации

N – число алгоритмов маршрутизации, поддерживаемых в отдельный момент времени маршрутизатором

На вход устройства управления блоком поступает информация о состоянии сети *d*, необходимая для работы алгоритма ограничения интенсивности потоков и алгоритмов маршрутизации, выполняемых непосредственно в устройстве управления, и запросы на маршрутизацию *r*. На выход устройства управления потоком поступают сообщения о состоянии маршрутизатора *g*,

необходимые для функционирования алгоритмов управления в других маршрутизаторах сети, и ответы маршрутизатора на запросы маршрутов от других маршрутизаторов *s*. Функции устройства управления маршрутизатора:

а) Управление интенсивностью потоков сообщений пользователей, поступающих на его вход;

б) Распределение запросов на маршрутизацию между программными модулями маршрутизации;

в) Распределение управляющей информации по программным модулям маршрутизации;

г) Сбор управляющей информации от программных модулей маршрутизации;

д) Запуск и останов алгоритмов маршрутизации (создание и уничтожение программных модулей маршрутизации) в зависимости от оценки состояния сети;

е) Переключение алгоритмов маршрутизации;

ж) Переназначение запросов с одного модуля маршрутизации на другой в зависимости от оценки состояния сети.

Основная информация, на основании которой функционируют алгоритмы маршрутизации, содержится в таблице маршрутизации. Таблица маршрутизации представляет собой матрицу, число строк которой равно числу узлов на том иерархическом уровне коммутации, на котором находится данный маршрутизатор и поддерживается данный алгоритм коммутации. Каждая строка таблицы маршрутизации содержит номера выходных каналов маршрутизатора для основного и альтернативных путей в пункт назначения, а также критерии эффективности этих путей.

Говоря об аппаратном обеспечении распределенных автоматизированных систем, нельзя забывать, что эти системы строятся на основе цифровых сетей интегрального обслуживания, которые на единой цифровой основе обеспечивают передачу различных видов информации. В настоящее время сети ЭВМ могут передавать следующие первичные информационные сигналы: телефонирования, звукового вещания, телеграфирования и передачи данных, факсимильные, телевизионного вещания. Рассмотрим основные параметры и характеристики этих сигналов.

§ 2. Сигналы телефонирования

Сигналы телефонирования представляют собой последовательности речевых импульсов, отделенных друг от друга паузами и состоят из комбинации передаваемых в разные отрезки времени речевых сигналов и сигналов управления и взаимодействия коммутационных устройств. Последние можно рассматривать как разновидность сигналов передачи данных, поэтому при описании сигналов телефонирования ограничимся парамет-

рами и характеристиками речевых сигналов. Речевые импульсы соответствуют звукам речи, произносимым слитно, и весьма разнообразны по форме и амплитуде. Длительности отдельных импульсов также отличаются друг от друга, но обычно они близки к 100 ÷ 150 мс. Паузы между импульсами изменяются в значительно большем диапазоне: от нескольких миллисекунд (межслоговые паузы) до нескольких минут или даже десятков минут (паузы при выслушивании ответа собеседника).

Частотный спектр речевого сигнала очень широк, однако экспериментально было установлено, что для передачи с достаточно высоким качеством (удовлетворительной натуральностью и разборчивостью слогов 90% и фраз 99%) можно ограничиться полосой частот 0,3 ÷ 3,4 кГц. Назовем $u_{ТФ}$ эффективным (среднеквадратическим) напряжением сигнала $u(t)$. Тогда можно записать:

$$u_{ТФ} = \sqrt{P_{ТФ} \cdot 1Ом} = \sqrt{\frac{1}{T_Н}\int_0^{T_Н} u^2(t)dt},$$

где $P_{ТФ}$ – мощность сигнала, усредненная за время наблюдения $T_Н$.

Отношение $y_{ТФ} = 10 \lg\frac{P_{ТФ}}{P_{ИЗМ}}$ дБмО называется динамическим уровнем (волюмом) телефонного сигнала. В этом выражении $P_{изм}$ – мощность измерительного сигнала в точке тракта, где проводится исследование. Согласно рекомендациям Международного консультативного комитета по телефонии и телеграфии (МККТТ) волюмы измеряются специальным прибором (волюмметром), обеспечивающим квадратичный закон суммирования колебаний различных частот и имеющим логарифмическую шкалу (в децибелах) и постоянную времени (время интегрирования) $T_Н$ = 200 мс. Статистическими исследованиями установлено, что разброс волюмов подчиняется гауссовому закону распределения со средним значением $y_{ТФ\ ср}$ = -12,7 дБмО и среднеквадратическим отклонением σ_y = 4,3 дБ. На основании указанных данных можно определить среднюю мощность телефонного сигнала $P_{ТФ\ ср}$. Для этого необходимо перейти от среднего логарифма ($y_{ТФ\ ср}$) к логарифму среднего $P_{ТФ\ ср}$, – уровню, соответствующему средней мощности: $P_{ТФ\ ср} = y_{ТФ\ ср} + 0{,}115\ \sigma_y^2 = -10{,}57$ дБмО. Тогда $P_{ТФ\ ср} = 1 \cdot 10^{0{,}1(-10{,}57)} = 88$ мкВтО -средняя мощность телефонного сигнала без учета пауз в точке нулевого относительного уровня по мощности.

Влияние пауз учитывается посредством коэффициента активности $K_А$ источника сигнала, равного отношению времени, в течение которого уровень сигнала на его выходе превышает установленное пороговое значение (обычно - 40 дБмО), к общему времени разговора. Для телефонных сигналов $K_А$ = 0,25. Тогда средняя мощность телефонного сигнала с учетом пауз

$P_{ТФ\ ср\ П} \approx 10K_A \cdot P_{ТФ\ ср}$ = 32 мкВтО (-15 дБмО), где первый сомножитель правой части, равный 10 мкВтО, вводится согласно рекомендациям Международного консультативного комитета по телефонии и телеграфии (МККТТ) как поправка на повышенную мощность сигналов, сопровождающих телефонный разговор (служебные переговоры персонала и сигналы управления и взаимодействия, передаваемые по тому же каналу). Установлено также, что $P_{ТФ\ max}$ = 2220 мкВтО ($P_{ТФ\ мах}$ = 3,5 дБмО) при $\varepsilon = 10^{-3}$. При определении величины флуктуационной помехи, действующей на входе оконечного аппарата, ее приводят к эффективно воздействующей на органы слуха взвешенной помехе. Суть взвешивания заключается в том, что на входе измерительного прибора устанавливается амплитудный корректор, частотная характеристика передачи которого повторяет среднестатистическую характеристику чувствительности системы «телефонный аппарат – слух». Очевидно, что взвешенное значение помехи будет меньше невзвешенного из-за меньшей чувствительности указанной системы на краях частотного диапазона, а значит, и большего затухания корректора на этих же частотах. Снижение действующего напряжения равномерно распределенной по спектру помехи определяется псофометрическим коэффициентом $K_{ПС}$, равным 1,33 для полосы частот 0,3 ÷ 3,4 кГц. Средняя мощность этой же помехи будет снижена в $1{,}33^2$ = 1,77 раза, а уровень - на 20 lg1,33 = 2,48 дБ. В размерности взвешенных (псофометрических) величин вводится буква «п», например дБмОп, пВтОп и т. д.

Экспериментально установлено, что качество приема телефонного сигнала еще достаточно при средней мощности помехи 178000 пВтО или 100000 пВтОп. При определении пик-фактора и помехозащищенности сигнала используют среднюю мощность сигнала без учета пауз: $Q_{ТФ} = 10 \lg\frac{2220 \cdot 10^{-6}}{88 \cdot 10^{-6}} \approx 14$ дБ и $A_{ЗТФ} = 10 \lg\frac{88 \cdot 10^{-6}}{178000 \cdot 10^{-12}} \approx 27$ дБ. Динамический диапазон телефонного сигнала $D_{ТФ} = 10 \lg\frac{2220 \cdot 10^{-6}}{178000 \cdot 10^{-12}} \approx 41$ дБ. При оценке потенциального информационного объема необходимо учитывать коэффициент активности источника сигнала. Тогда:

$$V_{ТФ\ max} = K_A\ F_B \log_2(1 + 10^{0,1A}) =$$
$$= 0{,}25 \cdot 3400 \cdot 3{,}32 \log_2(1+10^{0,1 \cdot 27}) = 7{,}6 \text{ кбит/с.}$$

Здесь множитель $3{,}32 = \frac{1}{\lg 2}$ - модуль перехода от двоичного логарифма к десятичному; F_B - верхняя эффективно передаваемая частота канала, кГц.

§ 3. Сигналы звукового вещания

Сигналы звукового вещания по своему характеру близки к речевым телефонным сигналам, поэтому их отличия от последних носят количественный характер. Частотный спектр сигналов звукового вещания ограничивают без заметного снижения качества передачи до 0,03 ÷ 15 кГц для каналов высшего класса и до 0,05 ÷ 10 кГц для каналов первого класса. Сигналы звукового вещания по сравнению с телефонными имеют значительно меньше пауз, а энергия отдельных импульсов, особенно музыкальных, существенно превышает энергию речевых импульсов телефонных сигналов. Поэтому средняя мощность сигналов звукового вещания намного больше средней мощности телефонных сигналов. Нормируются среднесекундная, среднеминутная и среднечасовая мощности $P_{ЗВ\ ср}$, равные соответственно 4500, 2230 и 923 мкВтО. Максимальная мощность определяется при вероятности превышения ε = 0,02 и составляет 8000 мкВтО. Минимальная мощность рассчитывается при вероятности превышения (1 - ε) = 0,98. Её значения различны для тех или иных видов сигналов и дают следующие значения динамического диапазона $D_{ЗВ}$ сигналов звукового вещания, дБ:

1. Речь диктора - до 35;
2. Художественное чтение - до 50;
3. Музыкальные и хоровые ансамбли - до 55;
4. Симфонический оркестр - до 65;

Взвешенная флуктуационная помеха на входе оконечного аппарата звукового вещания не должна превышать 16000 пВтОп. Поскольку спектр помехи в каналах звукового вещания шире, псофометрический коэффициент для них оказывается больше. Так, для канала первого класса он равен 2, т. е. мощность невзвешенной помехи может достигать 16000 · 2^2 = 64000 пВтО, следовательно, помехозащищенность сигналов звукового вещания должна быть не хуже $A_{ПЗ\ ЗВ} = 10\ \lg\frac{923\cdot10^{-6}}{64000\cdot10^{-12}} \approx 42$ дБ. Таким образом, потенциальная информационная емкость сигнала звукового вещания первого класса может достигать:

$$V_{ЗВ\ max} = 10000 \cdot 3{,}32\ \lg(1 + 10^{0{,}1\ \cdot\ 42}) \approx 140\ \text{кбит/с}.$$

§ 4. Сигналы телеграфирования и передачи данных

Сигналы телеграфирования и передачи данных чаще всего представляют последовательности униполярных или биполярных импульсов постоянной амплитуды, при этом положительный импульс обычно соответствует передаваемому знаку «1», а пропуск импульса или отрицательный - знаку «0». Частота следования «1» и «0» называется тактовой частотой F_T. Численно F_T соответствует скорости передачи информации в бодах, а в

данном случае (два разрешенных значения «1» и «0») - и скорости передачи в битах в секунду (бит/с).

Условно различают низкоскоростную (до 200 Бод), среднескоростную (300 ÷ 1200 Бод) и высокоскоростную (свыше 1200 Бод) передачу данных. Поскольку каждый передаваемый импульс занимает полностью тактовый интервал, его длительность находится в пределах до 5 мс при низкоскоростной, от 3,3 до 0,8 мс при среднескоростной и менее 0,8 мс при высокоскоростной передачах.

Из курса теории электросвязи известно, что спектральная плотность случайного сигнала такого вида максимальна на нулевой частоте и имеет первый минимум на частоте F_T. Если спектр сигнала ограничивать фильтром низких частот, близким к идеальному, то уверенный прием сигнала возможен при частоте среза фильтра, равной или более $0,5F_T$, т. е. можно считать, что эти сигналы занимают полосу частот $0 \div 0,5F_T$. Однако в реальных условиях верхнюю частоту спектра сигнала телеграфирования и передачи данных принимают равной F_T или даже $1,2F_T$. Это обусловлено тем, что при некоторых видах передачи информация заложена и в изменении длительности импульса (допускаются ограниченные краевые искажения принимаемых импульсов), а также мешающим воздействием помех.

При передаче сигналов телеграфирования и передачи данных допустимая вероятность ошибки равна около 10^{-5}. Это позволяет принять значение необходимой помехозащищенности, определяемой как отношение амплитуды импульса к действующему значению флуктуационной помехи, равным $A_{З\ ТП}$ = 12 дБ. Методы подобных расчетов подробно рассмотрены ниже в разделе цифровых систем передачи.

§ 5. Факсимильные сигналы

Факсимильные сигналы (сигналы передачи неподвижных изображений) получаются в результате преобразования светового потока, отражаемого элементами изображения, в электрические сигналы. Падающий световой поток перемещается по изображению в определенной последовательности (например, по принципу строчной развертки). В такой же последовательности в приемном устройстве перемещается элемент, воздействующий в соответствии с принимаемыми сигналами на носитель записи и окрашивающий соответственно его участки. Так, на передаче световое пятно можно перемещать по передаваемому рисунку, а отраженный поток воспринимать фотоэлементом, на выходе которого будет получаться электрический сигнал. На приеме этот сигнал возбуждает светодиод. Перемещая сфокусированный в световоде световой поток синфазно с потоком на передаче

по фоточувствительной бумаге, получаем фотокопию передаваемого изображения.

При передаче штриховых изображений (состоящих из черных и белых элементов, например газетной полосы) факсимильный сигнал (ФС) состоит из униполярных импульсов различной длительности, но одинаковой амплитуды. Принимается, что полоса частот такого сигнала находится в пределах 0 ÷ $F_Р$, причем $F_Р$ – частота рисунка - связана с длительностью самого короткого импульса $\tau_И$ соотношением $F_Р = 0{,}5\tau_И$. В свою очередь, $\tau_И$ определяется диаметром светового пятна $d_С$ и скоростью развертки $V_Р$ (скорость перемещения светового пятна по рисунку): $\tau_И = \frac{d_С}{V_Р}$. При передаче документов выбирают $d_С$ = 0,15 мм и $V_Р \le 440$ мм/с, тогда $\tau_И$ = 0,34 мс, а $F_Р$ = 1500 Гц. При передаче газетных полос $d_С < 0{,}06$ мм, а $V_Р \le 30$ м/с. Частота рисунка при этом достигает 250 кГц.

Помехозащищенность факсимильных сигналов $A_{ПЗ\ ФС}$ (отношение амплитуды сигнала к действующему напряжению флуктуационной помехи) принимается равной 35 дБ. При передаче штриховых изображений потенциальная информационная емкость факсимильных сигналов $V_{ФСШ\ max} = 2F_Р \cdot \log_2 2 = 2F_Р$ бит/с.

При передаче полутоновых изображений в копиях должны различаться 16 градаций яркости, при этом динамический диапазон сигнала $D_{ФС\ П} = 20 \lg\frac{16+1}{1} = 24{,}6$ дБ.

Оценим пик-фактор $Q_{ФС\ П}$, если число градаций яркости l = 16. Будем считать, что все напряжения сигнала u_i соответствующие i-м градациям яркости, имеют одинаковую вероятность появления $P = \frac{1}{l}$. Соответствующее i-й градации напряжение $u_i = \frac{iU_М}{l}$, где $U_М$ – амплитудное значение сигнала. В свою очередь значение среднеквадратического напряжения сигнала равно: $u_{СР}^2 = \sum_{i=1}^{l} u_i^2 p_i = \sum_{i=1}^{l} \frac{i^2 U_М^2}{l^2}\frac{1}{l} = \frac{U_М^2}{l^3}\sum_{i=1}^{l} i^2$

Известно, что: $\sum_{i=1}^{l} i^2 = \frac{l(1+l)(1+2l)}{6}$.

Тогда: $u_{СР}^2 = \frac{U_М^2(1+l)(1+2l)}{6l^2}$

Поскольку можно считать, что

$Q = 10 \lg\frac{U_М^2}{u_{СР}^2}$,

то $Q_{ФС\ П} = 10 \lg\frac{6l^2}{(1+l)(1+2l)}$.

При l = 16 $Q_{ФС\ П}$ = 4,4 дБ.

Заметим, что увеличение числа градаций яркости мало влияет на рост пик-фактора. Несложно показать, что при $l \to \infty$, пик-фактор $Q_{ФС\ П}$ стремится к 10 lg3 = 4,8 дБ.

Необходимая помехозащищенность полутоновых сигналов, как и штриховых, $A_{З\ ФС}$ = 35 дБ. При этом потенциальная информационная емкость полутоновых сигналов:

$V_{ФС\ П\ max} = 2F_P \log_2 l = 2F_P \log_2 16 = 8F_P$ бит/с, т. е. в 4 раза больше, чем штриховых.

§ 6. Сигналы телевизионного вещания

Сигналы телевизионного вещания состоят из сигналов передачи подвижных изображений и звукового сопровождения. Сигналы звукового сопровождения передаются по отдельным каналам и ничем не отличаются от сигналов звукового вещания, которые уже были рассмотрены выше. Поэтому можно считать, что сигналы телевизионного вещания являются сигналами передачи подвижных изображений и состоят из суммы сигналов яркости (изображения), аналогичных полутоновым факсимильным сигналам, сигналов цветности и так называемой «синхросмеси» - комбинации импульсов синхронизации строк, полукадров и импульсов гашения обратного хода луча. Частота рисунка F_P сигналов яркости может быть подсчитана исходя из того, что число элементов изображения в кадре равно $\frac{4}{3}m^2$, где m = 625 - число строк в кадре принятой системы цветного телевидения СЕКАМ, а $\frac{4}{3}$ - отношение размеров кадра по горизонтали и вертикали. Учитывая, что в секунду передается 25 кадров (50 полукадров, состоящих поочередно из четных и нечетных строк изображения), имеем $F_P = \frac{4}{3}m^2 \cdot \frac{25}{2}$ = 6,5 МГц. Однако практически вся энергия сигналов яркости сосредоточена в диапазоне 0 ÷ 1,5 МГц.

Защищенность сигналов яркости от флуктуационной помехи должна быть не хуже 48 дБ. Поскольку высокие частоты сигнала соответствуют мелким деталям изображения, Международный консультативный комитет по телефонии и телеграфии (МКТТ) рекомендует при оценке помехи пользоваться взвешивающим фильтром с падающей амплитудно-частотной характеристикой (АЧХ). Уровень псофометрической помехи ниже уровня помехи с равномерным спектральным распределением на 9 дБ ($K_{ПС}$ = 2,82), т. е. $A_{ЗП}$ = 57 дБ.

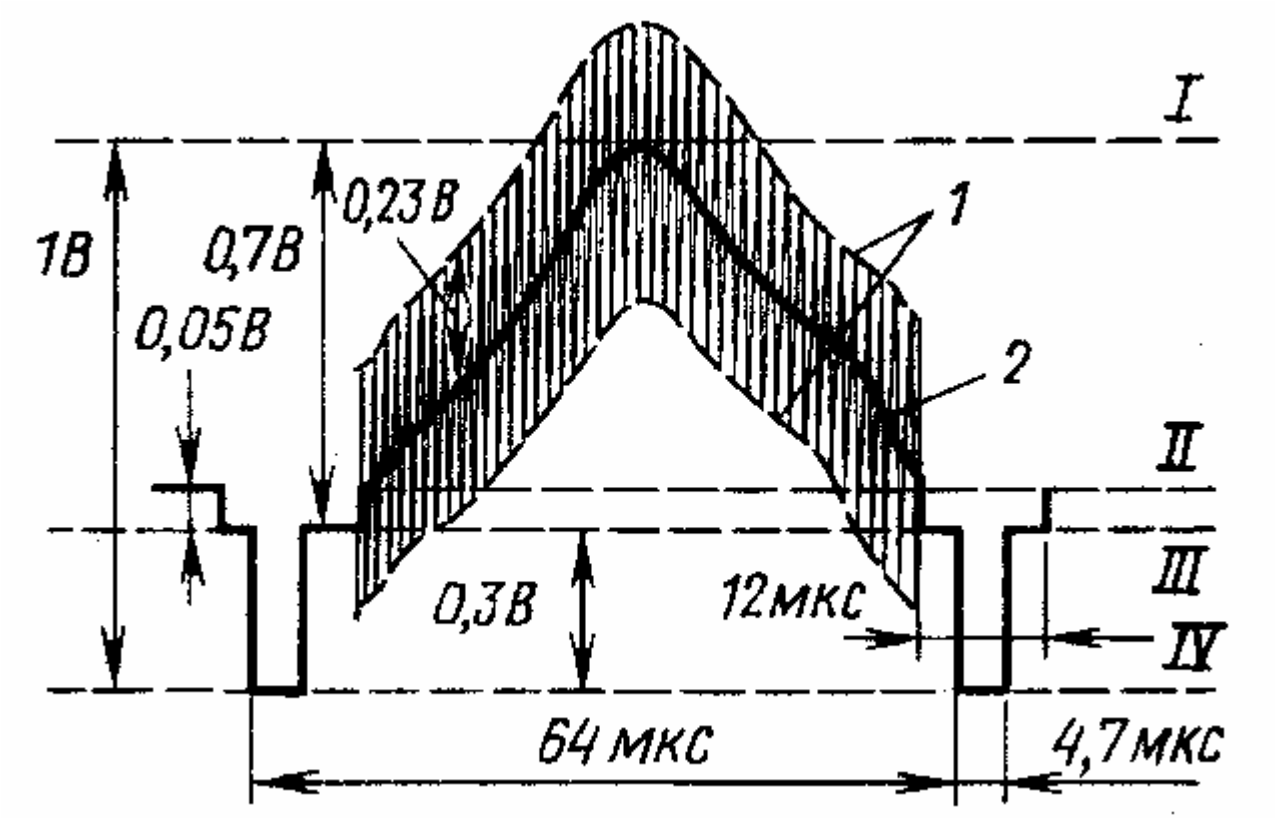


Рис. 3. Осцилограмма одной строки полного телевизионного сигнала.

I. – уровень белого; II. – уровень черного; III. – уровень гашения; IV. – уровень синхроимпульсов; 1 – сигнал цветности; 2 – сигнал яркости.

Число градаций яркости составляет 100, откуда $D_{ТВ}$ = 40 дБ. Пик-фактор сигнала, как было показано при рассмотрении полутонового факсимильного сигнала, не превысит 4,8 дБ, а потенциальный информационный объем

$V_{ТВ\ max} = 2 \cdot 6{,}5 \cdot 10^6 \cdot 3{,}32 \lg 100 = 86$ Мбит/с.

Сигналы цветности в этой системе представляют собой две поднесущие (4406,25 и 4250,00 кГц), промодулированные по частоте двумя чередующимися от строки к строке цветоразностными сигналами. Амплитуда поднесущих составляет 23% от размаха сигнала яркости. Частотный спектр сигналов цветности совмещается с верхней частью спектра сигнала яркости. Складываясь с сигналами яркости, сигналы цветности вызывают периодическое изменение яркости свечения экрана, что, однако, из-за инерционности зрения не влияет на восприятие изображения.

Нулевое напряжение сигнала яркости соответствует уровню черного, а максимальное - уровню белого. Импульсы синхронизации в этом случае передают отрицательным напряжением («чернее черного»), чтобы они не воспроизводились на приемном экране. Принято, что размах полного телевизионного сигнала на выходе телецентра составляет 1 В на нагрузке 75 Ом.

На рисунке приведена осциллограмма одной строки полного телевизионный сигнала, там же указаны соотношения между отдельными составляющими и длительности строки, импульса гашения и строчного синхроимпульса. Полукадры, состоящие примерно из 310 строк, отделяют друг от друга 25 чистыми строками (передаются только строчные синхроимпульсы и импульсы гашения). В этот промежуток через три строки после окончания полукадра вводится полукадровый синхроимпульс длительностью в три строки. При этом передача строчных синхроимпульсов и импульсов гашения не прекращается.

Спектр сигналов синхросмеси линейчатый с частотами $mf_{ПК}$, $nf_{СТ} \pm mf_{ПК}$, где m и n - целые числа; $f_{ПК}$ - частота следования полукадровых импульсов, равная 50 Гц; $f_{СТ}$ - частота следования строчных импульсов, равная 15625 Гц. Практически вся энергия этих сигналов сосредоточена в диапазоне 0,05 ÷ 300 кГц ($n \approx m < 18$).

§ 7. Уровни передачи сигналов в сетях ЭВМ

Любая информационная сеть (телефонная, радио, сеть ЭВМ и т.п.) предназначается для передачи сообщений посредством информационных сигналов. В общем случае *сообщением* является совокупность сведений о состоянии какого-либо материального объекта, поэтому в пункте передачи посредством оконечного (абонентского) аппарата должен быть сформирован сигнал, называемый первичным и соответствующий данному сообщению. В пункте приема абонентский аппарат осуществляет обратный процесс - в соответствии с принятым первичным сигналом формирует сообщение. Так, при передаче сигналов звукового вещания сообщением является изменение звукового давления, оконечным аппаратом передачи - микрофон, а приема - громкоговоритель.

Ввиду низкой стоимости и простоты инсталляции сред передачи данных, построенных на основе медного кабеля, наиболее распространены в настоящее время электрические сигналы. Количественно их можно характеризовать мощностью, напряжением и (или) током. Однако в технике электросвязи принято пользоваться логарифмическими характеристиками (уровнями передачи), что позволяет существенно упростить многие расчеты. Уровни передачи, вычисленные посредством десятичных логарифмов, называются децибелами (дБ), а посредством натуральных логарифмов - неперами (Нп). В настоящее время принято пользоваться децибелами.

Уровни передачи по мощности, напряжению и току определяются соответственно по формулам:

$$P_M = 10 \lg\left(\frac{P_X}{P_0}\right),\ P_H = 20 \lg\left(\frac{U_X}{U_0}\right),\ P_T = 20 \lg\left(\frac{I_X}{I_0}\right),$$

где P_X, U_X, I_X – величины мощности, напряжения и тока в рассматриваемой точке X; P_0, U_0, I_0 - величины, принятые за исходные. Если известны значения сопротивлений Z_X и Z_0, на которых выделяются мощности P_X и P_0, то на основании известного соотношения $P = \frac{U^2}{|Z|} = \frac{I^2}{|Z|}$ между уровнями передачи по мощности, напряжению и току могут быть найдены зависимости:

$$P_M = 10 \lg \frac{U_X^2}{|Z_X|} \cdot \frac{|Z_0|}{U_0^2} = P_H + 10 \lg \frac{|Z_0|}{|Z_X|},$$

$$P_M = 10 \lg \frac{I_X^2}{I_0^2} \cdot \frac{|Z_X|}{|Z_0|} = P_T - 10 \lg \frac{|Z_0|}{|Z_X|},$$

$$P_H = P_T - 20 \lg \frac{|Z_0|}{|Z_X|}.$$

Очевидно, что при $|Z_X| = |Z_0|$ уровни $P_M = P_H = P_T$.

Если за исходные величины мощности, напряжения и тока приняты соответственно P_0 = 1мВт (мВА), U_0 = 0,7746 В и I_0 = 1,291 мА, то вычисленные уровни называют абсолютными и обозначают дБм, дБн и дБт. Заметим, что указанные значения U_0 и I_0 получены в предположении, что $P_0 = 1$ мВА выделяется на сопротивлении $|Z_0|$ = 600 Ом.

При подаче на вход исправного и отрегулированного тракта синусоидального сигнала с абсолютным уровнем и частотой, рекомендованными для измерения этого тракта, в точках тракта устанавливаются абсолютные уровни, которые называются измерительными. Измерительные уровни содержатся в техническом паспорте тракта и удобны при проверке и настройке последнего.

Иногда в качестве исходных величин принимают значения P_H, U_H, I_H, установленные в начале тракта или в точке, принятой условно за начало. Тогда вычисленные уровни:

$$P_{M0} = 10 \lg(\frac{P_X}{P_H}),\ P_{H0} = 20 \lg(\frac{U_X}{U_H}),\ P_{T0} = 20 \lg(\frac{I_X}{I_H})$$

называют относительными и обозначают дБом, дБон и дБот соответственно. Эти уровни широко используют при измерениях передаточных характеристик трактов, поскольку их значения оказываются численно равными усилению по мощности, напряжению или току участка тракта от начала до данной точки. Очевидно, что отрицательные значения уровней при этом будут соответствовать не усилению, а затуханию данного участка.

При нормировании величин сигналов и помех в каналах и трактах используется понятие точки нулевого относительного уровня по мощности (ТНОУ). Абсолютный уровень P_{M0}, определенный в точке нулевого относительного уровня по мощности,

обозначается как дБм0. Для перехода от абсолютного уровня сигнала P_{M0} к уровню по мощности P_M в данной точке тракта пользуются соотношением $P_M = P_{M0} + P_{Мизм}$, где $P_{Мизм}$ - измерительный уровень по мощности в данной точке тракта.

§ 8. Параметры и характеристики сигналов в сетях ЭВМ

Сигналы передачи данных в сети ЭВМ во времени меняют свои мгновенные значения, причем эти изменения могут быть предсказаны лишь с некоторой (меньше единицы) вероятностью. Таким образом, сигналы связи являются случайными процессами и их описание, естественно, должно осуществляться посредством методов, аналогичных методам описания случайных процессов.

В общем случае сигналы передачи данных в сети ЭВМ соответствуют неэргодическому и нестационарному случайному процессу, что весьма усложняет методы их описания. Поэтому принято моделировать реальные сигналы эргодическим и стационарным (в широком смысле) случайным процессом, полученным в результате двойного усреднения - вначале по множеству реализации определяются числовые характеристики для достаточно большого числа моментов времени, а затем эти характеристики усредняются по времени. Полученная таким образом модель отображает некоторый среднестатистический сигнал, параметры которого и используются при практических расчетах. При этом очевидно, что в расчетах неизбежно возникают ошибки, которые преодолеваются некоторым завышением требований к рассчитываемым устройствам с помощью машинных и натурных экспериментов и т. д.

Стационарным случайным процессом называют случайный процесс, в котором вероятности появления случайных событий в одинаковые промежутки времени равны: $P(\tau_1) = P(\tau_2)$, где $\tau_1 = \tau_2$.

Эргодическим процессом называют стационарный случайный процесс, в котором среднее значение функции по времени совпадает со средним значением функции по множеству наблюдений с вероятностью, равной единице.

Следует отметить, что постоянно проводятся работы по накоплению статистических материалов с целью совершенствования моделей сигналов. Параметры моделей приводятся в рекомендациях Международного консультативного комитета по телефонии и телеграфии (МККТТ).

Рассмотрим основные параметры сигналов как числовые характеристики моделированного случайного процесса $u(t)$. При этом усреднение будем производить во времени на интервале от $-\frac{T}{2}$ до $\frac{T}{2}$, принимая усредненное значение как пре-

дел при T, стремящемся к бесконечности. Заметим, что это справедливо лишь для модели сигналов, поскольку реализации сигналов конечны, т. е. заданы на некотором интервале времени от T_1 до T_2.

Измерения также выполняются в конечных временных интервалах, что приводит к возникновению погрешности, которая оказывается тем больше, чем меньше интервал измерений. С учетом сказанного средние параметры сигналов нормируются по-разному на интервалах 1 с, 1 мин, 1 ч.

Рассмотрим электрические параметры сигналов:

1. Постоянная составляющая. Постоянная составляющая – это среднее значение случайного процесса:

$$U_{=} = \overline{u(t)} = \lim_{T\to\infty}\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} u(t)dt .$$

Постоянная составляющая во времени неизменна, но ее величина случайна. Для многих сигналов связи постоянная составляющая равна нулю.

2. Переменная составляющая. Переменная составляющая – это центрированный случайный процесс: $u_{\sim}(t) = u(t) - u(t)$.

3. Средняя мощность. Средняя мощность - это мощность переменной составляющей (постоянная составляющая при этом не учитывается, так как не несет информации):

$$P_{ср} = \frac{\overline{u_{\approx}^2(t)}}{1Ом} = \lim_{T\to\infty}\frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} u_{\approx}^2(t)dt .$$

Средняя мощность совпадает с дисперсией случайного процесса - мерой его разброса около среднего значения. Положительное значение $u_{ЭФ} = \sqrt{\frac{P_{СР}}{1Ом}}$ называют эффективным или действующим напряжением сигнала.

4. Максимальная мощность. Максимальная мощность P_{max} – это мощность синусоидального сигнала с амплитудой U_M , которая превышается мгновенными значениями переменной составляющей сигнала $u_{\sim}(t)$ с определенной, достаточно малой вероятностью ε. Для различных видов сигналов ε принимают равной 10^{-2},10^{-3}, а иногда и 10^{-5}.

5. Минимальная мощность. Минимальная мощность P_{min} чаще всего принимается равной допустимой среднеквадратической ошибке при приеме сигналов данного вида, которая устанавливается экспериментально. В свою очередь, среднеквадратическая ошибка обычно равна средней мощности допустимой флуктуационной помехи:

$P_{min} = P_{П\ ср}$.

Иногда минимальная мощность сигнала принимается равной мощности синусоидального сигнала с амплитудой $U_{M\ min}$, кото-

рая превышается мгновенными значениями переменной составляющей $u_{\sim}(t)$ с определенной, достаточно большой вероятностью $(1 - \varepsilon)$. Обычно принимают $(1 - \varepsilon) = 0{,}98$.

Возможно использование логарифмических отношений вышеназванных величин:

$10 \lg \frac{P_{\max}}{P_{ср}} = Q_C$ - пик-фактор сигнала;

$10 \lg \frac{P_{\max}}{P_{\min}} = D_C$ - динамический диапазон сигнала;

$10 \lg \frac{P_{ср}}{P_{Пср}} = A_{ПЗС}$ - помехозащищенность сигнала.

Две последние величины используются и для характеристик трактов передачи сигналов. При этом:

$D_T = 10 \lg \frac{P_{НМ}}{P_{Пср}}$, где $P_{НМ}$ - неискаженная мощность на выходе тракта;

$A_{ЗТ} = 10 \lg \frac{P_{изм}}{P_{Пср}}$, где $P_{изм}$ - мощность измерительного сигнала на выходе.

Тогда при передаче сигналов должны выполняться следующие неравенства:

$D_T \geq D_C$, $A_{ЗТ} > A_{ПЗС}$, $P_{ср} < P_{изм}$.

Для оценки скорости изменения сигнала используют функцию автокорреляции $R(\tau) = \lim\limits_{T\to\infty} \frac{1}{T} \int\limits_{-\frac{T}{2}}^{\frac{T}{2}} u_{\approx}(t) \cdot u_{\approx}(t+\tau) dt = u_{\sim}(t)\, u_{\sim}(t + \tau)$.

Очевидно, что при $\tau = 0$, $R(0) = P_{ср}$.

Величина $r(\tau) = \frac{R(\tau)}{R(0)}$ называется коэффициентом автокорреляции. Собственно мерой скорости изменения сигнала является интервал корреляции τ_0 - время, через которое утрачивается статистическая зависимость между $u_{\sim}(t)$ и $u_{\sim}(t+\tau)$:

$$\tau_0 = \frac{2\int\limits_0^{\infty} |R(\tau)| dt}{R(0)}.$$

Посредством косинус-преобразования Фурье можно получить спектральную плотность процесса $G(f)$ по функции автокорреляции:

$G(f) = 4 \int\limits_0^{\infty} R(\tau) \cos 2\pi f \tau d\tau$ или обратно:

$R(\tau) = \int\limits_0^{\infty} G(f) \cos 2\pi f \tau d\tau$, где $G(f)$ - спектральная плотность, или мощность процесса, определенная в бесконечно малой по-

лосе *df* вблизи частоты *f*. В конечной полосе частот средняя мощность равна $P_{f_1 f_2} = \int_{f_1}^{f_2} G(f)df$.

Эффективная ширина энергетического спектра сигнала равна:

$F_Э = \frac{\int_0^\infty G(f)df}{G_{\max}(f)} = \frac{P_{ср}}{G_{\max}(f)}$, где $G_{max}(f)$ – максимальное значение спектральной плотности.

Очевидно, что $F_Э = 0{,}5\tau_0$.

Эффективную ширину энергетического спектра сигнала не следует смешивать с эффективно передаваемой полосой частот сигнала, которая устанавливается экспериментально исходя из необходимо высокого качества передачи.

Потенциальный информационный объем цифрового сигнала $V_{ЦС\ max}$ может быть найден по формуле Шеннона для определения объема сигнала:

$V_{ЦС} = -F_T \sum_{i=1}^{l} P_i \log_2 P_i$, где F_T – тактовая частота, т. е. число передаваемых отсчетов сигнала в секунду, l – число разрешенных значений отсчетов (разрешенных уровней); p_i – вероятность появления отсчета с уровнем i, если положить, что все уровни отсчетов равновероятны, т. е. $p_j = p_i = \frac{1}{l}$.

Тогда $V_{ЦС\ max} = F_T \log_2 l$.

Аналоговый сигнал согласно теореме Котельникова может быть представлен последовательностью дискретных отсчетов, следующих с частотой $F_Д = 2F_B$, причем F_B – верхняя частота эффективно передаваемого спектра сигнала. Число уровней сигнала, которые можно различить на приеме, может быть найдено как:

$l = \sqrt{1 + \frac{P_{ср}}{P_{Пср}}} = \sqrt{1 + 10^{0{,}1A}_{ЗС}}$.

Тогда

$V_{ЗС\ max} = F_B \log_2(1 + 10^{0{,}1A}_{ЗС})$.

Часть II. Распределенные автоматизированные системы

Глава 1. Модель «клиент-сервер»

Согласно определению, в распределенной системе процессы распределены по разным ЭВМ сети. Распределенная обработка информации подразумевает, что по разным ЭВМ сети распределяются не просто любые не связанные друг с другом процессы, а процессы одной прикладной программы, то есть процессы, совместно выполняющие одну общую задачу. Для лучшего понимания механизмов распределенной обработки информации путем её декомпозиции на отдельные легко обозримые компоненты, существует модель «клиент-сервер». В базовой модели «клиент-сервер» все процессы в распределенных системах делятся на две возможно перекрывающиеся группы:

1. Сервер. Сервер – это процесс, реализующий некоторую сетевую службу. Существует второе определение сервера, как ЭВМ, осуществляющая управление доступом к ресурсам сети. Эти два определения не противоречат друг другу, а первое определение является даже более общим. С точки зрения первого определения, второе определение можно перефразировать, как ЭВМ, на которой работает процесс, реализующий сетевую службу предоставления доступа к ресурсам сети;

2. Клиент. Клиент – это процесс, запрашивающий службы у серверов. Взаимодействие клиента и сервера осуществляется в режиме «запрос-ответ».

Прикладные программы типа «клиент-сервер» принято делить на три логических уровня:

1. Уровень пользовательского интерфейса – обычно реализуется на рабочих станциях (клиентских ЭВМ). Этот уровень содержит средства взаимодействия (интерфейса) пользователя и прикладной программы.

2. Уровень обработки – реализует основную функциональную часть прикладной программы, может располагаться как на рабочей станции, так и на сервере.

3. Уровень данных – содержит программы обеспечения доступа к данным и их сохранности. Обычно уровень данных реализуется на серверах.

Прикладная программа может физически делиться на две, три или более частей и выполняться, соответственно, на двух (физически двухзвенная архитектура), трех (физически трехзвенная архитектура) или более (многопоточная технология) ЭВМ одновременно. Физическое разделение прикладной программы по двум ЭВМ (физически двухзвенная архитектура) – рабо-

чей станции и серверной ЭВМ, может производиться в пяти вариантах.

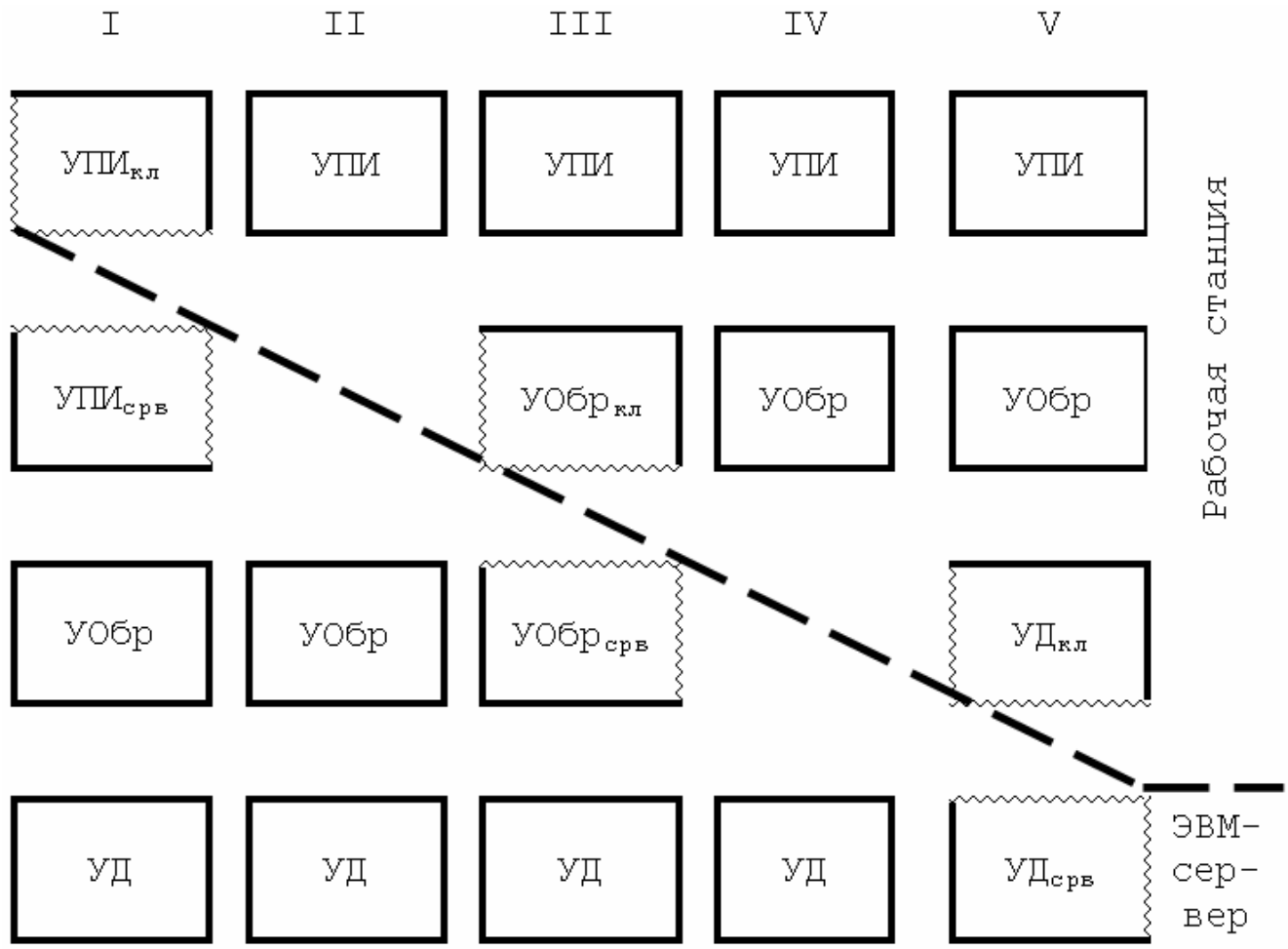


Рис. 4. Варианты физически двухзвенной архитектуры.
УПИ – уровень пользовательского интерфейса
УОбр – уровень обработки
УД – уровень данных
кл – клиентская часть соответствующего уровня
срв – серверная часть соответствующего уровня

В физически трехзвенной архитектуре прикладная программа физически разделяется на три части по уровням. При этом уровень пользовательского интерфейса реализуется на рабочих станциях, уровень обработки реализуется на ЭВМ, называемой сервер приложений, а уровень данных реализуется на ЭВМ, называемой сервер баз данных. В этой архитектуре уровень обработки выступает в качестве сервера по отношению к уровню пользовательского интерфейса и, одновременно, в качестве клиента по отношению к уровню данных. Именно по этой причине говорят, что все процессы в распределенных системах делятся на две возможно перекрывающиеся группы – клиент и сервер, то есть один и тот же процесс по отношению к другим процессам может быть одновременно и клиентом, и сервером.

Практика показывает, что простое физическое разделение прикладной программы на клиентскую и серверную части бывает недостаточно для повышения производительности распределенных вычислений. Решение заключается в более тонком дроблении прикладной программы в форме нескольких потоков выполнения (многопоточная технология). Поток выполнения – это комплекс информационно-независимых процессов одной прикладной программы, способных работать параллельно без взаимной блокировки.

На локальной ЭВМ преимущества многопоточных технологий проявляются только в многопроцессорных системах с общей (разделяемой) памятью. При этом и операционная система ЭВМ также должна поддерживать многопоточные технологии (напр.: ОС UNIX). Распределенная система, представляясь пользователю как виртуальная локальная многопроцессорная ЭВМ, является идеальной платформой для реализации многопоточной технологии. Важным свойством потоков выполнения является отсутствие блокировки других процессов при блокировке одного из них.

В соответствии с рассматриваемой моделью «клиент-сервер», в многопоточной технологии также существуют клиенты и серверы, причем в распределенных системах и те, и другие могут быть многопоточными. Многопоточные клиенты, как правило, реализуют пользовательский интерфейс в виде составного документа. Составной документ – это набор интерфейсных средств различных типов, предоставляемых различными прикладными программами, которые интегрируются в единый пользовательский интерфейс. В качестве примера можно привести судовую автоматизированную систему управления, где на устройстве отображения информации капитана в едином стиле, в одном окне, на адекватном картографическом фоне отображаются результаты решения различных задач (счисления пути, обсерваций, радиолокационной обстановки и т.д.), причем вся отображаемая информация эргономически оптимизирована для минимизации ошибок её восприятия. Пользовательский интерфейс, в котором представляется составной документ, скрывает тот факт, что с разными частями документа работают различные прикладные программы.

Многопоточные серверы, как правило, выгоднее всего реализовывать в виде серверов объектов. В отличие от других серверов, сервер объектов не предоставляет конкретной службы. Конкретные службы реализуются объектами, размещенными на сервере. Сервер предоставляет только средства обращения к локальным объектам по запросу удаленных клиентов. Таким образом можно легко изменять набор служб, просто добавляя или удаляя объекты. Правила обращения к объекту называются политикой активизации объекта. При этом объект перед обра-

щением к нему должен быть перемещен в оперативную память сервера, т.е. активизирован. Механизм группирования объектов в соответствии с политикой активизации каждого из них называется адаптер объектов или упаковщик объектов.

Адаптер объектов контролирует один или несколько объектов. Поскольку сервер должен поддерживать объекты с различной политикой активизации, на одном сервере могут работать несколько адаптеров объектов.

Глава 2. Организация связи между процессами

Обмен сообщениями между процессами является основой межпроцессного взаимодействия в распределенной системе. В то же время стандартные средства связи между процессами локальных и сетевых операционных систем не скрывают взаимодействия процессов, то есть не обеспечивают прозрачность доступа.

В настоящее время существует четыре способа связи между процессами, работающими на разных ЭВМ:

1. Удаленный вызов процедур;
2. Обращение к удаленным объектам;
3. Связь посредством сообщений;
4. Связь на основе потоков данных.

Рассмотрим последовательно каждый из этих видов связи между процессами.

§ 1. Удаленный вызов процедур

В любом языке программирования существует понятие процедуры. Процедура – это стандартно оформленный программный модуль, доступный для использования другими программами с помощью стандартных операций вызова процедур. Стандартные операции вызова процедур предусматривают передачу в процедуру исходных параметров для ее работы и возврат в вызывающую программу результатов работы процедуры. В локальной ЭВМ передача в процедуру исходных параметров для ее работы и возврат в вызывающую программу результатов работы процедуры производится через стек. Стек - специализированная область оперативной памяти ЭВМ, доступ к которой производится не по адресу (номеру) ячейки, а по очередности поступления в стек информации в соответствии с принципом FILO (First In, Last Out - первый пришел, последний вышел). В локальной ЭВМ передача параметров между вызывающей программой и процедурой происходит тремя способами:

1. Передача по значению – применяется при обмене отдельными переменными, когда в стек помещаются их непосредственные значения;

2. Передача по ссылке – применяется при обмене массивами данных, когда в стек помещаются начальные адреса (ссылки) массивов данных;

3. Передача через копирование-восстановление - применяется при необходимости модификации процедурой передаваемых ей переменных и возврата в вызывающую программу их новых значений. При вызове процедуры производится копирование переменной в стек, а при возврате в вызывающую программу – считывание нового значения этой переменной из стека и замена старого значения переменной в оперативной памяти на новое.

Удаленный вызов процедур подразумевает выполнение процедуры на другой ЭВМ сети, что делает невозможным передачу данных через стек.

Проблемы при выполнении удаленных процедур возникают вследствие следующих факторов:

1. Вызывающая программа и удаленная процедура размещаются на разных ЭВМ и выполняются в различных адресных пространствах;

2. Существует разница в представлении данных при размещении вызывающей программы и удаленной процедуры на ЭВМ с различными форматами данных;

3. Возможно возникновение сбоев на обеих ЭВМ в процессе выполнения удаленной процедуры.

Удаленный вызов процедур осуществляется с помощью технологии RPC (Remote Procedure Call - удаленный вызов процедур). Идея RPC состоит в том, чтобы с точки зрения пользователя (программиста) удаленный вызов процедур выглядел точно так же, как локальный. Это означает, что ни программист, ни вызывающая программа не должны уведомляться о том, что вызываемая процедура выполняется на другой ЭВМ, и наоборот, вызываемая процедура не должна уведомляться о том, что она вызывается программой, физически размещенной на другой ЭВМ сети. Иными словами, RPC призвана обеспечить прозрачность местоположения. RPC организует прозрачность местоположения с помощью механизмов клиентской и серверной заглушек. Клиентская заглушка – специальная версия функции вызова процедуры, работающая на стороне клиента и подменяющая аналогичную стандартную локальную функцию в случае размещения нужной процедуры на другой ЭВМ сети. Клиентская заглушка вызывается стандартной операцией вызова процедуры, но в отличие от оригинальной локальной функции вызова процедуры, она не помещает данные в стек, а упаковывает их в сообщение и требует у операционной системы переслать это сообщение на сервер.

При поступлении сообщения на сервер операционная система сервера передает его серверной заглушке. Серверная за-

глушка – специальная версия функции вызова процедуры, работающая на стороне сервера и подменяющая аналогичную стандартную локальную функцию в случае поступления по сети запроса на вызов удаленной процедуры, физически размещенной на данном сервере. Серверная заглушка преобразует приходящий по сети запрос в стандартный вызов локальной процедуры с передачей параметров через стек.

Таким образом, ни вызывающая программа, ни вызываемая процедура не уведомляются о том, что они физически находятся на различных ЭВМ сети. Удаленный вызов процедур происходит в десять этапов:

1. Вызывающая программа клиента стандартным образом вызывает клиентскую заглушку;
2. Клиентская заглушка создает сообщение и вызывает операционную систему рабочей станции;
3. Операционная система рабочей станции пересылает сообщение на сервер;
4. Операционная система сервера передает сообщение серверной заглушке;
5. Серверная заглушка извлекает из сообщения параметры и стандартным образом выполняет вызов процедуры;
6. После окончания своей работы процедура возвращает результаты серверной заглушке;
7. Серверная заглушка создает сообщение и вызывает операционную систему сервера;
8. Операционная система сервера пересылает сообщение на рабочую станцию;
9. Операционная система рабочей станции передает сообщение клиентской заглушке;
10. Клиентская заглушка извлекает из сообщения параметры и стандартным образом возвращает их вызывающей программе.

Базовая модель RPC предполагает, что вызывающий и вызываемый процессы связываются друг с другом для обмена сообщениями по сети ЭВМ. Однако существует вариант, когда клиент и сервер работают на одной машине. В стандартном случае в этом варианте применяются средства локального межпроцессного взаимодействия, имеющиеся во всех локальных операционных системах, поддерживающих многопрограммный режим.

Ряд локальных операционных систем предоставляют процессам, размещенным на одной ЭВМ, эквивалент RPC под названием “doors”.

В стандартном варианте вызова клиентом удаленной процедуры его работа приостанавливается до получения ответа. Такая приостановка работы клиентского процесса до получения ответа от удаленной процедуры называется блокировкой про-

цесса, а вызов процедуры, соответственно, синхронным вызовом процедур.

В ряде случаев для продолжения работы клиентского процесса ответ не нужен (например, добавление записи в базу данных, запуск удаленной службы и т.д.). Технология RPC предоставляет средства асинхронного вызова процедур, когда клиент получает возможность продолжить свою работу сразу после выполнения запроса на удаленный вызов процедуры. Для этого сервер немедленно по приходу запроса отсылает клиенту квитанцию о приеме запроса и только после этого вызывает запрашиваемую процедуру.

В случае, когда ответ необходим, но его отсутствие не влияет на продолжение работы клиентского процесса (например, циклические процессы в АСУ), организуются два асинхронных вызова: один – запрос со стороны клиента, а другой – ответ со стороны сервера. Комбинация из двух асинхронных вызовов называется отложенным синхронным вызовом процедур.

Дальнейшим развитием технологии RPC является система DCE RPC (Distributed Computing Environment – среда распределенных вычислений), адаптированная к программному обеспечению фирмы Microsoft. В настоящее время существуют версии DCE RPC для всех распространенных операционных систем.

§ 2. Обращение к удаленным объектам

С появлением объектно-ориентированного программирования стало очевидно, что принципы технологии RPC могут быть равно применимы и к программным объектам.

Программный объект – это стандартно оформленный программный модуль, содержащий данные и операции над этими данными. Данные, содержащиеся в объекте, в программировании называются состоянием объекта, а операции над этими данными – методами объекта. Доступ к методам можно получить через интерфейс объекта, предоставляемый системами программирования. Объект может реализовывать множество интерфейсов. В свою очередь, для описания интерфейса также может существовать несколько объектов.

Объекты в распределенных системах существуют в формах, принятых в том или ином языке программирования. Как правило, при выполнении распределенных вычислений интерфейс объекта находится на другой ЭВМ, чем сам объект. Такой объект называется распределенным объектом. При этом состояние объекта (т.е. содержащиеся в объекте данные) никогда не распределяется, а локализовано на одной ЭВМ. С других ЭВМ доступны только интерфейсы, реализованные в объекте.

При обращении клиента к распределенному объекту управление передается программе реализации интерфейса объекта,

аналогичной клиентской заглушке и называемой заместителем. Заместитель осуществляет транзит параметров объекта от вызывающей программы к операционной системе рабочей станции и обратно, так же как это делает клиентская заглушка.

На стороне сервера транзит параметров от операционной системы сервера к методам объекта и обратно осуществляет аналог серверной заглушки, называемый программа-скелетон.

В зависимости от степени связи со своим сервером объекты в распределенных системах могут быть сохранными (или резидентными) и нерезидентными. Сохранный объект продолжает существовать вне зависимости от состояния своего сервера. Сервер, управляющий таким объектом, может сохранить состояние объекта в оперативной памяти и завершить свою работу. Позже вновь запущенный сервер считывает состояние объекта из памяти и передает ему управление. Нерезидентный объект существует, только пока сервер им управляет. При завершении работы сервера объект выгружается из оперативной памяти.

Для привязки клиента к объекту, последние предоставляют ссылки, уникальные в пределах распределенной системы. Ссылка на объект должна содержать:

1. Сетевой адрес ЭВМ, на которой физически размещен программный объект;
2. Адрес сервера, который управляет этим объектом;
3. Указание на конкретный объект.

При возникновении сбоя на серверной ЭВМ и последующей её перезагрузке серверный процесс может загрузиться в другое место оперативной памяти. При этом все ссылки на объекты станут неправильными. Для решения данной проблемы необходимо создать сервер локализации, постоянно следящий за работой серверов, на которых размещены объекты. Сервер локализации требует указания в ссылке на объект не адреса, а уникального идентификатора объекта.

После того, как клиент свяжется с объектом, он может через заместителя обратиться к методам объекта. Такое удаленное обращение к методам поддерживается технологией RMI (Remote Method Invocation – удаленное обращение к методам). Технология RMI поддерживает два способа удаленного обращения к методам:

1. Статическое обращение, когда интерфейсы объекта при разработке клиентской прикладной программы заранее известны;
2. Динамическое обращение, когда сами методы и параметры обращения к ним выбираются в процессе выполнения прикладной программы.

Обращение к удаленным объектам может производиться как с использованием непосредственно самой технологии RMI, так и с использованием системы DCE RPC.

При обращении RMI клиент использует ссылку, содержащую следующие параметры:

1. Сетевой адрес сервера;
2. Полный путь к объекту на сервере;
3. Локальный идентификатор объекта в адресном пространстве сервера;
4. Указание на стек протоколов, используемых для взаимодействия клиента и сервера.

При обращении DCE RPC клиент передает серверу следующие данные:

1. Идентификатор объекта;
2. Идентификатор интерфейса, содержащего метод;
3. Идентификатор самого метода;
4. Параметры.

Сервер поддерживает таблицу объектов, с помощью которой по полученной информации идентификаторов объекта и интерфейса, идентифицирует объект, к которому обратился клиент. Затем сервер выбирает запрошенный метод и передает ему параметры. При получении сервером запроса с обращением к объекту, выполняются следующие действия:

1. Операционная система сервера передает запрос соответствующему адаптеру объектов;
2. Адаптер объекта извлекает из запроса ссылку на объект и передает запрос скелетону соответствующего объекта в соответствии с его политикой активизации.
3. Скелетон выполняет транзит параметров объекта и обращается к соответствующему методу объекта.

§ 3. Связь посредством сообщений

Связь посредством сообщений применяется при необходимости передачи данных без их обработки на сервере (напр.: электронная почта). При осуществлении распределенной обработки информации, связь посредством сообщений является основным способом связи между процессами, размещенными на различных ЭВМ. Классификация связи посредством сообщений:

1. По виду используемой коммуникации:

а) Сохранная (резидентная) связь посредством сообщений. При этом виде связи используется коммуникация, ориентированная на установление соединения. Отправитель не контролирует состояние процесса-получателя в момент отправки сообщения. В то же время, сообщение не будет потеряно и будет доставлено процессу-получателю сразу, как только он будет готов его принять;

б) Нерезидентная связь посредством сообщений. При этом виде связи используется коммуникация, не ориентированная на установление соединения. Сообщение существует в системе

только в момент его передачи процессом-отправителем. Если по какой-либо причине процесс-получатель не имеет возможности принять сообщение, оно теряется;

2. По виду блокировки процесса-отправителя:

а) Синхронная связь посредством сообщений. При этом виде связи процесс-отправитель блокируется до получения ответа от процесса-получателя о приеме сообщения;

б) Асинхронная связь посредством сообщений. При этом виде связи процесс-отправитель продолжает свою работу, не дожидаясь квитанции от процесса-получателя.

В системах нерезидентной связи посредством сообщений применяется стандарт пересылки сообщений MPI (Message Passing Interface – связь посредством сообщений), основанный на сокетах Беркли. Сокет Беркли – это абстрактная конечная точка коммуникации, в которую прикладная программа записывает данные, необходимые для передачи по сети, и из которой она может считывать поступающую из сети информацию. Сокеты Беркли были впервые реализованы в версии операционной системы UNIX, разработанной в университете Беркли. Операционная система предоставляет прикладной программе набор команд, называемых примитивами, с помощью которых прикладная программа может создать сокет, назначить ему локальный адрес, переслать или принять данные и разорвать соединение.

В системах сохранной связи посредством сообщений прикладная программа помещает отправляемое сообщение в локальную исходящую очередь, находящуюся на той же ЭВМ. Сообщение, помещенное в очередь, содержит описание очереди назначения, в которую оно должно быть перемещено. Системный процесс-отправитель помещает сообщение в очередь на коммуникационном сервере (маршрутизаторе), и оно перемещается последовательно по цепочке коммуникационных серверов до места назначения. Сообщение будет храниться в очереди последнего в цепочке маршрутизатора до тех пор, пока процесс-получатель не будет готов его принять. Процесс-отправитель в состоянии гарантировать только попадание сообщения во входящую очередь процесса-получателя. Будет ли это сообщение прочитано, определяется целиком поведением процесса-получателя. Очереди управляются программами, называемыми менеджеры очередей. Они взаимодействуют непосредственно с отправляющими и принимающими сообщения прикладными программами. Важнейшей областью применения очередей сообщений является интеграция существующих и новых прикладных программ в единые согласованные распределенные информационные системы. Интеграция требует, чтобы отправляемые сообщения имели единый формат внутри одной системы. Функции представительного уровня модели OSI в распределенной информационной системе выполняет специальная программа, называемая брокер со-

общений. Брокер сообщений работает как шлюз прикладного уровня в системе очередей сообщений, осуществляя преобразование входящих сообщений в формат целевой прикладной программы. Основой брокера сообщений является база данных с правилами перекодировки сообщений из одного формата в другой.

§ 4. Связь на основе потоков данных

Рассмотренные ранее способы связи между процессами, работающими на разных ЭВМ, касались обмена более или менее независимыми, законченными порциями информации, не привязанными к реальному времени (параметры процедур и объектов, сообщения). Однако в прикладных программах реального масштаба времени (аудио и видео) временные характеристики имеют решающее значение (аудио- и видеопотоки).

Для обмена критичной ко времени передачи информацией распределенные системы предоставляют поддержку потоков данных. Классификация потоков данных:

1. По физическому содержанию потока:

а) Дискретный поток данных – поток байт;

б) Непрерывный поток данных – поток бит;

2. По информационному содержанию потока:

а) Простой поток данных – содержит только одну последовательность данных;

б) Комплексный поток данных – содержит несколько связанных между собой простых потоков, называемых вложенными потоками данных (напр.: видео со стереозвуком).

Различают три режима передачи потоков данных:

1. Асинхронный режим передачи. В асинхронном режиме передачи временные ограничения на передачу потока данных не накладываются;

2. Синхронный режим передачи. В синхронном режиме передачи для каждого элемента потока данных определяется максимально возможная задержка передачи. Этот режим используется там, где необходимо передавать данные не реже какого-либо промежутка времени – например, при измерении параметров;

3. Изохронный режим передачи. В изохронном режиме передачи для каждого элемента данных определяется как максимально возможная, так и минимально возможная задержка передачи. Этот режим используется там, где недопустимо ни замедление ни ускорение передачи данных – например, при передаче аудио- и видеопотоков. Интервал времени между минимально возможной и максимально возможной задержками передачи называется интервалом дрожания.

Временные зависимости потоков данных выражаются в виде требований к качеству обслуживания, описывающих, что должна сделать распределенная система для того, чтобы гарантировать сохранение в потоке данных заданных временных соотношений. Требования к качеству обслуживания закрепляются в документе, называемом спецификация передачи. Спецификация передачи входит в пакет протоколов транспортного уровня модели OSI.

Для передачи потока данных распределенная система должна захватить ресурсы, удовлетворяющие требованиям к качеству обслуживания. Обычно такими ресурсами являются:

1. Пропускная способность каналов связи;
2. Буферная память маршрутизаторов;
3. Вычислительная мощность узлов обработки данных.

В мультимедийных системах важное значение имеет взаимная синхронизация вложенных потоков данных, т.е. поддержание между ними строгих временных соотношений. Различают два типа синхронизации вложенных потоков данных:

1. Простейшая синхронизация. Этот вид синхронизации осуществляется между дискретным и непрерывным потоками данных (напр.: показ слайдов со звуковым сопровождением);

2. Синхронизация артикуляции. Синхронизация артикуляции осуществляется между непрерывными потоками данных. Этот термин пришел в вычислительную технику из кинематографа, где означал синхронизацию движений губ актера с произносимыми им звуками. Примером синхронизации артикуляции может быть звуковое сопровождение любого видеофильма. Для того, чтобы визуально губы актера двигались в такт произносимым им словам, рассинхронизация артикуляции не должна превышать 20 мс.

Наиболее просто синхронизация вложенных потоков данных достигается путем их взаимного мультиплексирования (напр.: стандарт MPEG – Motion Picture Expert Group).

Глава 2. Миграция процессов

При наличии в распределенной системе бóльшего количества процессоров, чем необходимо для выполнения прикладной программы типа «клиент-сервер», возникает соблазн выровнять загрузку имеющихся процессоров за счет перемещения (миграции) работающих процессов между ЭВМ сети.

§ 1. Перенос кода (перенос процессов)

Перенос кода (перенос процессов) с одной ЭВМ на другую в распределенных системах производится с целью повышения производительности системы. Перенос производится с сильно

загруженной машины на слабо загруженную. Загрузка ЭВМ обычно определяется длиной очереди исполняемых программ к процессору. Перенос кода позволяет динамически конфигурировать модель «клиент-сервер», оперативно перемещая отдельные части прикладной программы, разбитой по уровням, с одной ЭВМ на другую.

Модель переноса кода состоит из трех компонент:

1. Сегмент кода. Сегмент кода содержит набор исполняемых инструкций переносимого программного модуля;

2. Сегмент ресурсов. Сегмент ресурсов содержит ссылки на внешние ресурсы, необходимые для работы переносимого программного модуля;

3. Сегмент исполнения. Сегмент исполнения содержит текущее слово состояния переносимой программы и стек промежуточных результатов вычислений.

Модель переноса кода существует в двух вариантах:

1. Модель слабой мобильности. В этом варианте допускается перенос только сегмента кода, переносимая программа запускается всегда из своего начального состояния;

2. Модель сильной мобильности. В этом варианте переносится сегмент кода и сегмент исполнения. Работающий процесс может быть приостановлен, перенесен на другую ЭВМ и его выполнение продолжено с места останова.

Перенос работающего процесса называется миграцией процесса. Помимо миграции процесса сильная мобильность может осуществляться за счет удаленного клонирования процесса. Клонирование создает точную копию процесса, которая выполняется на удаленной ЭВМ параллельно оригиналу.

Перенос кода может быть инициирован:

1. Отправителем, т.е. ЭВМ, на которой переносимый программный модуль постоянно размещен или выполняется;

2. Получателем, когда инициатива в переносе кода принадлежит ЭВМ-получателю.

С точки зрения переноса сегмента ресурсов различают три вида ресурсов:

1. Неприсоединенные ресурсы. Эти ресурсы могут быть с легкостью перенесены с одной ЭВМ на другую (напр.: файлы данных);

2. Связанные ресурсы. Данный вид ресурсов переносится с одной ЭВМ на другую с относительно большими затратами (напр.: базы данных и web-сайты, имеющие мощный ссылочный аппарат);

3. Фиксированные устройства. К фиксированным устройствам относятся локальные ресурсы, программный перенос которых с одной ЭВМ на другую не возможен (напр.: устройства печати, сканеры и другая аппаратура).

При переносе кода различают три типа привязки процесса к ресурсам:

1. Привязка по идентификатору. Данный вид привязки применяется, когда процесс требует в точности тот ресурс, на который ссылается (напр.: любимый web-сайт пользователя).

2. Привязка по значению. Такая привязка используется в случае, если процессу необходим только результат работы ресурса, а источник этого значения ему безразличен (напр.: стандартные библиотеки языков программирования – процессу все равно, кто ему посчитает результат какой-либо стандартной функции, ему важно лишь её значение).

3. Привязка по типу. Эта привязка применяется, когда процессу необходимо использовать ресурс определенного типа. Как правило, тип определяется для аппаратных ресурсов системы (напр.: устройство печати, устройство наглядного отображения и т.д.).

При переносе кода в гетерогенных системах, особенно, если различные ЭВМ системы построены на различных аппаратных платформах, требуется, чтобы сегмент кода выполнялся на всех этих платформах без перекомпиляции текста программы. В случае слабой мобильности в гетерогенных системах создаются различные варианты сегмента кода – по одному на каждую используемую аппаратную платформу. В случае сильной мобильности основная проблема заключается в переносе сегмента исполнения, вследствие различий в структуре слова состояния программы процессоров разных типов. Сильная мобильность в гетерогенной системе возможна только при использовании ЭВМ с одинаковыми аппаратными платформами.

§ 2. Программные агенты

Развитие средств искусственного интеллекта и их интеграция в распределенные системы обусловило появление нового класса процессов – программных агентов. Программный агент – это независимый процесс, обладающий признаками искусственного интеллекта, работающий автономно или совместно с другими агентами, способный своевременно реагировать на изменение в своем окружении и инициировать действия, влияющие на свое окружение.

Различают четыре основных вида программных агентов:

1. Кооперативный агент – это агент, являющийся частью мультиагентной системы, т.е. системы, в которой агенты, работая совместно, выполняют общие задачи;

2. Мобильный агент – это агент, способный перемещаться с одной ЭВМ на другую;

3. Интерфейсный агент – это агент, помогающий пользователям работать с одной или несколькими прикладными программами;

4. Информационный агент – это агент, управляющий информацией из множества различных источников.

Чаще всего агенты совмещают в себе несколько вышеперечисленных видов – напр.: кооперативный мобильный информационный агент – это программный агент, управляющий информацией из множества различных источников, способный перемещаться с одной ЭВМ на другую и являющийся частью мультиагентной системы.

Для того, чтобы лучше представить себе работу программных агентов в распределенной системе, существует обобщенная модель программных агентов. Эта модель включает в себя четыре уровня:

1. Программа-агент;

2. Компонент управления агентами. Данный компонент отслеживает агентов на конкретной ЭВМ, а также предоставляет механизм создания и уничтожения агентов;

3. Служба каталогов. При помощи службы каталогов агент может узнать о существовании других агентов;

4. Канал связи между агентами.

Программная реализация четырехуровневой обобщенной модели программных агентов называется платформой агента.

Связь между агентами происходит посредством коммуникационного протокола прикладного уровня, называющимся ACL (Agent Communication Language – язык взаимодействия агентов). Сообщение ACL состоит из заголовка и реального содержания. Заголовок сообщения ACL содержит следующие поля:

1. Поле цели сообщения;

2. Поле адреса отправителя;

3. Поле адреса получателя;

4. Поле онтологии (интерпретации содержания).

В ACL имеется жесткое разделение между целью сообщения и его содержанием. Идея ACL состоит в том, чтобы агент-отправитель и агент-получатель одинаково понимали цель сообщения. По этой причине количество целей в ACL ограничено. Цель сообщения определяет реакцию получателя.

ACL никак не задает формат или язык содержания сообщения. Формат или язык содержания сообщения определяют вступившие в связь агенты.

Мобильный код представляет собой серьезную угрозу безопасности информации в распределенной системе. Большинство компьютерных вирусов являются по своей сути мобильными агентами. Наиболее эффективной защитой от вредоносных последствий проникновения в систему мобильных агентов является недопущение их проникновения в систему извне с помощью

брандмауэра. В то же время возникают ситуации, когда надо пропустить в систему мобильного агента извне. При этом необходим тотальный контроль за абсолютно всеми действиями агента в распределенной системе. Имеются два механизма контроля за поведением агента:

1. Механизм сита. Механизм сита предусматривает покомандный контроль работы агента. При попытке выполнить запрещенную команду, обратиться к запрещенным регистрам или областям памяти, получить доступ к запрещенным ресурсам, выполнение программы агента прекращается. Механизм сита реализуется программой, называемой менеджер защиты, которая работает как стандартный интерпретатор, имеющий в качестве входной информации программный код мобильного агента. Стандартный менеджер защиты реализует очень жесткие правила защиты, независимо от того, откуда поступил мобильный код;

2. Механизм полигона. Механизм полигона обеспечивает бóльшую гибкость в реализации контроля за поведением агента за счет выделения специализированной ЭВМ для выполнения поступившего мобильного кода. Все ресурсы распределенной системы при этом физически отделены от ЭВМ полигона. Пользователи распределенной системы могут получить доступ к мобильному агенту стандартным способом (ACL, RPC и т.д.). Мобильный агент, в свою очередь, никуда переместиться с ЭВМ полигона не может. В механизме полигона используются три метода соблюдения правил защиты:

а) Метод подписания кода. Этот метод заключается в аутентификации программ вне зависимости от того, откуда они были получены. Мобильный код представляется как сообщение и аутентифицируется стандартным образом;

б) Метод расширенного анализа стека. Данный метод заключается в проверке прав доступа к каждому элементу стека и построении цепочки вызовов с целью предотвращения доступа к защищенному ресурсу через другие программные модули;

в) Метод управления пространством имен. Метод заключается в создании пространства разрешенных имен, не связанного с общим пространством имен распределенной системы. Связь разрешенного имени с именем реального ресурса системы осуществляется только после проверки соблюдения правил защиты.

При перемещении мобильных агентов с одной ЭВМ на другую возможны атаки на самих агентов. Полная защита агента от всех типов атак невозможна, так как нельзя спрогнозировать поведение всех процессов сети, способных оказывать влияние на мобильного агента. По этой причине для защиты мобильного агента применяется только контроль целостности его программного кода при попадании его на каждую ЭВМ сети. Для контроля целостности программного кода мобильного агента, создается дайджест кода агента, который перемещается

вместе с агентом и служит для других ЭВМ сети средством удостовериться в подлинности и целостности мобильного агента. На ЭВМ владельца агента по его возвращению также производится проверка с помощью дайджеста, был ли модифицирован код программного агента при его перемещениях между ЭВМ системы.

Кроме защиты непосредственно самого агента, необходимо защищать информацию, которую собирает этот мобильный агент. Для защиты информации, собираемой мобильным агентом, используется метод защищенных журналов. При отправке мобильного агента, его владелец помещает в пустой защищенный журнал контрольную сумму, значение которой равно открытому ключу владельца агента. Когда агент перемещается на сервер, который хочет передать агенту какие-либо данные, сервер добавляет эти данные в журнал, защищает их своей цифровой подписью и вычисляет контрольную сумму всего журнала, зашифровав её открытым ключом владельца агента. Когда агент возвращается к своему владельцу, владелец считывает данные с конца журнала, проверяя подлинность записавшего их сервера и их целостность по цифровой подписи сервера, целостность всего журнала по контрольной сумме и вычисляет предыдущую контрольную сумму. При несовпадении контрольных сумм или цифровых подписей серверов журнал считается сфальсифицированным.

Глава 3. Именование в распределенных системах

Любой человек, хотя бы раз имевший дело с персональным компьютером, должен был заметить, что любой файл, любая папка, и даже любой ресурс ЭВМ имеет свое уникальное название. В этой главе будет рассмотрено, каким образом эти названия организуются в единую систему, позволяющую находить всю необходимую пользователю информацию.

§ 1. Понятие сущности

В распределенных системах все, к чему можно получить программный доступ, называется сущностью. Любые процессы, аппаратные или информационные ресурсы, сетевые соединения, интерфейсные средства и т.д. и т.п. можно назвать сущностью, если к ним имеется программный доступ.

Каждая сущность имеет имя. Имя может быть адресом или идентификатором. Имена в распределенных системах могут быть:

1. Глобальными. Глобальное имя обозначает одну и ту же сущность вне зависимости от того, где в системе это имя используется;

2. Локальными. Интерпретация локального имени зависит от того, где в системе это имя используется.

Адрес является локально зависимым именем и указывает на конкретное размещение сущности, также как адрес места жительства гражданина указывает на конкретное его размещение на планете. Для доступа к сущности используется точка доступа, которая, в свою очередь, также является сущностью. Имя точки доступа является адресом сущности. Можно сказать, что адрес – это специальный тип имени, указывающий на точку доступа к сущности. Сущность может иметь более чем одну точку доступа. Сущность может поменять точку доступа, а точка доступа может быть перенацелена на другую сущность. Однако, применение адреса в распределенной системе неудобно, так как указание на локальное местоположение сущности не обеспечивает прозрачности переноса и смены местоположения.

В распределенных системах удобнее пользоваться локально независимым именем – идентификатором. Идентификатор обладает следующими свойствами:

1. Идентификатор ссылается не более чем на одну сущность;

2. На каждую сущность ссылается не более одного идентификатора;

3. Идентификатор всегда ссылается на одну и ту же сущность, то есть не может быть использован повторно.

Чтобы лучше понять указанные свойства, проведем ассоциацию: сущность – это гражданин, а идентификатор – его водительское удостоверение. В водительское удостоверение вписываются сведения только об одном гражданине, то есть идентификатор ссылается не более чем на одну сущность. Гражданин может иметь не более одного водительского удостоверения, два удостоверения – это уже преступление. Таким образом, на каждую сущность ссылается не более одного идентификатора. При замене водительского удостоверения старое уничтожается, а новое имеет совсем другой номер и реквизиты, то есть идентификатор всегда ссылается на одну и ту же сущность и не может быть использован повторно.

§ 2. Пространство имен

Имена в распределенных системах организуются в некоторую сущность, называемую пространство имен. Пространство имен представляется как граф с двумя типами узлов:

1. Листовой узел. Листовой узел представлен именованной сущностью и не имеет исходящих из него ребер, обычно содержит информацию о представляемой сущности.

2. Направляющий узел. В отличие от листового, этот тип узла имеет несколько исходящих из него именованных ребер. Направляющий узел хранит таблицу идентификаторов ребер, называемую направляющей таблицей.

Пространство имен становится доступным пользователям с помощью службы именования, реализуемой на серверах имен. Пространство имен, как правило, организуется иерархически. При этом оно разбивается на три логических уровня:

1. Глобальный уровень. Этот уровень является самым верхним и формируется узлами, направляющие таблицы которых изменяются крайне редко;

2. Административный уровень. Узлы административного уровня относительно стабильны, хотя изменяются чаще, чем узлы глобального уровня;

3. Управленческий уровень. Этот уровень является самым нижним и формируется из регулярно изменяемых узлов. В отличие от глобального и административного уровней, узлы управленческого уровня обслуживаются не только системными администраторами, но и пользователями.

Распределенность пространства имен по множеству серверов имен затрудняет процесс поиска информации, называемый разрешение имени. Каждый клиент имеет доступ к локальной процедуре разрешения имен, реализующей процесс разрешения. Имеется два способа разрешения имен:

1. Итеративное разрешение имени. В процессе разрешения процедура разрешения имен связывается последовательно с каждым сервером имен, начиная с корневого сервера глобального уровня, до тех пор, пока не найдет конечную сущность. Серверы имен каждый раз возвращают промежуточные результаты поиска процедуре разрешения имен;

2. Рекурсивное разрешение имени. В процессе разрешения процедура разрешения имен связывается с корневым сервером глобального уровня, который не возвращает процедуре промежуточные результаты поиска, а передает их следующему обнаруженному серверу имен, и далее каждый последующий сервер передает промежуточные результаты поиска очередному обнаруженному серверу имен, пока не будет найдена конечная сущность.

Недостаток рекурсивного разрешения имен заключается в требовании повышенной производительности каждого из серверов имен. По этой причине серверы имен глобального уровня поддерживают только итеративное разрешение имен.

В случае мобильных сущностей итеративное и рекурсивное разрешение имен не применимо, так как имена в этих способах разрешения привязаны к постоянному местоположению. Размещение мобильной сущности определяется посредством службы локализации. Служба локализации использует в качестве исход-

ных данных идентификатор сущности и возвращает текущий адрес соответствующей ему сущности. При наличии нескольких копий одной сущности будет возвращено несколько адресов. Реализация службы локализации возможна пятью способами:

1. Широковещательная рассылка. В случае локализации сущности с помощью широковещательной рассылки идентификатор сущности доносится до каждой ЭВМ, которые проверяют наличие у них запрашиваемой сущности. ЭВМ, которые могут предоставить точку входа к искомой сущности, посылают ответное сообщение, содержащее адрес точки входа (напр.: протокол разрешения адресов Интернета ARP). Широковещательная рассылка с ростом сети теряет эффективность;

2. Групповая рассылка. Групповая рассылка является вариантом широковещательной рассылки, но не для всей сети, а только для ограниченного числа ЭВМ сети;

3. Передача указателей. При передаче указателей сущность, перемещаясь в другое место системы, оставляет в старом месте ссылку на свое новое местоположение. Преимущество передачи указателей заключается в возможности использования традиционной службы именования (итеративной или рекурсивной) для поиска текущего адреса по цепочке оставленных указателей. В то же время, реализация службы локализации с помощью передачи указателей не лишена ряда недостатков:

а) При большой двигательной активности мобильной сущности цепочка указателей может стать излишне длинной, что резко снизит производительность работы службы локализации;

б) Имеется необходимость сохранения указателей все то время, пока существует мобильная сущность;

в) Высокая уязвимость к потере ссылок. При потере хотя бы одной ссылки сущность становится недоступной.

Рассмотренные первые три способа реализации службы локализации, а именно: широковещательная и групповая рассылки и передача указателей обеспечивают плохую масштабируемость распределенной системы;

4. С помощью базовой точки. Локализация сущности с помощью базовой точки реализуется путем создания в месте, где была создана сущность, базовой точки, из которой отслеживается текущее местоположение сущности. В базовой точке размещается агент базы, который регистрирует текущий адрес мобильной сущности при каждом её перемещении в новое место. Подход на основе базовой точки используется в качестве аварийного метода служб локализации, основанных на передаче указателей.

5. Иерархический подход. В иерархической схеме сеть делится на домены. Домен верхнего уровня охватывает всю сеть целиком. Каждый домен делится на несколько поддоменов, называемых дочерними доменами. Для дочерних доменов ближай-

ший домен более высокого уровня называется родительским доменом. Домен самого нижнего уровня называется листовым доменом и соответствует локальной сети или соте в мобильной сети. Каждый домен имеет ассоциированный с ним направляющий узел, который отслеживает сущности домена. Направляющий узел содержит таблицу идентификаторов всех сущностей своего домена.

Направляющий узел домена верхнего уровня называется корневым направляющим узлом и содержит сведения обо всех сущностях сети.

Для отслеживания местонахождения сущностей каждая из них представлена локализующей записью в своем направляющем узле. Локализующая запись содержит текущий адрес сущности в своем домене. Таким образом, текущий адрес конечной сущности содержится только в локализующей записи листового направляющего узла. Направляющие узлы более высоких уровней содержат только адреса направляющих узлов дочерних доменов.

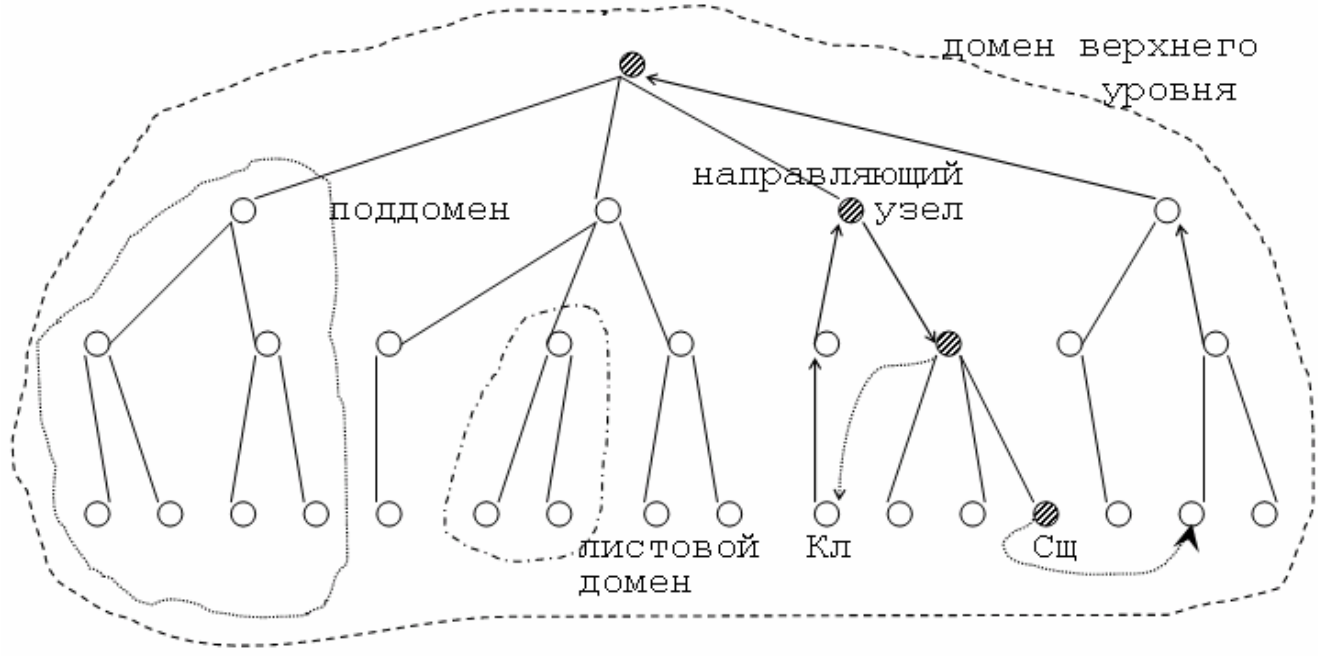


Рис. 5. Иерархическая схема локализации сущности.
Кл – клиент
Сщ - сущность

Клиент, желающий найти сущность, посылает запрос направляющему узлу своего домена. Если этот направляющий узел не содержит идентификатор запрашиваемой сущности, узел пересылает запрос своему родителю. Запрос передается вверх по уровням узлов до тех пор, пока не найдется направляющий узел, содержащий нужный идентификатор.

Ввиду того, что локализующие записи направляющих узлов содержат только адреса направляющих узлов дочерних доменов, далее запрос передается вниз по уровням узлов, содержащих идентификатор запрашиваемой сущности до тех пор, пока не

достигнет листового домена, содержащего запрашиваемую сущность. Адрес, содержащийся в локализующей записи направляющего узла листового домена запрашиваемой сущности, возвращается клиенту, инициировавшему поиск.

При перемещении сущности на другой листовой домен, направляющий узел этого домена посылает своему родительскому узлу сообщение о появлении в домене новой сущности. Если на родительском узле идентификатор этой сущности отсутствует, он пересылает данное сообщение вверх на следующий уровень.

Сообщение о появлении в домене новой сущности пересылается вверх по уровням с узла на узел до тех пор, пока на очередном направляющем узле не обнаружится идентификатор перемещенной сущности. Направляющий узел, получив сообщение о появлении в домене новой сущности, идентификатор которой на нем уже присутствует, изменяет значение локализующей записи этой сущности на новое и инициирует удаление идентификаторов перемещенной сущности по всей цепочке дочерних узлов вплоть до листового домена, на котором ранее находилась эта сущность.

Иерархическая служба локализации затрудняет масштабируемость сети вследствие необходимости хранения и обработки большого количества идентификаторов на корневом направляющем узле.

§ 3. Удаление сущностей, на которые нет ссылок

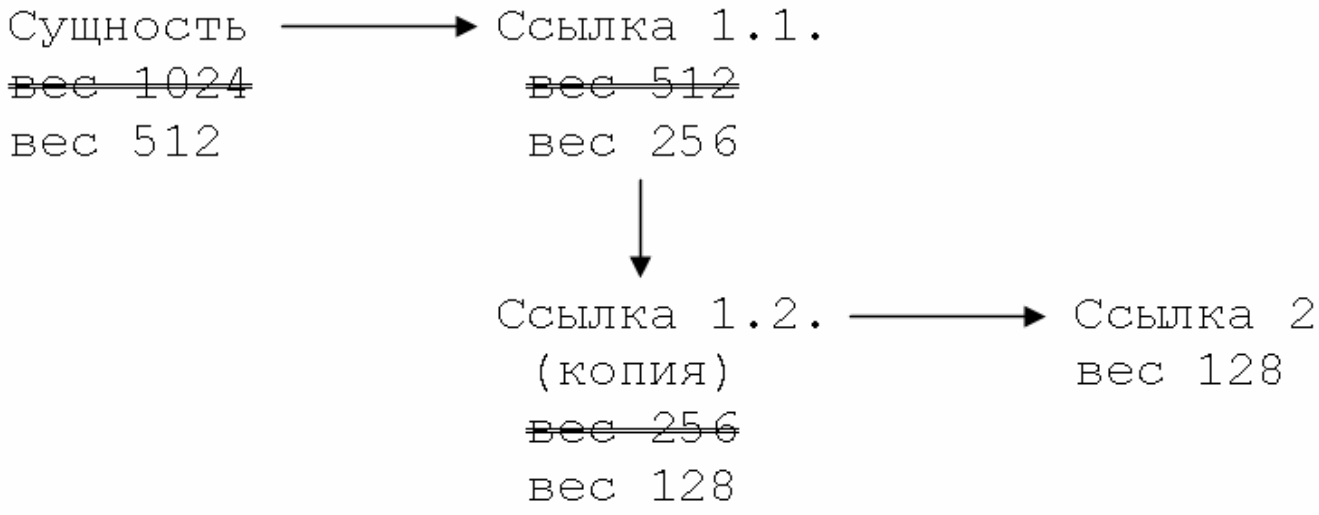
Если в распределенной системе имеются сущности, на которые отсутствуют ссылки, или несколько сущностей, взаимно ссылающихся друг на друга, то они непроизводительно занимают ресурсы системы и по этой причине подлежат удалению. В локальных ЭВМ критерий возможности удаления сущности основан на подсчете ссылок на эту сущность.

При создании ссылки на сущность организуется счетчик ссылок, который увеличивается на единицу при создании каждой новой ссылки на одну и ту же сущность. При уничтожении каждой ссылки на сущность соответствующий ей счетчик ссылок уменьшается на единицу. При обнулении счетчика ссылок соответствующая ему сущность удаляется.

В распределенных системах простой подсчет ссылок приводит к проблемам при потерях удаленных ссылок или подтверждений об изменении состояния счетчика ссылок. Потери вызваны низкой надежностью каналов связи. Для решения проблем при подсчете ссылок в распределенных системах применяют следующие методы:

1. Взвешенный подсчет ссылок. Каждой вновь создаваемой сущности присваивается определенный вес (как правило, это число, равное целой степени числа два). При создании уда-

ленной ссылки ей присваивается половина веса соответствующей сущности, а вес сущности уменьшается вдвое. При копировании ссылки вес копируемой ссылки уменьшается вдвое, а копии присваивается половина начального веса копируемой ссылки. При создании ссылки на ссылку также вес ссылки, на которую ссылаются, уменьшается вдвое, а новой ссылке присваивается половина начального веса ссылки, на которую ссылаются. Пример:

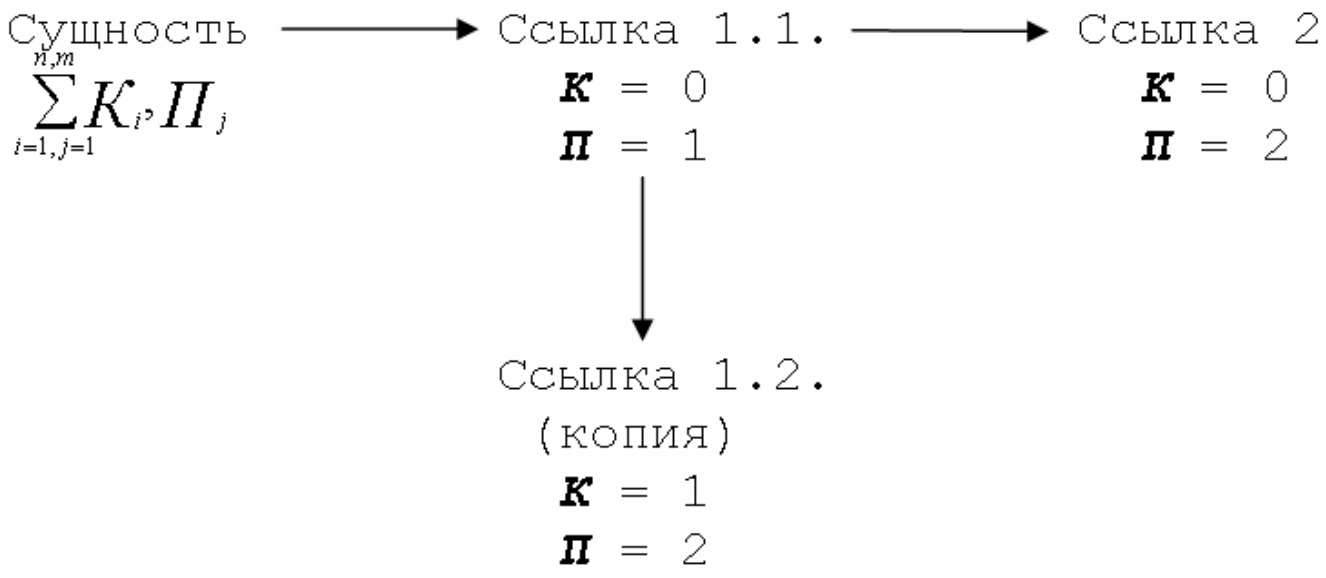


Необходимо заметить, что при взвешенном подсчете ссылок сумма весов ссылок равна весу сущности.

При уничтожении ссылки сущности посылается сообщение, уменьшающее значение её веса на значение веса удаляемой ссылки. Как только вес сущности становится равным нулю, она удаляется. Ввиду того, что счетчик веса работает только на уменьшение, ситуация, когда сущность будет уничтожена при наличии ссылок на неё, возникнуть не может;

2. Подсчет поколений ссылок. Для каждой ссылки создается два счетчика – счетчик копий ссылок и счетчик поколений ссылок. Эти счетчики сводятся в таблицу ссылок, хранящуюся на сущности. При создании непосредственной ссылки на сущность, номер поколения в соответствующем ей счетчике становится равным единице. При создании копии ссылки, счетчик копий скопированной ссылки увеличивается на единицу, а счетчик поколения копии ссылки увеличивается на единицу относительно скопированной ссылки. При создании ссылки на ссылку счетчик поколения новой ссылки устанавливается на единицу большим значения счетчика поколения ссылки, на которую ссылаются, а счетчик копий обнуляется. В таблице ссылок ведется сквозной подсчет общего числа поколений ссылок и их копий. При удалении ссылки из таблицы ссылок удаляются все сведения об удаляемой ссылке. Как только сквозная сумма общего числа поколений ссылок и их копий становится равной нулю, сущность уничтожается Пример: В этом примере *К* – текущее значение счетчика копий ссылок, *П* – текущее значение

счетчика поколений ссылок, *n* – максимальное количество копий ссылок, *m* – максимальное количество поколений ссылок;



3. Создание списка ссылок. На сущности создается полный список ссылок на сущность со следующими идемпотентными свойствами:

а) Добавление в список ссылки невозможно, если такая ссылка в списке уже существует;

б) Попытка удаления ссылки, отсутствующей в списке, ни к чему не приводит.

Достоинство метода – простота сохранения непротиворечивости ссылок при сбоях процессов. Недостаток метода – плохая масштабируемость системы.

§ 4. Файловые системы ОС РВ QNX

Одной из важнейших точек приложения служб разрешения имен являются файловые системы. Файловые системы в операционной системе реального масштаба времени реализуют процессы, называемые администраторами ресурсов. Как и все процессы, администраторы ресурсов, реализующие файловые системы, работают вне микроядра. Прикладные программы используют файловые системы посредством механизма обмена сообщениями, которые генерируются с помощью программного интерфейса POSIX. Каждая файловая система получает свою область в пространстве имен, называемую точкой монтирования, и предоставляет свои службы также посредством стандартного программного интерфейса POSIX. Администраторы ресурсов файловой системы получают точку монтирования и управляют структурой каталогов ниже этой точки. Администраторы ресурсов также осуществляют проверку прав доступа к файлам. Преимущества управления файловыми системами с помощью администраторов ресурсов:

1. Файловые системы можно запускать и останавливать динамически без перезагрузки системы;

2. Множество файловых систем могут работать одновременно;

3. Прикладные программы используют одно общее пространство имен и единый интерфейс независимо от конфигурации и количества применяемых файловых систем;

4. Обеспечивается прозрачность доступа и местоположения.

Файловые системы подразделяются на пять классов:

1. Образная файловая система. Специальная файловая система, которая существует в любой системе и содержит программные модули, включенные в состав загрузочного образа. Администратор процессов автоматически преобразует образную файловую систему в файловую систему в оперативной памяти;

2. Блочная файловая система. Традиционная файловая система, работающая с блочными устройствами типа жестких дисков и приводов компакт-дисков. К этому классу относятся файловые системы QNX4, DOS, CDFS;

3. Файловая система во флэш-памяти. Эта файловая система предназначена для устройств с флэш-памятью. К этому классу относятся файловые системы FFS3, ETFS;

4. Сетевая файловая система. Данная файловая система предоставляет сетевой доступ к файловым системам, находящимся на удаленных ЭВМ. К этому классу относятся файловые системы NFS, CIFS (SMB);

5. Виртуальная файловая система. Бывает двух видов:

а) Пакетная файловая система. Этот вид виртуальной файловой системы отображает клиенту выбранные файлы и каталоги в заданном пользователем представлении;

б) Распаковщик. Распаковщик – это менеджер ресурсов, выполняющий распаковку сжатых файлов для нужд других файловых систем.

Операционная система реального масштаба времени QNX поддерживает следующие виды файловых систем:

1. Образная файловая система;

2. Файловая система в ОЗУ. Эта файловая система применяется в миниатюрных встраиваемых системах, где не требуются постоянные запоминающие устройства, а необходима компактная быстродействующая файловая система с минимальными функциональными возможностями для временного хранения данных. В этой файловой системе нельзя создавать каталоги (папки) и задавать свойства файлов;

3. Файловая система ETFS (Embedded Transaction File System – встраиваемая транзакционная файловая система). Высоконадежная файловая система, предназначенная для постоянных перезаписываемых запоминающих устройств с полупроводниковой памятью. Файловая система ETFS основана на механизме транзакций. Каждая операция записи является транзакцией.

Если в момент выполнения операции случится какой-либо сбой, старые данные останутся в сохранности. Отказоустойчивость файловой системы ETFS обеспечивается благодаря реализации следующих функций:

а) Выравнивание динамического износа. Каждый блок флэш-памяти допускает ограниченное количество безотказных циклов стирания, которое, как правило, не превышает 100 тысяч. Файловая система ETFS ведет подсчет количества стираний по каждому блоку памяти для учета их износа при использовании и равномерного распределения нагрузки между блоками. Без выравнивания динамического износа среднее время наработки флэш-накопителя на отказ составляет десять дней, а при использовании выравнивания – сорок лет;

б) Выравнивание статического износа. Накопители часто содержат статичные файлы, используемые только для чтения (например: файлы программ). При этом остальные блоки флэш-памяти, в которых хранятся периодически изменяющиеся данные, будут изнашиваться значительно быстрее блоков, хранящих информацию только для чтения. Файловая система ETFS отслеживает статичные блоки с низким износом и периодически перемещает в них динамически изменяемые данные, а информацию только для чтения – на место, где ранее находились динамически изменяемые данные;

в) Выявление ошибок с помощью циклического контрольного кода;

г) Исправление одиночных ошибок с помощью кода Хэмминга;

д) Контроль количества операций чтения и автоматическое обновление. Большинство флэш-накопителей допускают не более 100 тысяч гарантированных операций чтения подряд, без обновления данных. Файловая система ETFS производит подсчет последовательных операций чтения подряд для каждого блока и до достижения 100 тысяч чтений автоматически обновляет данные в блоке памяти;

е) Откат транзакций. При запуске или перезагрузке файловой системы ETFS, она обрабатывает все транзакции и отменяет последние начавшиеся перед остановом или перезагрузкой частично выполненные или неуспешные транзакции. Операция отката сама является транзакцией, поэтому даже в случае возникновения сбоя при её выполнении, она будет повторена после вторичной перезагрузки. Это позволяет системе восстанавливаться даже после множественных последовательных сбоев;

ж) Атомарные операции с файлами. В отличие от других файловых систем, где перемещение файла из одного каталога в другой представляет собой многоэтапную операцию, в файловой

системе ETFS такое перемещение осуществляется посредством одной единственной операции записи;

з) Автоматическая дефрагментация файлов. Файловая система ETFS следит за уровнем фрагментированности каждого файла и периодически выполняет его дефрагментацию в фоновом режиме. Для уменьшения фрагментированности файлов также применяется буферизация операций;

4. Файловая система QNX4. Файловая система QNX4 имеет чрезвычайно высокую степень отказоустойчивости благодаря использованию экстент-ориентированной (extent - пространство) схемы распределения блоков на основе битовой карты, а также благодаря применению специализированных подписей, обеспечивающих защиту данных от потерь и возможность быстрого восстановления;

5. Файловая система DOS (Disk Operating System - дисковая операционная система). Использование этой файловой системы обусловлено исключительно соображениями совместимости с устаревшей средой DOS и средой Windows. В отличие от классической файловой системы DOS, в операционной системе QNX предусмотрено обеспечение прозрачности доступа к DOS-дискам в соответствии с требованиями стандарта POSIX;

6. Файловая система CDFS (Compact Disc File System - файловая система для компакт-дисков). Это стандартная файловая система на компакт-дисках, но, также как и в предыдущем случае, в операционной системе QNX предусмотрено обеспечение прозрачности доступа к компакт-дискам в соответствии с требованиями стандарта POSIX;

7. Файловая система FFS3 (Flash File System – файловая система для флэш-памяти). Драйверы Файловой системы FFS3 реализуют высоконадежную POSIX-подобную файловую систему на устройствах с флэш-памятью типа NOR. Механизмы функционирования файловой системы FFS3 подобны механизмам функционирования файловой системы ETFS, но, в отличие от последней, она более проста и компактна;

8. Файловая система NFS (Network File System - сетевая файловая система). Сетевая файловая система NFS обеспечивает клиентам прозрачный доступ к файлам через сеть, обеспечивая совместимость со многими современными операционными системами;

9. Файловая система CIFS (Common Internet File System - общая межсетевая файловая система). Ранее эта файловая система называлась SMB (Server Message Block - блок сообщений сервера). Во многих современных операционных системах последнее название так и сохраняется. Файловая система CIFS предоставляет клиентам прозрачный сетевой доступ к файлам на SMB-сервере, работающем под управлением операционных систем Windows или UNIX;

10. Файловая система Ext2. Это стандартная файловая система операционной системы Linux, но, также как и в случаях с файловыми системами DOS и CDFS, в операционной системе QNX предусмотрено обеспечение прозрачности доступа к разделам Linux в соответствии с требованиями стандарта POSIX;

11. Виртуальные файловые системы.

Глава 4. Синхронизация в распределенных системах

Ввиду того, что в распределенных системах процессы одной прикладной программы типа «клиент-сервер», работая на разных ЭВМ сети, выполняют одну общую задачу, синхронизация процессов между собой играет ключевую роль.

Все автоматизированные системы управления являются системами реального масштаба времени. Под системой реального масштаба времени понимается такая система, на время реакции которой на внешние события наложены жесткие ограничения извне. Системы реального масштаба времени могут быть:

1. Мягкая система реального масштаба времени. Превышение времени реакции мягкой системы реального масштаба времени на внешние события сверх установленных пределов приводит к снижению эффективности управления. Примером мягкой системы реального масштаба времени может служить система продажи железнодорожных или авиационных билетов. Предел времени реакции системы на внешнее событие – в данном случае запрос потенциального пассажира о желании приобрести билет – ограничивается терпением клиента. Если по какой-либо причине терпение потенциального пассажира заканчивается раньше, чем кассир выдает ему билет, происходит снижение эффективности управления – в данном случае, касса теряет клиента и лишается доли выручки от непроданного билета;

2. Жесткая система реального времени. Превышение времени реакции жесткой системы реального масштаба времени на внешние события сверх установленных пределов приводит к фатальным последствиям. Примером жесткой системы реального масштаба времени может служить система автоматического управления самолетом (автопилот). Предел времени реакции системы на внешние события в данном случае определяется инерционностью летательного аппарата, когда при незапланированном изменении каких-либо параметров внешней среды или самого самолета необходимо предпринять парирующие действия до того, как самолет начнет незапланированно изменять траекторию полета. Если система управления не успеет отработать рулями резкий переход летательного аппарата из плотной воздушной массы в разреженную до того, как самолет начнет падать, то он упадет и разобьется.

Все процессы реального масштаба времени требуют синхронизации с текущим временем.

§ 1. Синхронизация с текущим временем

Все современные ЭВМ имеют встроенные процессы для подсчета текущего времени, называемые системными часами ЭВМ. Точность измерения времени зависит от конкретных задач, решаемых системой, а также от инерционности управляемых системой объектов. Чем менее инерционны объекты управления системы, тем точнее необходимо измерять текущее время. Однако, некоторые процессы (например, в спутниковой навигации и связи) требуют синхронизации системных часов с Общемировым временем.

Еще в семидесятые годы прошлого столетия в качестве эталона времени принималось Гринвичское время, которое рассчитывалось астрономическими методами в Гринвичской обсерватории, расположенной на нулевом меридиане (Великобритания). Однако исследования показали, что неравномерность движения Земли по своей орбите, вызванная влиянием массивных планет Солнечной системы, не позволяет использовать Гринвичское время для нужд спутниковой навигации и связи. Спутник, являясь по отношению к Земле, а тем более к массивным планетам Солнечной системы, объектом микроскопических размеров, практически не испытывает их возмущающих влияний и движется по своей орбите значительно более равномерно, чем Земля вокруг Солнца. Точность определения координат с помощью искусственного спутника Земли прямо пропорциональна точности измерения времени спутниковой системой навигации. По этой причине возникла необходимость иметь более точное время, чем Гринвичское, в частности, более равномерно текущее, не подверженное астрономическим флуктуациям.

Вышеуказанная проблема была решена с помощью так называемых «атомных» часов, работа которых основывается на подсчете колебаний атома цезия-133. Одна секунда – это время, за которое атом цезия-133 совершает 9 192 631 770 колебаний. По всему миру были размещены пятьдесят лабораторий, оснащенных такими часами. Лаборатории расположены таким образом, чтобы взаимно компенсировать влияние гравитации, вращения Земли вокруг своей оси и вокруг Солнца на работу часов. Лаборатории производят подсчет времени с помощью подобных «атомных» часов и посылают значение времени в Париж в Международное бюро Мер и Весов. Международное бюро Мер и Весов усредняет данные этих пятидесяти лабораторий и выдает для пользователей значение Общемирового времени. Необходимо заметить, что из-за замедления вращения Земли вокруг своей

оси и вокруг Солнца средний солнечный день все время удлиняется, чего нельзя сказать о периоде колебаний атома цезия-133. 86 400 секунд общемирового времени на три миллисекунды меньше среднего солнечного дня. Когда разница между Общемировым и солнечным временем достигает восьмиста миллисекунд, Международное бюро Мер и Весов объявляет о добавлении так называемой «потерянной» секунды. В частности, «потерянная» секунда была добавлена на рубеже 2005 и 2006 годов.

В большинстве развитых стран мира существуют национальные службы точного времени, передающие значение общемирового времени и эталонных частот для национальных потребителей. В совокупности эти службы образуют Всемирную службу точного времени. В России национальной службой точного времени и частью Всемирной службы точного времени является Государственная служба Времени и Частоты (ГСВЧ), передающая значения точного времени и эталонных частот по сети специализированных радиостанций, разбросанных по всей территории России, а также по первому каналу телевизионного вещания, радиостанциям «Радио России» и «Радио Маяк».

По первому каналу телевизионного вещания информация о текущем значении времени в двоично-десятичном коде и эталонные сигналы частоты 1 МГц передаются непрерывно в составе каждой шестой строки телевизионного сигнала в течение 24 кадров.

Через сеть звукового вещания радиостанции «Радио России» и «Радио Маяк» в начале каждого часа передается группа из шести прямоугольных радиоимпульсов с частотой заполнения 1000 Гц. Первые пять импульсов имеют длительность 100 миллисекунд каждый с интервалом 900 миллисекунд, а длительность шестого импульса определяется в соответствии с выражением $\tau = (100 + 20h)$, где h - целочисленное значение часа. Начало шестого импульса соответствует началу целого часа (00 минут, 00 секунд).

В сети ЭВМ нельзя найти две машины с одинаковым системным временем. В мире не найдется ЭВМ, системное время которой точно совпадает с данными Всемирной службы точного времени.

Синхронизацию системных часов всех ЭВМ распределенной системы осуществляет служба времени, распределенная по всем машинам системы. Если одна из ЭВМ системы имеет приемник сигналов Общемирового времени, то эта машина называется сервером единого времени. В системах, не критичных к использованию Общемирового времени, но работающих в реальном масштабе времени, также используется сервер единого времени, с той только разницей, что он не имеет приемника сигналов Общемирового времени. Но в любом случае задача службы

времени состоит в синхронизации системных часов всех ЭВМ системы с сервером единого времени.

Рассмотрим принцип синхронизации системных часов ЭВМ с текущим временем. Пусть значение текущего времени на системных часах сервера единого времени равно t, а значение времени на системных часах рабочей станции равно $C(t)$.

В идеале $C(t) = t$, то есть $\frac{dC(t)}{dt} = 1$.

Реально системные часы рабочей станции либо отстают, либо спешат, то есть $1 - \rho \le \frac{dC(t)}{dt} \le 1 + \rho$, где ρ – это максимальная скорость дрейфа. Максимальная скорость дрейфа – это та величина, про которую мы говорим, произнося фразу, что «мои часы отстают на две минуты в сутки» или «мои часы спешат на тридцать секунд в неделю». Максимальную скорость дрейфа можно определить только экспериментально путем наблюдений за системными часами конкретной ЭВМ.

Если два процесса хотят гарантировать синхронность своей взаимной работы не хуже, чем на время δ, то синхронизация часов этих процессов должна производиться не реже, чем через каждые $\frac{\delta}{2\rho}$ единиц времени. Выбор конкретных единиц времени (секунд, минут, часов и т.д.) зависит от точности измерения времени в конкретной системе.

Для синхронизации системных часов рабочей станции с сервером единого времени используются два алгоритма:

1. Алгоритм Кристиана. Периодически, не реже чем через каждые $\frac{\delta}{2\rho}$ единиц времени, каждая рабочая станция посылает серверу единого времени запрос о текущем времени. Сервер так быстро, как это возможно, отвечает сообщением, содержащим значение текущего времени. Алгоритм имеет две проблемы:

а) Если системные часы рабочей станции спешат, то значение текущего времени, полученное от сервера единого времени, может оказаться меньшим, чем показания системных часов рабочей станции в этот момент. Эта ситуация может вызвать серьезные отказы в процессах, развивающихся эволюционно во времени (например, процесс счисления пути движущегося объекта). Решение данной проблемы заключается в приведении показаний системных часов рабочей станции в соответствие текущему времени не скачком, а путем искусственного замедления работы системных часов ЭВМ;

б) Передача сообщений к серверу единого времени и обратно требует определенного промежутка времени, длительность которого зависит от текущей загрузки сети. Эту задержку можно учесть путем фиксации моментов времени посылки запроса на сервер единого времени $T_П$ и получения ответа о

текущем времени T_O, определения среднего времени задержки передачи $\frac{T_O - T_П}{2}$ и исправления этим значением полученного точного времени. Однако и здесь имеется некоторая погрешность, обусловленная разным временем пересылки запросного и ответного сообщений вследствие разной загрузки сети. Эту погрешность учесть невозможно, так как невозможно с высокой долей вероятности спрогнозировать текущую загрузку сети ЭВМ;

2. Алгоритм Беркли. Периодически, не реже чем через каждые $\frac{\delta}{2\rho}$ единиц времени, сервер единого времени опрашивает системные часы каждой ЭВМ сети и предлагает им установить их системные часы на новое время. Проблемы алгоритма те же, что и у алгоритма Кристиана, но вторая проблема решается сложнее и с бóльшими погрешностями.

Всем известно, что существует возможность синхронизации системных часов ЭВМ с помощью Интернета. Во всемирной сети существует протокол сетевого времени NTP (Network Time Protocol). Однако вследствие неустановленного времени задержки передачи сообщений о точном времени, с помощью этого протокола можно обеспечить синхронизацию времени не точнее пятидесяти миллисекунд, и то только на выделенных линиях оптоволоконного качества. Такая точность, безусловно, более чем достаточна для большинства прикладных программ реального масштаба времени, но процессы, требующие более высокую точность измерения времени, вынуждены пользоваться непосредственно данными Всемирной службы точного времени.

§ 2. Синхронизация процессов в распределенных системах

Для процессов, не критичных к использованию точного времени, используются другие виды синхронизации, основным из которых является синхронизация, называемая «логические часы Лампорта». Любое пересылаемое сообщение содержит в своем заголовке информацию о времени отправки по системным часам ЭВМ-отправителя. Если при доставке сообщения системные часы ЭВМ-получателя показывают время более раннее, чем время отправки, получатель быстро подводит свои часы таким образом, чтобы они показывали время на единицу большее времени отправки.

С целью упорядочивания одновременно происходящих событий к значению времени справа через точку добавляется номер процесса (например: 42.1). Причинно-следственная связь между процессами может быть соблюдена с помощью так называемых векторных отметок времени, когда к значению времени и номера процесса добавляется ещё и номер сообщения, вызвавшего

активность данного процесса (например: 42.1.6). Метка времени, таким образом, получает кортеж (вектор) параметров.

Большинство распределенных алгоритмов требуют, чтобы один из процессов выполнял главенствующую роль над другими. В разных источниках такой процесс называется координатором, инициатором, диспетчером и так далее. В настоящем пособии процесс, выполняющий в распределенной системе главенствующую роль над другими, будем называть ведущим или пилотным процессом. Процессы, подчиненные ведущему процессу, будем называть ведомыми процессами. Для определения ведущего процесса существуют два алгоритма, построенных на наличии уникальных приоритетных номеров у всех процессов, функционирующих в системе:

1. Мажоритарный алгоритм. Когда один из процессов замечает, что ведущий процесс перестал отвечать на запросы, он посылает всем процессам с бóльшими, чем у себя номерами, сообщение «голосование». Если никто не отвечает, то процесс, пославший данное сообщение, становится ведущим, о чем уведомляет все остальные процессы. Если один из процессов с бóльшим номером ответил, то процесс, пославший данное сообщение, остается ведомым. В любой момент времени процесс может получить сообщение «голосование» от другого процесса с меньшим номером. При получении такого сообщения процесс дает квитанцию о готовности стать ведущим и сам, в свою очередь, организует отправку сообщения «голосование». ведущим процессом в мажоритарном алгоритме всегда является самый приоритетный процесс;

2. Кольцевой алгоритм. Этот алгоритм работает при наличии физического или логического упорядочивания процессов, то есть когда каждый процесс имеет информацию о том, какой процесс является его преемником. Когда один из процессов замечает, что ведущий процесс перестал отвечать на запросы, он посылает своему преемнику сообщение «голосование», содержащее свой уникальный в системе приоритетный номер. Процесс, получивший сообщение «голосование», добавляет к содержащемуся там номеру свой приоритетный номер и посылает сообщение дальше своему преемнику. Получение процессом сообщения «голосование», содержащее его собственный приоритетный номер означает, что кольцо голосования замкнулось, все активные процессы приняли участие в голосовании. При этом процесс, инициировавший голосование, получает сообщение «голосование», содержащее номера всех активных процессов, принявших участие в голосовании.

Получив сообщение «голосование» с номерами всех активных процессов в системе, процесс, инициировавший голосование, формирует сообщение «ведущий», содержащее номера всех активных процессов в системе, и вновь отправляет его по

кругу преемников. Процессы, принявшие сообщение «ведущий», получают информацию о ведущем процессе (процессе с наибольшим приоритетным номером) и список всех активных функционирующих в системе процессов.

§ 3. Взаимное исключение процессов

В распределенных системах существует опасность одновременного использования одних и тех же ресурсов различными процессами. Для исключения такой возможности применяются три алгоритма:

1. Централизованный алгоритм. Процесс, нуждающийся в каком-либо ресурсе, запрашивает разрешение на доступ к этому ресурсу у ведущего процесса. Если ни один из процессов не работает с данным ресурсом, ведущий процесс дает разрешение запрашивающему процессу на доступ к ресурсу. Если запрашиваемый ресурс занят, ведущий процесс организует очередь запросов и предоставляет ресурс последовательно каждому из запрашивающих процессов. Проблемы в централизованном алгоритме возникают при отказе ведущего процесса. Для решения этой проблемы необходимо переопределение ведущего процесса;

2. Распределенный алгоритм. Если процессу необходим какой-либо ресурс, он создает сообщение, содержащее имя ресурса, свой номер и значение текущего момента времени. Затем это сообщение рассылается всем активным в системе процессам, в том числе и самому себе. Когда процесс получает вышеуказанное сообщение, возможны три варианта действий:

а) Если процесс-получатель не работает с указанным ресурсом и не собирается его занимать, он отсылает отправителю квитанцию о приеме сообщения;

б) Если процесс-получатель работает с указанным ресурсом, он ничего не отвечает, а помещает запрос в очередь;

в) Если процесс-получатель собирается занять указанный ресурс, то есть он также разослал всем активным в системе процессам сообщение о намерении занять ресурс, но еще его не занял, он сравнивает метку времени полученного сообщения с меткой времени сообщения, которое он сам разослал. Если полученное сообщение послано раньше, процесс получатель отсылает отправителю квитанцию о приеме сообщения и ничего не делает. Если его собственное сообщение послано раньше, то процесс-получатель ничего не отвечает, ставит запрос в очередь и занимает нужный ему ресурс.

Проблемы в распределенном алгоритме возникают при отказе одного из процессов. Данная ситуация может быть воспринята неправильно другими процессами как отказ в доступе к ресурсу (занятость ресурса);

3. Алгоритм маркерного кольца. Все активные процессы в системе программно упорядочиваются и из них создается логическое кольцо. По кольцу от процесса к процессу запускается маркерное сообщение. Процесс может занять какой-либо ресурс, лишь получив маркерное сообщение. Отправить маркерное сообщение дальше по кольцу процесс может лишь после освобождения ресурса. Для каждого ресурса существует свое маркерное сообщение. Проблемы в алгоритме маркерного кольца возникают при потерях маркера.

§ 4. Распределенные транзакции

Транзакция – это сложная многоступенчатая операция, выполняющаяся как единый неделимый единовременный процесс.

Свойства транзакций:

1. Атомарность. Для окружающего мира транзакция неделима, то есть она либо полностью выполняется, либо полностью не выполняется, причем если она выполняется, то как одна неделимая одновременная операция;

2. Непротиворечивость. Транзакция не нарушает инвариантов (ключевых свойств) системы;

3. Изолированность. Одновременно выполняющиеся транзакции не влияют друг на друга;

4. Долговечность. После завершения транзакции результаты её работы неизменны.

Виды транзакций:

1. Плоская транзакция – это транзакция, строго удовлетворяющая четырем вышеизложенным свойствам транзакций;

2. Вложенная транзакция – это транзакция, состоящая из дочерних транзакций, способных работать параллельно без взаимных блокировок (у дочерних транзакций отсутствует свойство долговечности);

3. Распределенная транзакция – это набор плоских транзакций, совместно выполняющих одну общую задачу.

При реализации транзакции на локальной ЭВМ, процесс, её выполняющий, получает закрытое рабочее пространство оперативной памяти, в котором производит все промежуточные действия до тех пор, пока транзакция не выполнится или не прервется.

При реализации распределенных транзакций на различных ЭВМ сети, необходим алгоритм управления для сохранения свойств транзакций. Для управления распределенными транзакциями служат три иерархических процесса:

1. Менеджер данных. Менеджер данных находится на верхнем уровне иерархии и осуществляет транзактнонезависимые операции чтения и записи данных;

2. Планировщик. Планировщик определяет, в какой момент времени и какой транзакции разрешается передать операцию чтения или записи менеджеру данных;

3. Менеджер транзакций. Менеджер транзакций находится на нижнем уровне иерархии и обрабатывает команды транзакций, преобразуя их в запросы к планировщику.

Планировщик и менеджер данных размещаются на каждой из ЭВМ распределенной системы и, работая совместно, обеспечивают гарантии непротиворечивости локальных данных. Менеджер транзакций размещается только на сервере транзакций.

Для обеспечения свойства изолированности транзакций необходима синхронизация конфликтующих операций. Две операции конфликтуют, если они работают с одним и тем же элементом данных, и хотя бы одна из них является операцией записи. Синхронизация конфликтующих операций при выполнении транзакций производится двумя методами:

1. Двухфазная блокировка. Когда процесс в ходе транзакции нуждается в чтении или записи элемента данных, он делает запрос планировщику заблокировать для него этот элемент данных. Когда необходимость в этом элементе данных исчезает, процесс делает запрос планировщику снять блокировку. Задача планировщика состоит в том, чтобы устанавливать и снимать блокировку, не допуская некорректного перемежения операций, выполняемых различными транзакциями над одним и тем же элементом данных. При двухфазной блокировке различают две фазы:

а) Фаза подъема, на которой планировщик устанавливает все необходимые блокировки;

б) Фаза спада, на которой планировщик снимает все необходимые блокировки.

При этом выполняются три правила:

а) Если запрашивается операция, конфликтующая с операциями, уже получившими блокировку, то её выполнение откладывается;

б) Если менеджер данных уведомил планировщик, что он осуществляет операцию с элементом данных, планировщик не снимет блокировку с этого элемента данных до окончания работы менеджера данных с этим элементом данных;

в) Если планировщик снял блокировку с данных по требованию какой-либо транзакции, он никогда больше не заблокирует данные по требованию этой транзакции.

Варианты двухфазной блокировки:

а) Строгая двухфазная блокировка. В этом варианте блокировки фаза спада не начинается до тех пор, пока транзакция не завершится;

б) Централизованная двухфазная блокировка. В отличие от классического варианта, в этом случае за установку и

снятие блокировок отвечает один централизованный процесс-планировщик, называемый менеджер блокировок;

в) Первичная двухфазная блокировка. В этом варианте блокировки с каждого элемента данных снимается копия, на которую устанавливаются блокировки. В отличие от централизованной двухфазной блокировки, первичная двухфазная блокировка может быть распределена по нескольким ЭВМ сети;

г) Распределенная двухфазная блокировка. Этот вариант блокировки применяется, когда данные распределены по нескольким ЭВМ сети. При этом планировщики каждой ЭВМ сети отвечают не только за установку и снятие блокировок, но и за пересылку операций локальным менеджерам данных.

2. Упорядочивание по меткам времени. Упорядочивание по меткам времени может быть двух видов:

а) Пессимистическое упорядочивание по меткам времени. В момент начала каждой транзакции ей присваивается метка времени. Каждая операция, являющаяся частью транзакции, также получает метку времени, эквивалентную метке времени транзакции, частью которой она является. Каждый элемент данных получает метку времени записи и метку времени считывания. Метка времени записи эквивалентна метке времени транзакции, которая последней записывала этот элемент данных. Метка времени считывания эквивалентна метке времени транзакции, которая последней считывала этот элемент данных. В случае конфликта двух операций выполняется операция с меньшим значением метки времени, операция с бóльшим значением метки времени переводится в режим ожидания;

б) Оптимистическое упорядочивание по меткам времени. Параллельно выполняемые транзакции не обращают никакого внимания друг на друга. По окончании выполнения каждой транзакции результаты её работы анализируются на достоверность и либо принимаются, либо отбрасываются, причем в последнем случае инициируется повторное выполнение транзакции. Анализ результатов работы транзакции на достоверность основан на проверке факта изменения элемента данных, с которым работает транзакция, другой транзакцией. Если в процессе выполнения транзакции, элемент данных, с которым она работает, был изменен другой транзакцией, результаты работы обеих транзакций признаются недостоверными.

§ 5. Механизмы синхронизации процессов в ОС РВ QNX

Операционная система реального масштаба времени QNX использует связь посредством сообщений как средство взаимодействия между любыми процессами, как локальными, так и распределенными. Для синхронизации процессов в ОС РВ QNX используются следующие механизмы:

1. Переменные синхронизации (мутексы). В каждый момент времени переменной синхронизации может владеть только один поток выполнения. Другие потоки, которые пытаются захватить эту переменную, блокируются до момента её освобождения. После освобождения переменной синхронизации очередным потоком выполнения переменную синхронизации захватывает поток с наивысшим приоритетом из всех потоков, ожидающих возможности захватить эту переменную. В большинстве процессоров захват переменной синхронизации не требует обращения к микроядру, а выполняется аппаратно специальными командами процессора (например, такие команды имеются в процессорах x86, RISC и т.д.). Действие операционной системы при этом сводится только к управлению приоритетами. Если поток выполнения, который пытается захватить переменную синхронизации, имеет более высокий приоритет, чем поток, непосредственно владеющий этой переменной, то действующий приоритет текущего владельца устанавливается равным приоритету блокированного потока, ожидающего эту переменную синхронизации. Такой механизм обеспечивает минимальное время ожидания переменной синхронизации и решает проблему инверсии приоритетов;

2. Условные переменные. Условная переменная используется для блокировки выполняемого потока выполнения по какому-либо условию. Физически условная переменная представляет собой подпрограмму-функцию, ассоциированную с конкретным потоком выполнения и реализующую проверку какого-либо логического условия. В вызывающую программу условная переменная может возвращать только два значения – *TRUE* и *FALSE*. Условие может быть сколь угодно сложным и не зависит от условной переменной. Пока условие является истинным, поток выполнения, владеющий переменной синхронизации, блокирован. Условная переменная всегда используется совместно с переменной синхронизации. Разблокировать блокированный условной переменной поток выполнения может другой активный поток, изменив условие блокировки и сделав его ложным;

3. Барьеры. Барьер – это механизм синхронизации, позволяющий скоординировать работу нескольких взаимодействующих потоков выполнения таким образом, чтобы каждый из них блокировался в заданной точке в ожидании остальных потоков, прежде чем продолжить свою работу. После того, как заданное количество потоков достигает установленного барьера, все эти потоки разблокируются и продолжают свою работу;

4. Ждущие блокировки. Ждущая блокировка работает аналогично условной переменной, но не для одного потока выполнения, а одновременно для нескольких потоков. При этом используется только одна переменная синхронизации, а условные переменные создаются динамически для каждого потока выполнения;

5. Блокировки по чтению-записи. Блокировка по чтению-записи применяется при необходимости предоставить доступ к данным по чтению всем потокам выполнения, которые его запрашивают, а по записи – только одному потоку. Этот механизм синхронизации предоставляет доступ по чтению всем потокам выполнения, которые его запрашивают. Если поток запрашивает блокировку по записи, запрос отклоняется до тех пор, пока все потоки, выполняющие чтение, не снимут свои блокировки по чтению. Множество потоков выполнения, желающих произвести запись данных, помещаются в очередь в порядке своих приоритетов. Потоки выполняют запись после снятия всех блокировок по чтению в порядке очереди. Читающие потоки выполнения получают доступ к данным только после опустошения очереди. Приоритеты читающих потоков не учитываются. Реализация блокировок по чтению-записи происходит не в микроядре, а посредством переменных синхронизации и условных переменных, предоставляемых микроядром;

6. Семафоры. Семафор – это вариант условной переменной с единственным условием: поток выполнения блокируется при неположительном значении счетчика семафора. Изменить значение счетчика семафора может любой активный процесс. Семафоры специально предназначены для применения в программах обработки асинхронных событий и, в отличие от условных переменных, могут управляться обработчиками сигналов;

7. Синхронизация с помощью FIFO-планирования. Этот механизм синхронизации применяется при недостатке ресурсов процессоров для осуществления параллельной обработки потоков выполнения;

8. Синхронизация с помощью обмена сообщениями. Это единственный механизм синхронизации, способный работать в сети (смотри: связь посредством сообщений). Он осуществляется неявно с помощью блокировок;

9. Синхронизация с помощью атомарных операций. Этот механизм синхронизации применяется при необходимости гарантировать выполнение какой-либо операции, исключая возможность её вытеснения более приоритетным потоком.

Глава 5. Репликация в распределенных системах

§ 1. Понятие непротиворечивости

Цель репликации заключается в повышении надежности системы и её производительности. В то же время возникновение любой копии какого-либо ресурса ставит проблему сохранения непротиворечивости реплик. Под непротиворечивостью реплики ресурса понимается сохранение актуальности копий ресурса. Набор копий ресурса актуален, если операция чтения

данных дает одинаковые результаты для каждой из копий. Обновление данных должно распространяться на все копии до того, как начнется следующая операция чтения данных, то есть обновление данных должно являться транзакцией.

Как правило, в распределенных системах данные существуют не автономно, а в составе объектов. Поэтому правильнее говорить не о репликации данных, а о репликации объектов.

Репликация с целью увеличения производительности системы часто одновременно используется и в качестве способа масштабирования. Это происходит за счет сокращения времени доступа к реплике ресурса при её территориальном приближении к потребителю ресурса. Однако, с ростом сети и числа реплик одного ресурса, увеличивается время на обновление данных, вследствие чего общая производительность распределенной системы падает. Таким образом, при репликации необходим разумный компромисс между количеством реплик одного ресурса и значением общей производительности распределенной системы. Решение данной проблемы заключается в жертвовании требованием об атомарности обновления данных, то есть непротиворечивостью. Для этого производители сетевого программного обеспечения разрабатывают протоколы непротиворечивости, под которыми понимают соглашения о том, какие правила должны выполнять процессы, чтобы хранилище данных работало правильно. При этом внутри системы возможны отдельные состояния нарушения непротиворечивости данных, которые должны скрываться от клиентов. Существует два вида протоколов непротиворечивости:

1. Протоколы непротиворечивости, ориентированные на данные. Эта группа протоколов обеспечивает непротиворечивое представление данных;

2. Протоколы непротиворечивости, ориентированные на клиента. Эти протоколы допускают нарушения непротиворечивости данных, но обеспечивают сокрытие факта нарушений от клиента.

Рассмотренные протоколы определяют одноименные виды непротиворечивости.

§ 2. Непротиворечивость, ориентированная на данные

Непротиворечивость, ориентированная на данные, бывает двух видов:

1. Непротиворечивость, не требующая операций синхронизации. В свою очередь, этот вид непротиворечивости разбивается на четыре подвида:

а) Строгая непротиворечивость. При этой непротиворечивости всякое чтение элемента данных возвращает значение, соответствующее результату последней записи этого элемента

данных. Когда хранилище данных строго непротиворечиво, все операции записи мгновенно замечаются всеми процессами. Выдерживается абсолютный глобальный порядок во времени. Если элемент данных изменяется, все последующие операции чтения этого элемента данных возвращают новое значение. Строгая непротиворечивость – это идеализированный вид непротиворечивости, который в распределенной системе достигнуть очень сложно, а чаще всего невозможно. Поэтому на практике в зависимости от конкретной ситуации применяют другие, менее строгие виды непротиворечивости;

б) Последовательная непротиворечивость. Это менее строгая непротиворечивость. Хранилище данных последовательно непротиворечиво, если результат любого действия над элементом данных такой же, как если бы операции чтения и записи элемента данных выполнялись бы в некотором последовательном порядке. При последовательной непротиворечивости все процессы видят одно и то же чередование операций записи элемента данных. Процесс видит операции записи всех процессов, но только свои собственные операции чтения. Недостаток последовательной непротиворечивости заключается в том, что для всякого последовательно непротиворечивого хранилища данных увеличение скорости чтения вызывает падение скорости записи и наоборот;

в) Причинная непротиворечивость. Этот вид непротиворечивости представляет собой ослабленный вариант последовательной непротиворечивости, при котором проводится разделение между событиями, потенциально обладающими причинно-следственной связью, и событиями, ею не обладающими. Чтение связано с записью, предоставляющей данные для этого чтения, причинно-следственной связью. Операции, не имеющие причинно-следственной связи, называются параллельными. Хранилище данных поддерживает причинную непротиворечивость, если операции записи, потенциально связанные причинно-следственной связью, наблюдаются всеми процессами в одинаковом порядке, а параллельные операции записи могут наблюдаться в произвольном порядке. Реализация причинной непротиворечивости требует отслеживания, какие процессы какие операции записи видели. Одним из способов такого отслеживания является использование векторных меток времени для определения причинно-следственной связи;

г) Непротиворечивость FIFO. Данная непротиворечивость подчиняется следующему условию: операции записи, осуществляемые единичным процессом, наблюдаются всеми остальными процессами в том порядке, в котором они осуществляются, но операции записи, осуществляемые различными процессами, могут наблюдаться всеми остальными процессами в разном порядке. Непротиворечивость FIFO предполагает, что все операции

записи, осуществляемые различными процессами, являются параллельными. Реализация непротиворечивости FIFO осуществляется путем именования каждой операции парой чисел - номером процесса и номером операции в процессе – и осуществления операций записи каждого из процессов в порядке их номеров;

2. Непротиворечивость, использующая операции синхронизации. Этот вид непротиворечивости основан на использовании переменной синхронизации. Переменная синхронизации – это ассоциированная с хранилищем данных переменная, значение которой эквивалентно номеру процесса, имеющего право изменять данные в хранилище. С переменной синхронизации ассоциирована операция синхронизации, обеспечивающая сохранение актуальности всех реплик хранилища данных при каждом изменении данных в хранилище. Непротиворечивость, использующая операции синхронизации, бывает трех подвидов:

а) Слабая непротиворечивость. Эта непротиворечивость обладает тремя свойствами:

- доступ к переменным синхронизации производится на условии последовательной непротиворечивости, то есть внешние по отношению к хранилищу данных процессы наблюдают не отдельные операции записи или считывания, а только все операции над переменной синхронизации;

- с переменной синхронизации не может быть произведена ни одна операция до полного завершения всех операций владеющего этой переменной процесса (то есть процесса, номер которого присвоен этой переменной) с хранилищем данных;

- с элементами данных не может быть произведена ни одна операция до полного завершения всех операций с переменной синхронизации.

Слабая непротиворечивость реализует непротиворечивость не отдельных операций записи или считывания, а групп операций, выделение которых производится с помощью переменных синхронизации. В случае слабой непротиворечивости соблюдается последовательная непротиворечивость между группами операций. При этом допускается существование неверных значений отдельных данных внутри такой группы операций. Слабая непротиворечивость чаще всего используется при пакетировании процедур доступа к данным, когда много операций выполняются одна за другой в короткий срок (например, сортировка данных), а затем в течение длительного времени обращения к данным не происходит. Проблема слабой непротиворечивости состоит в том, что хранилище данных не знает, какую операцию над данными совершает процесс, владеющий переменной синхронизации. В то же время ряд реализаций хранилищ данных требует распознавать разницу между операциями чтения и записи данных;

б) Свободная непротиворечивость. Возможность распознавания разницы между операциями записи и чтения данных предоставляет свободная непротиворечивость. Свободная непротиворечивость использует две переменные синхронизации – одну для операций записи, а другую для операций чтения данных. В случае эквивалентности значений обеих переменных номеру одного и того же процесса, говорят о захвате процесса. Если значение одной из переменных синхронизации перестает быть эквивалентно номеру захваченного процесса, говорят об освобождении процесса. Распределенное хранилище данных является свободно непротиворечивым, если выполняются три условия:

- перед выполнением операций записи или считывания данных должен быть произведен захват процесса, работающего с этими данными;
- перед освобождением процесса все операции записи или считывания данных должны быть полностью завершены;
- доступ к переменным синхронизации должен обладать непротиворечивостью FIFO.

Хранилище данных со свободной непротиворечивостью гарантирует, что при захвате процесса все локальные реплики данных будут непротиворечивы относительно своих удаленных копий. Изменения, сделанные процессом в локальных данных, будут распространены на удаленные реплики только при освобождении процесса;

в) Поэлементная непротиворечивость. Эта непротиворечивость является вариантом свободной непротиворечивости, но не для групп, а для отдельных элементов данных. При этом каждому элементу данных ассоциируются две переменные синхронизации. В поэлементной непротиворечивости отсутствует связь между совместно используемыми элементами данных. В то же время, поэлементная непротиворечивость дает возможность независимого параллельного доступа к хранилищу данных.

§ 3. Непротиворечивость, ориентированная на клиента

Непротиворечивость, ориентированная на клиента, бывает пяти видов:

1. Потенциальная непротиворечивость. Хранилище данных потенциально непротиворечиво, если в отсутствии изменений все реплики постепенно становятся идентичными (например, кэширование Web-сайта прокси-сервером при условии отсутствия диалогового режима). Потенциальная непротиворечивость требует только того, чтобы все изменения гарантированно расходились по всем репликам. Потенциальная непротиворечивость используется, если вносить изменения в хранилище данных может лишь ограниченная небольшая группа процессов, а также, если клиент всегда осуществляет доступ только к од-

ной реплике. Мобильные клиенты, при перемещении в другую соту, могут быть переключены на другую реплику хранилища данных. В случае потенциальной непротиворечивости сделанные изменения в прежней реплике могут не успеть дойти до новой реплики данных. В этом случае применяют другие четыре вида непротиворечивости;

2. Непротиворечивость монотонного чтения. Хранилище данных обеспечивает непротиворечивость монотонного чтения, если каждая последующая операция чтения возвращает значение элемента данных, эквивалентное предыдущей операции чтения или более новое. При монотонном чтении процесс никогда не увидит более старого элемента данных, чем полученное при предыдущей операции чтения;

3. Непротиворечивость монотонной записи. Хранилище данных обеспечивает непротиворечивость монотонной записи, если операция записи процессом элемента данных завершается раньше любой из последующих операций записи этим же процессом того же элемента данных. Непротиворечивость монотонной записи подобна непротиворечивости FIFO с тем лишь различием, что в непротиворечивости FIFO участвует набор параллельных процессов, а в непротиворечивости монотонной записи – только один;

4. Непротиворечивость чтения собственных записей. Хранилище данных обеспечивает непротиворечивость чтения собственных записей, если результат операции записи процессом элемента данных всегда виден последующим операциям чтения этим же процессом того же элемента данных. Непротиворечивость чтения собственных записей должна обеспечиваться при обновлении Web-страниц с последующим просмотром результатов;

5. Непротиворечивость записи за чтением. Хранилище данных обеспечивает непротиворечивость записи за чтением, если операция записи процессом элемента данных, следующая за операцией чтения этим же процессом того же элемента данных, гарантирует, что будет выполняться над тем же самым или более новым значением элемента данных, чем то, которое было прочитано предыдущей операцией. Непротиворечивость записи за чтением должна обеспечиваться при реализации сетевой службы новостей, чтобы все пользователи данной службы видели ответы на заданный вопрос только после поступления этого вопроса, а не наоборот.

§ 4. Распространение обновлений

Основные проблемы, возникающие при проектировании распределенных хранилищ данных, заключаются в определении времени создания, места создания и владельца копии хранилища

данных, а также в распространении обновлений. Различают три типа реплик:

1. Постоянные реплики – это исходный набор реплик, образующих распределенное хранилище данных;

2. Реплики, инициируемые сервером. Эти реплики являются копиями хранилища данных, создаваемыми для повышения производительности системы;

3. Реплики, инициируемые клиентом. Эти реплики являются копиями хранилища данных, создаваемыми для сокращения времени доступа к данным.

При любом изменении данных в хранилище, необходимо актуализировать все реплики хранилища данных, то есть распространять обновления по всем репликам. Для распространения обновлений существуют три основные возможности:

1. Распространять только извещения об обновлении. Распространение извещений об обновлении производится в соответствии с протоколом о несостоятельности. Под протоколом о несостоятельности понимается соглашение о способах определения места обновления, то есть какая именно часть хранилища данных была изменена и перестала быть непротиворечивой своим репликам. При распространении извещений об обновлении не передается ничего, кроме собственно извещения. Конкретные действия по фактическому обновлению реплик зависят от поддерживаемого вида непротиворечивости. Преимущество передачи извещений об обновлении состоит в минимизации использования ресурсов сети. Извещения об обновлении чаще всего применяются при значительном преобладании операций записи по отношению к операциям чтения;

2. Передавать данные из одной копии в другую. Такое распространение обновлений применяется при значительном преобладании операций чтения по отношению к операциям записи. Для сокращения требуемых ресурсов сети можно передавать не сами данные, а так называемые журналы обновлений, представляющие собой несколько модификаций данных, упакованных в одно сообщение;

3. Распространять операции обновления по всем копиям. Этот вид распространения обновлений заключается в отказе от переноса модифицированных данных целиком, а указании каждой реплике, какую операцию с ней необходимо произвести. Для реализации распространения операций обновления по всем копиям необходимо, чтобы хранилище данных поддерживало активную репликацию. Активная репликация предполагает, что каждая реплика представлена процессом, способным сохранять актуальность своих данных при распространении операций обновления.

Распространение обновлений может инициироваться как сервером, так и клиентом. Серверное распространение обнов-

лений применяется для поддержания высокого уровня непротиворечивости, делая реплицируемые данные непротиворечивыми сразу после обновления какого-либо элемента данных. При обновлении элемента данных сервер немедленно инициирует распространение обновлений.

В случае клиентского распространения обновлений клиент при необходимости использования какого-либо элемента данных опрашивает сервер в поисках обновлений этого элемента данных.

Встречается смешанная форма инициализации распространения обновлений. В этом случае назначается промежуток времени, в течение которого сервер должен хотя бы один раз передать обновления клиенту. Если по истечении этого промежутка времени сервер ни разу не передал обновлений, клиент сам запрашивает обновления у сервера.

При распространении обновлений возникают проблемы при распространении удалений элементов данных. Если какой-либо элемент данных просто изъять из хранилища данных, то наличие этого элемента в более старой копии хранилища данных может быть интерпретировано как обновление, включающее в себя новый элемент данных. Проблема решается путем замены удаляемого элемента данных записью о его удалении. Но при этом возникает другая проблема – постепенное накопление хранилищем данных большого количества никому не нужных записей об удалении данных. Для решения данной проблемы запись об удалении содержит метку времени её создания, и по истечении заданного промежутка времени устаревания, устаревшие записи об удалении изымаются из хранилища данных.

Глава 6. Надежность распределенной обработки информации

Любая техническая система имеет свойство рано или поздно ломаться. Однако одни системы ломаются почти сразу после начала эксплуатации, другим же уготована долгая и безотказная служба на пользу человеку. Для того, чтобы можно было как-то оценить перспективы работы той или иной системы, была разработана теория надежности, изучающая факторы, влияющие на длительность безотказной эксплуатации технических систем и методы увеличения сроков такой эксплуатации.

§ 1. Основные понятия теории надежности

Надежность – это свойство системы сохранять свою работоспособность в заданных условиях эксплуатации. Понятие надежности включает в себя реализацию четырех требований к системе:

1. Доступность – это требование к системе постоянно находиться в состоянии готовности к работе. Данное требование определяется вероятностью того, что система в произвольный момент времени будет в состоянии выполнить свои функции;

2. Безотказность – это требование к системе безотказно работать в течение заданного промежутка времени. Данное требование определяется средним временем безотказной работы $T_0 = \frac{T_{набл}}{N_{отк}}$, где: $T_{набл}$ – время непрерывного наблюдения за системой, в течение которого определяется среднее время безотказной работы; $N_{отк}$ – количество отказов системы, произошедшее за время наблюдения $T_{набл}$. Необходимо четко различать понятия отказа системы и сбоя. Под отказом системы понимается устойчивое нарушение работоспособности системы, требующее для своего устранения вмешательства оператора. В свою очередь, сбой – это кратковременное самоустраняющееся нарушение работоспособности системы;

3. Безопасность – это требование к системе парировать ситуации возникновения сбоев и отказов в работе. Данное требование определяется степенью катастрофичности ситуации временной неспособности системы выполнять свои функции;

4. Ремонтопригодность – это требование к системе обладать максимальным удобством для осуществления обслуживающим персоналом функций восстановления системы после отказов. Данное требование определяется средним временем восстановления системы.

В автоматизированных системах управления классификация сбоев и отказов происходит как по фактору сохранения работоспособности системы, так и по временно́му фактору, зависящему от динамических характеристик системы и внешней среды:

1. Инерционности управляемых объектов, источников и потребителей информации;

2. Темпа решения задач по обработке информации;

3. Предельно допустимого времени реакции системы на внешние события.

Сбой программы, время самовосстановления которого превышает предельно допустимое время реакции системы на внешние события, в автоматизированных системах управления является отказом системы. Например, самовосстановление программы автоматизированного управления летательным аппаратом после того, как самолет уже начал неуправляемо падать, нельзя считать сбоем.

Надежность технических систем определяется двумя факторами: надежностью компонент и ошибками в конструкции. Для аппаратного обеспечения автоматизированной системы управления решающее значение в обеспечении надежности играет фак-

тор надежности компонент, а для программного обеспечения – ошибки конструкции, в данном случае – ошибки в программах. Отказы при исполнении программ в основном возникают в следующих случаях:

1. Нарушение целостности программного кода в памяти ЭВМ;
2. Искажение исходных данных;
3. Нарушение нормального хода вычислительного процесса.

Сбой происходит чаще всего при отсутствии физического разрушения программного кода, когда при обнаружении неправильной работы программы средствами операционной системы организуется повторное выполнение всей программы или её отдельного законченного участка. Такое действие операционной системы называется программным восстановлением. Программное восстановление становится возможным благодаря ниже перечисленным технологическим методам:

1. Введение избыточности в программные комплексы;
2. Законченность и малые размеры программных модулей;
3. Дублирование данных;
4. Резервирование аппаратных ресурсов.

Различают пять видов отказов распределенных систем:

1. Поломка. При поломке никаких признаков работы распределенной системы или её отдельных компонент не наблюдается;
2. Пропуск данных. Пропуск данных – это отказ в пересылке данных (например, при повреждении среды передачи данных), не позволяющий осуществлять распределенную обработку информации;
3. Ошибка синхронизации. Ошибка синхронизации – это нарушение временных зависимостей между процессами, не позволяющее осуществлять распределенную обработку информации;
4. Ошибка отклика. Ошибка отклика – это неадекватный ответ процесса на удаленный запрос, не позволяющий осуществлять распределенную обработку информации;
5. Произвольная ошибка. Произвольная ошибка – это генерация процессом некорректных сообщений, не позволяющая осуществлять распределенную обработку информации.

Причины отказов могут быть как аппаратными, то есть отказы оборудования, так и программными, возникающими вследствие ошибок программных комплексов. Различают четыре вида ошибок программных комплексов:

1. Технологическая ошибка – это ошибка, возникающая в результате искажения двоичных разрядов при фиксации программ в памяти ЭВМ. Общеизвестно, что программы, хранящиеся на долговременных носителях (дисковых, флэш-накопителях и т.п.), перед началом работы должны быть загружены в опера-

тивную память ЭВМ. Однако с вероятностью $10^{-3} \div 10^{-4}$ происходит искажение двоичных разрядов в процессе загрузки программ с долговременных носителей в оперативную память ЭВМ перед их выполнением. Специальные методы автоматического контроля записи позволяют снизить вероятность искажения двоичных разрядов при загрузке программ с долговременных носителей в оперативную память ЭВМ до значений $10^{-7} \div 10^{-8}$. Доля технологических ошибок среди всех прочих ошибок программных комплексов составляет 5% ÷ 10%;

2. Программная ошибка. Доля данного вид ошибок среди всех прочих ошибок программных комплексов зависит от квалификации программистов-разработчиков. Возникают программные ошибки в результате некорректного программирования. На начальном этапе разработки доля программных ошибок среди всех прочих ошибок программных комплексов составляет 30%. На этапе комплексной отладки эта доля падает до 15%, а на этапе эксплуатации – до 3%. Необходимо заметить, что приведенные цифры являются средними по отрасли разработки программ и могут колебаться в ту или другую сторону в зависимости от опыта разработчиков и сложности разрабатываемых программ;

3. Алгоритмическая ошибка. Этот вид ошибок обусловлен некорректной постановкой задач в техническом задании. Чаще всего в технических заданиях плохо прописывают диапазоны изменения переменных, неправильно оценивают точности используемых и получаемых величин, неправильно учитывают связи между переменными и т.п., что в результате приводит к возникновению алгоритмических ошибок. Доля алгоритмических ошибок среди всех прочих ошибок программных комплексов составляет 30%;

4. Системная ошибка. Системные ошибки – это ошибки, обусловленные неполнотой информации о реальных процессах внешней среды. Любая система, в том числе и автоматизированная система управления, функционирует в условиях внешней среды, учитывая факторы этой среды и каким-то образом оказывая влияние на эту среду. Разработка любой системы происходит на основе имеющейся у разработчика модели внешней среды. В то же время, любая модель представляет собой приближенное, упрощенное представление реального объекта или явления, и учесть в полной мере все проявления реальной внешней среды в модели просто невозможно. В начале процесса отладки, когда система минимально взаимодействует с внешней средой, доля системных ошибок среди всех прочих ошибок программных комплексов составляет 10%. На завершающих этапах комплексной отладки, когда система более тесно взаимодействует с реальной внешней средой, но это взаимодействие происходит в контролируемых, заранее заданных условиях, доля системных ошибок среди всех прочих ошибок программных ком-

плексов вырастает до 40%. В процессе эксплуатации, когда система в полной мере взаимодействует с реальной внешней средой, эта доля является основной среди всех прочих ошибок программных комплексов и составляет 80%.

Одним из важных показателей надежности является величина интенсивности отказов, обратно пропорциональная среднему времени безотказной работы $\lambda = \frac{1}{T_0}$. Зависимость интенсивности отказов от времени для аппаратного обеспечения автоматизированной системы управления имеет вид:

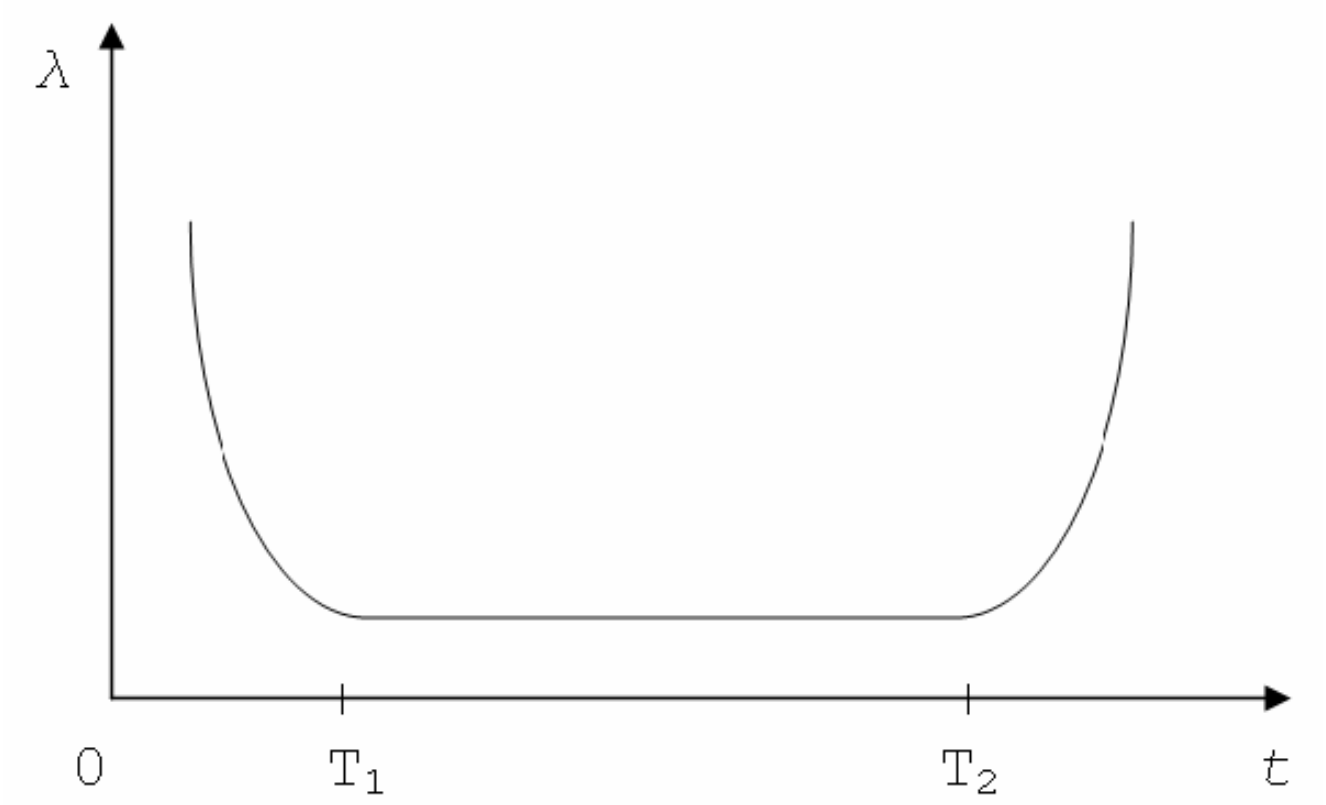


Рис. 6. Зависимость интенсивности отказов от времени для аппаратного обеспечения АСУ.

На этом графике можно выделить три различных участка:

1. От момента создания системы до момента времени T_1. Данный участок характеризуется уменьшением значения интенсивности отказов во времени и называется обкаткой, приработкой, наладкой и т.п. в зависимости от конкретного вида системы;

2. От момента времени T_1 до момента времени T_2. На этом участке значение интенсивности отказов во времени практически не изменяется, оставаясь на минимальном уровне. Такое поведение интенсивности отказов характерно для периода нормальной эксплуатации системы в штатных режимах;

3. От момента времени T_2 и до вывода системы из эксплуатации. Данный участок характеризуется ростом значения интенсивности отказов во времени и называется старением системы.

Для программного обеспечения автоматизированной системы управления зависимость интенсивности отказов от времени имеет несколько другой вид:

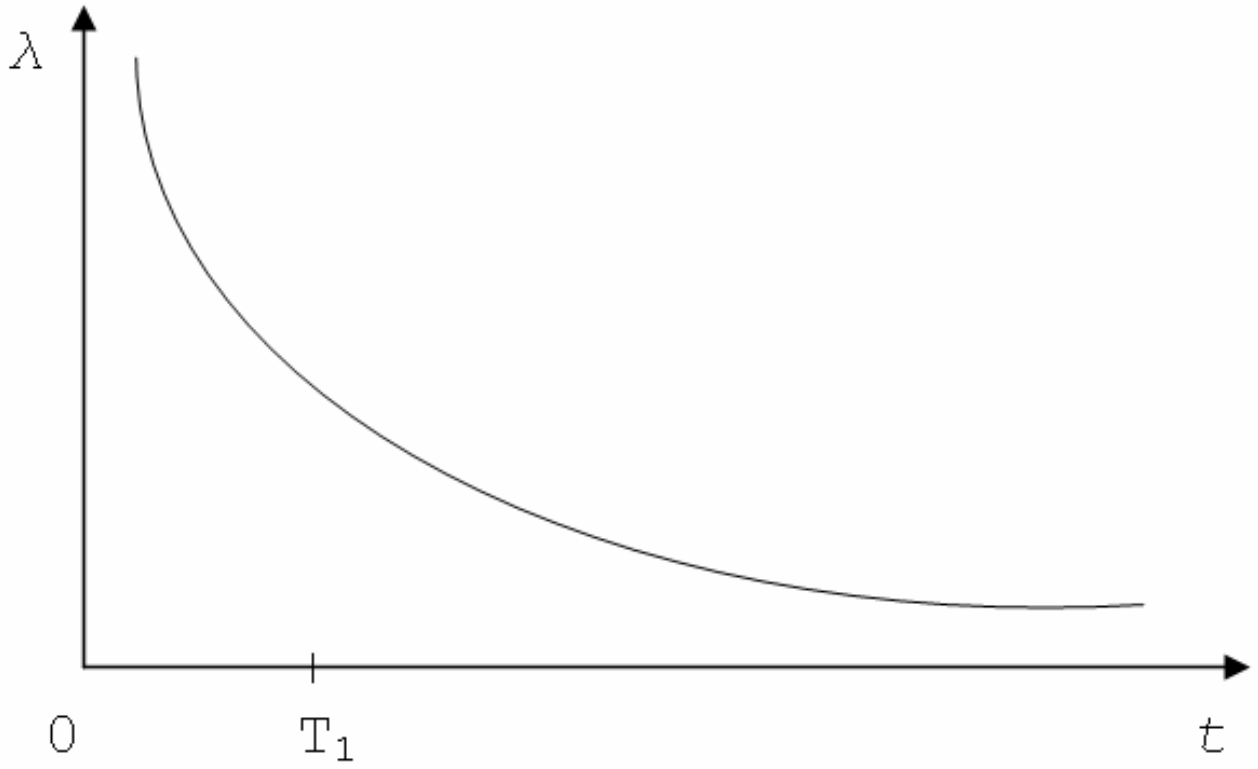


Рис. 7. Зависимость интенсивности отказов от времени для программного обеспечения АСУ.

В отличие от предыдущего графика, здесь можно выделить только два участка, и то достаточно условно ввиду большой пологости кривой:

1. От момента создания системы до момента времени T_1. Данный участок характеризуется уменьшением значения интенсивности отказов во времени и называется отладкой программного обеспечения;

2. От момента времени T_2 и до момента морального устаревания программы. В отличие от аппаратного обеспечения, ошибки в сложных программных комплексах находят в течение всего жизненного цикла программы. Поэтому данная кривая более пологая и никогда не достигает оси абсцисс. Ввиду отсутствия в программе механических компонент, программа может устареть только морально, но не физически. Программы, написанные многие десятилетия назад, будучи запущены на соответствующем им аппаратном обеспечении, работают, не проявляя тенденции к росту значения интенсивности отказов.

Еще одним не менее важным показателем надежности является вероятность безотказной работы $P_0 = e^{-\lambda t}$. Этот показатель определяется в техническом задании на разработку системы и задает вероятность, с которой система, имеющая интенсивность отказов λ, должна функционировать по прямому

назначению непрерывно и безотказно в течение времени t. Время t, входящее в данную формулу, различно для систем разного назначения. Например, для системы управления ядерным реактором это время может составлять годы, а для самолета, время непрерывного полета которого ограничено несколькими часами – соответственно, единицы или десятки часов.

Виды нарушения работоспособности программ и методы их парирования:

1. Зацикливание;
2. Останов (прекращение решения задачи);
3. Снижение темпа решения задачи;

Методом парирования первых трех видов нарушения работоспособности программ является контроль времени решения задачи. Для каждого исполняемого программного модуля задается максимально возможное время его решения, и операционная система жестко контролирует превышение этого времени. Если время решения задачи становится больше заданного предельного времени, программа снимается с обработки.

4. Пропуск программного модуля;
5. Нарушение внутрипрограммных переходов;

Методом парирования этих двух видов нарушения работоспособности программ является применение ключевых кодов. Каждому программному модулю или логически законченному участку программы присваивается уникальный ключевой код. Создается контрольный алгоритм выполнения программного комплекса или программного компонента, содержащий не сами программные модули или участки программы, а их ключевые коды. Контрольный алгоритм выполняется параллельно с основной программой, причем после каждого шага выполнения алгоритма и программы производится сравнение текущих значений ключевого кода контрольного алгоритма и ключевого кода программного модуля или участка программы. При их несовпадении предпринимаются меры по исправлению возникшей ситуации.

6. Нарушение взаимной синхронизации процессов. Данное нарушение парируется периодической посылкой контрольных сообщений с целью установления факта работоспособности механизмов взаимной синхронизации процессов. При выявлении каких-либо нарушений также предпринимаются меры по исправлению возникшей ситуации;

7. Искажение или потеря накопленной информации. Методом парирования этого вида нарушения работоспособности программ является контрольное суммирование. Контрольная сумма представляет собой сквозную сумму по модулю два всех слов информационного массива, записанную за последним словом массива. Таким образом, если просуммировать все слова информационного массива, уже имеющего контрольную сумму, по

модулю два, то в результате получится нулевой код. При искажении или потере информации массива, нулевого кода при контрольном суммировании уже не получится.

§ 2. Устойчивость вычислительного процесса

Любая автоматизированная система управления разрабатывается для поддержки принятия решений руководителями различных уровней управления, а также для непосредственного управления физическими объектами. По этой причине цена последствий отказов в функционировании системы может быть достаточно велика. В то же время нельзя забывать, что ни одна автоматизированная система не существует в своей первоначальной модификации без сложно переплетающихся ошибок проектирования. Для того, чтобы можно было учесть вероятные последствия отказов системы управления, теория надежности рассматривает вопросы устойчивости функционирования систем. Под устойчивостью в этом случае понимается свойство адаптации системы к изменениям во внешней среде. Понятно, что влияние внешней среды может вызвать изменения параметров системы, не предусмотренные нормами эксплуатации. Такие изменения называются кризисом. Если это изменение происходит в определенных границах, то система способна адаптироваться. В рамках способности системы к адаптации существует понятие живучести. Живучесть – это свойство системы выполнять заданные целевые функции при неблагоприятных воздействиях внешней среды, не предусмотренных нормами эксплуатации. Если изменение параметров системы превышает возможности её адаптивного развития, то система теряет устойчивость.

Различают три типа кризисов:

1. Критическая ситуация. Этот тип кризиса предусматривает адаптивное развитие системы. Например, занос автомобиля на скользкой дороге, при котором водитель сумел справиться с управлением и, не останавливаясь, вывел машину из заноса. Параметры системы (в данном случае, сцепление колес с дорожным покрытием) изменились на величину, не предусмотренную нормами эксплуатации. В то же время, благодаря грамотным действиям водителя, система успешно адаптировалась, и автомобиль поехал дальше;

2. Мягкая потеря устойчивости. Из названия понятно, что в этом случае возможности адаптивного развития системы превышены, но возможен возврат системы в работоспособное состояние. Примером может служить занос автомобиля на скользкой дороге с выносом в кювет, заполненный снегом. Никаких разрушений машина при этом не получила, но для продолжения движения необходима какая-либо внешняя сила, спо-

собная вытащить автомобиль из кювета (тягач или взвод солдат);

3. Катастрофа (жесткая потеря устойчивости). Этот тип кризиса сопровождается разрушением системы.

Устойчивость вычислительного процесса АСУ определяется:

1. Надежностью и живучестью технических средств;
2. Защищенностью информации;
3. Надежностью программного обеспечения;
4. Квалификацией операторов систем.

Для обеспечения живучести автоматизированных систем управления служит специализированное программное обеспечение, которое включает в себя:

1. Программное обеспечение резервирования элементов автоматизированной системы;
2. Программное обеспечение самодиагностики на всех уровнях иерархии автоматизированной системы;
3. Программное обеспечение организации альтернативных маршрутов передачи информации для обхода отказавших элементов системы;
4. Программное обеспечение информирования персонала об ошибках функционирования системы;
5. Программное обеспечение локализации ошибок функционирования системы;
6. Программное обеспечение самовосстановления элементов системы.

Современный подход к обеспечению надежности автоматизированных систем управления основан на сплошном непрерывном контроле функционирования системы и краткосрочном пошаговом прогнозировании состояния устройств с соответствующей реакцией на каждом шаге. Такой подход называется сценарным и реализует принцип «не реагировать и исправлять, а предвидеть и упреждать». В то же время, говоря о живучести систем, необходимо знать и учитывать особенности поведения систем в аварийных ситуациях:

1. Изменение целей функционирования системы – главной целью становится сохранение целостности самой системы;
2. Активное отклонение многих регулируемых параметров от нормальных пределов в целях самосохранения системы;
3. Изменение структуры внешних связей.

§ 3. Методы обеспечения надежности

Основным методом обеспечения надежности является введение избыточности. Различают три вида избыточности:

1. Информационная избыточность;
2. Временна́я избыточность;

3. Физическая избыточность:
- аппаратное резервирование;
- репликация.

Разберем указанные виды избыточности. Информационная избыточность заключается в введении дополнительных избыточных элементов в информационные единицы. Ввиду того, что такие элементы вводятся на уровне двоичных кодов, реализация информационной избыточности осуществляется с помощью корректирующих машинных кодов.

Идея корректирующих машинных кодов состоит в том, что в n-разрядном двоичном коде используются не все $N = 2^n$ возможных комбинаций двоичных цифр (от 00…00 до 11…11), а только их некоторая разрешенная часть $N_P < N$. Остальные $N - N_P = N_З$ комбинаций являются запрещенными. Ошибки в двоичных кодах проявляются путем замены одного или нескольких двоичных разрядов на противоположные значения. Если в результате ошибок рассматриваемая кодовая комбинация перешла в разряд запрещенных, то тем самым обнаруживается наличие ошибок. Если рассматриваемая кодовая комбинация в результате ошибок перешла в другую разрешенную комбинацию, то такая ошибка не обнаруживается при данном уровне избыточности. Чем больше избыточность, тем выше корректирующая способность данного избыточного кода.

Таким образом, из n разрядов двоичного кода l разрядов являются информационными, а k – избыточными, или их еще называют контрольными. Характеристики избыточности:

1. Относительная избыточность первого рода – это отношение количества контрольных разрядов к количеству информационных разрядов некоторой кодовой комбинации:

$$0 \le \frac{k}{l} = \frac{k}{n-k} < \infty;$$

2. Относительная избыточность второго рода – это отношение количества контрольных разрядов к общему количеству разрядов некоторой кодовой комбинации: $0 \le \frac{k}{n} = \frac{k}{l+k} < 1$;

3. Кодовый вес – это количество единиц в некоторой кодовой комбинации. Например, число 11010110001_2 имеет кодовый вес $\omega = 6_{10}$;

4. Кодовое расстояние – это количество разрядов, в которых не совпадают двоичные цифры двух кодовых комбинаций. Кодовое расстояние определяется поразрядным сложением по модулю два с последующим определением кодового веса полученной суммы. Например:

$$\begin{array}{r} 11010110001 \\ \oplus\ 10111011010 \\ \hline 01101101011 \end{array}$$

Кодовый вес полученной суммы, а, следовательно, и кодовое расстояние двух кодовых комбинаций $\omega = d = 7_{10}$;

5. Минимальное кодовое расстояние двоичного кода – это самое малое кодовое расстояние, возможное между двумя любыми разрешенными кодовыми комбинациями в этом коде.

Любой код, обладающий ненулевой избыточностью, называется корректирующим вне зависимости от того, позволяет ли он только обнаруживать, или ещё и исправлять ошибки. Корректирующие коды бывают:

1. Посылочные корректирующие коды. Эти коды применяются для контроля правильности передачи и хранения информации. В свою очередь, посылочные коды могут быть:

а) Коды, обнаруживающие ошибки;

б) Коды, исправляющие ошибки;

2. Арифметические корректирующие коды. Эти коды применяются для контроля правильности выполнения арифметических операций.

В обычном безизбыточном двоичном не корректирующем коде минимальное кодовое расстояние $d_{min} = 1$. Корректирующий код с $d_{min} = 2$ не может исправить даже одиночную ошибку, так как любая запрещенная комбинация является равноотстоящей на единицу от двух разрешенных. Однако такой код широко применяется в вычислительной технике и называется кодом с контролем по четности (или нечетности). При $d_{min} = 3$ одиночная ошибка приводит к такой запрещенной комбинации, которая будет ближе находиться к исходной разрешенной комбинации и дальше от любой другой разрешенной комбинации. Задача исправления ошибки сводится только к поиску того единственного разряда, инвертирование которого приведет запрещенную комбинацию в разряд разрешенных. Когда в кодовой комбинации могут появиться ошибки любой кратности $i \leq n$, то для обнаружения такой ошибки потребуется корректирующий код с минимальным кодовым расстоянием $d_{min} = i + 1$, а для исправления этой ошибки потребуется корректирующий код с минимальным кодовым расстоянием $d_{min} = 2i + 1$.

Методика исправления одиночной ошибки основана на идее обнаружения. При этом считается, что в кодовых комбинациях возможны только одиночные ошибки. Примером корректирующего кода, способного исправлять одиночные ошибки, может служить код Хэмминга.

В коде Хэмминга, как и в любом избыточном корректирующем коде, в n-разрядной кодовой комбинации имеется l информационных разрядов и k контрольных, причем $n = l + k$. Контрольные разряды располагаются в разрядах кодовой комбинации, номера которых кратны целой степени числа два. В остальных разрядах кодовой комбинации располагаются информационные разряды. Сумма единиц всех разрядов кодовой комби-

нации, включая контрольные, двоичные номера которых имеют единицы в одинаковых разрядах, должна быть четной. Требуемое количество контрольных разрядов равно $k \geq \log_2(n + 1)$, округленное до ближайшего большего целого. Пример: Необходимо сформировать код Хэмминга шестнадцатеричного числа $ABCD_{16}$.

$ABCD_{16} = 1010\ 1011\ 1100\ 1101_2$

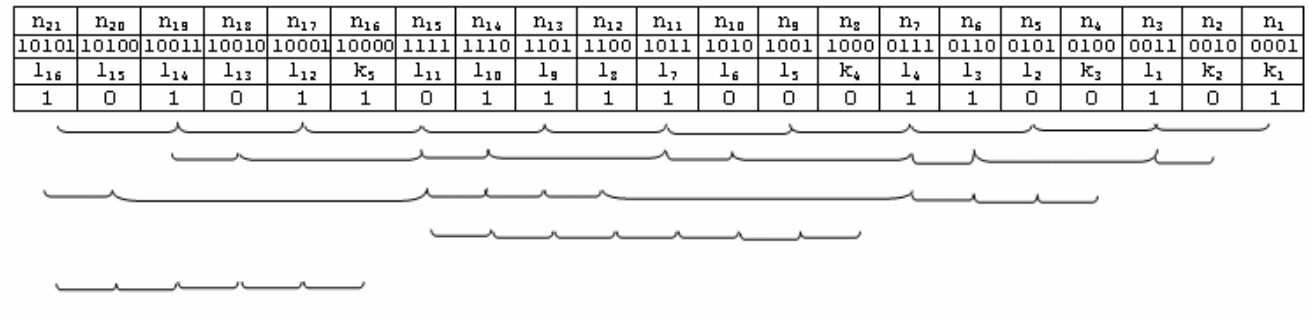


Рис. 8. Формирование кода Хэмминга.

На рисунке видно, что:

1. В формировании первого контрольного разряда участвуют 1, 2, 4, 5, 7, 9, 11, 12, 14 и 16 информационные разряды. Обозначим сумму по модулю два всех этих разрядов, включая контрольный, как E_1;

2. В формировании второго контрольного разряда участвуют 1, 3, 4, 6, 7, 10, 11, 13 и 14 информационные разряды. Обозначим сумму по модулю два всех этих разрядов, включая контрольный, как E_2;

3. В формировании третьего контрольного разряда участвуют 2, 3, 4, 8, 9, 10, 11, 15 и 16 информационные разряды. Обозначим сумму по модулю два всех этих разрядов, включая контрольный, как E_3;

4. В формировании четвертого контрольного разряда участвуют 5, 6, 7, 8, 9, 10 и 11 информационные разряды. Обозначим сумму по модулю два всех этих разрядов, включая контрольный, как E_4;

5. В формировании пятого контрольного разряда участвуют 12, 13, 14, 15 и 16 информационные разряды. Обозначим сумму по модулю два всех этих разрядов, включая контрольный, как E_5.

При отсутствии ошибок, то есть когда сумма единиц всех разрядов кодовой комбинации, включая контрольные, двоичные номера которых имеют единицы в одинаковых разрядах, четная, значения сумм E_i равно нулю. Число, составленное из этих сумм, $E_5E_4E_3E_2E_1$, называется корректирующим числом Хэмминга, и при отсутствии ошибок оно равно нулю. При возникновении одиночной ошибки, то есть при инвертировании значения какого-либо одного разряда кодовой комбинации, суммы по модулю два E_j, в формировании которых участвует данный разряд, по-

меняют свое значение с нуля на единицу. Тогда корректирующее число Хэмминга $E_5E_4E_3E_2E_1$ будет равно номеру ошибочного разряда кодовой комбинации. Для исправления возникшей ошибки достаточно будет инвертировать значение этого разряда.

Если к рассмотренному нами коду Хэмминга добавить еще контрольный разряд по нечетности, то получится модифицированный код Хэмминга, позволяющий не только устранять одиночные ошибки, но и обнаруживать двойные ошибки.

Корректирующее число Хэмминга $E_kE_{k-1}\ \ldots\ E_2E_1$	Контрольный разряд по нечетности	Вывод
нули	0	Нет ошибок
≠ 0	1	Одиночная ошибка
≠ 0	0	Двойная ошибка
нули	1	Ошибка в контрольном разряде по нечетности

Наряду с корректирующими кодами, позволяющими исправлять ошибки, в современной вычислительной технике широко используются корректирующие коды, обнаруживающие множественные ошибки, но не способные их исправлять. Примером такого кода может служить код CRC (Cyclic Redundancy Check – контроль циклическим избыточным кодом). Этот код представляет собой остаток от двоичного деления сообщения, взятого в виде длинной последовательности единиц и нулей независимо от границ байтов, на порождающий многочлен $X^{16} + X^{12} + X^5 + X + 1$, называемый полиномом CRC-CCITT.

Коды CRC с полиномом CRC-CCITT обнаруживают следующее:

1. Все одиночные и двойные поразрядные ошибки;
2. Все ошибки нечетного числа разрядов;
3. Все пакеты ошибок из 16 и менее разрядов, при этом длина пакета ошибки равна числу разрядов между первым и последним ошибочным разрядом включительно, причем в промежутке между этими разрядами может быть любое число ошибочных разрядов;
4. До 99.998% всех остальных ошибок.

Благодаря таким возможностям обнаружения ошибок применение 16-разрядных кодов CRC обычно ограничивается сообщениями длиной менее 4 Кбайт, поскольку существует достаточно вариантов искажения сообщений длиной более 4 Кбайт, и поэтому улавливание только 99.998% ошибок считается неприемлемым.

Для сообщений длиной более 4 Кбайт используются 32-разрядные коды CRC. Подобные коды CRC обнаруживают 99.999999977% всех ошибок. Порождающий многочлен для 32-разрядных кодов CRC, используемый в сетях Ethernet и Token

Ring, представляет собой полином $X^{32} + X^{26} + X^{23} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$.

Посылочные корректирующие коды нельзя непосредственно применить для контроля правильности арифметических операций, так как у них не существует однозначной связи между контрольными разрядами исходных операндов и контрольными разрядами результата арифметической операции. Идея обнаружения арифметических ошибок заключается в следующем: остаток от деления результата арифметической операции на какое-либо число равен результату той же операции над остатками от деления операндов на то же самое число. Например, возьмем число, на которое будем делить, равное трем, числа $X_1 = 13$ (остаток от деления числа 13 на три равен 1), $X_2 = 16$ (остаток от деления числа 16 на три равен 1). $X_1 + X_2 = 13 + 16 = 29$ (остаток от деления числа 29 на три равен 2). Сумма остатков чисел X_1 и X_2 также равна 2, что говорит о правильности выполнения данной арифметической операции. Если перемножить указанные числа, то получится $X_1 \cdot X_2 = 13 \cdot 16 = 208$ (остаток от деления числа 208 на три равен 1). Произведение остатков чисел X_1 и X_2 также равно 1, что говорит о правильности выполнения данной арифметической операции.

Рассмотренный метод справедлив только для операций сложения и умножения. Операция вычитания заменяется операцией сложения со вторым операндом, взятым в дополнительном коде, а операция деления заменяется последовательностью операций сложения и сдвига. В качестве модуля для деления нельзя брать число, равное целой степени основания системы счисления, в которой производятся вычисления. Для применения в ЭВМ рекомендуется брать число, на которое делят операнды, равное пятнадцати.

Временна́я избыточность заключается в повторении передачи данных или выполнения какого-либо процесса с последующим сравнением результатов. При совпадении полученных результатов передачи данных или работы процесса, эти результаты считаются достоверными.

§ 4. Физическая избыточность

Первым и традиционным методом обеспечения надежности с помощью физической избыточности является аппаратное резервирование. Аппаратное резервирование заключается в дублировании аппаратуры системы с целью обеспечения возможности оперативного переключения на резервное оборудование при отказе основного. В зависимости от заданных параметров безопасности системы, переключение на резерв может быть автома-

тическим или ручным. Различают две схемы резервирования систем:

1. Помашинное резервирование. При помашинном резервировании резервируется вся система целиком. В случае выхода из строя какого-либо элемента основной системы, вся основная система выводится из действия, и происходит переключение целиком на работу резервной системы. В данной схеме имеется единственная нерезервируемая точка отказа - устройство переключения на резерв, которое само по себе должно обладать повышенной надежностью во избежание снижения показателей надежности системы в целом;

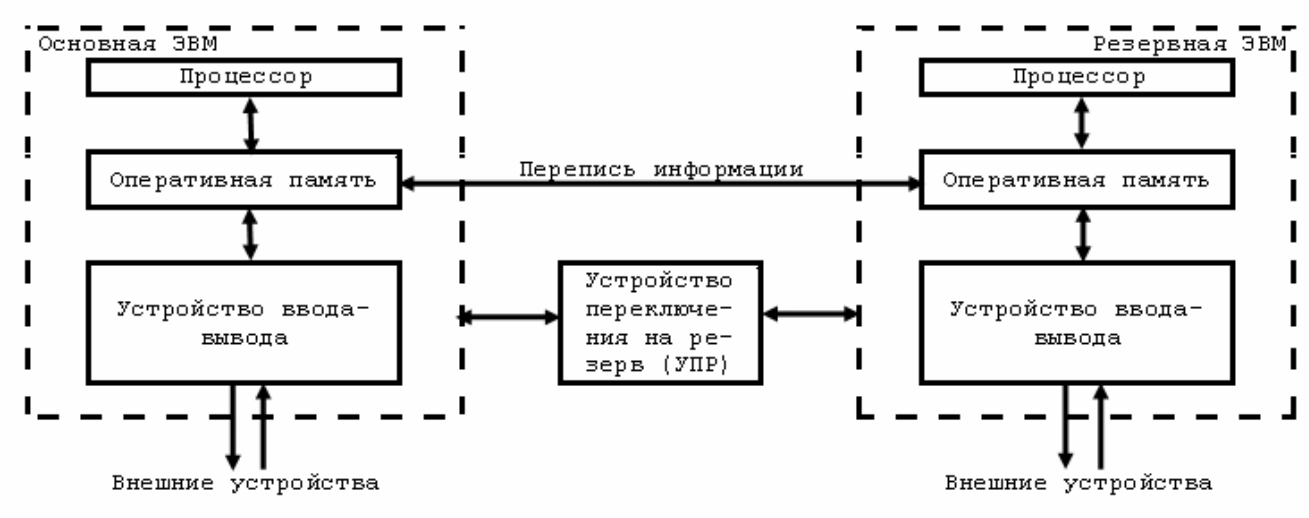


Рис. 9. Помашинное резервирование

2. Помодульное резервирование. При помодульном резервировании резервируются отдельные подсистемы. В случае выхода из строя какого-либо элемента основной системы, основная подсистема, в которой находится отказавший элемент, выводится из действия, и происходит переключение на работу резервной подсистемы. В данной схеме на каждую пару взаимно резервируемых подсистем требуется нерезервируемое устройство переключения на резерв. Однако, несмотря на наличие бóльшего количества нерезервируемых элементов, помодульное резервирование обеспечивает лучшую надежность по сравнению с помашинным резервированием. Применение этой схемы ограничивает её повышенная сложность, обуславливаемая необходимостью организации альтернативных маршрутов передачи информации для обхода выводимых из действия подсистем.

Необходимо заметить, что при резервировании средств вычислительной техники обязательно производится постоянная перепись информации оперативной памяти основной системы в оперативную память резервной с целью обеспечения возможности продолжения вычислений на резервном оборудовании после отказа основного.

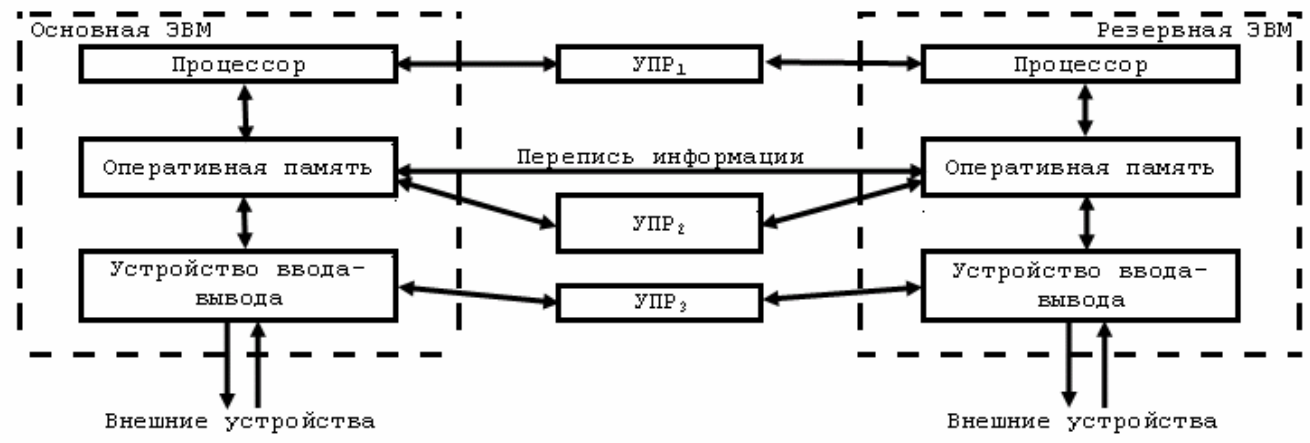


Рис. 10. Помодульное резервирование

В прошлом веке в системах, построенных на дискретных полупроводниковых элементах, ограниченно применялось ещё и поэлементное резервирование, являющееся вариантом помодульного резервирования, с тем лишь отличием, что уровень резервируемой подсистемы в этой схеме понижался до отдельных дискретных элементов. Поэлементное резервирование обеспечивало наилучшую надежность по сравнению с рассмотренными схемами резервирования, но ввиду большой сложности реализации в настоящее время нигде не применяется.

В реальных системах, как правило, используются смешанные схемы резервирования, когда вся система резервируется помашинно, отдельные наиболее ответственные подсистемы резервируются помодульно, а вспомогательные подсистемы, непосредственно не влияющие на выполнение системой основных возложенных на неё задач, не резервируются вообще.

Распределенные системы, обладая практически неограниченными ресурсами, позволяют создавать системы высокой степени надежности за счет многократного резервирования ресурсов. Устойчивые хранилища данных, будучи распределены по сети ЭВМ, создают условия абсолютной надежности хранения данных, способной противостоять даже физическому уничтожению отдельных ЭВМ.

Устойчивые хранилища данных строятся на основе использования идеологии RAID-систем (Redundant Array of Inexpensive Disks – матрица недорогих дисков с избыточностью). Исторически, еще во времена ЭВМ первых поколений, RAID-системы представляли собой несколько дисковых накопителей, подключенных к одному процессору. Даже сегодня можно встретить RAID-системы, базирующиеся на одной локальной ЭВМ. Однако, распределенная система, физически представляющая собой сеть из нескольких ЭВМ, позволяет по-новому подойти к реализации RAID-системы. Общеизвестно, что любой компьютер обязательно имеет дисковый накопитель. Распределенная система, представляясь пользователю как виртуальная локальная ЭВМ, таким образом, имеет столько дисковых накопителей,

сколько имеется ЭВМ в сети (или больше, если отдельные ЭВМ сети имеют более одного дискового накопителя). Поэтому ничего не мешает создать в распределенной системе RAID-систему на основе имеющихся на каждой ЭВМ сети дисковых накопителей. RAID-системы стандартизованы и классифицируются по шести уровням:

1. Уровень 0 – чередование дисков. Массив данных делится на блоки размером 64 Кбайт и равномерно распределяется по всем дискам системы (от двух до 32 дисков). Преимущество данного уровня заключается в повышении производительности системы за счет обеспечения одновременного доступа к различным блокам хранилища данных. В то же время, нулевой уровень не дает выигрыша в надежности хранения данных ввиду отсутствия избыточности;

2. Уровень 1 – зеркализация дисков. Разделы диска копируются на другом физическом диске. Преимущество первого уровня в высокой надежности – всегда существует резервная копия данных на другом физическом носителе;

3. Уровень 2 – чередование дисков с записью кода коррекции ошибок. Этот уровень подобен нулевому уровню, но для каждого байта создается корректирующий код Хэмминга, который записывается на отдельный диск. Преимущества второго уровня те же, что и в нулевом уровне, а недостаток заключается в больших объемах корректирующих кодов;

4. Уровень 3 – чередование дисков с контролем по четности. Этот уровень также подобен нулевому уровню, но для каждого байта создается разряд контроля по четности, который записывается на отдельный диск. По сравнению со вторым уровнем, третий уровень обеспечивает более высокий коэффициент использования дискового пространства;

5. Уровень 4 – чередование дисков большими блоками. Четвертый уровень является комбинацией первого и третьего уровней, то есть производится дублирование всего массива данных на другой диск с контролем по четности. Этот уровень применяется для резервирования больших массивов данных;

6. Уровень 5. Данный уровень не имеет собственного названия и представляет собой комбинацию второго, третьего и четвертого уровней, реализованную фирмой Microsoft.

Физическая избыточность процессов осуществляется путем объединения в группу нескольких одинаковых процессов. Любое сообщение, адресованное одному из процессов группы, получают все процессы этой группы. Если один из процессов группы перестает работать, его место занимает другой процесс группы. Один процесс может входить одновременно в несколько групп. Классификация групп процессов:

1. По составу процессов:

а) Статическая группа процессов. Данная группа имеет постоянный состав процессов;

б) Динамическая группа процессов. Состав процессов этой группы может изменяться в зависимости от текущей задачи, решаемой системой;

2. По иерархической структуре:

а) Одноранговая группа процессов. Недостаток одноранговой группы заключается в сложности синхронизации процессов;

б) Иерархическая группа процессов. Иерархическая группа процессов предусматривает наличие в ней ведущего процесса. Недостаток иерархической группы заключается в наличии единственной нерезервируемой точки отказа – ведущего процесса, при останове которого вся группа теряет работоспособность.

Ввиду того, что реплики хранилища данных существуют, как правило, в виде объектов, то для обеспечения надежности они также могут объединяться в группы. Для обеспечения надежности функционирования группы процессов, в группе должно быть $3m + 1$ процессов, где m – прогнозируемое число неправильно работающих процессов. Неправильно работающие процессы в группе выявляются по мажоритарному принципу, когда сравниваются отклики всех процессов группы на одно сообщение, и процессы, дающие отклики, не совпадающие с откликами большинства процессов в группе, признаются дефектными (неправильно работающими).

При взаимодействиях клиента с сервером могут возникнуть следующие ситуации:

1. Клиент не в состоянии обнаружить сервер. Данная ситуация может возникнуть в двух случаях:

а) Сервер отключен;

б) При длительном отсутствии запросов от клиента сервер при перезагрузке после отказа или сбоя может создать новый интерфейс с новыми серверными заглушками.

Выхода из ситуации, когда клиент не в состоянии обнаружить сервер, без полной потери прозрачности не существует;

2. Потеря сообщения с запросом от клиента к серверу. Данная ситуация разрешается путем контроля времени получения ответа от сервера. При превышении заданного предельного времени получения ответа клиент посылает повторное сообщение. Для предотвращения повторного выполнения неидемпотентных операций, все повторные сообщения нумеруются;

3. Сбой сервера после получения запроса от клиента. В этой ситуации операционная система сервера посылает клиенту сообщение об ошибке. Проблема состоит в том, что операционная система рабочей станции не в состоянии понять, что

именно произошло – надо просто ждать ответа сервера или необходимо повторно посылать запрос. Существуют две методики действий по выходу из данной ситуации:

а) Клиент, получив сообщение об ошибке, ожидает перезагрузки сервера и повторяет запрос. Попытки клиента повторяются до тех пор, пока сервер не выдаст нужный клиенту ответ;

б) Клиент, получив сообщение об ошибке, отказывается от всяких дальнейших попыток связаться с сервером;

4. Потеря ответного сообщения от сервера к клиенту. В данной ситуации проблему решить можно также путем осуществления контроля времени получения ответа от сервера и в случае идемпотентных операций повторять запросы. В случае неидемпотентных операций сервер должен иметь средства учета запросов от каждого из клиентов и возможность идемпотентных ответов на повторные запросы клиентов;

5. Сбой клиента после посылки запроса на сервер. При сбое клиента после посылки запроса на сервер может оказаться, что операция сервером будет выполнена, но не окажется заказчика, ожидающего результата запроса. Ответы сервера, не имеющие адресата, а также порождаемые ими процессы, называются сиротами. Сироты могут связывать полезные ресурсы распределенной системы, снижая тем самым её производительность. Для борьбы с сиротами существуют четыре способа:

а) Истребление сирот. Перед посылкой запроса на сервер клиентская заглушка создает запись в специальном журнале на запоминающем устройстве, способном пережить перезагрузку. После перезагрузки клиента журнал анализируется, все сироты ищутся и уничтожаются. Данный способ имеет два недостатка:

- в случае генерации сиротами своих запросов на сервер, образуются внучатые сироты, которых этим способом вообще невозможно обнаружить;

- при разделе сети на фрагменты отказавшим маршрутизатором, истребить сирот в других фрагментах сети невозможно;

б) Реинкарнация. Время работы клиента разбивается на последовательно пронумерованные эпохи. После перезагрузки клиента, он путем широковещательной рассылки объявляет о начале новой эпохи. Все удаленные вычисления, проводимые на сервере по заказу этого клиента из прошлой эпохи, прекращаются, а все ответы на запросы из прошлой эпохи уничтожаются. Общий недостаток первых двух способов (истребления сирот и реинкарнации) заключается в возможности уничтожения сирот, заблокировавших какие-либо ресурсы системы. В этом случае снять блокировку без перезагрузки ресурса практически невозможно;

в) Мягкая реинкарнация. Время работы клиента разбивается на последовательно пронумерованные эпохи. После пере-

загрузки клиента, он путем широковещательной рассылки объявляет о начале новой эпохи. Когда приходит сообщение о смене эпох, сервер проверяет, происходят ли на нем какие-либо удаленные вычисления. Если да, то сервер пытается найти их владельца. Если владельца найти не удалось, вычисления прекращаются;

г) Истечение срока. Каждому клиентскому запросу назначается стандартная продолжительность его работы. Если за указанное время удаленный процесс не закончил свою работу на сервере, клиент должен в явном виде затребовать следующий срок. Если в отведенное время клиент не затребовал следующий срок, вычисления прекращаются. Недостаток последних двух способов (мягкой реинкарнации и истечения срока) заключается в отсутствии механизма борьбы с сиротами.

§ 5. Надежная групповая рассылка

Групповая рассылка, то есть отправка одного сообщения всем процессам группы, считается надежной, если можно гарантировать получение сообщения всеми правильно работающими членами группы. Если предположить, что все процессы, как передающие, так и приемные, безотказны, а используемые коммуникации не ориентированы на соединение, то используется следующий алгоритм: передающий процесс присваивает каждому пересылаемому сообщению последовательный номер. Если сообщения принимаются в том же порядке, что и передаются, процесс-получатель легко обнаруживает пропажу сообщения и запрашивает повторную передачу. В свою очередь, процесс-отправитель, не получивший в течение заданного предельного времени ожидания квитанцию о приеме сообщения, автоматически повторяет передачу сообщения. Этот алгоритм обладает плохой масштабируемостью, так как при большом количестве процессов-получателей процесс-отправитель может быть блокирован ответными сообщениями (квитанциями о приеме и запросами повторной передачи). Такая блокировка процесса-отправителя ответными сообщениями называется обратным ударом. Для решения проблемы масштабируемости надежной групповой рассылки существуют два способа:

1. Иерархическое управление обратной связью. Все процессы-получатели разбиваются на несколько подгрупп, размеры которых позволяют воспользоваться любой схемой групповой рассылки внутри подгруппы. Подгруппы организуются в виде дерева, в корне которого находится процесс-отправитель. В каждой подгруппе выделяется ведущий процесс, осуществляющий обмен информацией с процессом-отправителем. Ведомые процессы в подгруппе обмениваются только с ведущим процессом своей подгруппы. Ведущие процессы организуют свой буфер исто-

рии для осуществления повторных передач без связи с процессом-отправителем.

2. Неиерархическое управление обратной связью. В этом способе управления используется модель подавления откликов. Процесс-получатель никогда не подтверждает успешного приема сообщения, посылая только запросы на повторную передачу при потере сообщения. Отслеживание факта потери сообщения осуществляется процессом-получателем. Когда процесс-получатель обнаруживает, что потерял сообщение, он осуществляет групповую рассылку запроса на повторную передачу всем членам группы. Процесс-получатель, потерявший сообщение, но получивший от другого процесса группы запрос на повторную передачу того же сообщения, сам отказывается от посылки запроса на повторную передачу. Для предотвращения одновременной посылки запроса на повторную передачу несколькими процессами, посылка такого запроса осуществляется со случайной по времени задержкой. В идеале до процесса-отправителя доходит только один запрос на повторную передачу. Повторная передача при этом осуществляется также групповой рассылкой.

Недостаток неиерархического управления обратной связью заключается в необходимости обработки групповых рассылок запроса на повторную передачу и самой повторной передачи процессами, успешно получившими исходное сообщение. Данный недостаток преодолевается выделением процессов-получателей, потерявших сообщение, в отдельную группу, что, в свою очередь, сложно реализуемо в больших системах.

При наличии отказов в процессах может возникнуть ситуация, что обновление состояния работающего объекта произойдет в момент его отказа, и после перезагрузки процесса, он продолжит работать со старой не обновленной версией состояния объекта. Для предотвращения этой ситуации применяется атомарная групповая рассылка. Атомарность групповой рассылки означает, что операция обновления реплик, начатая до того, как произошла поломка одной из них, будет выполнена корректно на всех работающих репликах или не будет выполнена ни на одной из них. Атомарность групповой рассылки требует, чтобы при отказе одного из процессов группы, оставшиеся работающие процессы заключали новое соглашение о членстве в группе. После перезагрузки отказавшего процесса он должен снова войти в группу, так как иначе в его адрес не будет производиться групповая рассылка. Вход в группу требует приведение состояния объекта в соответствие с состоянием остальных членов группы.

Проблема атомарной групповой рассылки заключается в возможности одновременной рассылки обновлений и извещений об отказах процессов, что может привести к неоднозначной интерпретации процессами ситуации в системе. Атомарная

групповая рассылка, гарантирующая, что ситуация отказа процесса в ходе групповой рассылки представляется как ситуация того, что групповая рассылка вообще не производилась, называется виртуально синхронной групповой рассылкой. Виртуальная синхронность реализуется с помощью файлов представления группы, содержащих список процессов, входящих в группу. Использование файлов представления группы подобно использованию переменных синхронизации в распределенных хранилищах данных, то есть все групповые рассылки возможны только в период между изменениями файлов представления группы, и, наоборот, изменение файлов представления возможно только после завершения всех групповых рассылок. Порядок следования надежных групповых рассылок может быть четырех вариантов:

1. Неупорядоченная надежная групповая рассылка – это виртуально синхронная групповая рассылка, не дающая никаких гарантий порядка прихода сообщений к различным процессам;

2. Надежная групповая рассылка в порядке FIFO. Эта рассылка гарантирует доставку сообщений каждому из процессов группы в том же порядке, в котором они были отправлены;

3. Причинно упорядоченная надежная групповая рассылка. В этом варианте сообщения доставляются в порядке потенциальной причинной связи между ними. Потенциальная причинная связь означает, что если одно сообщение причинно предшествует другому, то независимо от того, посылались они одним отправителем или разными, получатель получит эти сообщения в порядке их причинного следования;

4. Полностью упорядоченная надежная групповая рассылка. Этот вариант рассылки означает, что независимо от того, как упорядочена доставка сообщений для отдельных процессов, сообщения доставляются всем членам группы в одинаковом порядке.

Большинство прикладных программ, использующих атомарную групповую рассылку, требуют также подтверждения готовности выполнить операцию. В простейшем случае распределенные подтверждения выполняются с помощью ведущего процесса, который осуществляет иерархическую групповую рассылку. Однако, при выполнении распределенных транзакций, простое подтверждение готовности выполнения транзакции не обеспечивает необходимого уровня надежности. Для обеспечения надежности распределенных подтверждений применяются протоколы двухфазного и трехфазного подтверждений.

Алгоритм двухфазного подтверждения:

1. Ведущий процесс рассылает всем процессам группы сообщение «голосование»;

2. Ведомый процесс группы, получив сообщение «голосование», сообщает ведущему процессу о своей готовности или неготовности выполнить свою часть транзакции;

3. Если все ведомые процессы готовы выполнить свои части транзакции, ведущий процесс рассылает всем процессам группы разрешение выполнить транзакцию. Если хотя бы один ведомый процесс не готов выполнить свою часть транзакции, ведущий процесс рассылает всем процессам группы запрет на выполнение транзакции;

4. Ведомый процесс, получив разрешение выполнить транзакцию, выполняет свою часть транзакции.

Недостаток алгоритма двухфазного подтверждения заключается в возможности блокировки механизма подтверждения при отказе процессов в ходе подтверждения готовности выполнить транзакцию. Протокол трехфазного подтверждения является неблокирующим и основан на выполнении всеми процессами, как ведущим, так и ведомыми, двух условий:

1. Не существует такого состояния, из которого возможен прямой переход в состояние готовности или неготовности выполнить транзакцию;

2. Не существует такого состояния, в котором невозможно принять итоговое решение, но возможен переход в состояние готовности выполнить транзакцию.

Отличие протокола трехфазного подтверждения от уже рассмотренного состоит в наличии дополнительной фазы предварительной готовности процессов.

Алгоритм трехфазного подтверждения:

1. Ведущий процесс рассылает всем процессам группы сообщение «голосование»;

2. Ведомый процесс, получив сообщение «голосование», сообщает ведущему процессу о своей предварительной готовности выполнить транзакцию;

3. При наличии предварительной готовности всех ведомых процессов, ведущий процесс рассылает всем процессам группы предварительное разрешение выполнить транзакцию;

4. Получив предварительное разрешение, ведомые процессы сообщают ведущему процессу о своей окончательной готовности выполнить транзакцию;

5. Только после наличия окончательной готовности всех ведомых процессов, ведущий процесс дает окончательное разрешение на выполнение транзакции;

6. Ведомый процесс, получив разрешение выполнить транзакцию, выполняет свою часть транзакции.

Достоинство протокола трехфазного подтверждения заключается в отсутствии тупиковых путей принятия решения о выполнении или прерывании распределенной транзакции.

§ 6. Восстановление после ошибок

В основе программной надежности лежит возможность восстановления процессов после ошибок. Существуют два способа восстановления процессов после ошибок:

1. Обратное исправление. Этот способ заключается в возврате системы из текущего ошибочного состояния к предыдущему исправному состоянию. Для этого периодически записывается состояние системы, называемое контрольной точкой или точкой восстановления. Примером обратного восстановления может служить временна́я избыточность. Недостаток обратного исправления состоит в значительной потере производительности системы при откате в предыдущее исправное состояние;

2. Прямое исправление. Этот способ заключается в попытке при возникновении ошибки перевести систему в новое исправное состояние. Примером прямого исправления может служить исправление ошибок кодами Хэмминга.

Для повышения производительности систем с обратным исправлением дополнительно используют протоколирование сообщений как процессом-отправителем, так и процессом-получателем. При этом отпадает необходимость повторной генерации сообщений после отката процесса к последней контрольной точке. Информация контрольных точек должна храниться в устойчивых хранилищах данных, способных пережить отказы и перезагрузки системы. Как правило, устойчивые хранилища данных строятся на основе энергонезависимых запоминающих устройств различных типов с двух или трехкратным резервированием на основе RAID-систем. В распределенных системах при работе взаимодействующих процессов на различных ЭВМ, совокупность локальных контрольных точек называется распределенным снимком состояния системы.

Ввиду того, что отказ является случайным событием, то для получения непротиворечивого распределенного снимка состояния системы необходима глобальная синхронизация создания контрольных точек. При этом запись информации о состоянии процессов, работающих на различных ЭВМ сети, в устойчивые хранилища данных, должна производиться одновременно. Для координированного создания контрольных точек применяется двухфазная блокировка как способ синхронизации.

Приложения:

В приложения вынесен материал, напрямую не касающийся тематики настоящего пособия, но без глубокого знания этого материала невозможно адекватно воспринимать рассматриваемые в пособии темы.

Приложение 1. Классификация сетей ЭВМ

§ 1. По принадлежности аппаратных средств.

1. Локальные сети. Все аппаратные средства локальной сети принадлежат одному владельцу – частному лицу или организации;

2. Глобальные сети. Аппаратные средства глобальной сети принадлежат различным владельцам.

Необходимо отметить, что далеко не все авторы делят сети на глобальные и локальные по признаку принадлежности аппаратных средств. В литературе, особенно переводной, часто можно встретить деление сетей на глобальные и локальные по признаку территориального охвата. При этом вводится еще понятие сетей кампусов. Эти авторы утверждают, что если сеть по своему территориальному охвату не выходит за пределы одного здания или одного предприятия, то это локальная сеть. Если сеть охватывает территорию какого-либо населенного пункта или района крупного города, то это сеть кампуса. Если же сеть по своим размерам выходит за пределы населенного пункта или района крупного города, то это глобальная сеть. Приведенная здесь позиция хороша для небольших западно-европейских государств, где выстрел из пушки грозит международным скандалом, так как снаряд может в случае промаха улететь на территорию соседней страны.

На просторах нашей необъятной Родины существуют ряд необъятных организаций, таких, как, например, Газпром. Не секрет, что эта фирма осуществляет разведку и добычу газа с дальнейшей его продажей иностранным потребителям. Все подразделения Газпрома оснащены современной вычислительной техникой. Газпром достаточно богатая организация, чтобы покупать и запускать свои собственные спутники связи. Теперь представим: компьютер на буровой вышке где-нибудь в тюменской тайге, другой компьютер в московском офисе Газпрома, третий – на газоперекачивающей станции на границе с Западной Европой, сверху летает спутник связи, объединяющий эти компьютеры в сеть ЭВМ. Все аппаратные средства, включая спутник связи, принадлежат одному владельцу – Газпрому. Спрашивается: какая это сеть – локальная или глобальная? По признаку территориального охвата данная сеть имеет размеры

нескольких Европ, и должна именоваться глобальной. Но кроме Газпрома этой сетью никто не пользуется! Поэтому логичнее такую сеть классифицировать все-таки по признаку принадлежности аппаратных средств и считать её локальной, тем более, что в нашем Отечестве организаций подобных Газпрому не одна, и не две, а значительно больше.

Другой пример: два студента живут в одном доме на одной лестничной клетке, оба имеют компьютеры и связываются друг с другом в режиме гипертерминала через городскую телефонную сеть общего пользования. Аппаратные средства такой сети принадлежат сразу трем владельцам: двум студентам и телефонной сети общего пользования, но по размерам эта сеть не выходит за пределы одного здания. Спрашивается: какая это сеть – локальная или глобальная. По признаку территориального охвата данная сеть имеет размеры одной лестничной клетки должна именоваться локальной. Но у нее нет одного хозяина! Поэтому логичнее такую сеть классифицировать все-таки по признаку принадлежности аппаратных средств и считать её глобальной.

§ 2. По иерархической структуре.

1. Одноранговые сети. В одноранговой сети отсутствует централизованное управление, все ресурсы сети равномерно распределены по сети и все ЭВМ сети имеют к ним одинаковые права доступа;

2. Серверные сети. В составе серверной сети имеются одна или несколько ЭВМ, осуществляющих управление доступом к ресурсам сети. Такие ЭВМ называются серверами. В свою очередь, если в сети есть ЭВМ, которая не является сервером, то эта ЭВМ будет называться рабочей станцией;

3. Гибридные сети. Все ЭВМ гибридной сети имеют одинаковые права доступа к ресурсам сети, находящимся на серверах.

Ресурсы сети могут быть информационными и аппаратными. К информационным ресурсам относятся разного рода программы и данные, архивы, документы и т.п., то есть всё, что хранится на различных носителях информации и что можно увидеть только с помощью компьютера. Аппаратные ресурсы, напротив, имеют конкретные физические размеры и вес, их можно переносить с места на место, подключать к компьютеру и отключать от него. Это различного рода накопители информации, печатающие устройства, сканеры и т.п.

В сети ЭВМ один сервер может управлять доступом к нескольким видам ресурсов. Такой сервер является сервером общего назначения и, как правило, не имеет своего специфического названия. Если сервер управляет доступом только к од-

ному конкретному ресурсу, то он носит название по виду того ресурса, доступом к которому управляет. Например: файловый сервер, сервер баз данных, сервер печати, почтовый сервер и т.д.

§ 3. По топологической структуре.

Топология – это метод соединения ЭВМ в сеть.

1. Сети с шинной топологией (шинные сети). Если все узлы сети подключить к единой среде передачи данных, как например, это делается в радиосети, то получится сеть с шинной топологией или, как еще говорят, сеть с общей шиной. В сетях с шинной топологией используется широковещательный метод доступа. Это означает, что одновременно передавать информацию может только одна ЭВМ сети, остальные узлы сети должны эту информацию принимать. Если по какой-либо причине две или более ЭВМ начнут одновременно передавать, то их данные смешаются и будет невозможно выделить, где чья информация. Такая ситуация называется коллизией, и для предотвращения её возникновения любая передача начинается со случайной по времени задержкой.

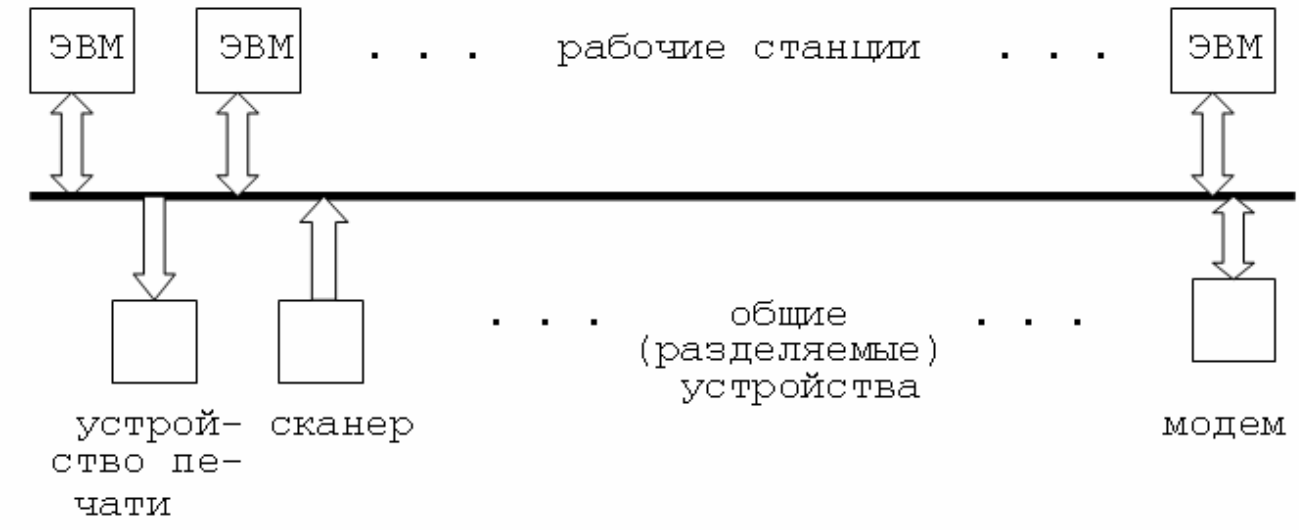


Рис. 11. Шинная топология сети

По причине того, что одновременно в сети может передаваться только одно сообщение, сети с шинной топологией обладают наименьшей из всех топологий пропускной способностью. В то же время, если имеется высоконадежная среда передачи данных, как, например, в радиосети, выход из строя любого узла сети не приводит к потере работоспособности сети в целом. Таким образом, сети с шинной топологией обладают наивысшей по сравнению с сетями других топологических структур надежностью.

Исторически развитие сетей ЭВМ начиналось как раз с сетей с шинной топологией. Общая шина этих сетей строилась на основе медного кабеля, низкая надежность которого как среды передачи данных обусловила тот факт, что в настоящее время сети с шинной топологией, построенные на основе медного кабеля, практически нигде не применяются. В то же время высокая надежность радиочастотного излучения как среды передачи данных позволяет радиосетям с шинной топологией бурно развиваться;

2. Сети со звездообразной топологией (звездообразные сети). Если все узлы сети подключить к одному центральному устройству, называемому концентратором или хабом, то получится сеть с звездообразной топологией. Для того чтобы получить выигрыш в пропускной способности сети, недостаточно просто соединить все узлы сети по схеме монтажного «или», то есть между собой. При этом получится псевдозвезда. Фактически это будет всё та же общая шина. Концентратор должен обеспечивать одновременный независимый обмен информацией между несколькими парами узлов сети. Таким образом, одновременно в сети может передаваться количество сообщений, равное числу пар узлов сети. Налицо существенное повышение пропускной способности сети со звездообразной топологией по отношению к пропускной способности сети с шинной топологией.

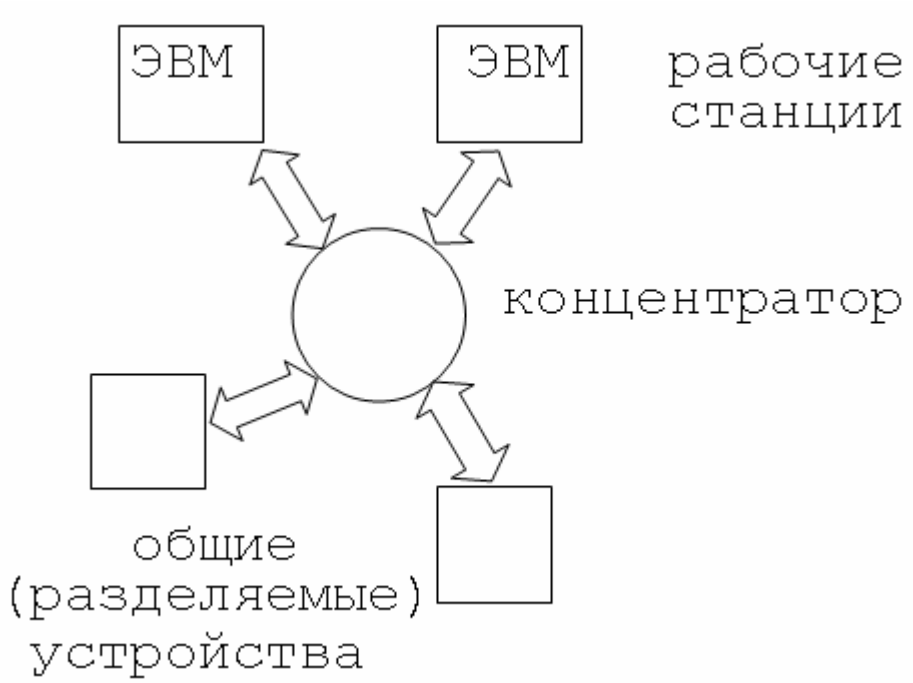


Рис. 12. Звездообразная топология сети

В то же время появление концентратора резко снижает надежность сети со звездообразной топологией. При выходе из строя этого устройства вся сеть теряет работоспособность. Таким образом, сети со звездообразной топологией обладают самой низкой надежностью по сравнению с сетями других топо-

логических структур. Надежность звездообразной сети сравнима с надежностью первых сетей с шинной топологией, построенных на основе медного кабеля, где повреждение кабеля также лишало работоспособности всю сеть. Но если в шинных сетях в настоящее время медный кабель практически не применяется, то в звездообразной сети вне зависимости от природы среды передачи данных концентратор как единственная точка отказа все равно присутствует;

3. Сети с кольцевой топологией (кольцевые сети). Сетевые адаптеры, используемые в сетях с кольцевой топологией, имеют раздельные вход и выход. Выход одного сетевого адаптера соединяется со входом другого до образования замкнутого кольца. В сетях с кольцевой топологией используется маркерный метод доступа. По кольцу в одном направлении запускается одно или несколько служебных сообщений, называемых маркерами. Передающая ЭВМ прикрепляет сообщение пользователя к маркеру и отправляет его по кольцу. Каждый узел сети проверяет адресную группу сообщения. Если сообщение адресовано не этому узлу, узел отправляет сообщение дальше по кольцу. Если сообщение адресовано данному узлу, узел забирает сообщение и отправляет пустой маркер дальше по кольцу.

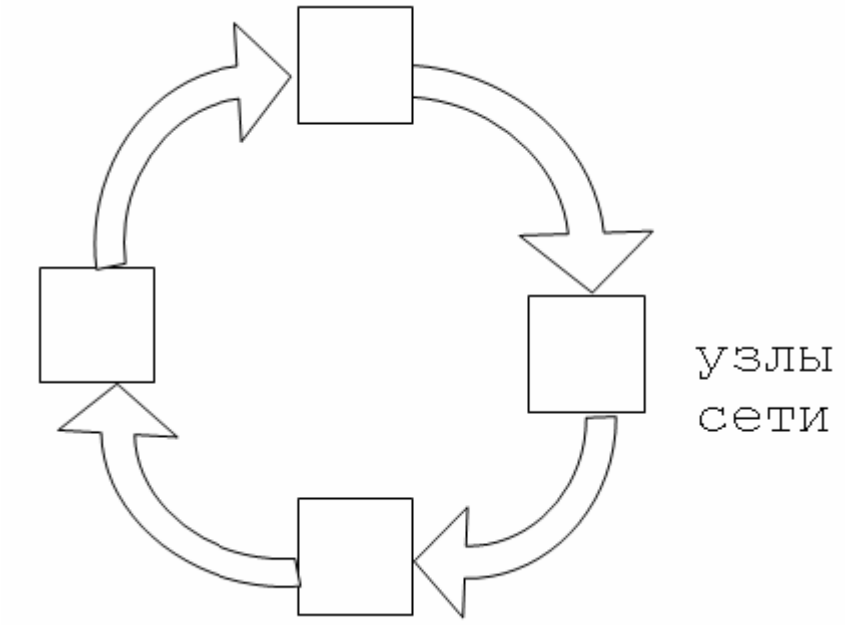


Рис. 13. Кольцевая топология сети

Возвращение на передающую ЭВМ пустого маркера означает успешную доставку сообщения адресату. Возвращение на передающую ЭВМ маркера с сообщением означает невозможность адресата принять сообщение (адресат выключен, перегружен, неисправен и т.п.). Потеря маркера означает повреждение сети.

Ввиду того, что сеть физически разделена на отдельные несвязанные участки между узлами сети, одновременно в сети может передаваться количество сообщений, равное числу узлов сети. Таким образом, при прочих равных условиях сети с

кольцевой топологией обладают наивысшей пропускной способностью по сравнению с сетями других топологических структур. При выходе из строя какого-либо узла сети или повреждении связывающего узлы участка сети сеть частично теряет работоспособность, так как только часть узлов, находящихся за местом повреждения становятся недоступными.

Если отранжировать сетевые топологии по пропускной способности и надежности, то результат будет следующим:

- пропускная способность:
- - наивысшая – кольцевая топология;
- - средняя – звездообразная топология;
- - наихудшая – шинная топология;
- надежность:
- - наивысшая – шинная топология;
- - средняя – кольцевая топология;
- - наихудшая – звездообразная топология.

Для повышения надежности сетей с кольцевой топологией применяют двойное кольцо. Маркеры в кольцах запускаются в противоположных направлениях. При выходе из строя какого-либо узла сети или повреждении связывающего узлы участка сети, на ближайших к месту повреждения узлах сети кольца соединяются друг с другом, и сеть превращается в одинарное кольцо. Отличием такого кольца от обычного является только несколько видоизмененная траектория движения маркера;

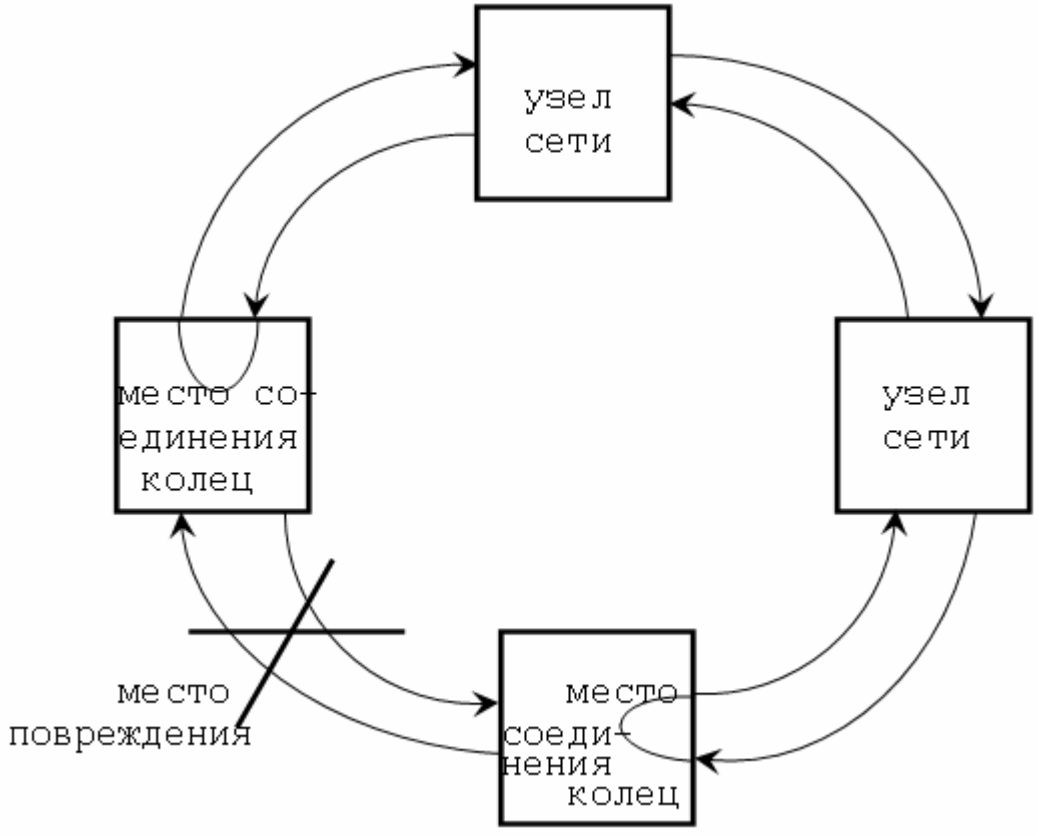


Рис. 14. Кольцевая топология повышенной надежности

4. Гибридные сети. Шинная, звездообразная и кольцевая топологии называются базовыми сетевыми топологиями. Гибридные сети представляют собой любую комбинацию сетей с базовыми сетевыми топологиями. Например, если в учебном заведении имеется несколько компьютерных классов с базовыми сетевыми топологиями каждый, то будучи соединенными между собой, в совокупности они будут являться гибридной по топологической структуре сетью.

§ 4. По среде передачи данных.

Среда передачи данных – это комплекс аппаратных средств, обеспечивающих обмен информацией между ЭВМ. Кроме непосредственно физической среды распространения сигнала (медного кабеля, оптоволокна, радиочастотного излучения и т.д.), в понятие среды передачи данных входят устройства, преобразующие сигналы соответствующих шин ЭВМ, к которым эти устройства подключаются (PCI, USB и т.д.), в сигналы, соответствующие природе физической среды их распространения и обратно. Такие устройства называются сетевыми адаптерами или сетевыми платами.

1. Сети, построенные на основе медного кабеля. Медный кабель, используемый при построении сетей ЭВМ, может быть трех видов: коаксиальный кабель, неэкранированная и экранированная витая пара.

Рассмотрим характеристики каждого из этих видов медного кабеля:

а) Коаксиальный кабель представляет собой центральную медную жилу, со всех сторон покрытую полиэтиленовой изоляцией. Поверх изоляции имеется электростатический экран из медной проволочной оплетки или медной фольги (а бывает – и того, и другого вместе). Электростатический экран покрыт защитной изолирующей оболочкой, предохраняющей кабель от повреждений. В быту с помощью коаксиального кабеля выполняют подключение телевизионных приемников к коллективной антенне. Коаксиальный кабель имеет низкую стоимость и прост в применении. Еще несколько лет назад для подключения коаксиального кабеля требовались элементарные паяльные навыки. В настоящее время для подключения коаксиального кабеля используются разъёмы, не требующие пайки. Пропускная способность коаксиального кабеля не превышает 10 Мбод.

Бод – это величина, используемая в технике связи для измерения пропускной способности. 1 бод = 1 бит в секунду. Кратная приставка «М» означает 2^{20}. Таким образом, 10 Мбод = $10 \cdot 2^{20}$ бит в секунду.

В вычислительной технике нельзя путать кратные приставки, имеющие одинаковое написание с кратными приставка-

ми, использующимися с единицами системы СИ. В системе СИ кратные приставки имеют основанием число десять. Например: к (кило) = 10^3, М (мега) = 10^6, Г (гига) = 10^9 и т. д. Применяются эти приставки только с единицами системы СИ. Например: кг – килограмм, МВт – мегаватт, Гл – гигалитр. В вычислительной технике, где применяется двоичная система счисления, кратные приставки имеют основанием число 2. Например: К = 2^{10}, М = 2^{20}, Г = 2^{30} и т.д. Применяются эти приставки с внесистемными единицами, использующимися в вычислительной технике, и не имеют названий. В разговоре такие приставки произносятся просто буквами «к», «м», «г» и т.д. Например: Кбод, Мбайт, Гбит.

Коаксиальный кабель имеет среднее затухание. Длина непрерывного участка кабеля без ретрансляции может достигать 1 км. Кабель слабо подвержен электромагнитным помехам и перехвату сигнала (несанкционированному съему информации). Последняя характеристика обусловлена конструкцией кабеля. Имеющийся вокруг центральной сигнальной жилы электростатический экран препятствует распространению электромагнитной энергии, наводимой сигнальной жилой, в окружающее пространство, и аналогично, препятствует проникновению электромагнитной помехи из окружающего кабель пространства внутрь кабеля к центральной сигнальной жиле;

б) Витая пара представляет собой два скрученных по всей длине проводника. Один проводник скрутки обязательно заземляется, другой является сигнальным. Современные кабели содержат четыре скрутки проводов в одной защитной оболочке и позволяют передавать четыре информационных сигнала. Второй провод каждой скрутки заземляется. В экранированной витой паре четыре скрутки проводов помещаются в электростатический экран, такой же, как и у коаксиального кабеля.

Стоимость витой пары сравнима со стоимостью коаксиального кабеля, подключение витой пары также не требует никаких специальных навыков. Однако в отличие от коаксиального кабеля, кроме непосредственно разъёма для подключения витой пары требуются специальные обжимные клещи. При наличии таких клещей монтаж разъёма на витую пару не представляет никаких затруднений.

Пропускная способность витой пары более чем на порядок выше, чем у коаксиального кабеля и достигает 155 Мбод. Коаксиальный кабель исторически был первой средой передачи данных, применявшейся в сетях ЭВМ. Но появление витой пары со значительно более высокой пропускной способностью при прочих равных условиях предопределило тот факт, что в настоящее время коаксиальный кабель в качестве среды передачи данных сетей ЭВМ практически не применяется. В то же время затухание витой пары на порядок сильнее, чем у коаксиально-

го кабеля. Длина непрерывного участка кабеля без ретрансляции не превышает 100 метров.

Неэкранированная витая пара сильно подвержена электромагнитным помехам и перехвату сигнала, вследствие чего применяется только в условиях отсутствия промышленных (индустриальных) помех, т.е. в быту и в офисах, удаленных от мест производства. При наличии промышленных (индустриальных) помех используется экранированная витая пара, обладающая слабой подверженностью электромагнитным помехам и перехвату сигнала (несанкционированному съёму информации) по той же причине присутствия электростатического экрана, что и коаксиальный кабель.

В условиях сильного уровня промышленных (индустриальных) помех применяют специальные виды медных кабелей (коаксиального и витой пары), имеющих двойную электростатическую экранировку. Кабель покрыт двумя электростатическими экранами, электрически не связанными друг с другом. Один из экранов подключается к общему проводу (массе) устройства, другой – к электротехническому заземлению, имеющемуся в помещении;

2. Сети, построенные на основе волоконно-оптического (оптоволоконного, стекловолоконного) кабеля. Волоконно-оптический кабель представляет собой вытянутую в нить стеклянную массу, в которую добавлены специальные пластификаторы для того, чтобы при изгибах нить не ломалась. Стеклянная нить покрыта слоем серебряной амальгамы, такой же, какой покрывают листовое стекло при изготовлении самых обычных зеркал. Поверх амальгамы имеется защитная оболочка, предохраняющая кабель от механических повреждений. Носителем информации в волоконно-оптическом кабеле является световое излучение. Волоконно-оптический кабель на порядок легче медного, что делает привлекательным его применение в тех местах, где необходимо прокладывать большое количество кабельных линий (подземные кабельные коллекторы, тоннели метрополитена, телевизионные и радиовышки и т.д.). В то же время, волоконно-оптический кабель значительно дороже медного, что ограничивает его применение в быту.

Пропускная способность волоконно-оптического кабеля достигает 2 Гбод, что неизмеримо выше, чем у витой пары. Кабель имеет очень низкое затухание. Длина непрерывного участка кабеля без ретрансляции может достигать 10 км. Волоконно-оптический кабель не подвержен электромагнитным помехам, т.к. электромагнитная помеха, имея другую природу, чем световое излучение, не оказывает никакого влияния на носитель информации в кабеле. Перехват сигнала (несанкционированный съём информации) невозможен, потому что любое повреждение слоя серебряной амальгамы, покрывающей стеклян-

ную жилу, приводит к полному выходу из строя всей линии связи, после чего съем информации с неё становится бессмысленным.

Основным препятствием применения волоконно-оптического кабеля является сложность соединения двух кусков светопроводящей стеклянной жилы. В целях предотвращения возникновения высокого затухания сигнала в месте соединения, сращивание двух жил кабеля должно производиться точно в стык (торец – в – торец) жилы, причем между торцами двух кусков жилы не должно оставаться воздушного промежутка. Чтобы обеспечить такое соединение, применяют специальные прецизионные (высокоточные) разъёмы, высокочастотную сварку и склейку специальными высокопрозрачными эпоксидными смолами. Любое из этих применений требует наличия специального оборудования и материалов, высокой квалификации персонала.

Несмотря на высокую стоимость и сложность применения, низкие вес волоконно-оптического кабеля и затухание, высокая пропускная способность и помехозащищенность предопределили широкое его применение в технике связи;

3. Радиосети. Носителем информации в радиосети служит электромагнитное излучение. Различают радиосети на традиционно низких частотах ниже 1 ГГц и радиосети в микроволновом диапазоне (микроволновые сети) на частотах свыше 1 ГГц. Оборудование радиосетей имеет высокую и сверхвысокую (в спутниковых системах) стоимость, но, приобретя такое оборудование, войти в сеть очень просто – достаточно только его включить и настроить на соответствующую частоту. Чаще всего оборудование радиосетей настраивается автоматически и не требует от пользователя никаких дополнительных действий, кроме простого включения.

Пропускная способность любой радиосети не превышает 10 Мбод, то есть такая же, как у коаксиального кабеля. Затухание радиочастотного сигнала зависит от длины волны. Чем короче длина волны, тем выше затухание сигнала. В технике связи принято делить радиочастотный диапазон на длинные, средние, короткие и ультракороткие волны. На длинных волнах и на низкочастотном участке средневолнового диапазона волн распространяется поверхностная волна, имеющая низкое затухание и без труда огибающая земной шар. В этом диапазоне осуществляются дальние и сверхдальние связи.

На высокочастотном участке средневолнового диапазона и на коротких волнах распространяется пространственная волна, испытывающая многократные отражения от ионосферы и от земной поверхности. Дальность связи в этом диапазоне сильно зависит от состояния ионосферы. При наличии хорошей отражательной способности ионосферы, что бывает достаточно редко, возможны дальние и сверхдальние связи.

В ультракоротковолновом диапазоне дальность связи равна дальности прямой видимости. По этой причине антенны ультракоротковолновых передатчиков (телевидение, УКВ-радиовещание, сотовая связь и т.д.) стараются размещать на высоких объектах, доминирующих над окружающим ландшафтом. Чем выше будет размещена такая антенна, тем на бóльшую дальность будет распространяться излучаемый ею сигнал.

В микроволновом диапазоне затухание зависит от метеоусловий. При наличии неблагоприятных метеоусловий (дождь, снег, туман, морось и т.п.) дальность связи может быть значительно меньше дальности прямой видимости.

Радиочастотное излучение сильно подвержено электромагнитным помехам и перехвату сигнала (несанкционированному съёму информации). Все радиосети являются сетями с шинной топологией, поэтому для несанкционированного съема информации достаточно просто включить радиоприемник, настроенный на рабочую частоту соответствующей радиосети, чтобы знать всё, что в этой сети происходит. Для уменьшения влияния электромагнитных помех и в целях защиты передаваемой информации применяют специальные методы модуляции и кодирования, шифрование сигнала.

Несмотря на столь низкие характеристики радиочастотного излучения, основным достоинством радиосетей является высокая мобильность узлов сети, которую не в состоянии обеспечить никакая другая среда передачи данных. Благодаря этому радиосети в настоящее время бурно развиваются и находят все более широкое применение;

4. Инфракрасные сети. Носителем информации в инфракрасной сети является инфракрасное излучение, такое же, какое применяется в пультах дистанционного управления различной бытовой аппаратуры (телевизоров, видеомагнитофонов, DVD-проигрывателях и т.д.). По этой причине стоимость оборудования для построения инфракрасных сетей достаточно низкая, а для того, чтобы войти в сеть, достаточно просто включить это оборудование, никакой настройки не требуется.

Пропускная способность инфракрасной сети на порядок ниже, чем даже у коаксиального кабеля и не превышает 1 Мбод. Затухание зависит от прозрачности атмосферы. Непрозрачные предметы являются непреодолимым препятствием на пути распространения инфракрасного излучения. На распространение инфракрасного излучения оказывают помехи источники интенсивной засветки и нагревательные приборы. Инфракрасные сети также, как и радиосети, являются сетями с шинной топологией, поэтому перехват сигнала (несанкционированный съём информации) не представляет затруднений.

Несмотря на столь низкие характеристики, инфракрасные сети находят широкое применение в качестве временных сетей

в закрытых помещениях при проведении различного рода заседаний, конференций, семинаров и тому подобных мероприятий, когда на первое место ставится фактор обеспечения безопасности передаваемой по сети информации, и по этой причине использование радиосетей нежелательно. Большинство моделей современных носимых и карманных персональных ЭВМ (ноутбуков и лэптопов) имеют инфракрасный порт, вследствие чего инсталляция (установка) инфракрасной сети из таких узлов не представляет никаких затруднений. Инфракрасное излучение за пределы закрытого помещения не выходит, обеспечивая тем самым без дополнительных затрат сохранение конфиденциальности передаваемой по сети информации. Единственным дополнительным оборудованием, требующимся для организации инфракрасной сети, являются ретрансляторы, устанавливаемые по углам помещения, в котором проводится данное мероприятие;

5. Гибридные сети. Гибридная сеть представляет собой любую комбинацию из выше рассмотренных. Например, для выхода с ноутбука в глобальную сеть Internet используется мобильный телефон с инфракрасным портом. В этом случае имеется инфракрасный участок передачи между ноутбуком и сотовым телефоном, радиочастотный участок передачи между сотовым телефоном и мобильным оператором, от мобильного оператора далее в сеть сигнал пойдет, скорее всего, по волоконно-оптической среде передачи. В этом случае комбинация сразу трех сред передачи данных в одной сети являет собой пример гибридной сети.

§ 5. По виду передаваемых сигналов.

1. Аналоговые сети. В аналоговой сети форма передаваемых по среде передачи данных сигналов отлична от прямоугольного импульса. Все радиосети являются аналоговыми сетями. Также аналоговыми являются модемные окончания цифровых сетей.

В аналоговых сетях в большинстве случаев не представляется возможности непосредственной передачи полезного сигнала ввиду несоответствия частоты и формы этого сигнала природе или параметрам среды передачи данных. Чтобы передать такой сигнал, используют модуляцию, то есть изменение высокочастотного несущего сигнала, удовлетворяющего природе или параметрам среды передачи данных, по закону изменения полезного сигнала. В современных аналоговых сетях используют чаще всего один из трех видов модуляции:

а) Частотная модуляция. При частотной модуляции по закону изменения полезного сигнала изменяется частота высокочастотного несущего сигнала. Амплитуда и фаза несущего сигнала остаются неизменными;

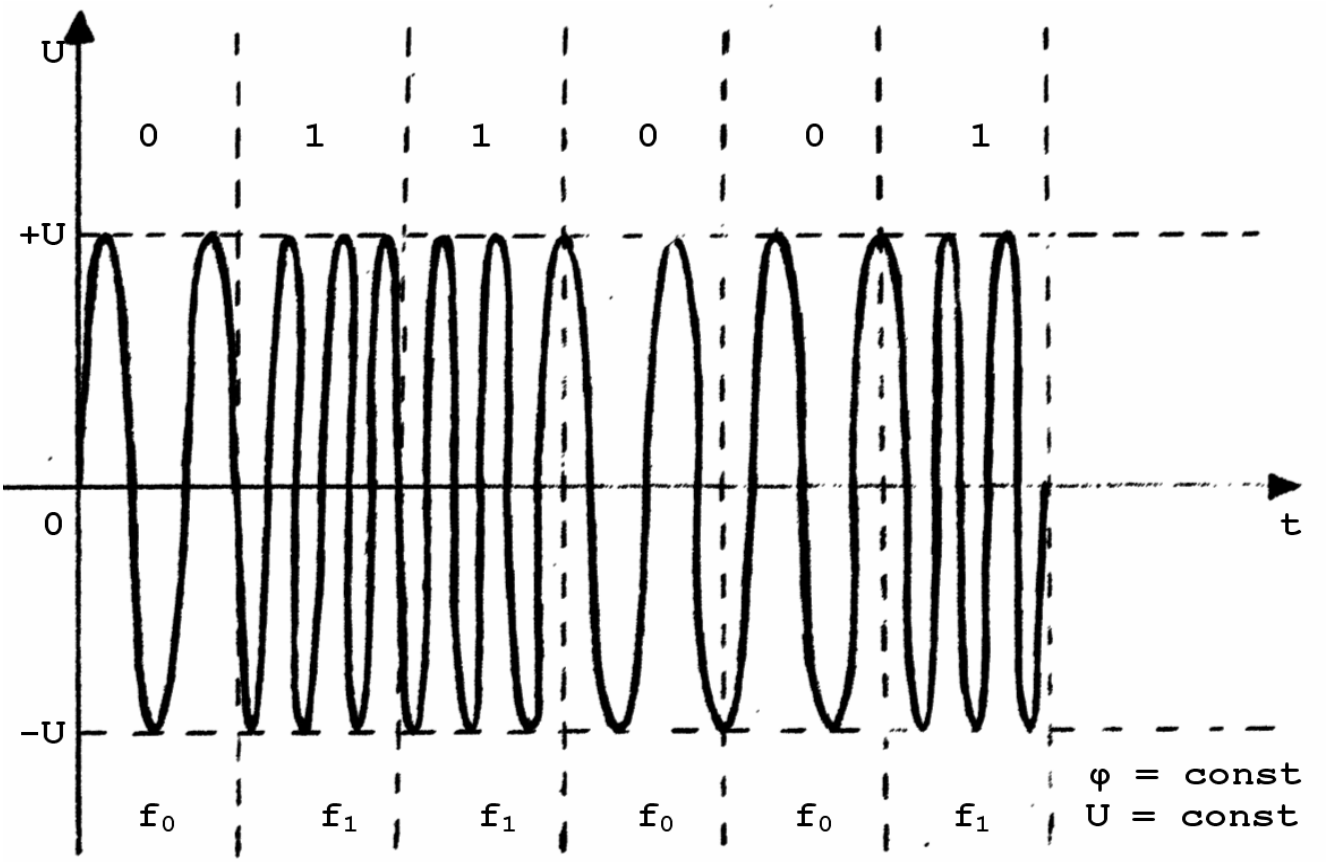


Рис. 15. Частотная модуляция.

б) Амплитудная модуляция. При амплитудной модуляции по закону изменения полезного сигнала изменяется амплитуда высокочастотного несущего сигнала. Частота и фаза несущего сигнала остаются неизменными;

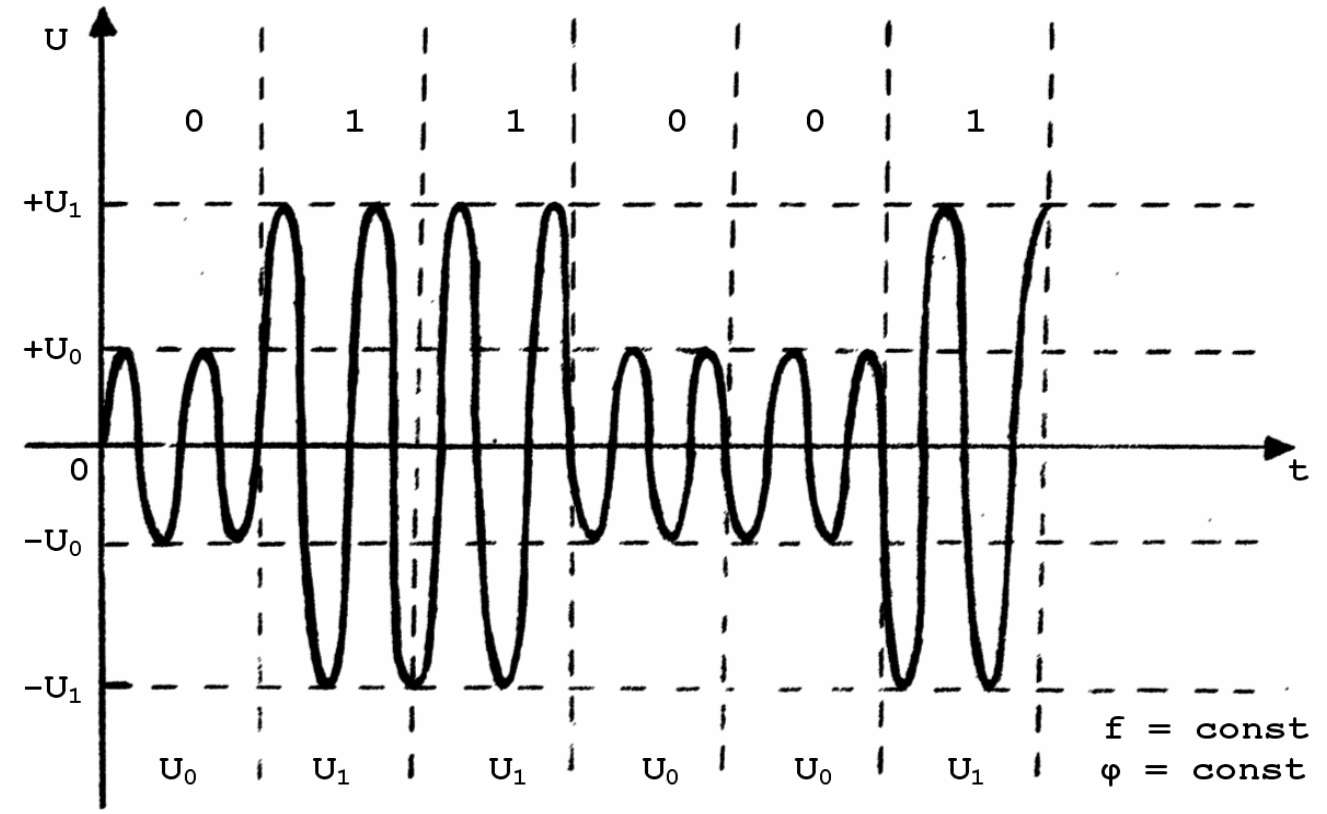


Рис. 16. Амплитудная модуляция.

в) Фазовая модуляция. При фазовой модуляции по закону изменения полезного сигнала изменяется фаза высокочастотного несущего сигнала. Амплитуда и частота несущего сигнала остаются неизменными.

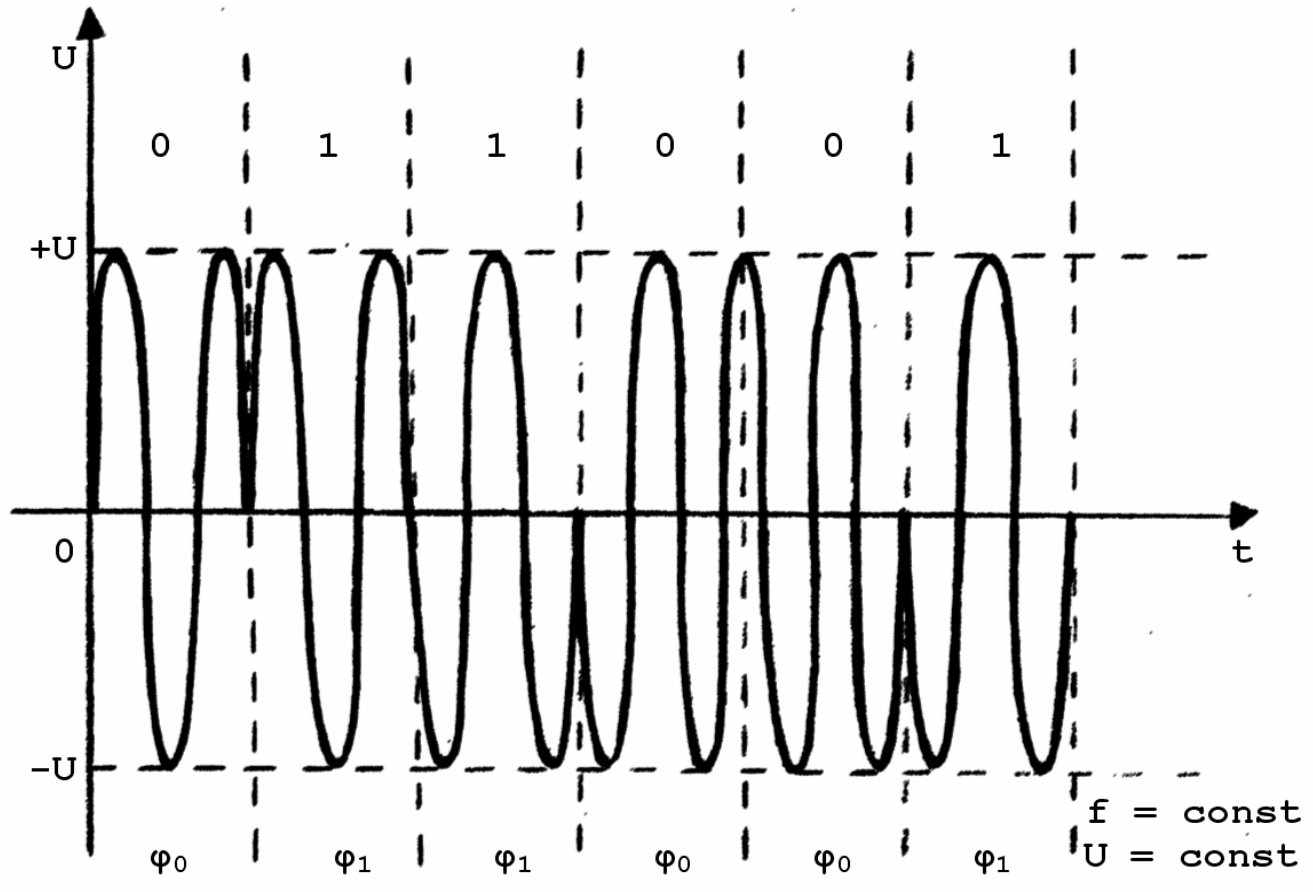


Рис. 17. Фазовая модуляция.

Амплитудная и фазовая модуляции имеют неизменную частоту, узкий энергетический спектр и поэтому могут использоваться в низкочастотных диапазонах. При частотной модуляции частота несущего сигнала изменяется, энергетический спектр расширяется и по этой причине частотная модуляция используется в основном в высокочастотных диапазонах.

Шумовая помеха имеет амплитудную природу и не оказывает влияние на изменение полезного сигнала в частотной и фазовой модуляции. В то же время наложение шумовой помехи на амплитудно-модулированный сигнал сразу проявляется в полезном сигнале. Таким образом, частотная и фазовая модуляции обладают наилучшей помехозащищенностью по сравнению с амплитудной модуляцией;

2. Цифровые сети. В цифровых сетях информация по среде передачи данных передается в виде прямоугольных импульсов. Существует два вида кодирования передаваемой информации прямоугольными импульсами:

а) Кодирование уровнем. В этом виде кодирования существуют два уровня напряжения – напряжение нуля и напряжение единицы. При передаче каждого бита информации напряжение

сигнала меняется в соответствии со значением передаваемого бита. Существуют четыре способа кодирования уровнем:

– однополярное (униполярное) кодирование. При униполярном кодировании оба уровня напряжений (нуля и единицы) имеют одинаковую (чаще положительную) полярность;

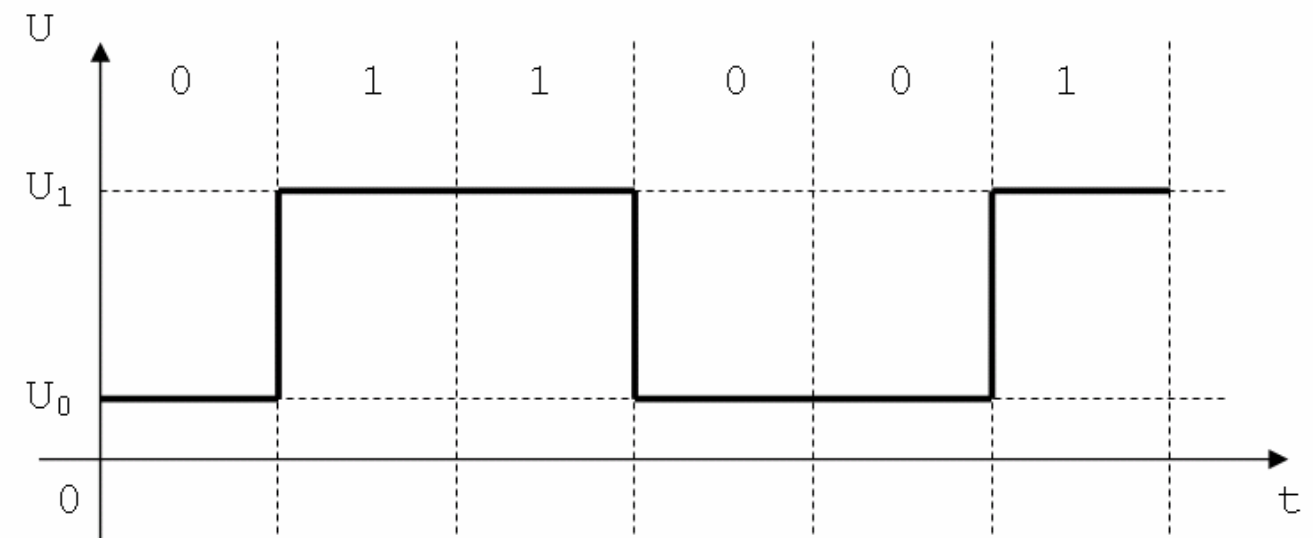


Рис. 18. Униполярное кодирование.

– двуполярное (биполярное) кодирование. При биполярном кодировании уровни напряжений нуля и единицы имеют различную полярность. Как правило, напряжение нуля отрицательно, а напряжение единицы положительно;

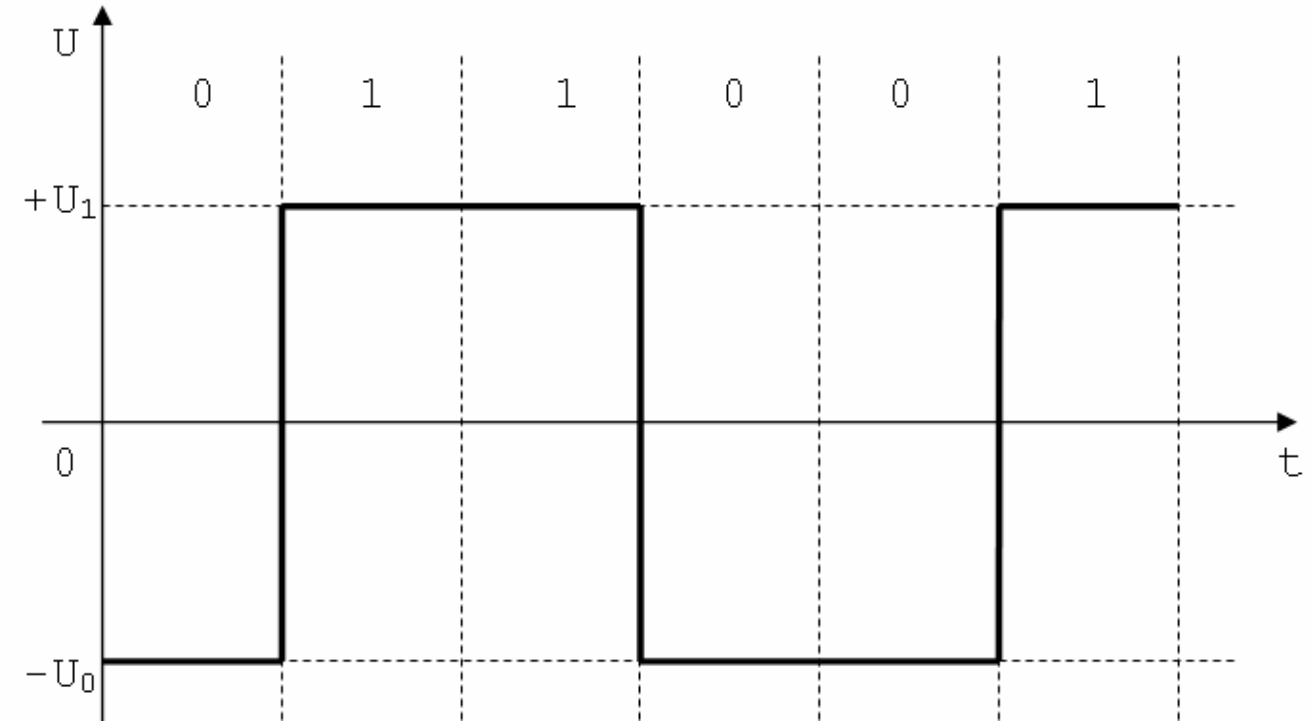


Рис. 19. Биполярное кодирование.

– кодирование с возвратом к нулю. При кодировании с возвратом к нулю информация кодируется короткими импульсами в начале каждого битового интервала. Как правило, единица кодируется положительным импульсом, а ноль – отрицательным.

Все остальное время в пределах битового интервала напряжение в линии связи равно нулю;

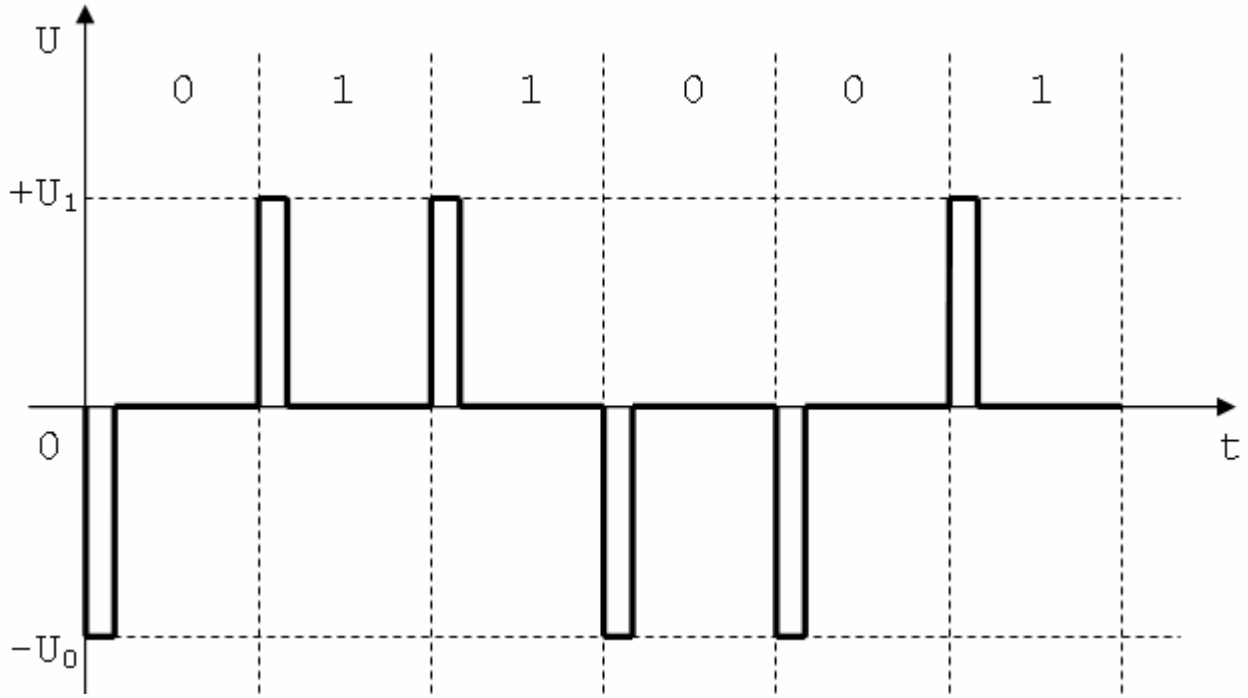


Рис. 20. Кодирование с возвратом к нулю.

– двухфазное (парафазное) кодирование. Парафазное кодирование применяется на трехпроводных линиях, где во входных каскадах используются дифференциальные усилители. Дифференциальный усилитель – это электронное устройство, имеющее два входа, работающих на вычитание сигнала. Положительная фаза полезного сигнала подается на вход «+», а отрицательная – на вход «–».

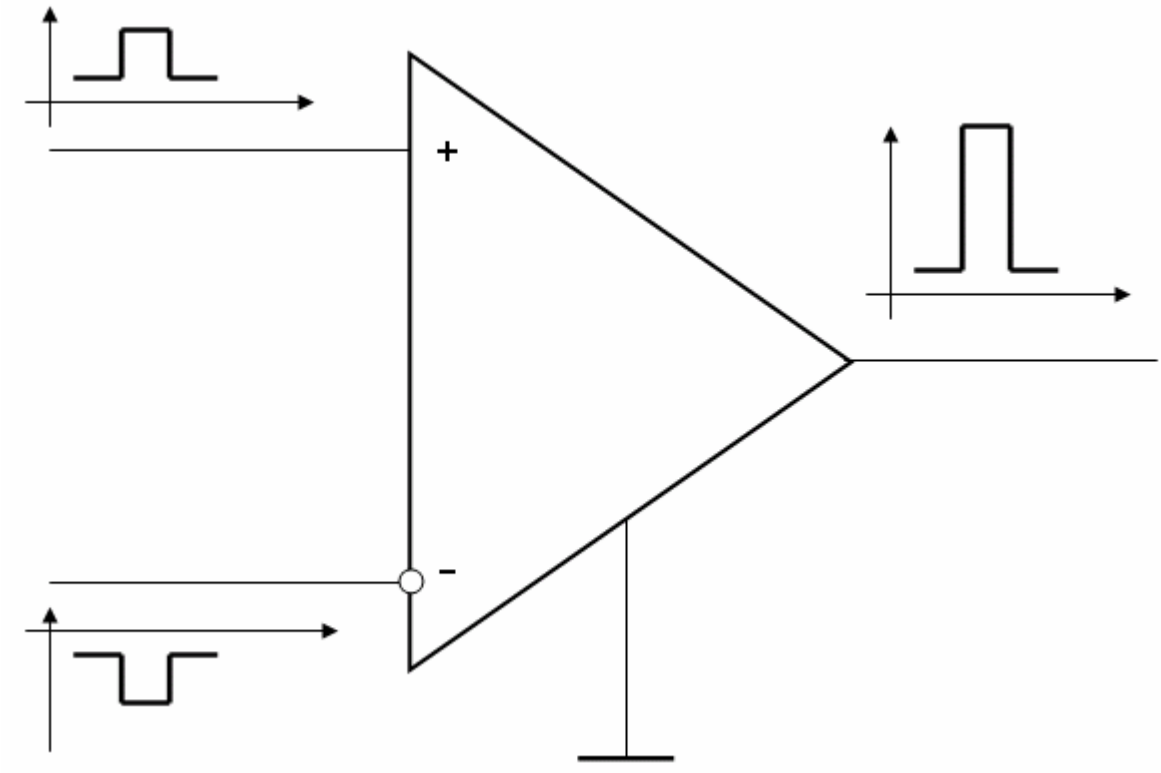


Рис. 21. Принцип работы дифференциального усилителя.

При вычитании из положительной фазы сигнала отрицательной, на выходе получается сигнал с амплитудой, в два раза превышающей амплитуду исходного сигнала. Шумовая помеха, наводимая во всех проводах трехпроводной линии связи, имеет одинаковую по знаку и величине амплитуду. При вычитании двух одинаковых по знаку и амплитуде сигналов, на выходе получается ноль. Таким образом, парафазное кодирование обладает высокой помехозащищенностью, полностью уничтожая шумовую помеху на выходе дифференциального усилителя;

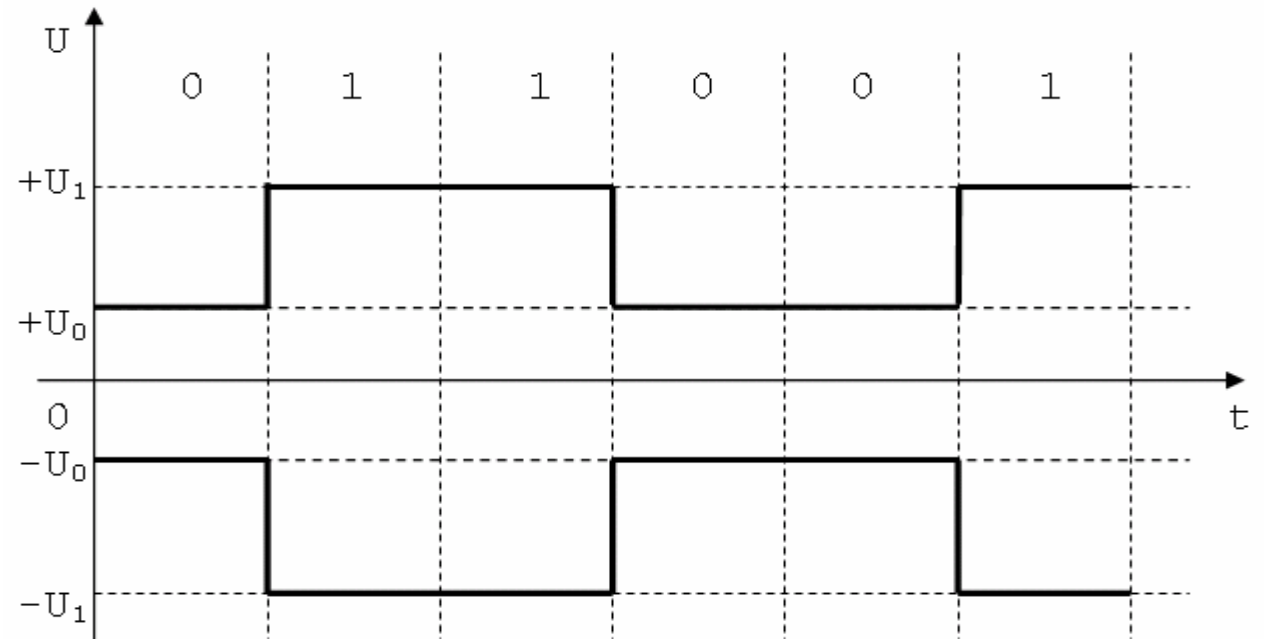


Рис. 22. Парафазное кодирование.

б) Кодирование фронтом. Существуют четыре способа кодирования уровнем:

- без возврата к нулю. В этом способе кодировании единица кодируется изменением уровня напряжения на противоположный, а при кодировании нуля уровень напряжения остается без изменений;

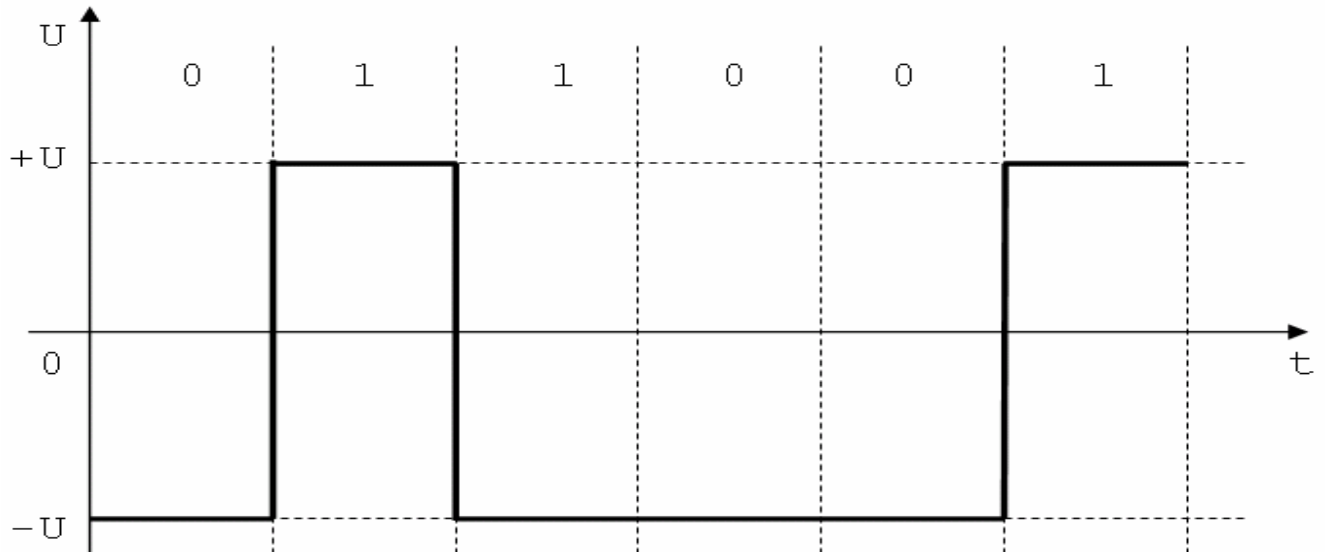


Рис. 23. Кодирование без возврата к нулю.

- код Манчестер. В коде Манчестер единица кодируется изменением уровня напряжения с положительного на отрицательный в середине битового интервала, а ноль – изменением уровня напряжения с отрицательного на положительный также в середине битового интервала. При передаче подряд нескольких нулей или единиц, уровень напряжения изменяется на противоположный на границе битового интервала.

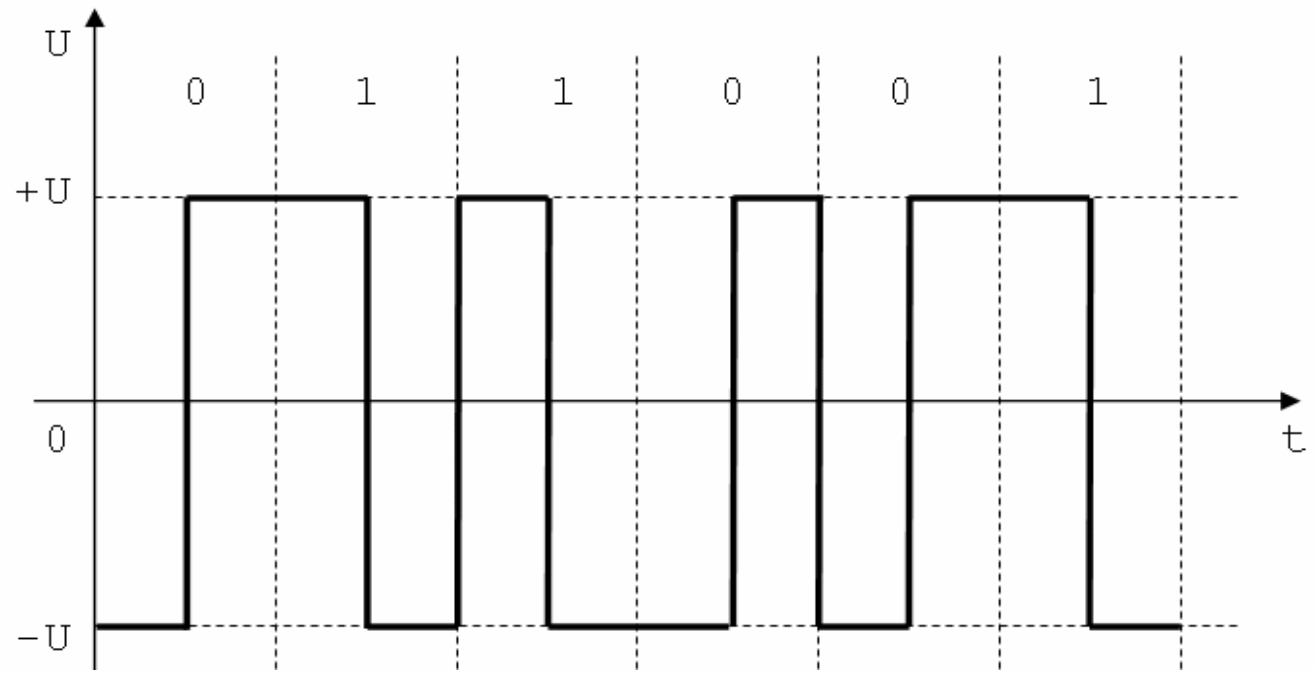


Рис. 24. Код Манчестер

Униполярное, биполярное, парафазное способы кодирования и кодирование без возврата к нулю при передаче длинной последовательности нулей или единиц образуют постоянное напряжение. По этой причине указанные способы кодирования не применяются в линиях связи, содержащих емкостные или индуктивные элементы. В основном эти способы кодирования используются в межблочных и внутриблочных линиях связи в пределах одного сетевого устройства. По этой же причине для подсчета количества переданных подряд нулей или единиц указанные способы кодирования требуют наличия внешней синхронизации. Кодирование с возвратом к нулю и код Манчестер не образуют постоянного напряжения и обладают свойством самосинхронизации. Благодаря этому указанные способы кодирования широко используются в линиях связи, соединяющих различные сетевые устройства, а также содержащих емкостные и индуктивные элементы. Код Манчестер является в настоящее время наиболее употребимым при передаче данных в сетях ЭВМ.

Приложение 2. Теоретическая модель сети

В 1984 году Международная организация по стандартизации ISO (International Standard Organization) закончила начатую в 1977 году разработку модели открытого системного взаимодействия OSI (Open System Interface), которая является в настоящее время международным стандартом для передачи данных по сетям ЭВМ. Модель OSI определяет:

1. Способы установки связи и обмена данными между сетевыми устройствами при использовании ими различных систем кодирования данных;

2. Методы определения момента начала передачи данных;

3.Методы обеспечения передачи нужной информации конкретным адресатам;

4. Организацию коммутации элементов физической среды передачи данных;

5. Поддержание необходимой скорости передачи данных всеми сетевыми устройствами;

6. Методы представления двоичных битов в среде передачи данных.

Модель OSI не описывает ничего реального – это концептуальная основа, с помощью которой общая задача передачи данных разделяется на отдельные легко обозримые компоненты. Модель OSI реализована в виде сетевых протоколов. Под сетевым протоколом понимается соглашение между производителями сетевого оборудования и программного обеспечения о способах обмена информацией между ЭВМ. Сетевые протоколы бывают:

1. Аппаратный протокол. Аппаратный протокол определяет, как функционируют и взаимодействуют между собой устройства ЭВМ (например: тип среды передачи данных, уровни сигналов в среде передачи данных, способы кодирования информации и т.п.);

2. Программный протокол. Программный протокол определяет способы взаимодействия программ друг с другом (например: способы распознавания типа передаваемой информации, контроля целостности передаваемой информации и т.п.).

Существуют два способа коммуникаций между ЭВМ в сети:

1. Коммуникации, не ориентированные на установление соединения. Эти коммуникации реализуют сквозную передачу потока данных, заранее предполагая, что потеря данных маловероятна и они гарантированно достигнут адресата. Как правило, этот способ коммуникаций применяется в сетях с высоким качеством среды передачи данных и низкой вероятностью возникновения перегрузок в сети;

2. Коммуникации, ориентированные на установление соединения. Этот способ коммуникации предполагает, что в процессе передачи данные могут теряться или поступать в некор-

ректном порядке. Как правило, данный способ коммуникаций применяется в сетях с низким качеством среды передачи данных и высокой вероятностью возникновения перегрузок в сети.

Сетевые протоколы, как программные, так и аппаратные, подразделяются по тем же признакам, что и способы коммуникаций – сетевые протоколы, ориентированные на установление соединения, и сетевые протоколы, не ориентированные на установление соединения.

Модель OSI имеет семь иерархических уровней:

1. Физический уровень. Это самый нижний уровень модели OSI, определяющий физические аспекты передачи двоичной информации в среде передачи данных. Протоколы физического уровня детально описывают природу среды передачи данных, напряжения, частоты, виды синхронизации и т.п.;

2. Канальный уровень. Данный уровень обеспечивает безошибочную передачу данных через среду передачи данных, то есть через физический уровень, которая при передаче может их искажать. Протоколы канального уровня определяют порядок оформления исходящей информации в блоки данных стандартного размера, называемые кадрами, и алгоритмы коррекции возникающих при передаче ошибок. Канальный уровень принимает пакеты данных от вышележащего сетевого уровня, делит их на кадры и добавляет к каждому кадру управляющую информацию. При приеме информации канальный уровень распознает кадры, адресованные данной ЭВМ, выявляет испорченные и потерянные кадры и дополнительно запрашивает их у корреспондента. Канальный уровень реализует так называемое прямое соединение, то есть соединение двух сетевых устройств, взаимодействующих непосредственно друг с другом без помощи третьего устройства. Ввиду большой сложности реализации канального уровня модели OSI, он был разбит на два подуровня:

а) Подуровень управления доступом к среде передачи (нижний подуровень);

б) Подуровень управления логическим каналом (верхний подуровень);

3. Сетевой уровень. Сетевой уровень обеспечивает проводку сообщений по сети, которая может иметь более одного прямого соединения. Протоколы сетевого уровня определяют порядок деления информации на пакеты и алгоритмы маршрутизации сообщений. Под маршрутизацией понимается процесс определения пути, по которому следует пересылать данные между ЭВМ;

4. Транспортный уровень. Этот уровень обеспечивает надежность передачи потоков данных в сети и контроль соединения между конечными точками маршрута. Протоколы транспортного уровня определяют организацию передачи данных по мар-

шруту, определенному на сетевом уровне (например: комплектность данных, порядок передачи данных и т.п.);

5. Сеансовый уровень. Данный уровень обеспечивает совместную работу программ на различных ЭВМ. Протоколы сеансового уровня дают возможность двум программам найти друг друга и установить соединение, а в случае отказа в сети повторно передать только данные, посланные после возникновения отказа;

6. Представительный уровень. Представительный уровень обеспечивает преобразование данных из формата представления данных конкретной ЭВМ в формат сети и обратно. Протоколы представительного уровня определяют алгоритмы перекодировки двоичной, текстовой информации и изображений, сжатия и распаковки данных, преобразования различных сетевых стандартов;

7. Прикладной уровень. Это самый верхний уровень модели OSI. Он обеспечивает интерфейс между прикладной программой и сетью. Ни пользователь, ни прикладная программа не имеют доступа ни к одному из уровней модели OSI, кроме прикладного.

Каждый уровень модели OSI реализуется соответствующими сетевыми протоколами. Каждый уровень модели OSI имеет не менее одного связанного с ним сетевого протокола. Таким образом, для реализации коммуникационного процесса необходимо иметь не менее семи сетевых протоколов. Поэтому существует понятие стека протоколов, то есть набора сетевых протоколов, упорядоченных в виде уровней для реализации коммуникационного процесса. Физически протоколы реализованы в виде пакетов сетевых программ, входящих в состав той или иной операционной системы, функционирующей на конкретной ЭВМ.

На передающей ЭВМ данные, которые необходимо передать по сети, поступают на верхний, прикладной уровень модели OSI. Далее эти данные передаются вниз по уровням модели OSI от прикладного до физического. Каждый уровень модели OSI, кроме физического, добавляет к поступившим от вышележащего уровня данным заголовок, содержащий управляющую информацию для соответствующего уровня принимающей ЭВМ. На сетевом уровне совокупная информация (данные и заголовки) делится на пакеты по числу маршрутов передачи, а на канальном уровне эти пакеты делятся на кадры стандартного размера, пригодные для непосредственной передачи на физическом уровне.

На принимающей ЭВМ поступившая по сети информация передается вверх по уровням модели OSI от физического до прикладного. Каждый уровень модели OSI, кроме физического, удаляет соответствующий адресованный ему заголовок. На канальном уровне принятые кадры упорядочиваются, так как могут быть приняты не в той последовательности, чем посыла-

лись, дополнительно запрашиваются потерянные и искаженные кадры, и из них формируется пакет, который передается затем на вышележащий сетевой уровень. На сетевом уровне после прихода всех пакетов одного сообщения, передававшихся по разным маршрутам, они упорядочиваются, и из них формируется полная информация, переданная с другой ЭВМ. Полученные от другой ЭВМ данные, выдаются прикладным уровнем модели OSI запрашивающему процессу.

Приложение 3. Управление сетями ЭВМ

Одной из существенных современных тенденций является усиливающаяся зависимость мировой и национальной экономики от информационных услуг, предоставляемых цифровыми сетями интегрального обслуживания. В свою очередь, экономичность, конкурентоспособность и качественные показатели самих сетей и систем телекоммуникаций все более зависят от организации управления ими. Под управлением сетью ЭВМ понимаются все действия, относящиеся к планированию, сооружению и эксплуатации сети, и приводящие к более экономически эффективному использованию ресурсов сети, при сохранении требуемого качества предоставляемых сетью услуг. Управление цифровыми сетями рассматривается сегодня как одна из наиболее проблемных и сложных задач. Усложнение задач управления сетями ЭВМ обуславливается следующими причинами:

1. Необходимость уменьшения стоимости информационных услуг, предоставляемых цифровыми сетями, с одновременным повышением их качества;
2. Необходимость постоянного расширения рынка новых услуг, предоставляемых сетью;
3. Сложность и большой диапазон услуг, предоставляемых сетью;
4. Необходимость создания в глобальных сетях ЭВМ услуг общемировых телекоммуникаций;
5. Необходимость обеспечения работы в цифровой сети оборудования различных производителей, различных типов и разнообразных сетевых технологий.

В настоящее время главным требованием к цифровой сети интегрального обслуживания является предоставление пользователям необходимых им услуг в кратчайшее время. По этой причине сети не должны обслуживаться вручную, а только с помощью специализированных высокопроизводительных ЭВМ. Основная задача системы управления состоит в уменьшении времени реакции сети на изменение её состояния. Для этого требуется:

1. Программное управление и дистанционная коммутация на всех уровнях иерархии сети;

2. Высокоскоростная сеть управления связью для выполнения функций технического обслуживания цифровой сети;

3. Удобное для пользователей программное обеспечение управления сетью.

Система управления сети ЭВМ должна обеспечивать адаптацию сети к возникающим в процессе функционирования непредвиденным ситуациям, а при нормальной работе - обеспечивать максимальное использование избыточности технических средств сети с целью увеличения её коммерческих показателей. Система управления сетью по своей сути является вычислительной сетью, логически наложенной на управляемую информационную сеть. Физически они используют одни и те же каналы связи, коммутационное оборудование и управляющие ЭВМ. Международная организация по стандартизации ISO (International Standard Organization) определила обязательным применение в цифровых сетях асинхронного режима передачи данных ATM (Asynchronous Transfer Mode – асинхронный режим передачи) с одновременным использованием синхронных цифровых иерархий SDH (Synchronous Digital Hierarchy – синхронная цифровая иерархия). Это объясняется тем, что одним из основных свойств синхронных цифровых иерархий является заложенные в них широкие потенциальные возможности сетевого управления.

До недавнего времени коммутируемые телефонные сети общего пользования и сети ЭВМ развивались как самостоятельные структуры. При разработке сетевых стандартов Международная организация по стандартизации ISO рекомендовала начать интеграцию этих структур на основе модели OSI и цифровых автоматических телефонных станций (маршрутизаторов), управляемых специализированными ЭВМ. В крупных городах России практически все автоматические телефонные станции переведены на цифровую основу, и коммутируемая телефонная сеть общего пользования в настоящее время являет собой пример классической цифровой сети интегрального обслуживания.

Во многих развитых странах мира к сегодняшнему дню уже созданы национальные информационные инфраструктуры, базирующиеся на принципах широкого внедрения цифровых сетей интегрального обслуживания. Так, повсеместное пользование такими услугами, как телекоммуникации, электронные покупки, телеконференции, электронная почта и т.п., заменяют от 20% до 50% соответствующих транспортных услуг и приносят экономический эффект около 100 млрд долларов в год, что почти вдвое больше, чем ежегодная экономия, ожидаемая от использования альтернативных видов топлива. Благодаря внедрению цифровых сетей, деловая активность и продуктивность предпринимателей и всего общества повысилась в полтора раза.

Создание национальных информационных инфраструктур идет по четырем направлениям развития:

1. Персональный доступ к цифровой сети как в стационарных, так и в мобильных условиях;

2. Внедрение технологии «информационных полей», обеспечивающей одинаковый доступ ко всем ресурсам и услугам цифровых сетей интегрального обслуживания;

3. Интеграция предприятий с точки зрения совместного использования их общих информационных и технических ресурсов;

4. Формирование политических, государственных и образовательных процессов с учетом новых возможностей цифровых сетей интегрального обслуживания.

Управление сетями ЭВМ строится на базе модели TMN (Telecommunications Management Network – сеть управления связью), разработанной Международной организацией по стандартизации ISO. Модель TMN включает в себя четыре иерархических уровня:

1. Уровень управления сетевыми элементами. Это самый нижний уровень модели TMN. Он определяет действия управления по типам управляемого оборудования (например: системы передачи, маршрутизаторы и т.п.);

2. Уровень управления телекоммуникационной сетью. Данный уровень модели TMN определяет действия управления на сеть в целом;

3. Уровень управления услугами. Этот уровень модели TMN определяет формы обслуживания пользователей (например: финансовые расчеты, рассмотрение жалоб и т.п.);

4. Уровень управления бизнесом. Это самый верхний уровень модели TMN. Он определяет формы управления предприятиями, предоставляющими услуги цифровых сетей и осуществляющими эксплуатацию оборудования сетей.

Физическая реализация модели TMN осуществляется с помощью пяти функциональных блоков:

1. Сетевые управляющие операционные системы;

2. Блоки преобразования протоколов. Как правило, эти блоки входят в состав сетевых программ представительного уровня модели OSI;

3. Сетевые блоки. Сетевые блоки обеспечивают работу с сетевым оборудованием;

4. Блоки адаптеров. Блоки адаптеров обеспечивают работу с сетевым оборудованием, не имеющим стандартного интерфейса;

5. Рабочие станции.

Информационный обмен между блоками модели TMN осуществляется с помощью семиуровневой модели OSI. Модель TMN, будучи логически наложенной на модель OSI, нижним своим уров-

нем взаимодействует с верхним прикладным уровнем модели OSI. Функции модели TMN:

1. Общие функции:

а) Транспортная функция (обмен управляющей информацией между элементами модели TMN);

б) Хранение и отображение управляющей информации;

в) Обеспечение безопасности, контроль доступа к управляющей информации;

г) Доступ к управляющей информации;

д) Обработка управляющей информации;

е) Поддержка пользовательских рабочих станций (управление обменом информации);

2. Прикладные функции:

а) Управление работой:

- контроль изменения работы;
- управление нагрузкой на сеть;
- контроль качества обслуживания пользователей;

б) Управление техническим обслуживанием:

- обнаружение сбоев и отказов оборудования;
- локализация неисправностей оборудования;
- тестирование линий передачи данных;

в) Управление конфигурацией сети:

- оперативная реконфигурация сети;
- сбор информации о состоянии элементов сети;
- развитие сети;

г) Управление учетом и распределением ресурсов сети:

- сбор учетных и статистических данных;
- определение стоимости пользовательских услуг;
- начисление платы пользователям;

д) Управление защитой от несанкционированного доступа.

Модель TMN в качестве перспективы интеграции сетей связи рассматривает создание интеллектуальных сетей при условии реализации элементов искусственного интеллекта в звене управления цифровыми сетями. Примеры технической реализации модели TMN:

1. Система управления сетью Французской республики ABC (Alcatel Business Communications);

2. Оптимизированная система сетевого управления фирмы Сименс (Германия) ONMS (Optimized Network Management System);

3. Система управления цифровой коммутируемой телефонной сетью Швейцарии PTT Telecom;

4. Система управления объединенной сетевой архитектурой Великобритании British Telecom;

5. Отечественная система сигнализации ОКС 7. Система сигнализации ОКС 7 реализована на базе модели TMN и применяется в качестве системы управления в элементах отечест-

венных коммутируемых телефонных сетей общего пользования, совместимых со стандартом модели OSI и в отечественных сетях цифровой сотовой связи. Название «система сигнализации ОКС 7» является больше историческим, чем отражающим реальную суть системы. На рубеже 19 и 20 столетий, когда коммутация в телефонных сетях осуществлялась вручную, возникла необходимость в организации канала связи между коммутаторами для облегчения руководства телефонными барышнями. Этот канал связи соединял все городские ручные коммутаторы и был некоммутируемый, поэтому его назвали «системой сигнализации ОКС (общий канал сигнализации)». В дальнейшем, по мере совершенствования и автоматизации телефонных сетей появилась система сигнализации ОКС 1, затем ОКС 2 и т.д. При разработке и внедрении современной системы управления, очередным номером общего канала сигнализации была семерка, поэтому исторически систему управления назвали ОКС 7. По своей сути сеть ОКС 7 является вложенной пакетной сетью, где пакет – это сигнальная единица информации. Конфигурация сети ОКС 7 и маршруты передачи управляющей информации не всегда совпадают с конфигурацией управляемой сети и маршрутами передачи пользовательской информации. В качестве физического канала передачи используется цифровой канал производительностью 64 Кбод. Передаваемые управляющие пакеты внешне ничем не отличаются от пользовательских пакетов за исключением кода типа сообщения в заголовке пакета. Для передачи по физическому уровню пакеты стандартным образом делятся на кадры.

Нижние два уровня модели TMN, а именно уровень управления сетевыми элементами и уровень управления телекоммуникационной сетью, реализуются средствами, встроенными в сетевые операционные системы. Настройку этих операционных систем производят специалисты, называемые администраторами сетей, или предприятия, поддерживающие нормальную работу сети, и называемые операторами сети. И те, и другие, осуществляют административное управление сетью. Администраторы и операторы сетей решают следующие задачи:

1. Контроль и настройка средств обеспечения высокой производительности сетевого оборудования;
2. Выявление и устранение проблем, возникающих в сети;
3. Установка и настройка программного обеспечения;
4. Резервное копирование и восстановление данных;
5. Автоматизация сети;
6. Корректировка параметров конфигурации и управления сетью;
7. Поддержание целостности сети и решение проблем защиты информации;
8. Сопровождение учетных записей пользователей;
9. Помощь и поддержка пользователей и рабочих групп;

10. Запуск и останов элементов сети.

Верхние два уровня модели TMN, а именно уровень управления услугами и уровень управления бизнесом, реализуются специальными прикладными программами, не входящими в стандартный пакет программ сетевой операционной системы. С этими программами работают специалисты – менеджеры (управляющие) операторами сетей, а также предприятиями или организациями, предоставляющими услуги цифровых сетей. Такие предприятия и организации, которые сами сетевое оборудование не эксплуатируют, а только предоставляют услуги сети, называются провайдерами (provider – поставщик). Необходимо заметить, что на рынке существуют как предприятия и организации, которые являются чисто операторами или чисто провайдерами сетей, так и предприятия и организации, совмещающие в себе обе эти функции. Вышеуказанные менеджеры осуществляют оперативное управление сетью.

Перечень терминов и их определений

QNX-процесс – это работа микроядра распределенной операционной системы по обработке одного или нескольких потоков выполнения.

Автоматизированная система управления – это человеко-машинная система, обеспечивающая автоматизированный сбор и обработку информации, и выработку на её основе рекомендаций для поддержки принятия управленческих решений и (или) управляющих воздействий на физические объекты.

Авторизация – это процедура предоставления объекту, успешно прошедшему процедуры идентификации и аутентификации, соответствующих полномочий и прав доступа к ресурсам системы.

Адаптер объектов – это механизм группирования программных объектов в соответствии с политикой активизации каждого из них.

Административная масштабируемость – это вид масштабируемости, определяющий легкость в управлении множеством независимых компонент системы.

Адрес – это специальный тип имени, указывающий на точку доступа к сущности.

Актуальность копий ресурса – это состояние, когда операция чтения данных дает одинаковые результаты для каждой из копий ресурса.

Алгоритмизированная деятельность – это деятельность, выполняемая в соответствии с заранее заданными указаниями.

Алгоритмическая ошибка – это ошибка, обусловленная некорректной постановкой задач в техническом задании.

Аналитическая модель – это модель, в которой процесс функционирования изучаемого объекта, процесса или явления записывается в виде функциональных отношений или логических условий.

Аналоговая модель – это модель, берущая в основу существующую наглядную аналогию.

Аппаратное обеспечение АСУ – это комплекс технических средств, предназначенных для обеспечения работы АСУ.

Аппаратный протокол – это сетевой протокол, который определяет, как функционируют и взаимодействуют между собой устройства ЭВМ.

Асинхронный вызов процедур – это вариант вызова клиентом удаленной процедуры, когда клиент получает возможность продолжить свою работу сразу после выполнения запроса на удаленный вызов процедуры.

Асинхронный режим передачи – это режим передачи, в котором временные ограничения на передачу потока данных не накладываются.

Атака на систему – это действие, предпринимаемое злоумышленником с целью поиска и использования той или иной уязвимости системы.

Атомарная групповая рассылка – это такая рассылка, при которой операция обновления реплик, начатая до того, как произошла поломка одной из них, будет выполнена корректно на всех работающих репликах или не будет выполнена ни на одной из них.

Аутентификация – это проверка подлинности объекта.

Базовая сетевая топология – это шинная, звездообразная или кольцевая топология.

Безопасная система – это система, успешно и эффективно противостоящая угрозам безопасности информации.

Безопасность – это требование к системе парировать ситуации возникновения сбоев и отказов в работе.

Безотказность – это требование к системе безотказно работать в течение заданного промежутка времени.

Блокировка процесса – это приостановка работы процесса.

Брандмауэр – это монитор ссылок, работающий как шлюз прикладного уровня между локальным сегментом сети и остальной глобальной сетью, и предназначенный для тотальной авторизации всех пакетов, поступающих в локальный сегмент сети, и выходящих из него.

Брокер сообщений – это программа, выполняющая функции представительного уровня модели OSI в распределенной информационной системе.

Ведомый процесс – это процесс, подчиненный ведущему процессу.

Ведущий процесс – это процесс, выполняющий в распределенной системе главенствующую роль над другими.

Вероятность безотказной работы – это вероятность, с которой система, имеющая заданную интенсивность отказов, будет функционировать по прямому назначению непрерывно и безотказно в течение заданного времени.

Вертикаль – это множество всех подсистем, вышестоящих и подчиненных по отношению к данной.

Виртуально синхронная групповая рассылка – это атомарная групповая рассылка, гарантирующая, что ситуация отказа процесса в ходе групповой рассылки представляется как ситуация того, что групповая рассылка вообще не производилась.

Вложенная транзакция – это транзакция, состоящая из дочерних транзакций, способных работать параллельно без взаимных блокировок (у дочерних транзакций отсутствует свойство долговечности).

Вложенный поток данных – это несколько связанных между

собой простых потоков.

Внешняя среда – это множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под её воздействием в условиях рассматриваемой задачи.

Возмущающие воздействия – это воздействия на систему факторов внешней среды.

Геоинформационная система – это интегрированная информационная автоматизированная система управления территориального уровня, предназначенная для поддержки принятия решений, основанных на анализе пространственно-временной информации.

Геоинформационная технология – это совокупность средств и методов сбора, обработки, накопления и использования пространственно-временной информации.

Гетерогенная распределенная система – это распределенная система, использующая одновременно несколько сетевых технологий на ЭВМ с одинаковыми или различными аппаратными платформами.

Гибкость – это характеристика открытости, показывающая, насколько легко конфигурируются системы, состоящие из различных компонент от разных производителей.

Гибридная сеть (по иерархической структуре) – это сеть ЭВМ, в которой все ЭВМ имеют одинаковые права доступа к ресурсам сети, находящимся на серверах.

Гибридная сеть (по топологической структуре) – это любая комбинация сетей с базовыми сетевыми топологиями.

Гипотетическая модель – это гипотеза исследователя о причинно-следственных связях между входом и выходом изучаемого объекта, процесса или явления.

Глобальная сеть – это сеть ЭВМ, в которой аппаратные средства сети принадлежат различным владельцам.

Глобальное имя – это тип имени, обозначающего одну и ту же сущность вне зависимости от того, где в системе это имя используется.

Гомогенная распределенная система – это распределенная система, использующая одну сетевую технологию на ЭВМ с одинаковыми аппаратными платформами.

Государственная тайна – это информация, доступ к которой ограничен Законом РФ «О государственной тайне».

Датаграмма – это пакет, передаваемый через сеть независимо от других пакетов без установления логического соединения и подтверждения приема.

Дерево целей – это результат выделения целей по всем подсистемам с указанием зависимостей между ними.

Детерминированная модель – это модель, предполагающая отсутствие случайных воздействий в моделируемом процессе.

Динамическая группа процессов – это группа, состав процессов которой может изменяться в зависимости от текущей задачи, решаемой системой.

Динамическая модель – это модель, отражающая поведение объекта во времени.

Дискретная модель – это модель, описывающая дискретные процессы.

Дискретно-непрерывная модель – это модель, описывающая процессы, в которых можно выделить как дискретные, так и непрерывные составляющие.

Дискретный поток данных – это поток байт.

Доверенная вычислительная база – это комплекс всех механизмов защиты информации.

Достоверность информации – это свойство информации строго принадлежать тому субъекту, от которого она была принята или который является её источником.

Доступ к информации – это процедура предоставления возможности ознакомления с этой информацией и её обработки.

Доступность – это требование к системе постоянно находиться в состоянии готовности к работе.

Доступность ресурса – это свойство ресурса быть доступным законным пользователям системы.

Дочерний домен – это поддомен родительского домена.

Жесткая система реального масштаба времени – это такая система, превышение времени реакции которой на внешние события сверх установленных пределов приводит к фатальным последствиям.

Живучесть – это свойство системы выполнять заданные целевые функции при неблагоприятных воздействиях внешней среды, не предусмотренных нормами эксплуатации.

Жизненный цикл программы – это период времени от начала её разработки до вывода программы из эксплуатации по причине морального устаревания.

Заместитель – это программа реализации интерфейса объекта, осуществляющая транзит параметров объекта от вызывающей программы к операционной системе рабочей станции и обратно.

Замкнутая система – это система, в которой любой элемент имеет связи только с элементами самой системы.

Защита информации – это комплекс мероприятий, направленный на обеспечение целостности и конфиденциальности информационного обеспечения АСУ.

Знаковая модель – это модель, реализуемая при использовании условных обозначений отдельных понятий (знаков), а также определенных операций между этими знаками.

Знания – это формализованные оценки экспертов на конкретные события в своей предметной области.

Идемпотентная операция – это такая операция, многократное повторение которой не влияет на конечный результат.

Идентификация – это процедура распознавания объекта.

Иерархическая группа процессов – это группа процессов, которая предусматривает наличие в ней ведущего процесса.

Изохронный режим передачи - это режим передачи, в котором для каждого элемента данных определяется как максимально возможная, так и минимально возможная задержка передачи.

Имитационная модель – это модель, которая реализуется при помощи средств вычислительной техники.

Инверсия приоритетов – это ситуация, когда поток выполнения с низким приоритетом использует ресурс процессора с более высоким, чем у себя, приоритетом.

Инженерная психология – это отрасль эргономики, изучающая информационные взаимодействия технических систем с внешней средой, в качестве которой выступает человек.

Интегрированная система – это пакет системных программ, дающий пользователю одинаковые возможности доступа к различным пакетам программ из одной операционной среды и стыковки программ пакетов по информационным данным.

Интервал дрожания – это интервал времени между минимально возможной и максимально возможной задержками передачи потока данных.

Интерфейсный агент – это программный агент, помогающий пользователям работать с одной или несколькими прикладными программами.

Информационная безопасность – это такое состояние информационного обеспечения АСУ, при котором исключается возможность ознакомления с этим информационным обеспечением, его изменения или уничтожения лицами, не имеющими на это право.

Информационное обеспечение АСУ – это совокупность единой системы классификации и кодирования технико-экономической информации, унифицированных систем документации и массивов информации, используемых в АСУ.

Информационный агент – это программный агент, управляющий информацией из множества различных источников.

Искусственный интеллект – это комплексное научно-техническое направление, имеющее целью создание и применение программно-аппаратных средств, позволяющих моделировать процессы человеческого мышления и обеспечить диалог с ЭВМ на языке, естественном для человека.

Клиент – это процесс, запрашивающий службы у серверов.

Клиентская заглушка – это специальная версия функции вызова процедуры, работающая на стороне клиента и подменяющая аналогичную стандартную локальную функцию в случае раз-

мещения нужной процедуры на другой ЭВМ сети.

Клонирование процесса – это создание точной копии процесса, которая выполняется на удаленной ЭВМ параллельно оригиналу.

Когнитолог – это специалист по методам формализованного представления знаний.

Кодовое расстояние – это количество разрядов, в которых не совпадают двоичные цифры двух кодовых комбинаций.

Кодовый вес – это количество единиц в некоторой кодовой комбинации.

Комбинированная модель – это модель, реализуемая при помощи декомпозиции процесса функционирования изучаемого объекта, процесса или явления на подпроцессы, при изучении которых используется как аналитическое, так и имитационное моделирование.

Коммерческая тайна – это информация, имеющая действительную или потенциальную коммерческую ценность в силу неизвестности её третьим лицам.

Коммуникация, не ориентированная на установление соединения – это коммуникация, которая реализуют сквозную передачу потока данных, заранее предполагая, что потеря данных маловероятна и они гарантированно достигнут адресата.

Коммуникация, ориентированная на установление соединения – это коммуникация, которая предполагает, что в процессе передачи данные могут теряться или поступать в некорректном порядке.

Коммутация каналов – это метод статической коммутации, когда между отправителем и получателем устанавливается выделенное физическое соединение, поддерживаемое на все время передачи информации.

Коммутация пакетов – это метод динамической коммутации, когда сообщение разбивается на короткие пакеты, которые передаются адресату одновременно по разным маршрутам.

Коммутация сообщений – это метод динамической коммутации, когда каждое сообщение интерпретируется как независимая единица, передающаяся от одного узла сети к другому.

Комплексные испытания – это выяснение соответствия заявленных разработчиком (производителем) и фактических характеристик какого-либо изделия.

Комплексный поток данных – это поток, содержащий несколько связанных между собой простых потоков.

Контроль доступа – это формальное подтверждение прав доступа к ресурсам.

Конфиденциальная информация – это вид информации, доступ к которой ограничен кругом лиц, которым она была доверена по службе или стала известна в процессе работы.

Концентратор – это устройство в составе сети с звездо-

образной топологией, обеспечивающее одновременный независимый обмен информацией между несколькими парами узлов сети.

Кооперативный агент – это агент, являющийся частью мультиагентной системы.

Корневой направляющий узел – это направляющий узел домена верхнего уровня.

Кризис – это изменение параметров системы, не предусмотренное нормами эксплуатации.

Криптография – это замена исходного сообщения его нечитабельным вариантом с целью сохранения конфиденциальности содержащейся в сообщении информации.

Критерий эффективности второго рода – это математическое выражение, позволяющее количественно оценить пути достижения цели.

Критерий эффективности первого рода – это математическое выражение, позволяющее количественно оценить степень достижения цели системой.

Кэширование – это создание копии ресурса потребителем этого ресурса.

Лингвистическое обеспечение АСУ – это совокупность научно-технических терминов и других языковых средств, используемых в АСУ, а также правил формализации естественного языка.

Листовой домен – это домен самого нижнего уровня.

Листовой узел – это узел пространства имен, представленный именованной сущностью и не имеющий исходящих из него ребер, обычно содержит информацию о представляемой сущности.

Локальная сеть – это сеть ЭВМ, в которой все аппаратные средства сети принадлежат одному владельцу – частному лицу или организации.

Локальное имя – это тип имени, интерпретация которого зависит от того, где в системе это имя используется.

Макет – это наглядная аналогия объекта.

Мандат – это нефальсифицируемая структура данных, относящаяся к некоторому ресурсу и точно определяющая права доступа владельца мандата к этому ресурсу.

Мандат – это нефальсифицируемая структура данных, относящаяся к некоторому ресурсу и точно определяющая права доступа владельца мандата к этому ресурсу.

Маршрутизатор – это сетевое устройство, предназначенное для соединения двух или более сегментов сети.

Маршрутизация – это процесс определения пути, по которому следует пересылать данные между ЭВМ.

Масштабируемость – это свойство расширения системы.

масштабируемость по размеру – это вид масштабируемости, определяющий легкость подключения дополнительных поль-

зователей и ресурсов.

Математическая модель – это модель, которая устанавливает соответствия реальному объекту, процессу или явлению символических высказываний в терминах математической логики.

Математическое обеспечение АСУ – это совокупность математических методов, моделей и алгоритмов для решения задач обработки информации в АСУ.

Менеджер данных – это процесс, который осуществляет транзактнонезависимые операции чтения и записи данных.

Менеджер транзакций – это процесс, который обрабатывает команды транзакций, преобразуя их в запросы к планировщику.

Миграция процесса – это перенос работающего процесса.

Минимальное кодовое расстояние двоичного кода – это самое малое кодовое расстояние, возможное между двумя любыми разрешенными кодовыми комбинациями в этом коде.

Мобильный агент – это программный агент, способный перемещаться с одной ЭВМ на другую.

Моделирование – это процесс построения модели и её использования в системном анализе.

Моделирование в нереальном масштабе времени – это моделирование, которое проводится при изучении быстротекущих или вялотекущих процессов.

Моделирование в реальном масштабе времени – это моделирование, которое проводится при соответствии скорости протекания исследуемых процессов возможностям исследователя по наблюдению и регистрации этих процессов.

Модель – это приближенное, упрощенное представление объекта, процесса или явления, помогающее лучше понять его функционирование и устройство, его характеристики.

Модель сильной мобильности – это вариант модели переноса кода, в котором переносится сегмент кода и сегмент исполнения, а работающий процесс может быть приостановлен, перенесен на другую ЭВМ и его выполнение продолжено с места останова.

Модель слабой мобильности – это вариант модели переноса кода, в котором допускается перенос только сегмента кода, а переносимая программа запускается всегда из своего начального состояния.

Модернизация – это стадия разработки АСУ, которая проводится специалистами предприятия-разработчика (изготовителя) на основе предъявленных в процессе промышленной эксплуатации рекламаций и имеет целью совершенствование отдельных характеристик системы.

Модуляция – это изменение высокочастотного несущего сигнала, удовлетворяющего природе или параметрам среды пе-

редачи данных, по закону изменения полезного сигнала.

Монитор ссылок – это программа реализации контроля доступа к ресурсам системы.

Мультиагентная система – это система, в которой программные агенты, работая совместно, выполняют общие задачи.

Мультиплексирование – это процедура предоставления доступа нескольким медленным процессам к одному быстродействующему ресурсу в режиме разделения времени таким образом, что у каждого из процессов создается иллюзия монопольного владения этим ресурсом.

Мысленная модель – это модель, которая применяется в случае невозможности реализовать моделируемый объект, процесс или явление в заданном интервале времени, или при отсутствии условий, возможных для их физического создания.

Мягкая система реального масштаба времени – это такая система, превышение времени реакции которой на внешние события сверх установленных пределов приводит к снижению эффективности управления.

Наглядная модель – это модель, которая создается на базе представлений человека о реальном объекте, процессе или явлении и наглядно отображает те или иные свойства оригинала.

Надежная групповая рассылка в порядке FIFO – это надежная групповая рассылка, которая гарантирует доставку сообщений каждому из процессов группы в том же порядке, в котором они были отправлены.

Надежность – это свойство системы сохранять свою работоспособность в заданных условиях эксплуатации.

Направляющая таблица - это таблица идентификаторов ребер графа пространства имен.

Направляющий узел – это узел пространства имен, имеющий несколько исходящих из него именованных ребер и хранящий таблицу идентификаторов ребер.

Натурное моделирование – это моделирование, которое проводится полностью на реальном объекте.

Научный эксперимент – это вмешательство человека в исследуемый процесс с целью определения границ его устойчивости.

Непрерывная модель – это модель, описывающая непрерывные процессы.

Непрерывный поток данных – это поток бит.

Неприсоединенный ресурс – это ресурс, который может быть с легкостью перенесен с одной ЭВМ на другую.

Непротиворечивость реплик ресурса – это сохранение актуальности копий ресурса.

Непротиворечивость, ориентированная на данные – это вид непротиворечивости, который обеспечивает непротиворечи-

вое представление данных.

Непротиворечивость, ориентированная на клиента– это вид непротиворечивости, который допускает нарушения непротиворечивости данных, но обеспечивают сокрытие факта нарушений от клиента.

Нерезидентный объект – это программный объект, который существует, только пока сервер им управляет.

Несанкционированный доступ – это совокупность приемов и порядок действий с целью получения конфиденциальной информации незаконным, противоправным путем.

Неупорядоченная надежная групповая рассылка – это виртуально синхронная групповая рассылка, не дающая никаких гарантий порядка прихода сообщений к различным процессам.

Обратный удар – это блокировка процесса-отправителя ответными сообщениями.

Обследование – это определение в самом общем виде основных целей и ограничений разрабатываемой системы, возможностей повышения эффективности управления при внедрении АСУ.

Общее устройство – это аппаратный ресурс, непосредственно подключенный в сеть.

Одноранговая сеть – это сеть ЭВМ, в которой отсутствует централизованное управление, все ресурсы сети равномерно распределены по сети и все ЭВМ сети имеют к ним одинаковые права доступа.

Оперативное управление – это управление, обеспечивающее функционирование системы в соответствии с намеченным планом.

Оперативность доступа к информации – это способность информации быть доступной конечному пользователю в соответствии с его оперативными потребностями.

Оператор – это человек-специалист, выполняющий работу по эксплуатации АСУ и непосредственно участвующий в процессе автоматизации управления.

Операционная система – это совокупность системных программ, предназначенных для обеспечения определенного уровня эффективности систем обработки информации, за счет автоматизированного управления её работой и предоставляемого пользователю набора услуг.

Операционная среда – это совокупность интерфейсных средств, предназначенных для управления ходом работы программы и отражения результатов её работы.

Оптимальное управление – это выбор наилучших по некоторому критерию эффективности управляющих воздействий из множества возможных в соответствии с установленной целью управления.

Опытная эксплуатация – это стадия разработки АСУ, ко-

торая проводится совместно специалистами предприятий разработчика и заказчика и имеет целью выявление ошибок разработки и накопление опыта эксплуатации системы.

Организация – это функция управления, устанавливающая постоянные и временные взаимоотношения между всеми элементами системы и определяющая порядок и условия их функционирования.

Отказ – это устойчивое нарушение работоспособности системы, требующее для своего устранения вмешательства оператора.

Открытая система – это система, в которой по крайней мере один элемент имеет связь с внешней средой.

Открытость – это свойство стандартизации доступа к ресурсам системы.

Отложенный синхронный вызов процедур – это комбинация из двух асинхронных вызовов процедур.

Относительная избыточность второго рода – это отношение количества контрольных разрядов к общему количеству разрядов некоторой кодовой комбинации.

Относительная избыточность первого рода – это отношение количества контрольных разрядов к количеству информационных разрядов некоторой кодовой комбинации.

Ошибка отклика – это неадекватный ответ процесса на удаленный запрос, не позволяющий осуществлять распределенную обработку информации.

Ошибка синхронизации – это нарушение временных зависимостей между процессами, не позволяющее осуществлять распределенную обработку информации.

Пакет программ – это система программ для решения задач определенного класса.

Переменная синхронизации – это ассоциированная с хранилищем данных переменная, значение которой эквивалентно номеру процесса, имеющего право изменять данные в хранилище.

Переносимость – это характеристика открытости, показывающая, насколько прикладная программа, разработанная для одной распределенной системы, может без изменения выполняться в другой распределенной системе, реализуя одни и те же интерфейсные средства.

Пилотный процесс – это процесс, выполняющий в распределенной системе главенствующую роль над другими.

Планирование – это выбор целей системой.

Планировщик – это процесс, который определяет, в какой момент времени и какой транзакции разрешается передать операцию чтения или записи менеджеру данных.

Платформа агента – это программная реализация четырехуровневой обобщенной модели программных агентов.

Плоская транзакция – это транзакция, строго удовлетворяющая четырем основным свойствам транзакций.

Подсистема – это выделенное из системы по определенному правилу целенаправленное подмножество элементов любой природы.

Показатель эффективности – это количественная оценка какого-либо отдельного свойства изучаемого объекта или явления.

Политика активизации объекта – это правила обращения к программному объекту.

Политика безопасности – это совокупность норм, правил и практических рекомендаций, на которых строятся управление, защита и распределение информации в АСУ.

Полностью упорядоченная надежная групповая рассылка – это вариант рассылки, который означает, что независимо от того, как упорядочена доставка сообщений для отдельных процессов, сообщения доставляются всем членам группы в одинаковом порядке.

Поломка – это вид отказа, при котором никаких признаков работы распределенной системы или её отдельных компонент не наблюдается.

Поток выполнения – это комплекс информационно-независимых процессов одной прикладной программы, способных работать параллельно без взаимной блокировки.

Прикладная программа – это часть программного обеспечения, предназначенная для решения задачи или класса задач в определенной области применения систем обработки информации.

Причинно упорядоченная надежная групповая рассылка – это надежная групповая рассылка, при которой сообщения доставляются в порядке потенциальной причинной связи между ними.

Программа обслуживания – это системная программа, предназначенная для оказания услуг общего характера пользователям и обслуживающему персоналу систем обработки информации.

Программная ошибка – это ошибка, возникающая в результате некорректного программирования.

Программное обеспечение АСУ – это совокупность программ и программных документов для реализации целей и задач АСУ.

Программный агент – это независимый процесс, обладающий признаками искусственного интеллекта, работающий автономно или совместно с другими агентами, способный своевременно реагировать на изменение в своем окружении и инициировать действия, влияющие на свое окружение.

Программный комплекс – это программа, состоящая из

двух или более программных компонент и (или) программных комплексов, выполняющих взаимосвязанные функции, и применяемая самостоятельно или в составе другого программного комплекса.

Программный компонент – это программа, рассматриваемая как единое целое, выполняющая законченную функцию и применяемая самостоятельно или в составе программного комплекса.

Программный объект – это стандартно оформленный программный модуль, содержащий данные и операции над этими данными.

Программный протокол – это сетевой протокол, который определяет способы взаимодействия программ друг с другом.

Продукция Поста – это принцип взаимооднозначного однонаправленного соответствия между фактами, выраженными каким-либо формализованным способом.

Прозрачность – это свойство сокрытия факта того, что процессы и ресурсы физически распределены по различным ЭВМ сети.

Прозрачность доступа – это вид прозрачности, скрывающий разницу в представлении данных и в способах доступа пользователей к ресурсам.

Прозрачность местоположения – это вид прозрачности, скрывающий от пользователей, где именно физически расположен в системе нужный им ресурс.

Прозрачность отказа – это вид прозрачности, скрывающий от пользователей факт отказа и восстановления системы.

Прозрачность параллельного доступа – это вид прозрачности, скрывающий факт возможного совместного использования ресурса несколькими конкурирующими пользователями.

Прозрачность переноса – это вид прозрачности, скрывающий от пользователей факт перемещения ресурса в другое место системы.

Прозрачность репликации – это вид прозрачности, скрывающий от пользователей тот факт, что в системе существует несколько копий ресурса.

Прозрачность смены местоположения - это вид прозрачности, скрывающий от пользователей факт перемещения ресурса в процессе обработки в другое место системы.

Прозрачность сохранности – это вид прозрачности, скрывающий от пользователей местоположение информационных ресурсов – в оперативной памяти или на долговременных носителях.

Производительность сети ЭВМ - это количество информации пользователей, содержащихся во всех кадрах, обслуженных сетью полностью и с заданным качеством, за единичный интервал времени её функционирования.

Производственный эксперимент – это целенаправленные

действия, преследующие улучшение технологии производства какого-либо изделия.

Произвольная ошибка – это генерация процессом некорректных сообщений, не позволяющая осуществлять распределенную обработку информации.

Прокси-шлюз – это тип брандмауэра, скрывающий работу ряда прикладных программ.

Промышленная эксплуатация – это стадия разработки АСУ, которая проводится специалистами предприятия-заказчика и имеет целью применение системы по прямому предназначению.

Пропуск данных – это отказ в пересылке данных, не позволяющий осуществлять распределенную обработку информации.

Простейшая синхронизация – это вид синхронизации, который осуществляется между дискретным и непрерывным потоками данных.

Простой поток данных – это поток, содержащий только одну последовательность данных.

Пространственная масштабируемость – это вид масштабируемости, определяющий легкость разнесения пользователей и ресурсов в пространстве.

Процедура – это стандартно оформленный программный модуль, доступный для использования другими программами с помощью стандартных операций вызова процедур.

Процесс – это любая работа, выполняемая процессором ЭВМ.

Прямое соединение – это соединение двух сетевых устройств, взаимодействующих непосредственно друг с другом без помощи третьего устройства.

Рабочая станция – это ЭВМ, подключенная в сеть и не являющаяся сервером.

Рабочий проект – это комплект технической документации, содержащий уточненные и детализированные общесистемные проектные решения, программы и инструкции по решению задач, уточненную оценку экономической эффективности АСУ и утвержденный перечень мероприятий по подготовке объекта к внедрению.

Разделяемое устройство – это аппаратный ресурс, непосредственно подключенный в сеть.

Разрешение имени – это процесс поиска информации в распределенной системе.

Распределение – это разбиение информационных ресурсов на части с последующим разнесением этих частей по системе.

Распределенная система – это сеть ЭВМ, ресурсы которой представляются пользователям рабочих станций сети как виртуальная локальная ЭВМ с неограниченными ресурсами.

Распределенная транзакция – это набор плоских транзакций, совместно выполняющих одну общую задачу.

Распределенный объект – это такой объект, интерфейс которого находится на другой ЭВМ, чем сам объект.

Распределенный процесс – это работа микроядра распределенной операционной системы по обработке одного или нескольких потоков выполнения.

Распределенный снимок состояния системы – это совокупность локальных контрольных точек взаимодействующих процессов, работающих на различных ЭВМ.

Реальное моделирование – это моделирование, которое использует возможность исследования различных характеристик изучаемого объекта, процесса или явления либо на реальном объекте целиком, либо на отдельной его части.

Резидентный объект – это программный объект, который продолжает существовать вне зависимости от состояния своего сервера.

Ремонтопригодность – это требование к системе обладать максимальным удобством для осуществления обслуживающим персоналом функций восстановления системы после отказов.

Реплика – это копия ресурса.

Репликация – это создание копии ресурса его владельцем.

Робот – это программируемая техническая система, способная к автономному самоуправлению и выполнению достаточно сложных операций в пространстве и времени, воспринимающая и приспосабливающаяся к изменениям во внешней среде.

Родительский домен - это ближайший домен более высокого уровня для дочернего домена.

Роль – это поименованный набор полномочий

Сбой – это кратковременное самоустраняющееся нарушение работоспособности системы.

Связанный ресурс – это ресурс, который переносится с одной ЭВМ на другую с относительно большими затратами.

Связь – это передача сведений о состоянии объекта и внешней среды в центры управления системой, взаимообмен информацией между этими центрами, а также между системой и внешней средой.

Сеанс – это совместная работа программ на различных ЭВМ.

Сегмент исполнения – это компонента модели переноса кода, которая содержит текущее слово состояния переносимой программы и стек промежуточных результатов вычислений.

Сегмент кода – это компонента модели переноса кода, которая содержит набор исполняемых инструкций переносимого программного модуля.

Сегмент ресурсов – это компонента модели переноса кода, которая содержит ссылки на внешние ресурсы, необходимые для работы переносимого программного модуля.

Сервер – это процесс, реализующий некоторую сетевую службу.

Сервер – это ЭВМ, осуществляющая управление доступом к ресурсам сети.

Сервер единого времени – это ЭВМ, с системными часами которой производится синхронизация системных часов всех ЭВМ распределенной системы.

Серверная заглушка – это специальная версия функции вызова процедуры, работающая на стороне сервера и подменяющая аналогичную стандартную локальную функцию в случае поступления по сети запроса на вызов удаленной процедуры, физически размещенной на данном сервере.

Серверная сеть – это сеть ЭВМ, в составе которой имеются одна или несколько ЭВМ, осуществляющих управление доступом к ресурсам сети.

Сетевая служба – это сетевой компонент, реализующий некоторый набор услуг.

Сетевая технология – это совокупность аппаратных и программных методов передачи информации по сети ЭВМ.

Сетевой протокол – это соглашение между производителями сетевого оборудования и программного обеспечения о способах обмена информацией между ЭВМ.

Сеть ЭВМ – это две или более электронно-вычислительные машины, соединенные между собой для передачи информации.

Символическая модель – это модель, описывающая свойства изучаемого объекта, процесса или явления с помощью системы символов.

Синхронизация – это установление и поддержание строгих временных соотношений между событиями, возникающими в системе и во внешней среде.

Синхронизация артикуляции – это вид синхронизации, который осуществляется между непрерывными потоками данных.

Синхронизация вложенных потоков данных – это поддержание временных соотношений между вложенными потоками данных.

Синхронный вызов процедур – это вариант вызова клиентом удаленной процедуры, когда работа клиентского процесса приостанавливается до получения ответа от удаленной процедуры.

Синхронный режим передачи это режим передачи, в котором для каждого элемента потока данных определяется максимально возможная задержка передачи.

Система – это целенаправленное множество взаимосвязанных элементов любой природы.

Система обработки информации – это совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации.

Система реального масштаба времени – это такая система, на время реакции которой на внешние события наложены жесткие ограничения извне.

Системная ошибка – это ошибка, обусловленная неполнотой информации о реальных процессах внешней среды.

Системная программа – это часть программного обеспечения, предназначенная для поддержания работоспособности системы обработки информации или повышения эффективности её использования в процессе выполнения прикладных программ.

Системный анализ – это всестороннее, систематизированное, то есть построенное на основе определенного набора правил, изучение сложного объекта в целом, проводимое для выяснения возможностей улучшения функционирования этого объекта.

Скелетон – это программа, осуществляющая транзит параметров от операционной системы сервера к методам объекта и обратно.

Слово состояния программы – это машинное слово, служащее для фиксации и индикации состояния триггеров и регистров процессора по отношению к выполняемой программе в какой-либо момент времени.

Служебная тайна – это служебные сведения, доступ к которым ограничен органами государственной власти и федеральными законами.

Сокет Беркли – это абстрактная конечная точка коммуникации, в которую прикладная программа записывает данные, необходимые для передачи по сети, и из которой она может считывать поступающую из сети информацию.

Соподчиненные подсистемы – это подсистемы, принадлежащие одной вертикали.

Составной документ – это набор интерфейсных средств различных типов, предоставляемых различными прикладными программами, которые интегрируются в единый пользовательский интерфейс.

Сохранный объект – это программный объект, который продолжает существовать вне зависимости от состояния своего сервера.

Спецификация передачи – это документ, в котором закрепляются требования к качеству обслуживания.

Способность к взаимодействию – это характеристика открытости, показывающая, насколько две реализации систем или их компонент от разных производителей в состоянии совместно работать, полагаясь только на то, что их интерфейс соответствует стандарту.

Среда передачи данных – это комплекс аппаратных средств, обеспечивающих обмен информацией между ЭВМ.

Среднее время задержки пакета в сети – это среднее по

всем пакетам время от момента передачи пакета отправителем до момента его успешного приема получателем.

Среднее время задержки сообщения пользователя в сети – это среднее по всему множеству сообщений время от момента приема первого бита сообщения от пользователя в рабочей станции-отправителе до передачи последнего бита сообщения рабочей станцией-получателем пользователю.

Среднее время установления соединения – это среднее по всем устанавливаемым соединениям время от момента передачи запроса на соединение отправителем до момента получения им подтверждения о том, что соединение установлено.

Средняя вероятность искажения информации – это средняя вероятность искажения информационных бит кодовой комбинации кадра.

Средняя дисперсия времени задержки сообщения пользователя в сети – это среднее по сети среднеквадратическое отклонение времени задержки сообщения пользователя для осуществления запроса на передачу информации определенного вида и приоритета при заданном методе коммутации.

Статическая группа процессов – это группа, которая имеет постоянный состав процессов.

Статическая модель – это модель, описывающая поведение объекта в какой-либо фиксированный момент времени.

Стационарный случайный процесс - это случайный процесс, в котором вероятности появления случайных событий в одинаковые промежутки времени равны.

Стек – это специализированная область оперативной памяти ЭВМ, доступ к которой осуществляется не по адресу (номеру ячейки), а по очередности поступления в стек информации в соответствии с принципом FILO (First In, Last Out – первым зашел, последним вышел).

Стек протоколов – это набор сетевых протоколов, упорядоченных в виде уровней для реализации коммуникационного процесса.

Стохастическая модель – это модель, отображающая вероятностные процессы и события в моделируемом процессе.

Стратегическое планирование – это планирование, определяющее конечные цели системы.

Структура – это совокупность связей между элементами системы, отражающих их взаимодействие.

Сущность – это все, к чему в распределенной системе можно получить программный доступ.

Тактическое планирование – это планирование, определяющее промежуточные цели и траектории движения системы.

Территориальный объект – это организационно-техническая система, функционирующая на определенном участке земной поверхности или в пространстве относительно зем-

ной поверхности.

Технический проект – это комплект технической документации, содержащий общесистемные проектные решения, алгоритмы решения задач, предварительную оценку экономической эффективности АСУ и примерный перечень мероприятий по подготовке объекта к внедрению.

Техническое задание – это официальный документ, определяющий требования к разрабатываемой системе.

Техническое обеспечение АСУ – это комплекс технических средств, предназначенных для обеспечения работы АСУ.

Технологическая ошибка – это ошибка, возникающая в результате искажения двоичных разрядов при фиксации программ в памяти ЭВМ.

Топология – это метод соединения ЭВМ в сеть.

Транзакция – это сложная многоступенчатая операция, выполняющаяся как единый неделимый единовременный процесс.

Трафик – это сетевой обмен.

Угроза безопасности – это возможные воздействия, способные прямо или косвенно нанести ущерб безопасности системы.

Узел сети – это ЭВМ и общие (разделяемые) устройства, подключенные в сеть.

Упаковщик объектов – это механизм группирования программных объектов в соответствии с политикой активизации каждого из них.

Управление – это воздействия, направленные на поддержание или улучшение функционирования управляемого объекта в соответствии с имеющейся целью управления.

Управление сетью ЭВМ – это все действия, относящиеся к планированию, сооружению и эксплуатации сети, и приводящие к более экономически эффективному использованию ресурсов сети, при сохранении требуемого качества предоставляемых сетью услуг.

Устойчивость – это свойство адаптации системы к изменениям во внешней среде.

Утечка информации – это бесконтрольный выход конфиденциальной информации за пределы организации или круга лиц, которым она была доверена по службе или стала известна в процессе работы.

Ущерб безопасности – это нарушение состояния защищенности информации, хранящейся и обрабатывающейся в системе.

Уязвимость системы – это любая характеристика системы, использование которой может привести к реализации угрозы безопасности.

Физическое моделирование – это моделирование, которое проводится на установках, сохраняющих природу явлений и обладающих физическим подобием.

Фиксированное устройство – это локальный ресурс, программный перенос которого с одной ЭВМ на другую не возможен.

Функциональная часть АСУ – это комплекс административных, организационных и математических методов, обеспечивающих решение задач поддержки принятия управленческих решений и управления физическими объектами.

Целостность данных – это механизм безопасности информации, обеспечивающий защиту сообщения от изменений.

Целостность информации – это свойство информации сохранять свою структуру и содержание в процессе передачи и хранения.

Целостность ресурса – это свойство информационного ресурса быть семантически неизменным при функционировании системы в условиях возможной реализации угроз безопасности информации.

Цифровая сеть интегрального обслуживания – это совокупность информационно-технологических методов и аппаратно-программных средств доставки информации территориально удаленным пользователям, позволяющая на единой цифровой основе обеспечить различные виды информационных услуг.

Шифрование – это замена исходного сообщения его нечитабельным вариантом с целью сохранения конфиденциальности содержащейся в сообщении информации.

Шлюз прикладного уровня – это тип брандмауэра, анализирующий всю информацию пакетов и на основе этого анализа принимающий решение об авторизации пакета.

Шлюз фильтрации пакетов – это тип брандмауэра, анализирующий только заголовки пакетов и принимающий решение об авторизации пакета только на основании адресов отправителя и получателя.

Эвристическая деятельность – это деятельность, выполняемая на основе личного опыта, умений и навыков, опирающаяся на интуицию.

Эксперт – это человек, являющийся признанным специалистом в своей предметной области и на основании собственных знаний и практического опыта способный решать сложные задачи, относящиеся к этой предметной области.

Экспертная система – это комплекс программных средств для выработки рекомендаций по решению трудноформализуемых задач в условиях дефицита времени, противоречивой и недостоверной информации о внешней среде и в непредсказуемых ситуациях на основе обобщенного коллективного опыта экспертов, хранящегося в памяти ЭВМ.

Эргодический процесс - это стационарный случайный процесс, в котором среднее значение функции по времени совпадает со средним значением функции по множеству наблюдений с

вероятностью, равной единице.

Эргономика – это наука, изучающая любые взаимодействия технических систем с внешней средой.

Эскизный проект – это документированное описание вариантов предлагаемой системы.

Юридическая значимость информации – это свойство информации обладать юридической силой.

Языковая модель – это модель, реализуемая при введении фиксированного набора входящих понятий, лишенных неоднозначности определения.

Литература

1. ISO 5807:1985. Обработка информации. Символы, применяемые в документации и обозначения для блок-схем программ. - Москва, изд-во ГК по стандартам, 1987 г.
2. Автоматизированные системы управления в народном хозяйстве. – Москва, изд-во «Экономика», 1987 г.
3. Автоматизированные системы управления, эффективность их использования. Методические рекомендации. – Сумы, 1984 г.
4. Алексеев А.Г., Евсеев Г.А., Симонович С.В. Специальная информатика. Учебное пособие. – Москва, изд-во «АСТ-Пресс», 2000 г.
5. Анфилатов В.С., Емельянов А.А., Кукушкин А.А. Системный анализ в управлении. Учебное пособие для студентов ВУЗов. – Москва, изд-во «Финансы и статистика», 2003 г.
6. Биденко С.И., Лямов Г.В., Яшин А.И. Геоинформационные технологии. Учебное пособие. - Петродворец, изд-во ВМИРЭ, 2004 г.
7. Буассо М., Деманж М., Мюнье Ж.-М. Введение в технологию АТМ. – Пер. с англ., Москва, изд-во «Радио и Связь», 1997 г.
8. Ван Тассел Л. Стиль, разработка, эффективность, отладка и испытание программ. – Пер. с англ., Москва, изд-во «Мир», 1985 г.
9. Волкова С. Р. Пособие по автоматизированным системам управления. – Алма-Ата, изд-во «Мектеп», 1982 г.
10. Выгодский М.Я. Справочник по высшей математике. - Москва, изд-во «Наука», 1997 г.
11. ГОСТ 15971-90г. Системы обработки информации. Термины и определения. - Москва, изд-во ГК по стандартам, 1992 г.
12. ГОСТ 19701-90. Обработка информации. Символы, применяемые в документации и обозначения для блок-схем программ. - Москва, изд-во ГК по стандартам, 1992 г.
13. ГОСТ 19781-90г. Системы обработки информации. Программное обеспечение. Термины и определения. - Москва, изд-во ГК по стандартам, 1992 г.
14. Губарев В. В., Иванов Л.Н. Технические средства и системы информатизации. – Москва, изд-во «Энергоатомиздат», 1989 г.
15. Джеймс Челлис, Чарльз Перкинс, Мэттью Стриб. Основы построения сетей. Учебное руководство для специалистов MCSE. – Пер. с англ., Москва, изд-во «Лори», 1997 г.
16. Доценко С.М., Шпак В.Ф. Методы и средства защиты компьютерной информации. Учебное пособие для курсантов военно-морских институтов. - Петродворец, изд-во ВМИРЭ, 2002 г.
17. Емельянова Н.З., Партыка Т.Л., Попов И.И. Основы по-

строения автоматизированных информационных систем. Учебное пособие для студентов учреждений среднего профессионального образования, обучающихся по специальности 2203 «Программное обеспечение вычислительной техники и автоматизированных систем». – Москва, изд-во «Форум – Инфра-М», 2005 г.

18. Зевеке Г.В., Ионкин П.А., Нетушил А.В., Страхов С.В. Основы теории цепей. Учебник для студентов ВУЗов. – Москва, изд-во «Энергоатомиздат», 1989 г.

19. Зима В.М., Молдовян А.А., Молдовян Н.А. Безопасность глобальных сетевых технологий. – С.-Пб, изд-во «БХВ-Петербург», 2001 г.

20. Зыль С.Н. Операционная система реального времени QNX: от теории к практике.: - 2-е изд., перераб. и доп., С.-Пб, изд-во «БХВ-Петербург», 2004 г.

21. Инженерные расчеты на ЭВМ. Под ред. Троицкого В.А. - Москва, изд-во «Мир», 1985 г.

22. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. - Москва, изд-во «Наука», 1984 г.

23. Лавров А.А., Федоров С.А. Общие положения программного обеспечения вычислительных систем. - Петродворец, изд-во ВВМУРЭ, 1995 г.

24. Липаев В.В. Проектирование программных средств. - Москва, изд-во «Высшая школа», 1990 г.

25. Липаев В.В., Малиновский Б.Н., Слободянюк Т.Ф. Справочник по цифровой вычислительной технике (программное обеспечение). - Киев, изд-во «Техника», 1981 г.

26. Лоптин К.К., Марова Е.Л. Программное обеспечение терминальных устройств ЕС ЭВМ. - Ленинград, изд-во ВМА им. Гречко, 1985 г.

27. Мамиконов А.Г. Основы построения АСУ. – Москва, изд-во «Высшая школа», 1981 г.

28. Математическое обеспечение БИУС пл. Учебник для курсантов военно-морских институтов под ред. Лаврова А.А. - Петродворец, изд-во ВМИРЭ, 2001 г.

29. Многоканальные системы передачи. Учебник для студентов ВУЗов под ред. Баевой Н.Н. и Гордиенко В.Н. - Москва, изд-во «Радио и Связь», 1997 г.

30. Операционная система реального времени QNX Neutrino 6.3. Системная архитектура. - Пер. с англ., С.-Пб, изд-во «БХВ-Петербург», 2006 г.

31. Панов А.В. Разработка управленческих решений: информационные технологии. Учебное пособие для студентов ВУЗов. – Москва, изд-во «Горячая линия – Телеком», 2004 г.

32. Советов Б.Я. АСУ. Введение в специальность. – Москва, изд-во «Высшая школа», 1989 г.

33. Советов Б.Я., Яковлев С.А. Моделирование систем. - Москва, изд-во «Высшая школа», 1985 г.

34. Советов Б.Я., Яковлев С.А. Построение сетей интегрального обслуживания. - Ленинград, изд-во «Машиностроение», 1990 г.
35. Справочник - словарь терминов АСУ. – Москва, изд-во «Радио и связь», 1990 г.
36. Справочник по оптимизационным задачам в АСУ. – Ленинград, изд-во «Машиностроение», 1984 г.
37. Таненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы. – Пер. с англ., С.-Пб, изд-во «Питер», 2003 г.