

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО
ТРАНСПОРТА**

Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Самарский государственный университет путей сообщения» в г.Саратове
Филиал СамГУПС в г. Саратове

КУРС ЛЕКЦИЙ

**ПМ 03 Использование программного обеспечения в процессе эксплуатации
микропроцессорных устройств. МДК 03.01 Технология программирования,
инсталляции и ввода в действие транспортного радиоэлектронного
оборудования (по видам транспорта).**

***Раздел 1. Использование информационных технологий в профессиональной
деятельности.***

для студентов специальности

Техническая эксплуатация транспортного радиоэлектронного оборудования(по
видам транспорта)

РАССМОТРЕНО

на заседании методического совета Протокол
№1 _____ от «21 » 09 2023 г.

СОГЛАСОВАНО

Протокол №1 от «31» 08 2023 г.

Председатель ЦМК 1 1.02.23 Техническая эксплуатация транспортного радиоэлектронного оборудования (по видам транспорта)

_____/Ю.А. Солдатенко/

УТВЕРЖДАЮ

Заместитель директора по учебной работе

_____/С.А. Крижановский/

Автор (составитель):

А.Г. Волков преподаватель филиала СамГУПС в г. Саратове

Рецензент:

А. Б. Соболева — преподаватель высшей квалификационной категории филиала
СамГУПС в г. Саратове

Классификация ИС

Классификация – система распределения объектов (предметов, явлений, процессов, понятий) по классам в соответствии с определенным признаком. Под объектом понимается любой предмет, процесс, явление материального или нематериального свойства. Задача классификации – создать некие удобные образы, позволяющие, например, при выборе систем ограничиться определенным классом или типом.

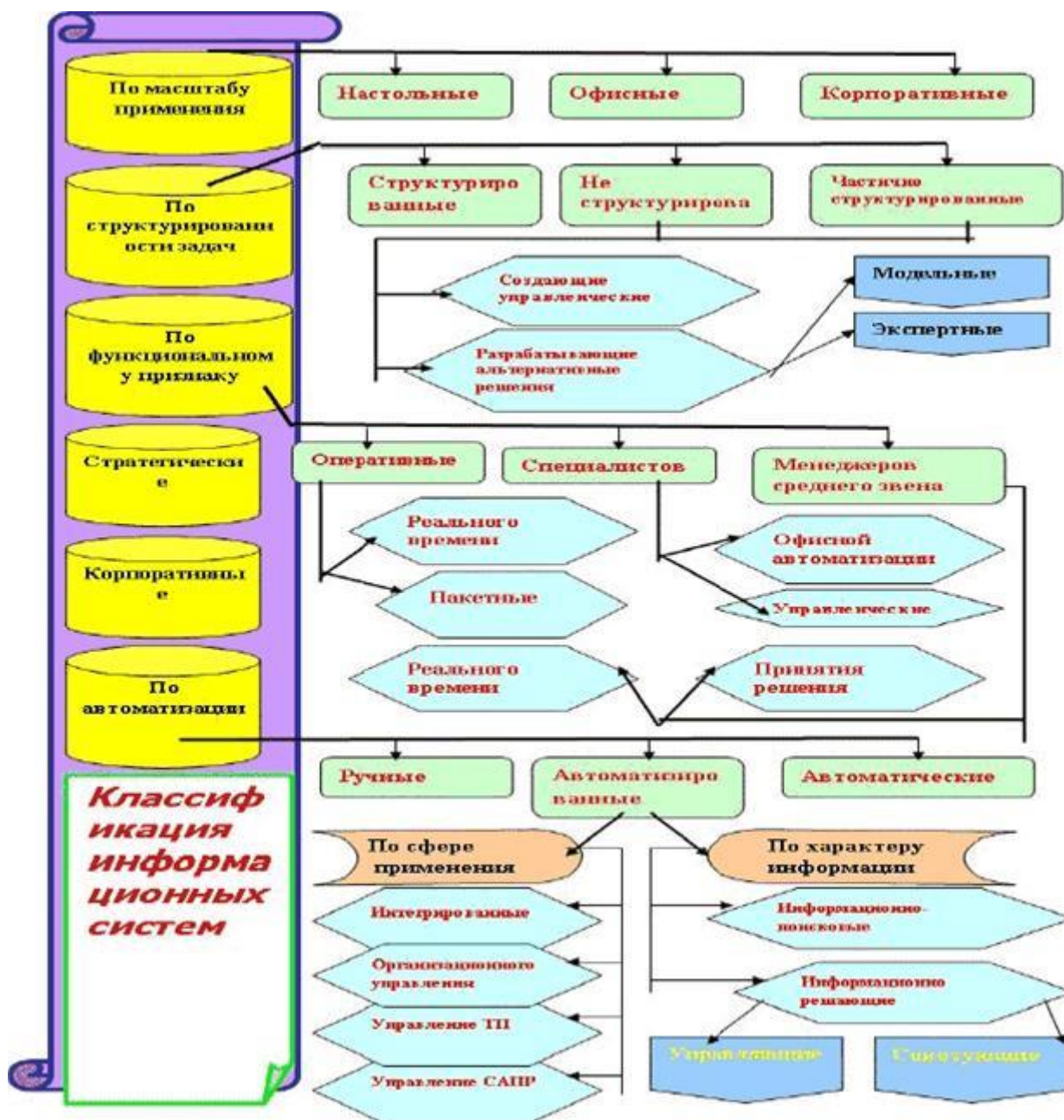
Информационные системы могут значительно различаться по типам объектов, характером и объемом решаемых задач и рядом других признаков.

Общепринятой классификации ИС (информационная система) до сих пор не существует, поэтому их можно классифицировать по разным признаками, что вызвало существование нескольких различных классификаций ИС.

Согласно общепринятой классификации информационные системы подразделяются:

- по масштабам применения – настольные и офисные;
- по признаку структурированности задач – структурированные (формализуемые), не структурируемые (не формализуемые), частично структурируемые. Частично-структурированные делятся на: ИС **репортинга** и ИС разработки альтернативных решений (модельные, экспертные). Экспертные в свою очередь делятся на: централизованные, децентрализованные и коллективного использования; с интеграцией по уровням управления, по уровням планирования и т.д.;
- по функциональному признаку – производственные, маркетинговые (анализа рынка, рекламные, снабженческие и т.п.), финансовые (бухгалтерские, статистические, и т.п.), кадровые;
- по квалификации персонала и уровням управления – стратегические (топ-менеджеров), функциональные (менеджеров среднего звена) и оперативные (специалистов);
- по характеру обработки информации: системы обработки данных, системы управления, система поддержки принятия решений;
- по оперативности обработки данных – пакетной обработки и оперативные;
- по степени автоматизации – ручные, автоматические, автоматизированные;

- по характеру использования информации – на информационно-поисковые, информационно-справочные, информационно-решающие, управляющие, советующие и т.п.;
- по степени централизации обработки информации – на централизованные, децентрализованные, информационные системы коллективного использования;
- по характеру использования вычислительных ресурсов – на локальные и распределенные;
- по сфере деятельности – на государственные, территориальные (региональные), отраслевые, объединений, предприятий или учреждений, технологических процессов;
- по классу реализуемых технологических операций – на системы с текстовыми редакторами, системы с табличными редакторами, СУБД, СУБЗ, системы с графикой, мультимедиа, [гипертекстом](#);
- по месту в процессе управления предприятия – на АРМ специалиста, ИС руководителя, ИС внешнего контролера, интегрированные системы, объединяющие в себе часть или все из этих функций;
- по концепции построения – файловые, автоматизированные банки данных, банки знаний, хранилища данных (ХД);
- по режиму работы – на пакетные, диалоговые и смешанные.



Классификация ИС по масштабности применения.

По масштабам применения современные ИС подразделяются на:

Настольные (одиночные) ИС. Для работы одного человека. К ним можно отнести автоматизированное рабочее место (АРМ) специалиста (конструктора, технолога, расчетчика на прочность, бухгалтера малого предприятия, расчетчика заработной платы и т.д.) ИС этого уровня позволяют специалистам, работающим с данными повысить продуктивность и производительность работы.

Внедрение таких программ не вызывает особых трудностей и для хороших систем может исчисляться днями. Настольные ИС реализуются на автономном компьютере, как правило, ПК. Такая система может содержать несколько простых приложений, связанных общим информационным фондом, и рассчитана на работу

одного пользователя или группы пользователей, разделяющих по времени одно рабочее место. Подобные приложения создаются с помощью так называемых «настольных СУБД» (FoxPro, Paradox, dBase, MS Access) или с помощью файловой системы и диалоговой оболочки для ввода, редактирования и обработки данных.

Офисные (групповые) ИС. Основная цель информатизации офиса – обработка данных, повышение эффективности их работы и упрощение канцелярского труда. Групповые ИС ориентированы на коллективное использование информации членами рабочей группы (одного подразделения). Чаще всего строятся как локальная вычислительная сеть ПК или реже как многотерминальная централизованная вычислительная система.

ИС офисной автоматизации вследствие своей простоты и многопрофильности активно используются работниками любого организационного уровня. Наиболее часто их применяют работники средней квалификации: бухгалтеры, секретари, клерки и т.п. Их деятельность в основном охватывает управление документацией, коммуникации, составление расписаний и т.д. Они позволяют повысить производительность труда секретарей и конторских работников и дают им возможность справляться с возрастающим объемом работ. Однако это преимущество является второстепенным по сравнению с возможностью использования автоматизации офиса в качестве инструмента для решения проблем. Улучшение принимаемых менеджерами решений в результате их более совершенной коммуникации способно обеспечить экономический рост фирмы.

Классификация ИС по концепции построения.

По концепции построения ИС подразделяются на следующие системы:

Файловые системы. Информационное обеспечение построено в виде файловых систем. В современных ЭВМ операционная система берет на себя распределение внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным. Программное обеспечение ИС напрямую использует функции ОС для работы с файлами. Файловые системы обычно обеспечивают хранение слабо структурированной информации, оставляя дальнейшую структуризацию прикладным программам. В таких системах сложно решить проблемы согласования данных в разных файлах, коллективного доступа к данным, модификации структуры данных

Автоматизированные банки данных. Банк данных – это система специальным образом организованных БД, программных, технических, языковых и организационно – методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

В отличие от файловых систем, структура БД меньше зависит от прикладных программ, а все функции по работе с БД сосредоточены в специальном компоненте – системе управления базами данных (СУБД), которая играет центральную роль в функционировании банка данных, так как обеспечивает связь прикладных программ и пользователей, данными. Сведения о структуре БД сосредоточены в словаре-справочнике ([репозитории](#)). Этот вид информации называется метайнформацией. В состав метайнформации входит семантическая информация, физические характеристики данных и информация об их использовании. С помощью словарей данных автоматизируется процесс использования метайнформации в ИС.

Интеллектуальные банки данных (банки знаний, БЗ). Это сравнительно новый способ построения ИС, при котором информация о предметной области условно делится между двумя базами. Если БД содержит сведения о количественных и качественных характеристиках конкретных объектов, то БЗ содержит сведения о закономерностях в ПО, позволяющие выводить новые факты из имеющихся в БД; метайнформацию; сведения о структуре предметной области; сведения, обеспечивающие понимание запроса и синтез ответа.

Если в традиционном банке данных знания о предметной области заложены программистом в каждую прикладную программу, а также в структуру БД, то в интеллектуальном банке данных (ИБД) они хранятся в базе знаний и отделены от прикладных программ. В отличие от данных, знания активны: на их основе формируются цели и выбираются способы их достижения. Например, ИБД в системе складского учета может автоматически реагировать на такое событие, как уменьшение количества деталей на складе до критической нормы, при этом ИБД без участия пользователя генерирует документы для заказа этих деталей и отправляет их по электронной почте поставщику.

Другое характерное отличие знаний от данных – связность, причем знания отражают как структурные взаимосвязи между объектами предметной области, так и вызванные конкретными бизнес-процессами, например такие связи, как «происходит одновременно», «следует из...», «если – то» и др.

Наконец, существенную роль в ИБД играет форма представления информации для пользователя: она должна быть как можно ближе к естественным для человека способам обмена данными (профессиональный естественный язык, речевой ввод / вывод, графическая форма).

Хранилища данных (ХД). В настоящее время в корпоративных БД накоплены гигантские объемы информации, однако она недостаточно эффективно используется в процессе управления бизнесом, поэтому бурно развивается новая форма построения ИС – склады (хранилища) данных.

ХД представляет собой автономный банк данных, в котором база данных разделена на два компонента: оперативная БД хранит текущую информацию, квазипостоянная БД содержит исторические данные, например, в оперативной БД

могут содержаться данные о продажах за текущий год, а в квазипостоянной БД хранятся систематизированные годовые отчеты и балансы за все время существования предприятия. Подсистема оперативного анализа данных позволяет эффективно и быстро анализировать текущую информацию. Подсистема принятия решений пользуется обобщенной и исторической информацией, применяет методы логического вывода. Для общения с пользователем служит универсальный интерфейс.

Классификация ИС по оперативности обработки данных.

По оперативности обработки данных ИС подразделяют на:

Информационные системы пакетной обработки. В чистом виде можно встретить на больших централизованных ЭВМ. Это большинство из ныне существующих ИС. Они решают задачи управления банковскими счетами, учета материальных ценностей, информационного поиска, мониторинга безопасности сетей на основе БД безопасности и т.д.

ИС оперативного (операционного) уровня. Предназначены для аналитической работы с информацией и поддержки специалистов-исполнителей в обработке оперативных данных (счета, накладные, зарплата, кредиты, поток сырья и материалов). Назначение оперативной ИС – при функционировании в режиме реального времени отвечать на запросы о текущем состоянии и отслеживать поток сделок в фирме, что соответствует оперативному управлению. Чтобы с этим справляться, информационная система должна быть легкодоступной, непрерывно действующей и предоставлять точную информацию. Задачи, цели и источники информации на операционном уровне заранее определены и в высокой степени структурированы. Решение запрограммировано в соответствии с заданным алгоритмом.

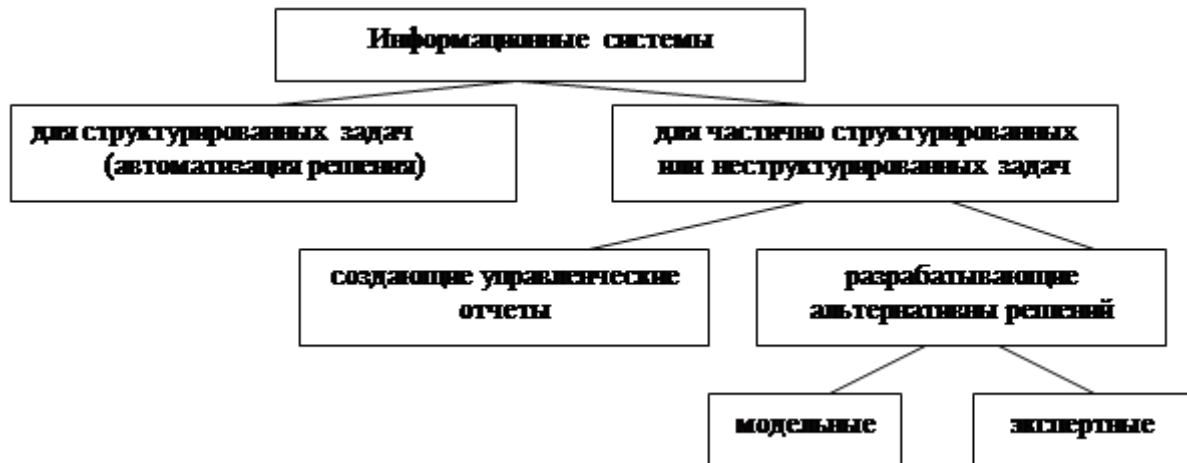
ИС оперативного уровня является связующим звеном между фирмой и внешней средой. Если система работает плохо, то организация либо не получает информации извне, либо не выдает информацию. Кроме того, система – это основной поставщик информации для остальных типов ИС в организации, так как содержит и оперативную, и архивную информацию. Отключение этой ИС привело бы к необратимым негативным последствиям.

Классификация ИС по признаку структурированности задач.

При создании или при классификации ИС неизбежно возникают проблемы, связанные с формальным – математическим и алгоритмическим описанием решаемых задач.

Степень формализации - это степень математического описания задачи, от которой, во многом зависит эффективность работы всей системы, а также уровень

автоматизации, определяемый степенью участия человека при принятии решения на основе получаемой информации. Чем точнее математическое описание задач, тем выше возможности компьютерной обработки данных и тем меньше степень участия человека в процессе ее решения. Это и определяет степень автоматизации задачи.



Разл

ичают три типа задач, для которых создаются ИС: структурированные (формализуемые); не структурируемые (не формализуемые); частично структурируемые.

Классификация ИС по признаку структурированности решаемых задач:

Структурированные задачи. Содержание может быть выражено в форме математической модели, имеющей алгоритм решения. Подобные задачи обычно приходится решать многократно, и они носят рутинный характер (например, расчет на прочность стандартизированных деталей). Целью использования информационной системы для решения структурированных задач является полная автоматизация их решения, т.е. сведение роли человека к нулю.

Например, в ИС необходимо реализовать задачу расчета заработной платы. Это структурированная задача, где полностью известен алгоритм решения. Рутинный характер этой задачи определяется тем, что расчеты всех начислений и отчислений весьма просты, но объем их очень велик, так как они должны многократно повторяться ежемесячно для всех категорий работающих.

Неструктурированные задачи. Это задачи, в которых решение связано с большими трудностями из-за невозможности создания математического описания и разработки алгоритма. Возможности использования здесь информационной системы невелики. Решение в таких случаях принимается человеком из эвристических соображений на основе своего опыта и, возможно, косвенной информации из разных источников.

Например, формализация взаимоотношений в студенческой группе. Задача, не решаемая в связи с тем, что для нее существен психологический и социальный факторы, которые очень сложно описать алгоритмически.

Частично структурированные задачи. Это задачи, в которых известна лишь часть их элементов и связей между ними. В практике работы любого предприятия существует сравнительно немного полностью структурированных или совершенно неструктурированных задач. На большинстве предприятий персонал сталкивается с частично структурированными задачами.

Например, требуется принять решение по устранению ситуации, когда потребность в трудовых ресурсах для выполнения в срок одной из работ комплекса превышает их наличие. Пути решения этой задачи могут быть разными, например: выделение дополнительного финансирования на увеличение численности работающих; отнесение срока окончания работы на более позднюю дату и т.д. В данной ситуации ИС может помочь человеку принять то или иное решение, если снабдит его информацией о ходе выполнения работ по всем необходимым параметрам.

ИС, используемые для решения частично структурированных задач, обычно подразделяются на два вида:

- создание отчета (**репортинг**). Создание отчета путем обработки данных (поиск, сортировку, агрегирование, фильтрацию). Используя сведения, содержащиеся в этих отчетах, специалист принимает решение. ИС, создающие отчеты обеспечивают информационную поддержку пользователя, т.е. предоставляют доступ к информации БД и ее частичную обработку.

- ИС, разрабатывающие альтернативы решения. Принятое решение при этом сводится к выбору одной из предложенных альтернатив. ИС, разрабатывающие альтернативы решений, могут быть модельными или экспертными.

Экспертные информационные системы. Обеспечивают выработку и оценку возможных альтернатив пользователем за счет создания ИС, связанных с обработкой знаний. Экспертные системы основаны на использовании искусственного интеллекта, и дают возможность менеджеру или специалисту получать консультации экспертов по любым проблемам, о которых этими системами накоплены знания.

Подразделяются:

- по степени централизации обработки – на информационно-централизованные, децентрализованные, информационной системы коллективные использования;
- по степени интеграции функций – многоуровневые ИС с интеграцией за уровнями управления (цеха – производство, управления - предприятие,

предприятие - объединение, объединение – корпорация и т.д.), многоуровневые с интеграцией по уровням планирования и т.п.

Модельные информационные системы. Модельные ИС предоставляют пользователю математические, статистические, финансовые и другие модели, использование которых облегчает выработку и оценку альтернатив решения. Пользователь может получить недостающую ему для принятия решения информацию путем установления диалога с моделью в процессе ее исследования.

Классификация ИС по сфере деятельности.

ИС подразделяют на следующие сферы деятельности.

Государственные информационные системы. Государственные ИС предназначены для решения важнейших народнохозяйственных проблем страны. На базе использования вычислительных комплексов и экономико-математических методов в них составляют перспективные и текущие планы развития страны, ведут учет результатов и регулируют деятельность отдельных звеньев народного хозяйства, разрабатывают Государственный бюджет, контролируют его выполнения и т.п. К ним относятся автоматизированная система государственной статистики (АСДС), автоматизированная система плановых расчетов (АСПР), государственная ИС финансовых расчетов (АСФР) при Министерстве финансов Украины, система обработки информации из цен (АСОИ цен), система управления национальным банком АСУ банк), система обработки научно-технической информации (АСО НТИ), система управления Национальным банком (АСУ банк), система обработки научно-технической информации (АСО НТИ) и другие.

Территориальные (региональные) ИС. Территориальные ИС предназначены для управления административно-территориальным регионом. Сюда относятся ИСУ области, города, района. Эти системы обрабатывают информацию, которая необходима для реализации функций управления регионом, формирования отчетности и выдачи оперативных данных местным и руководящим государственным и хозяйственным органам.

Отраслевые информационные системы управления (ИСУ). Отраслевые ИСУ предназначены для управления подведомственными предприятиями и организациями. Отраслевые ИСУ действуют в промышленности и сельском хозяйстве, строительстве на транспорте и др. В них решаются задачи информационного обслуживания аппарата управления отраслевых министерств и их подразделов. Отраслевые ИС различаются по сферам применения – промышленная, непромышленная, научная и т.п.

Классификация ИС по режиму работы.

ИС подразделяют по следующим режимам работы:

Пакетные информационные системы. Пакетные ИС работают в пакетном режиме: вначале данные накапливаются, и формируется пакет данных, а затем пакет последовательно обрабатывается рядом программ. Недостаток этого режима – низкая оперативность принятия решений и обособленность пользователя от системы.

Диалоговые информационные системы. Диалоговые ИС работают в режиме обмена сообщениями между пользователями и системой (например, система продажи авиабилетов). Этот режим особенно удобен, когда пользователь может выбирать перспективные варианты из числа предлагаемых системой.

Классификация ИС по степени автоматизации.

В зависимости от степени автоматизации информационных процессов ИС определяются как ручные, автоматические, автоматизированные.

Ручные ИС. Ручные ИС характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной ИС.

Автоматизированные информационные системы (АИС). Автоматизированные ИС (АИС) предполагают участие в процессе обработки информации и человека, и технических средств, причем главная роль отводится компьютеру. В современном толковании в термин «ИС» вкладывается обязательно понятие автоматизируемой системы. Автоматизированные ИС, учитывая их широкое использование в организации процессов управления, имеют различные модификации и могут быть классифицированы, например, по характеру использования информации и по сфере применения.

Например, роль расчетчика деталей и узлов машин на прочность в информационной системе заключается в создании расчетной схемы нагрузок и исходных данных. ИС обрабатывает их по заранее известному алгоритму с выдачей результатной информации в виде расчетно-пояснительной записки, напечатанной на принтере.

Основу АИС составляет банк данных, в котором хранится большая по объему информация о какой-либо области человеческих знаний. Территориально этот банк может быть распределенным. Важно, что для пользователя этот банк представляется, как единое хранилище информации, куда он может обратиться с запросом.

Автоматические ИС. Автоматические информационные системы, в которых автоматизация является полной, то есть вмешательство персонала не требуется или требуется только эпизодически.



Классификация ИС по степени автоматизации информационных процессов

Классификация ИС по функциональности.

Функциональный признак определяет назначение системы, а также ее основные цели, задачи и функции. Структура ИС может быть представлена как совокупность ее функциональных подсистем, поэтому функциональный признак может быть использован при классификации ИС.

Тип ИС зависит от того, чьи интересы она обслуживает и на каком уровне управления. На рисунке показан вариант классификации ИС по функциональному признаку с учетом уровней управления и уровней квалификации персонала.



Типы ИС в зависимости от функционального признака с учетом уровней управления и квалификации персонала

Из рисунка видно, что чем выше по значимости уровень управления, тем меньше объем работ, выполняемых специалистом и менеджером с помощью ИС. Однако при этом возрастают сложность и интеллектуальные возможности ИС, и ее роль в принятии менеджером решений. Любой уровень управления нуждается в информации из всех функциональных систем, но в разных объемах и с разной степенью обобщения

Основание пирамиды составляют ИС, с помощью которых сотрудники-исполнители занимаются операционной обработкой данных, а менеджеры низшего звена – оперативным управлением. Наверху пирамиды на уровне стратегического управления ИС изменяют свою роль и становятся стратегическими, поддерживающими деятельность менеджеров высшего звена по принятию решений в условиях плохой структурированности поставленных задач.

В хозяйственной практике производственных и коммерческих объектов типовыми видами деятельности, которые определяют функциональный признак классификации ИС, являются:

- производственная, связана с непосредственным выпуском продукции и направлена на создание и внедрение в производство научно-технических новшеств;
- маркетинговая, которая включает в себя: анализ рынка производителей и потребителей выпускаемой продукции, анализ продаж; организацию рекламной кампании по продвижению продукции; рациональную организацию материально-технического снабжения;
- финансовая, связанная с организацией контроля и анализа финансовых ресурсов фирмы на основе бухгалтерской, статистической, оперативной информации;
- кадровая, направленная на подбор и расстановку необходимых фирме специалистов, а также ведение служебной документации по различным аспектам.

Указанные направления деятельности определили типовой набор ИС:

- производственные системы;
- системы [маркетинга](#);
- финансовые и учетные системы;
- системы кадров (человеческих ресурсов);
- прочие типы, выполняющие вспомогательные функции в зависимости от специфики деятельности фирмы.

В крупных фирмах основная ИС функционального назначения может состоять из нескольких подсистем для выполнения подфункций.

Например, подсистемы производственной информационной системы:

- конструкторской подготовки производства;
- технологической подготовки производства;
- управления материально-техническим снабжением;
- управления производственным процессом;
- компьютерного инжиниринга и т.д.

Классификация ИС по квалификации персонала и управления.

ИС подразделяют на следующие уровни:

Информационные системы оперативного уровня. Информационные системы оперативного уровня помогают специалистам, работающим с данными, повышают продуктивность и производительность работы. Задача – интеграция новых сведений в организацию и помощь в обработке бумажных документов.

В этом классе ИС можно выделить две группы ИС:

- Системы офисной автоматизации. Вследствие своей простоты и многопрофильности активно используются работниками любого организационного уровня. Наиболее часто их применяют работники средней квалификации: бухгалтеры, секретари, клерки. Основная цель – обработка данных, повышение эффективности их работы и упрощение канцелярского труда. Их деятельность в основном охватывает управление документацией, коммуникации, составление расписаний и т.д.
- Системы обработки знаний. Эти ИС вбирают в себя знания, необходимые инженерам, юристам, ученым при разработке или создании нового продукта. Их работа заключается в создании новой информации и нового знания. Так, например, существующие специализированные рабочие станции по инженерному и научному проектированию позволяют обеспечить высокий уровень технических разработок.

Информационные системы функционального уровня. ИС менеджеров среднего звена используются для мониторинга (постоянного слежения), контроля, принятия решений и администрирования. Основные функции этих ИС: сравнение текущих показателей с прошлыми; составление периодических отчетов за определенное время, а не выдача отчетов по текущим событиям, как на оперативном уровне; обеспечение доступа к архивной информации ит.д.

Диапазон программ, применяемых для систем этого уровня весьма широк.

Среди них можно выделить следующие основные классы:

- анализ маркетинга. Моделирование стратегии, анализ положения компании на рынке, разработка плана маркетинга
- анализ продаж. Информационная поддержка и анализ процесса продаж, моделирование каналов сбыта.

Информационные системы стратегического уровня. Эти системы предназначены для топ-менеджеров, поэтому для них существует второе, даже более часто применяемое название КИС (корпоративные информационные системы). Помогают высшему звену управленцев решать неструктурированные задачи осуществлять долгосрочное планирование. Развитие и успех любого предприятия, организации (фирмы) во многом определяются принятой в ней стратегией. Стратегия – это набор методов и средств решения перспективных долгосрочных задач.

В настоящее время возрос интерес к развитию стратегических систем. Они стали расцениваться как стратегически важные системы, которые влияют на изменение выбора целей фирмы, ее задач, методов, продуктов, услуг, позволяя опередить конкурентов, а также наладить более тесное взаимодействие с потребителем и поставщиками.

Основная задача стратегических ИС – сравнение происходящих во внешнем окружении изменений с существующим потенциалом организации. Они призваны создать общую среду компьютерной и телекоммуникационной поддержки решений в неожиданно возникающих ситуациях. Используя самые совершенные программы, эти системы способны в любой момент предоставить информацию из многих источников. Для некоторых стратегических систем характерны ограниченные аналитические возможности. На данном организационном уровне ИС играют вспомогательную роль и используются как средство предоставления менеджеру необходимой информации для принятия решений.

Например, в качестве стратегической цели определено повышение качества выпускаемой продукции. Взаимосвязь предприятия с внешним окружением обуславливается использованием ИС в условиях конкуренции с другими фирмами, выпускающими продукцию аналогичную уже имеющуюся на потребительском рынке.

Для того, чтобы понять эффект от использования ИС, нужно понять взаимосвязь фирмы с ее внешним окружением.

На фирму оказывают воздействие следующие внешние факторы:

- конкуренты, проводящие на рынке свою политику;

- покупатели, обладающих разными возможностями по приобретению товаров и услуг;
- поставщики, которые проводят свою ценовую политику.

Фирма может обеспечить себе конкурентное преимущество, если будет учитывать эти факторы и придерживаться следующих стратегий:

- создание новых товаров и услуг, которые выгодно отличаются от аналогичных;
- отыскание рынков, где товары и услуги фирмы обладают рядом отличительных признаков по сравнению с уже имеющимися там аналогами;
- создание таких связей, которые закрепляют покупателей и поставщиков за данной фирмой и делают невыгодным обращение к другой;
- снижение стоимости продукции без ущерба качества.

Классификация ИС по способу распределения вычислительных ресурсов.

Подразделяются на:

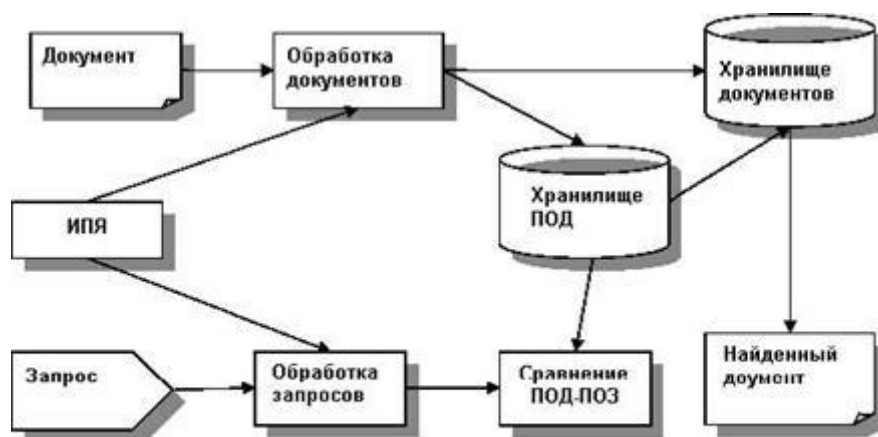
Локальные информационные системы. Локальные ИС используют одну ЭВМ и предназначены для автоматизации отдельных функций управления на отдельных уровнях управления. Такая ИС может быть однопользовательской, функционирующей в отдельных подразделениях системы управления.

Распределенные ИС. В распределенных ИС взаимодействуют несколько ЭВМ, связанных сетью. Отдельные узлы сети обычно территориально удалены друг от друга, решают разные задачи, но используют общую информационную базу.

Классификация ИС по характеру использования информации.

Все ИС, классифицируемые по характеру используемой информации, относятся к автоматизированным информационным системам (АИС).

Информационно-поисковые системы (ИПС). Информационно-поисковые системы ориентированы на решение задач поиска информации, документа или факта в множестве источников информации (документов). Содержательная обработка информации в таких системах отсутствующая. Производят ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных. Например, информационно-поисковая система в библиотеке билетов. В таких системах хранится информационный массив, из которого по требованиям пользователей выдается нужная информация. Поиск информации по требованию пользователя осуществляется либо автоматически, либо вручную.



Структура информационных потоков при поиске документов в информационно-поисковой системе

Поисковый образ документа (ПОД) получается в результате процесса индексирования, который состоит из двух этапов: выявление смысла документа и описание смысла на специальном информационно-поисковом языке (ИПЯ). Запрос к ИПС описывается также на этом языке. Поиск документа состоит в сравнении множества хранящихся в системе ПОД и текущего поискового образа запроса (ПОЗ), в результате чего пользователю выдается требуемый документ или отказ.

Информационно-поисковые системы делятся на два типа:

- **Документальные** (документографические). Это системы, в которой объектом сохранения и обработки есть собственно документы.. В такой ИПС все хранимые документы индексируются некоторым специальным образом. Каждому документу (статье, отчету, протоколу и т.п.) присваивается индивидуальный код, составляющий поисковый образ документа. Поиск идет не по самим документам, а по их поисковым образам, которые содержат информацию (адрес) о местонахождении документа. Именно так ищут книги по заказам читателя в больших библиотеках (в маленьких библиотеках библиотекарь обычно ищет книги сам).

По требованию читателя сначала находят карточку в каталоге, а потом по шифру, указанному на ней, отыскивается и сама книга. Различия документографических ИПС определяются тем, как устроен поисковый образ документа. В простейшем случае это просто его индивидуальное название (например, название, автор, год издания книги). В более сложных случаях нет однозначного соответствия между поисковым образом документа и самим документом. Вполне возможен случай, когда поисковый образ документа соответствует нескольким различным документам и, наоборот, один и тот же документ соответствует не одному, а нескольким поисковым образам.

- **Фактографическая информационная поисковая система (ИПС)**. Это система, где, объектом или сущностью есть то, что представляет для проблемной сферы многосторонний интерес (сотрудник, договор, изделие и т.п.). Ведомости об этих

сущностях могут находиться во множестве разных входных и исходных сообщений. В отличие от документографических ИПС в ИПС такого типа хранятся не документы, а факты, относящиеся к какой-либо предметной области.

Хранимые факты могут быть извлечены из различных документов. В базе фактов они связываются между собой системой разнообразных отношений. Такая сеть в ИПС носит название тезауруса предметной области. Запросы, поступающие в фактографические ИПС, используют тезаурус для поиска ответов на запросы. Поиск осуществляется методом поиска по образцу, широко применяющемуся в базах знаний систем искусственного интеллекта. ИПС фактографического типа постепенно приближаются по своей организации и функционированию к развитым базам данных и знаний.

Информационно-решающие системы. В таких ИС по результатам поиска вычисляют значения арифметических функций. Осуществляют все операции переработки информации по определенному алгоритму. Среди них можно провести классификацию по степени воздействия выработанной совместной результатной информации на процесс принятия решений и выделить два класса:

- **Управляющие ИС.** Информационно-управляющие, или управленческие, системы (известные в отечественной литературе под названием «автоматизированной системы организационные управления») представляют собой организационно-технической системы, которые обеспечивают получение решения на основе автоматизации информационных процессов в сфере управления, на основе которой человек принимает решение. Итак, они предназначены для автоматизированного решения широкого круга задач управления. Для этих систем характерны тип задач расчетного характера и обработка больших объемов данных. Примером могут служить система оперативного планирования выпуска продукции, система бухгалтерского учета.

- **Советующие ИС.** Вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, так как для них характерна обработка знаний, а не данных. Например, существуют медицинские ИС для постановки диагноза больного и определения предполагаемой процедуры лечения. Врач при работе с подобной системой может принять к сведению полученную информацию, но предложить иное по сравнению с рекомендуемым решение.

Классификация ИС по характеру обработки информации.

В соответствии с характером обработки информации в ИС на различных уровнях управления экономической системой (оперативном, тактическом и стратегическом) выделяются несколько типов ИС.

Системы обработки данных (СОД). (EDP – Electronic Data Processing, СОД) предназначены для учета и оперативного регулирования хозяйственных операций,

подготовки стандартных документов для внешней среды (счетов, накладных, платежных поручений, расчета заработной платы, статистической отчетности и т.п.). Такие системы наряду с функциями ввода, выборки, коррекции информации выполняют математические расчеты без применения методов оптимизации. Горизонт оперативного управления хозяйственными процессами составляет от одного до несколько дней и реализует регистрацию и обработку событий (оформление и мониторинг выполнения заказов, приход и расход материальных ценностей на складе, ведение табеля учета рабочего времени и т.д.). Эти задачи имеют итеративный, регулярный характер, выполняются непосредственными исполнителями хозяйственных процессов (рабочими, кладовщиками, администраторами и т.д.) и связаны с оформлением и пересылкой документов в соответствии с четко определенными алгоритмами. Результаты выполнения хозяйственных операций через экранные формы вводятся в базу данных.

Информационные системы управления (ИСУ). (MIS – Management Information System, ИСУ) ориентированы на тактический уровень управления: среднесрочное планирование, анализ и организацию работ в течение нескольких недель (месяцев), например анализ и планирование поставок, сбыта, составление производственных программ. Для данного класса задач характерны регламентированность (периодическая повторяемость) формирования результатных документов и четко определенный алгоритм решения задач, например свод заказов для формирования производственной программы и определение потребности в комплектующих деталях и материалах на основе спецификации изделий. Решение подобных задач предназначено для руководителей различных служб предприятий (отделов материально-технического снабжения и сбыта, цехов и т.д.). Задачи решаются на основе накопленной базы оперативных данных.

Системы поддержки принятия решений (СППР). (DSS – Decision Support System, СППР) используются в основном на верхнем уровне управления (руководства фирм, предприятий, организаций), имеющего стратегическое долгосрочное значение в течение года или нескольких лет. К таким задачам относятся формирование стратегических целей, планирование привлечения ресурсов, источников финансирования, выбор места размещения предприятий и т.д. Реже задачи класса СППР решаются на тактическом уровне, например при выборе поставщиков или заключении контрактов с клиентами. Задачи СППР имеют, как правило, нерегулярный характер. Для задач СППР свойственны недостаточность имеющейся информации, ее противоречивость и нечеткость, преобладание качественных оценок целей и ограничений, слабая формализованность алгоритмов решения. В качестве инструментов обобщения чаще всего используются средства составления аналитических отчетов произвольной формы, методы статистического анализа, экспертных оценок и систем, математического и имитационного моделирования. При этом используются базы обобщенной информации, информационные хранилища, базы знаний о правилах и моделях принятия решений.

Идеальной считается ИС, которая включает все три типа перечисленных ИС.

Контрольные вопросы для проверки знаний по теме:

1. Каковы тенденции в развитии информатизации в обществе?
2. Назовите основные черты информационного общества.
3. Что такое система?
4. Каковы особенности рынка средств информатизации в России?
5. В чем состоят назначение и необходимость каждой из обеспечивающих подсистем ИС?
6. По каким признакам можно классифицировать ИС?
7. Что понимается под функциональной подсистемой ИС? Приведите примеры.
8. Назовите существующие принципы создания ИС.
9. Что такое система автоматизированного проектирования?
10. Что такое экспертные системы? Назовите области их применения.

Тест 1

Вопрос 1. Что такое информационное общество?

- А. человеческое общество;
- Б. Российское общество;
- В. общество, в котором большинство работающих занято преобразованием информации.

Вопрос 2. Под информатизацией Российского общества понимается?

- А. модернизация информационно-телекоммуникационной инфраструктуры в России;
- Б. организованный процесс создания оптимальных условий для удовлетворения информационных потребностей физических и юридических лиц на основе формирования и использования информационных ресурсов;
- В. обучение и подготовка к жизни и работе.

Вопрос 3. Информационный кризис это?

- А. противоречие между ограниченными возможностями человека по восприятию и обработке информации и нарастающими ее потоками;
- Б. рост абсолютной численности управленческого персонала при невозможности оперативно обработать возникающий объем учетных данных;
- В. возрастание информационных потоков при низком качестве учебного процесса в школах.

Вопрос 4. Что такое информационный ресурс?

- А. документированные знания для создания информационных продуктов и предоставления информационных услуг;
- Б. результат интеллектуальной деятельности человека;
- В. сырье для деятельности информационной индустрии.

Вопрос 5. Что такое информационный продукт?

- А. информационная услуга, предоставляемая пользователю;
- Б. результат интеллектуальной деятельности человека;
- В. доведение до пользователя сведений.

Вопрос 6. Информационный рынок – это?

- А. система экономических, правовых и организационных отношений по торговле продуктами интеллектуального труда на коммерческой основе;
- Б. производство, продажа и покупка ЭВМ и устройств ЭВМ при активном государственном регулировании;
- В. предоставление платных сетевых услуг, прежде всего, через Интернет.

Вопрос 7. Составляющими информационного рынка являются?

- А. технические, программные средства и информационные технологии;
- Б. справочные средства о поставщиках информационных продуктов и услуг, а также информационно-правовые документы по информации;
- В. все вышеперечисленное.

Вопрос 8 Что такое система?

А. совокупность людей, находящихся в пределах национальных границ того или иного государства;

Б. совокупность взаимосвязанных элементов, подчиненных единой цели;

В. некая общность элементов, объединенных в пространстве и во времени.

Вопрос 9. Какие функции управления реализует система управления?

А. учет, контроль и регулирование;

Б. планирование, анализ и учет;

В. планирование, учет, анализ, контроль и регулирование.

Вопрос 10. По уровню в системе государственного управления выделяют такие ИС?

А. отраслевые, территориальные и межотраслевые;

Б. банков, финансовых органов и статистики;

В. управленческие и обучающие.

Вопрос 11. Главный признак, отличающий одно поколение ЭВМ от другого?

А. технико-эксплуатационные параметры (быстродействие, объемы памяти и др.);

Б. элементная база;

В. вид сборки (красная, белая или желтая).

Вопрос 12. CASE-технологии – это?

А. совокупность методов обучения персонала, анализа и прогнозирования развития организации;

Б. совокупность методов анализа, проектирования, разработки и сопровождение АИС с максимальной автоматизацией процессов разработки и функционирования систем;

В. комплекс приемов совершенствования управления организации на базе внедрения новой информационной технологии.

Вопрос 13. Сколько этапов можно выделить в развитии ИС?

А. пять;

Б. четыре;

В. три.

Классификация информационных систем по признаку структурированности задач

Классификация **информационных систем** по признаку структурированности задач

Понятие структурированности задач

При создании или при классификации информационных систем неизбежно возникают проблемы, связанные с формальным - математическим и алгоритмическим описанием решаемых задач. От степени формализации во многом зависят эффективность работы всей системы, а также уровень автоматизации, определяемый степенью участия человека при принятии решения на основе получаемой информации.

Чем точнее математическое описание задачи, тем выше возможности компьютерной обработки данных и тем меньше степень участия человека в процессе ее решения. Это и определяет степень автоматизации задачи.

Различают три типа задач, для которых создаются информационные системы:

- структурированные (формализуемые),
- неструктурированные (не формализуемые),
- частично структурированные.

Структурированная (формализуемая) задача - задача, где известны все ее элементы и взаимосвязи между ними.

Неструктурированная (не формализуемая) задача - задача, в которой невозможно выделить элементы и установить между ними связи.

В структурированной задаче удастся выразить ее содержание в форме математической модели, имеющей точный алгоритм решения. Подобные задачи обычно приходится решать многократно, и они носят рутинный характер. Целью использования информационной системы для решения структурированных задач является полная автоматизация их решения, т. е. сведение роли человека к нулю.

Пример. В информационной системе необходимо реализовать задачу расчета [заработной платы](#).

Это структурированная задача, где полностью известен алгоритм решения. Рутинный характер этой задачи определяется тем, что расчеты всех начислений и отчислений весьма просты, но объем их очень велик, так как они должны многократно повторяться ежемесячно для всех категорий работающих.

Решение неструктурированных задач из-за невозможности создания математического описания и разработки алгоритма связано с большими трудностями. Возможности использования здесь информационной системы невелики. Решение в таких случаях принимается человеком из эвристических соображений на основе своего опыта и, возможно, косвенной информации из разных источников.

Пример. Попробуйте формализовать [взаимоотношения](#) в вашей студенческой группе. Наверное, вряд ли вы сможете это сделать. Это связано с тем, что для данной задачи существен психологический и социальный факторы, которые очень сложно описать алгоритмически.

Заметим, что в практике работы любой организации существует сравнительно немного полностью структурированных или совершенно неструктурированных задач. О большинстве задач можно сказать, что известна лишь часть их элементов и связей между ними. Такие задачи называются частично структурированными. В этих условиях можно создать информационную систему. Получаемая в ней информация анализируется человеком, который будет играть определяющую роль. Такие информационные системы являются автоматизированными, так как в их функционировании принимает участие человек.

Пример. Требуется принять решение по устранению ситуации, когда потребность в трудовых ресурсах для выполнения в срок одной из работ комплекса превышает их наличие. Пути решения этой задачи могут быть разными, например:

- выделение дополнительного финансирования на увеличение численности работающих;

- отнесение срока окончания работы на более позднюю дату и т. д. Как видно, в данной ситуации информационная система может помочь человеку принять то или иное решение, если снабдит его информацией о ходе **выполнения работ** по всем необходимым параметрам.

Типы информационных систем, используемые для решения частично структурированных задач

Информационные системы, используемые для решения частично структурированных задач, подразделяются на два вида:

- создающие управленческие отчеты и ориентированные главным образом на обработку данных (поиск, сортировку, **агрегирование**, фильтрацию). Используя сведения, содержащиеся в этих отчетах, управляющий принимает решение;
- разрабатывающие возможные альтернативы решения.

Принятие решения при этом сводится к выбору одной из предложенных альтернатив.

Информационные системы, создающие управленческие отчеты, обеспечивают информационную поддержку пользователя, т. е. предоставляют доступ к информации в **базе данных** и ее частичную обработку. Процедуры манипулирования данными в информационной системе должны обеспечивать следующие возможности:

- составление комбинаций данных, получаемых из различных источников;
- быстрое добавление или исключение того или иного источника данных и автоматическое переключение источников при поиске данных;
- управление данными с использованием возможностей систем управления базами данных;
- логическую независимость данных этого типа от других баз данных, входящих в подсистему **информационного обеспечения**;
- автоматическое отслеживание потока информации для наполнения баз данных.

Информационные системы, разрабатывающие альтернативы решений, могут быть **модельными или экспертными**.

Модельные информационные системы предоставляют пользователю математические, статистические, финансовые и другие модели, использование которых облегчает выработку и оценку альтернатив решения. Пользователь может получить недостающую ему для принятия решения информацию путем установления диалога с моделью в процессе ее исследования.

Основными функциями модельной информационной системы являются:

- возможность работы в среде типовых математических моделей, включая решение основных задач моделирования типа "как сделать, чтобы?", "что будет, если?", анализ чувствительности и др.;
- достаточно быстрая и адекватная интерпретация результатов моделирования;
- оперативная подготовка и корректировка входных параметров и ограничений модели;
- возможность графического отображения динамики модели;

- возможность объяснения пользователю необходимых шагов формирования и работы модели.

Экспертные информационные системы обеспечивают выработку и оценку возможных альтернатив пользователем за счет создания экспертных систем, связанных с обработкой знаний. Экспертная поддержка принимаемых пользователем решений реализуется на двух уровнях.

Работа первого уровня экспертной поддержки исходит из концепции "типовых управленческих решений", в соответствии с которой часто возникающие в процессе управления проблемные ситуации можно свести к некоторым однородным классам управленческих решений, т. е. к некоторому типовому набору альтернатив. Для реализации экспертной поддержки на этом уровне создается информационный фонд хранения и анализа типовых альтернатив. Если возникшая проблемная ситуация не ассоциируется с имеющимися классами типовых альтернатив, в работу должен вступать второй уровень экспертной поддержки управленческих решений. Этот уровень генерирует альтернативы на базе имеющихся в информационном фонде данных, правил преобразования и процедур оценки синтезированных альтернатив.

При разработке и создании АИС необходимо руководствоваться следующими принципами: новых задач, системности, обратной связи, первого руководителя, типизации проектных решений, одноразовости **ввода данных**, согласованности пропускных способностей частей системы.

Принцип новых задач заключается в том, что в ходе аналитического и информационного анализа системы управления предприятием выявляются новые задачи, которые до внедрения АИС не могли решаться из-за их сложности. Новые задачи могут быть внедрены взамен стереотипных. Например в условиях АИС становится возможной разработка в реальном масштабе времени при непосредственном и совместном участии линейного персонала на своих автоматизированных рабочих местах, соединенных компьютерными сетями, расписаний работ (от недельно-суточных до годовых) на программу строительного треста с оптимизацией по обширной номенклатуре трудовых и материально-технических ресурсов.

Последовательная автоматизация процессов формирования планов снизу вверх (от прораба до треста, как это имеет место при ручном управлении) не дала бы существенного эффекта. Малоэффективной была бы также, например, автоматизация решения задач прикрепления потребителей - строек к поставщикам - заводам в рутинной их постановке. Однако те же задачи в оптимизационном режиме решения дают высокий экономический эффект с одновременным сокращением затрат ручного труда управленческого персонала предприятия.

Принцип системности и комплексного подхода к проектированию АИС состоит в том, что все вопросы, связанные с проектированием, должны решаться на основе определенной цели и критериев функционирования системы, взаимной увязки организационно-технологических решений, программно-математического, информационного, правового и технического обеспечения АИС. С этим принципом тесно связан принцип субоптимизации, заключающийся в том, что несистемная оптимизация конкретной подсистемы нередко дает эффект, но не позволяет оптимизировать систему в целом. Успешное решение взаимоувязанных задач АИС возможно только при условии открытости и доступности электронных баз данных предприятий отрасли по всем управленческим вертикалям и горизонталям.

Функционирование системы Internet позволяет получить для решения задачи АИС любую требуемую информацию (исключая "защищенные" сведения) из баз данных указанных предприятий., если они подключены к глобальной компьютерной сети. Системный подход неразрывно связан с эффективным использованием человеко-машинного диалога, с системой программ, обеспечивающих четкое управление диалогом. При реализации человеко-машинного диалога необходимо предусматривать тесное взаимодействие управленческого персонала с компьютерным комплексом АИС, передачу системе возрастающих объемов рутинных работ с

тем, чтобы большую часть времени менеджеры использовали для решения творческих задач управления.

Принцип обратной связи заключается в том, что процесс разработки и внедрения АИС следует рассматривать как непрерывный с использованием предшествующего опыта. Так, например, после выполнения комплекса работ по цепочке: "аналитическое обследование предприятия; консалтинг по мероприятиям, необходимым для реорганизации; проектирование информационной системы; настройка уже существующего **программно-математического обеспечения** и разработка нового под конкретные условия строительной организации; развертывание (внедрения) АИС; сопровождение системы" должно быть организовано плавное перетекание последнего этапа в первый. Это означает, что работа по созданию АИС начнется вновь, только на другом - более высоком уровне.

Принцип первого руководителя означает, что руководитель организации (низовой **строительной фирмы**, крупной строительной корпорации, министр) единолично ответственен за своевременность и качество разработки АИС соответствующего уровня и ее эффективное функционирование. Первый руководитель отвечает за четкое взаимодействие заказчика и разработчика АИС, рациональное распределение обязанностей между ними. Когда создание АИС передоверяется второстепенным лицам, эта система используется, как правило, для решения рутинных задач и, в конечном счете, оказывается малоэффективной.

Принцип типизации проектных решений предусматривает максимальное использование при проектировании АС **типовых проектных** решений. Учитывая, что наибольший объем работ по созданию АС связан с подготовкой программно-математического обеспечения, особое внимание следует уделять типовым программным комплексам (автоматизированным рабочим местам - АРМам), чтобы на их основе разрабатывать как отдельные управленческие задачи, так и целые подсистемы. Например, задачи **бухгалтерского учета** и отдела кадров, подсистемы разработки расписаний и планирования планов поставок материально-технической продукции, и др. Единство АС, расположенных по горизонтали, достигается использованием общих подходов к их построению, а по вертикали - использованием общих форм документов и современных стандартов электронных баз данных, общих принципов формирования комплексов технических и программных средств, систем коммуникации и связи.

Существенный экономический эффект, высокое качество, сокращение сроков разработки, возможность активного участия управленческого персонала организации в создании АС достигается применением интегрированного программно-математического обеспечения. Проектирование АС на базе интегрированных программных систем, значительно упрощает процессы связывания и встраивания электронных документов, их передачи как внутри предприятия, так и другим информационным системам. Прикладные программы, созданные на основе интегрированных программных средств, отличаются максимально-возможной открытостью и достаточно просто могут улучшаться непосредственно инженерно-техническими работниками предприятия. Интегрированные программные системы максимально упрощают и эксплуатацию АС, так как все задачи решаются с применением единообразного пользовательского интерфейса.

Принцип одноразовости ввода данных в **орган управления** означает, что информация, введенная один раз в компьютерную систему, используется затем для решения нескольких задач данного или другого предприятия, оснащенного соответствующими **информационными технологиями** и коммуникациями. Соблюдение этого принципа позволяет избежать дублирования информации, исключить несурзности и ошибки, уменьшить потоки вводимой и обрабатываемой информации. Сокращение потоков информации также достигается в результате исключения из вводимых данных сведений нормативно-справочного характера, имеющих в машинных базах данных.

Принцип полной информационной совместимости между **автоматизированными системами** различных уровней управления предусматривает применение согласованных подходов к разработке машинных баз данных, входным и выходным документам, программным комплексам для АС различных предприятий. Это упростит использование разделённых (общих) баз данных строительными предприятиями отрасли, снизит затраты на разработку и поддержку информационного обеспечения (в т. ч. систем управления базами данных), придаст корпоративным информационным технологиям большую гибкость и адаптируемость.

Принцип независимости структуры **автоматизированных систем управления** от используемой техники и базовых технологий заключается в применении таких технологий создания информационной среды АС, которые были бы инвариантны по отношению к техническому обеспечению и легко трансформировались при создании новых программных инструментариев информатизации. На реализацию этого принципа, в частности, направлены объектные подходы к формированию информационной среды, основанные на CASE-технологиях.

Принцип согласованности пропускной способности частей системы заключается в том, что пропускная способность последующего устройства должна быть не ниже пропускной способности предыдущего. Например, компьютерные сети АС должны иметь пропускную способность, соответствующую быстродействию ЭВМ.

1. В соответствии с характером **обработки информации** в ИС на различных уровнях управления экономической системой (оперативном, тактическом и стратегическом) выделяются следующие типы информационных систем:

- *системы обработки данных (EDP - electronic data processing);*
- *информационная система управления (MIS - management information system);*
- *система поддержки принятия решений (DSS - decision support system);*

Системы обработки данных (СОД) предназначены для учета и оперативного регулирования хозяйственных операций, подготовки стандартных документов для внешней среды (счетов, накладных, **платежных поручений**). Горизонт **оперативного управления** хозяйственными процессами составляет от одного до несколько дней и реализует регистрацию и обработку событий, например оформление и мониторинг выполнения заказов, приход и расход материальных ценностей на складе, ведение табеля учета **рабочего времени** и т. д. Эти задачи имеют итеративный, регулярный характер, выполняются непосредственными исполнителями хозяйственных процессов (рабочими, кладовщиками, администраторами и т. д.) и связаны с оформлением и пересылкой документов в соответствии с четко определенными алгоритмами. Результаты выполнения хозяйственных операций через экранные формы вводятся в базу данных.

Информационные системы управления (ИСУ) ориентированы на тактический уровень управления: среднесрочное планирование, анализ и организацию работ в течение нескольких недель (месяцев), например анализ и планирование поставок, сбыта, составление производственных программ. Для данного класса задач характерны регламентированность (периодическая повторяемость) формирования результатных документов и четко определенный алгоритм решения задач, например свод заказов для формирования производственной программы и определение потребности в комплектующих деталях и материалах на основе спецификации изделий. Решение подобных задач предназначено для руководителей различных служб предприятий (отделов материально-технического снабжения и сбыта, цехов и т. д.). Задачи решаются на основе накопленной базы оперативных данных.

Системы поддержки принятия решений (СППР) используются в основном на верхнем уровне управления (руководства фирм, предприятий, организаций), имеющего стратегическое долгосрочное значение в течение года или нескольких лет. К таким задачам относятся формирование стратегических целей, планирование привлечения ресурсов, **источников**

финансирования, выбор места размещения предприятий и т. д. Реже задачи класса СППР решаются на тактическом уровне, например при **выборе поставщиков** или заключении контрактов с клиентами. Задачи СППР имеют, как правило, нерегулярный характер. Для задач СППР свойственны недостаточность имеющейся информации, ее противоречивость и нечеткость, преобладание качественных оценок целей и ограничений, слабая формализованность алгоритмов решения. В качестве инструментов обобщения чаще всего используются средства составления аналитических отчетов произвольной формы, методы статистического анализа, экспертных оценок и систем, математического и имитационного моделирования. При этом используются базы обобщенной информации, информационные хранилища, базы знаний о правилах и моделях принятия решений. Идеальной считается ИС, которая включает все три типа перечисленных информационных систем.

2. В зависимости от охвата функций и уровней управления различают корпоративные (интегрированные) и локальные ИС.

Корпоративная (интегрированная) ИС автоматизирует все функции управления на всех уровнях управления. Такая ИС является многопользовательской, функционирует в распределенной вычислительной сети.

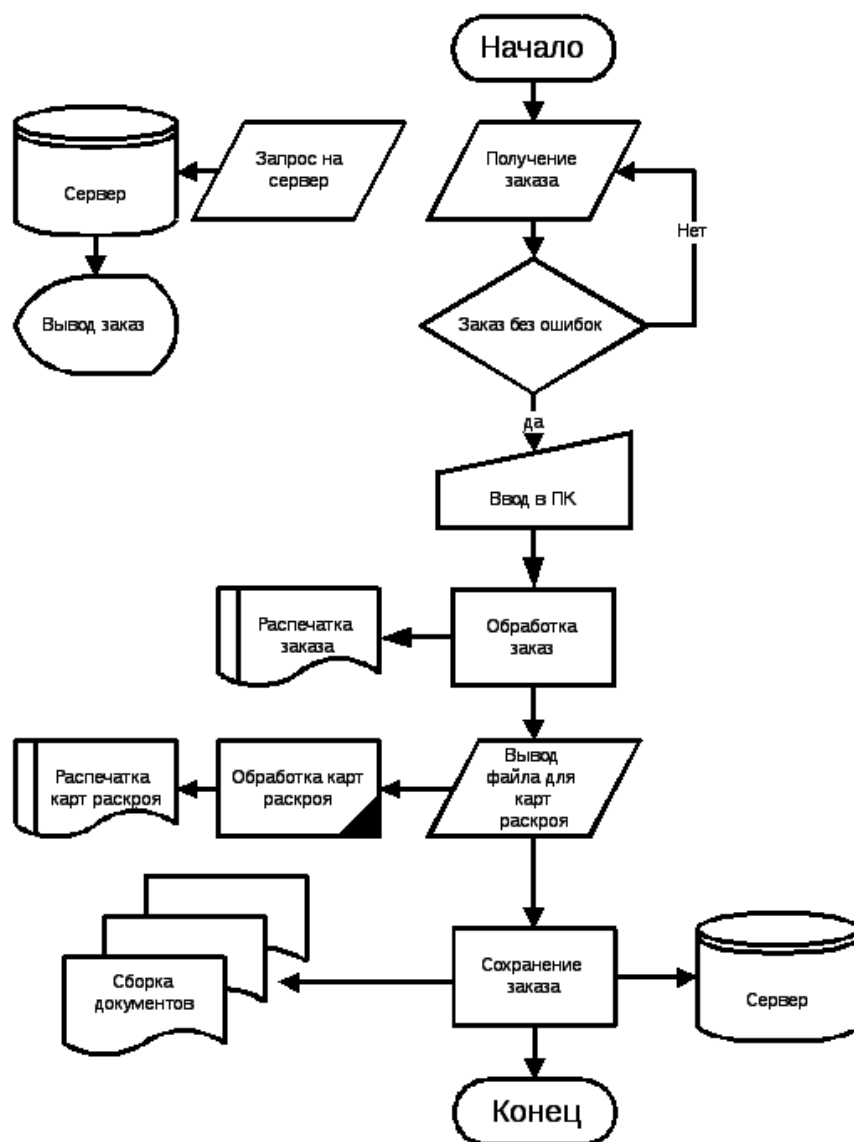
Локальная ИС автоматизирует отдельные функции управления на отдельных уровнях управления. Такая ИС может быть однопользовательской, функционирующей в отдельных подразделениях системы управления.

Способы описания информационной технологии.

Графическая модель.

Опишем графически информационный процесс обработки заказов.

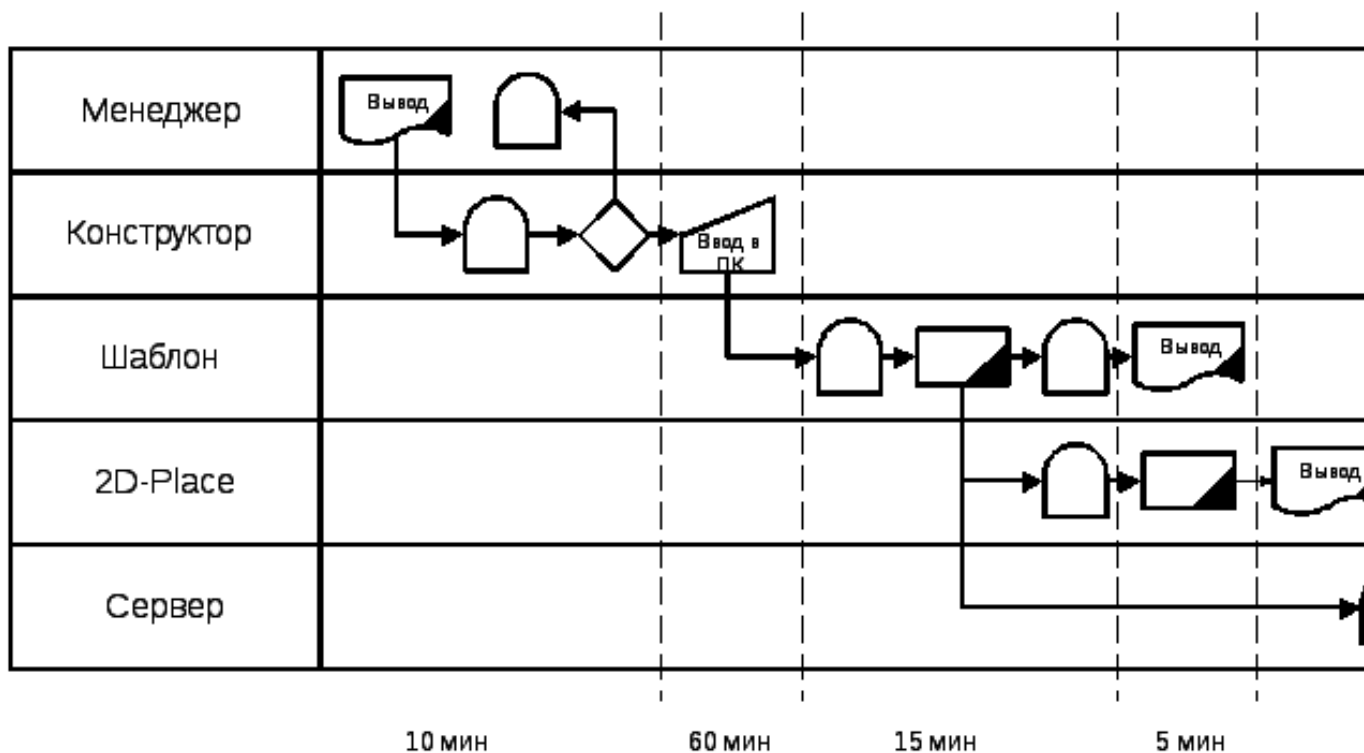
Конструктор получает заказ и проверяет его на ошибки, при обнаружении которых заказ возвращается менеджеру. Заказ вводится в шаблон «Копия». После полного ввода данных заказ обрабатывается с помощью специальных макросов. Выводится файл для создания карт раскроя. Файл импортируется, обрабатывается. Готовые карты раскроя распечатываются. Обработанный заказ также распечатывается для участков производства. Заказ сохраняется на сервере для дальнейшего хранения. При поступлении рекламации осуществляется доступ к сохраненному файлу.



Обобщенные структурные информационно-временные схемы (осивс).

Модели типа обобщенных структурных информационно-временных схем (ОСИВС) предназначены для содержательного описания технологии обработки данных с указанием места и периода времени выполнения всякой операции. Процесс обработки данных изображается с помощью совокупности символов на прямоугольной сетке, где столбцы соответствуют временным интервалам, а строки – подразделениям, где выполняется операция. Условные символы, отображающие операции и данные, помещаются в клетках, образованных пересечением строк и столбцов.

Обобщенная структурная информационно-временная схема представлена на рисунке.



Обобщённая структурная информационно-временная схема.

Информационный процесс на жд транспорте

- 1. Структура инф. процесса.
- 2. Способы описания ИТ. Классификация моделей.
- 3. Характеристики и показатели качества информ. процессов.
- 4. Информационные потоки.
- 1 Структура инф. процесса.

Информационные технологии определяют методы, способы, приемы реализации инф. процесса.

Элементарные процессы:

- - сбор
- - преобразование и ввод в ЭВМ

Реализация каждой операции над информацией может осуществляться с использованием различных методов, поэтому можно говорить о различных технологиях сбора, преобразования информации в ЭВМ.

перечисленные функции называется элементарными. Более сложные функции образуются с помощью элементарных. Обмен сообщений между пользователями осуществляется в результате реализации упорядоченной совокупности операции ввода, передачи, хранения, обработки информации.

Способы описания ИТ. Классификация моделей

вычислительная машина технология информационный процесс

Чтобы дать представление ИТ, нужно указать ее назначение, перечислить вып. операции и определить их взаимосвязь и последовательность, указать метод реализации каждой операции.

Основные функции эл. почты:

- 1. Ввод сообщений в систему и получение сообщений
- 2. Передача сообщений в почтовые ящики пользователя и обмен между ящиками, проверка исправления ошибок, возникающих при передаче, пересылке подтверждения о доставке сообщения
- 3. Рассылка сообщений по спискам адресов, изменение адресов

Приведенная технология работы эл. почты является описательной неформализованной моделью. Такие модели дают общие представления о технологии, однако они не особенно наглядны. Более сложное взаимодействия между элементарными операциями представить нельзя.

Более понятные формализованные описательные модели, то есть такие модели, где сведения представлены с помощью специальных документов: бланков, форм, анкет, таблиц.

Еще большей наглядностью обладают графические модели, особенно в ситуациях, когда требуется описать взаимосвязь между отдельными операциями, например i-последовательность. К графическим моделям относят схемы, графы, чертежи.

Схемы информ. процессов и обобщенные структурные информационно-временные схемы применяются для описания информ. технологий с использованием спец. системы граф. символов.

Графические модели типа логических схем, граф-состояний используется при анализе характеристик информационных процессов. Эти модели позволяют перейти к математическим моделям, представляющим инф процесс на языке мат. отношений.

Мат. модели могут иметь вид функциональной зависимости систем алгебраических или диф. уравнений, логических выражений. Мат модели делят на аналитические и имитационные.

Аналитические модели ориентированы на исследование аналитических или числены методов (решение систем уравнений, поиск сочетаний факторов, где описываемый моделью показатель принимает экстремальное значение)

Имитационные модели представляются в виде совокупности алгоритмов воспроизводящих процесс функционирования системы, при этом анализ имитационных моделей осуществляется путем экспериментирования на модели. Многократные имитации изучаемого инф. процесса при воздействии на него различного рода изучаемых явлений, воспроизводимых при моделировании, то есть при исследовании инф. процесса с использованием его модели.

Информационный процесс на жд транспорте

1. Структура инф. процесса.

2. Способы описания ИТ. Классификация моделей.

3. Характеристики и показатели качества информ. процессов.

4. Информационные потоки.

1 Структура инф. процесса.

Информационные технологии определяют методы, способы, приемы реализации инф. процесса.

Элементарные процессы:

- сбор

- преобразование и ввод в ЭВМ

Реализация каждой операции над информацией может осуществляться с использованием различных методов, поэтому можно говорить о различных технологиях сбора, преобразования информации в ЭВМ.

перечисленные функции называются элементарными. Более сложные функции образуются с помощью элементарных. Обмен сообщений между пользователями осуществляется в результате реализации упорядоченной совокупности операции ввода, передачи, хранения, обработки информации.

Способы описания ИТ. Классификация моделей

вычислительная машина технология информационный процесс

Чтобы дать представление ИТ, нужно указать ее назначение, перечислить вып. операции и определить их взаимосвязь и последовательность, указать метод реализации каждой операции.

Основные функции эл. почты:

1. Ввод сообщений в систему и получение сообщений

2. Передача сообщений в почтовые ящики пользователя и обмен между ящиками, проверка исправления ошибок, возникающих при передаче, пересылке подтверждения о доставке сообщения

3. Рассылка сообщений по спискам адресов, изменение адресов

Приведенная технология работы эл. почты является описательной неформализованной моделью. Такие модели дают общие представления о технологии, однако они не особенно наглядны. Более сложное взаимодействия между элементарными операциями представить нельзя.

Более понятные формализованные описательные модели, то есть такие модели, где сведения представлены с помощью специальных документов: бланков, форм, анкет, таблиц.

Еще большей наглядностью обладают графические модели, особенно в ситуациях, когда требуется описать взаимосвязь между отдельными операциями, например i-последовательность. К графическим моделям относят схемы, графы, чертежи.

Схемы информ. процессов и обобщенные структурные информационно-временные схемы применяются для описания информ. технологий с использованием спец. системы граф. символов.

Графические модели типа логических схем, граф-состояний используется при анализе характеристик информационных процессов. Эти модели позволяют перейти к математическим моделям, представляющим инф процесс на языке мат. отношений.

Мат. модели могут иметь вид функциональной зависимости систем алгебраических или диф. уравнений, логических выражений. Мат модели делят на аналитические и имитационные.

Аналитические модели ориентированы на исследование аналитических или численны методов (решение систем уравнений, поиск сочетаний факторов, где описываемый моделью показатель принимает экстремальное значение)

Имитационные модели представляются в виде совокупности алгоритмов воспроизводящих процесс функционирования системы, при этом анализ имитационных моделей осуществляется путем экспериментирования на модели. Многократные имитации изучаемого инф. процесса при воздействии на него различного рода изучаемых явлений, воспроизводимых при моделировании, то есть при исследовании инф. процесса с использованием его модели.

Автоматизированные системы управления технологическими процессами

Материал темы лекции содержит содержание следующих вопросов: структура АСУТП; назначение, цели и функции АСУТП; примеры информационных и управляющих АСУТП; основные разновидности АСУ ТП; состав АСУ ТП.

Структура АСУТП.

При построении средств современной промышленной **автоматики** (обычно в виде АСУ ТП) используется иерархическая информационная структура с применением на разных уровнях вычислительных средств различной мощности. Примерная общая современная структура АСУ ТП показана на рисунке 14.1:



ИП - измерительные преобразователи (датчики),

ИМ - исполнительные механизмы,

ПЛК - программируемый логический контроллер,

ПрК - программируемый (настраиваемый) контроллер,

ИнП- интеллектуальные измерительные преобразователи,

ИнИМ - интеллектуальные исполнительные устройства,

Модем - модулятор/демодулятор сигналов,

ТО - техническое обеспечение (аппаратная часть, «железо»),

ИО - информационное обеспечение (базы данных),

ПО - программное обеспечение,

КО - коммуникационное обеспечение (последовательный порт и ПО).

ПОпл - программное обеспечение пользователя,

ПОпр - программное обеспечение производителя,

Инд - индикатор.

Рисунок 14.1 - Типовая функциональная схема современной АСУ ТП.

В настоящее АСУ ТП обычно реализуются по схемам:

1. 1-уровневой (локальная система), содержащей ПЛК, или моноблочный настраиваемый контроллер (МНК) обеспечивающие индикацию и сигнализацию состояния контролируемого или регулируемого ТП на передней панели,

2. 2-уровневой (централизованная система), включающих:

1. На нижнем уровне несколько ПЛК с подключенными к ним датчиками и исполнительными устройствами,

2. На верхнем уровне – одна (возможно несколько) операторских (рабочих) станций (автоматизированных рабочих мест (АРМ) оператора).

Обычно рабочая станция или АРМ - это [ЭВМ](#) в специальном промышленном исполнении, со специальным программным обеспечением, – системой сбора и визуализации данных (SCADA-системы).

Типовая функциональная схема одноуровневой АСУ ТП показана на рисунке 14.2



Рисунок 14.2 - Типовая функциональная схема одноуровневой системы автоматического управления САУ.

Основные функции элементов:

ПЛК:

1. Прием дискретных сигналов от преобразователей технологического оборудования,
2. Аналого-цифровое преобразование (АЦП) аналоговых сигналов, поступающих на входы из преобразователей,
3. Масштабирование и цифровая фильтрация данных после АЦП,
4. Обработка принятых данных по программе функционирования,
5. Генерация (в соответствии с программой) управляющих дискретных сигналов и подача их на исполнительные устройства,
6. Цифро-аналоговое преобразование (ЦАП) выходных информационных данных в выходные аналоговые сигналы,
7. Подача управляющих сигналов на соответствующие исполнительные устройства,
8. Защита от потери работоспособности из-за зависания процессора с помощью сторожевого таймера,
9. Сохранение работоспособности при временном отключении электропитания (за счет источника бесперебойного питания с аккумулятором достаточной емкости),
10. Контроль за работоспособностью датчиков и достоверностью измеренных величин,
11. Индикация текущих и интегральных значений измеряемых величин,

12. Контрольная сигнализация состояния управляемого процесса,
13. Контрольная световая и символьная сигнализация состояния контроллера,
14. Возможность конфигурации (настройки параметров) через ПК подключаемый, к специальному порту.

Преобразователи (Пр):

1. Преобразование значения измеряемой величины (температуры, давления, перемещения и т.д.) в непрерывный или импульсный (для счетных входов ПЛК) электрический сигнал.

Исполнительные устройства (ИУ):

1. Преобразование управляющих электрических непрерывных или импульсных сигналов в механическое перемещение исполнительных механизмов, электронное управление током в силовых цепях и т.д.

Устройство согласования (при необходимости):

1. Гальваническая или другие виды развязки между ПЛК и исполнительными устройствами (ИУ),
2. Согласование допустимых значений выходного тока управляющих каналов ПЛК и тока, необходимого для нормальной работы ИУ.

При недостаточном числе каналов одного ПЛК используется схема распределенного ввода/вывода с использованием других (управляемых, ведомых ПЛК) или дополнительных контроллеров (модулей) ввода/вывода.

Типовая функциональная схема одноуровневой АСУ ТП с распределенным вводом/выводом показана на рисунке 14.3:



Рисунок 14.3 - Типовая функциональная схема одноуровневой АСУ ТП с распределенным вводом/выводом

Типовая функциональная схема 2-уровневой АСУ ТП показана на рисунке 14.4.

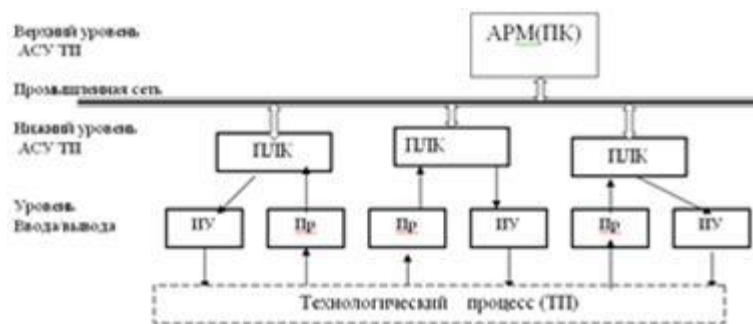


Рисунок 14.4 - Типовая функциональная схема 2-уровневой АСУТП

Все ПЛК и АРМы объединены промышленной информационной сетью, обеспечивающей непрерывный обмен данными. Преимущества: позволяет распределить задачи, между узлами системы, повысив надежность ее функционирования.

Основные функции нижнего уровня:

1. Сбор, электрическая фильтрация и АЦП сигналов с преобразователей (датчиков);
2. Реализация локальных АСУ технологического процесса в объеме функций ПЛК одноуровневой системы;
3. Реализация аварийной и предупредительной сигнализации;
4. Организация системы защит и блокировок;
5. Обмен текущими данными с ПК верхнего уровня через промышленную сеть по запросам ПК.

Основные функции верхнего уровня:

1. Визуализация состояния технологического процесса;
2. Текущая регистрация характеристик технологического процесса;
3. Оперативный анализ состояния оборудования и **технологического процесса**;
4. Регистрация действий оператора, в том числе при аварийных сообщениях;
5. Архивация и длительное хранение значений протоколов технологического процесса;
6. Реализация алгоритмов «системы советчика»;
7. Супервизорное управление;

8. Хранение и ведение баз данных:

- параметров техпроцессов,
- критических параметров оборудования,
- признаков аварийных состояний **технологического процесса**,
- состава допускаемых к работе с системой операторов (их паролей).

Таким образом, нижний уровень реализует алгоритмы **управления** оборудованием, верхний - решение стратегических вопросов функционирования. Например, решение включить или выключить насос принимается на верхнем уровне, а подача всех необходимых управляющих сигналов, проверка состояния насоса, реализация механизма блокировок выполняется на нижнем уровне.

Иерархическая структура АСУ технологического процесса подразумевает:

1. Поток команд направлен от верхнего уровня к нижнему,
2. Нижний отвечает верхнему по его запросам.

Это обеспечивает предсказуемое поведение ПЛК при выходе из строя верхнего уровня или промышленной сети, поскольку такие неисправности воспринимаются нижним уровнем как отсутствие новых команд и запросов.

При конфигурации ПЛК устанавливается: до какого времени после получения последнего запроса ПЛК продолжает функционировать, поддерживая последний заданный режим, после чего переходит в нужный при данной аварийной ситуации режим работы.

Например, структура организации АСУТП некоторого производства бетона на бетоносмесительных установках по логике построения можно разделить на два основных уровня :

- нижний уровень – уровень реализации задачи на базе промышленных контроллеров (PLC);
- верхний уровень – уровень реализации задачи визуализации процессов, происходящих при производстве бетона на БСУ (SCADA).

На нижнем уровне система решает следующие основные задачи:

- сбор первичной информации с исполнительных узлов БСУ;
- анализ собранной информации;

- отработка логики технологического процесса при производстве бетона с учетом всех современных требований;
- выдача управляющих воздействий на исполнительные устройства.

На верхнем уровне система решает другие задачи:

- визуализация основных технологических параметров с БСУ (состояние исполнительных органов, ток потребления миксера, вес дозируемых материалов и т.д.);
- архивирование всех параметров процесса производства бетона;
- выдача команд на воздействие исполнительными органами БСУ;
- выдача команд на изменение параметров внешних воздействий;
- разработка и хранение рецептур бетонных смесей.

Назначение АСУТП. АСУТП предназначена для выработки к реализации управляющих воздействий на технологический объект управления.

Технологический объект управления (АСУТП) представляет собой совокупность технологического оборудования и реализованного на нем по соответствующим инструкциям или регламентам технологического процесса производства продуктов, полупродуктов, изделий или энергии,

К технологическим объектам управления относятся:

- технологические агрегаты и установки (группы станков), реализующие самостоятельный технологический процесс;
- отдельные производства (цехи, участки), если управление этим производством носит, в основном, технологический характер, то есть заключается в реализации рациональных режимов работы взаимосвязанного технологического оборудования (агрегатов, участков).

Совместно функционирующие ТОУ и управляющая ими АСУТП образуют автоматизированный технологический комплекс (АТК). В машиностроении и других дискретных производствах в качестве АТК выступают гибкие производственные системы (ГПС).

Термины АСУТП, ТОУ и АТК должно употребляться только в приведенных сочетаниях. Совокупность других систем управления с управлением ими технологическим оборудованием не является АТК. Система управления в других случаях (не в АТК) не является АСУТП и т.д. АСУТП - это организационно-техническая система управления объектом в целом в соответствии с принятым

критерием (критериями) управления, в которой сбор и обработка необходимой информации осуществляется с применением средств вычислительной техники.

Приведенная формулировка подчеркивает:

- во-первых, использование в АСУТП современных средств вычислительной техники;
- во-вторых, роль человека в системе как субъекта труда, принимающего содержательное участие в выработке решений по управлению;
- в-третьих, что АСУТП - это система, осуществляющая обработку технологической и технико-экономической информации;
- в-четвертых, что цель функционирования АСУТП заключается в оптимизации работы технологического объекта управления в соответствии с принятым критерием (критериями) управления путем соответствующего выбора управляющих воздействий.

Критерий управления в АСУТП - это соотношение, характеризующее степень достижения целей управления (качество функционирования технологического объекта управления в целом) и принимающее различные числовые значения в зависимости от используемых управляющих воздействий. Отсюда следует, что критерий обычно является технико-экономическим (например, себестоимость выходного продукта при заданном его качестве, производительность ТОУ при заданном качестве выходного продукта и т.п.) или техническим показателем (параметра процесса, характеристики выходного продукта).

В случае, если ТОУ управляется АСУТП, весь участвующий в управлении оперативный персонал ТОУ и все средства управления, предусмотренные документацией на АСУТП и взаимодействующие при управлении ТОУ, входят в состав системы, независимо от того, каким путем (нового строительства или модернизации системы управления) создавался АТК.

АСУТП создается путем капитального строительства, т.к. независимо от объёма поставки для её ввода в действие, необходимо проведение строительно-монтажных и наладочных работ на объекте.

АСУТП как компонент общей системы управления промышленным предприятием предназначена для целенаправленного ведения технологических процессов и обеспечения смежных и вышестоящих систем управления оперативной и достоверной технико-экономической информацией. АСУТП созданные для

объектов основного и (или) вспомогательного производства, представляют собой низовой уровень автоматизированных систем управления на предприятии.

АСУТП могут использоваться для управления отдельными производствами, включающими в свой состав взаимосвязанные ТОУ, в том числе управляемые собственными АСУТП низового уровня.

Для объектов с дискретным характером производства в состав гибких производственных систем могут включаться автоматизированные системы технологической подготовки производства (или их соответствующие подсистемы) и системы автоматизированного проектирования технология (САПР-технология).

Организация взаимодействия АСУТП с вышестоящими уровнями управления определяется наличием на промышленном предприятии автоматизированной системы управления предприятием (АСУП) и автоматизированных систем оперативно-диспетчерского управления (АСОДУ).

При их наличии АСУТП в совокупности с ними образуют интегрированную автоматизированную систему управления (ИАСУ). В этом случае АСУТП получает от соответствующих подсистем АСУП или служб управления предприятием непосредственно или через ОСОДУ задания и ограничения (номенклатуру подлежащих выпуску продуктов или изделий, объем производства, технико-экономические показатели, характеризующие качество функционирования АТК, сведения о наличии ресурсов) и обеспечивает подготовку и передачу этим системам необходимой для их работы технико-экономической информации, в частности о результатах работы АТК, основных показателях выпускаемой продукции, оперативной потребности в ресурсах, состоянии АТК (состоянии оборудования, ходе технологического процесса, его технико-экономических показателях и т.п.),

При наличии на предприятии автоматизированных систем технической и технологической подготовки производства должно быть обеспечено необходимое взаимодействие АСУТП с этими системами. При этом АСУТП получают от них техническую, технологическую и другую информацию, необходимую для обеспечения заданного проведения технологических процессов, и направляют в названные системы фактическую оперативную информацию, необходимую для их функционирования.

При создании на предприятия комплексной системы управления качеством продукции автоматизированные системы управления технологическими процессами выступают в роли ее исполнительных подсистем, обеспечивающих заданное качество продукции ТОУ и подготовку оперативной фактической информации о ходе технологических процессов (статистический контроль и т.д.)

Операционные системы

Частью программного обеспечения, наиболее тесно взаимодействующей с аппаратной частью компьютера, является системное программное обеспечение и, прежде всего операционная система.

Операционная система играет роль посредника между пользователем, программами и оборудованием компьютера. Она обеспечивает возможность запуска программ, поддерживает работоспособность устройств, предоставляет средства проверки и настройки различных компонентов. Чем гибче и многофункциональнее операционная система, тем больше возможностей она предоставляет, тем удобнее работать с компьютером.

Операционная система (ОС) – это комплекс (набор) программ, который обеспечивает взаимодействие всех устройств ЭВМ и позволяет пользователю осуществлять общее управление ЭВМ.

Главное назначение ОС – управление ресурсами, а главные ресурсы, которыми она управляет, – это аппаратура компьютера. ОС управляет вычислительным процессом и информационным обменом между процессором, памятью, внешними устройствами.

Поскольку все устройства компьютера работают одновременно, ОС обеспечивает разделение ресурсов, предотвращая тем самым опасность возникновения конфликтных ситуаций между компонентами вычислительной системы, способных привести к сбою в работе, потере или искажении информации.

ОС реализует много различных функций, в том числе:

- создает рабочую среду и поддерживает пользовательский [интерфейс](#);
- обеспечивает выполнение команд пользователя и программных инструкций;
- управляет аппаратными средствами компьютера;
- обеспечивает разделение аппаратных ресурсов между программами;
- планирует доступ пользователей к общим ресурсам;
- обеспечивает выполнение операций ввода–вывода, хранения информации и управление файловой системой;
- осуществляет восстановление информации в случае аппаратных сбоев и программных ошибок.

Развитие операционных систем всегда следовало за развитием аппаратуры.

Операционная система - это комплекс взаимосвязанных системных программ, назначение которого - организовать взаимодействие пользователя с компьютером и выполнение всех других программ.

Операционная система выполняет роль связующего звена между аппаратурой компьютера, с одной стороны, и выполняемыми программами, а также пользователем, с другой стороны.

Операционную систему составляют:

- управляющие программы;
- набор утилит, необходимых для эксплуатации операционной системы.

Операционная система обычно хранится во внешней памяти компьютера - на диске. При включении компьютера она считывается с дисковой памяти и размещается в [ОЗУ](#). Этот процесс называется загрузкой операционной системы.

В функции операционной системы входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, принтера и др.).

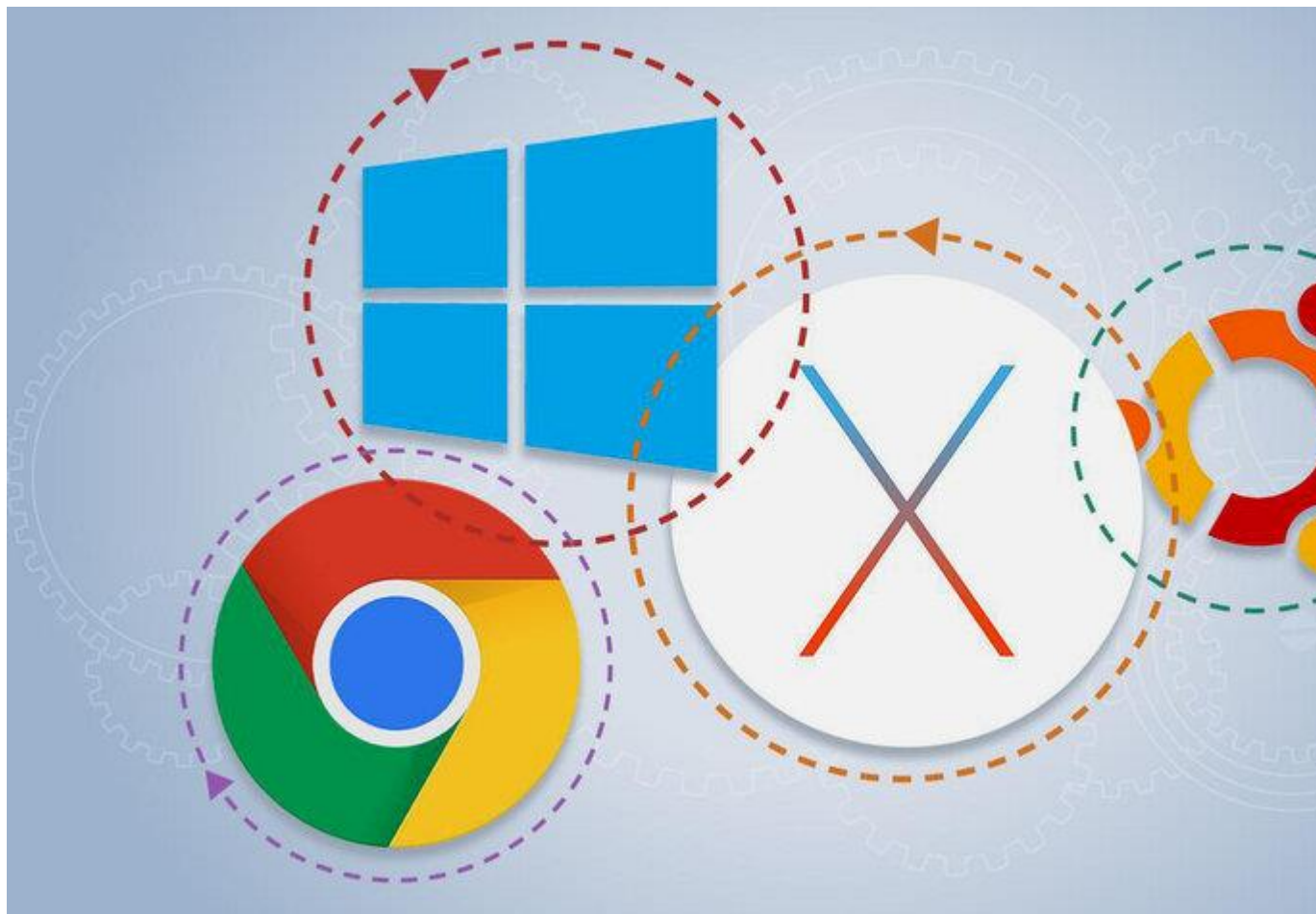
Операционную систему можно назвать программным продолжением устройства управления компьютера. Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера.

Требования к современным операционным системам:

- совместимость – ОС должна включать средства для выполнения приложений, подготовленных для других ОС;

ТОП-10 ОПЕРАЦИОННЫХ СИСТЕМ ДЛЯ ПК: ЧЕМ ОТЛИЧАЮТСЯ И КАКИЕ У НИХ ПРЕИМУЩЕСТВА

Список самых популярных операционных систем для компьютеров на рынке со сравнением особенностей и возможностей



Стремительное развитие технологий — наибольшее чудо нашего времени. Человеческая жизнь становится все проще, быстрее и интереснее, и за это нужно поблагодарить компьютеры в любом из их проявлений. Они прошли внушительный эволюционный путь, который превратил громоздкие настольные коробки в более компактные и удобные ноутбуки и смартфоны. В сети то и дело обсуждают их характеристики, но часто забывают про то, что заставляет их работать, — операционные

системы. В этой статье собраны самые популярные из их настольных вариантов с уточнением особенностей и возможностей.

1. Windows



Источник: MicrosoftNews

- **Предназначение:** приложения, интернет-серфинг, персональное использование, игры.
- **Стоимость:** от 199 долларов за Home до 289 долларов за Pro.
- **Веб-сайт:** [Microsoft](https://www.microsoft.com/windows).

Начиная с Windows 95 и заканчивая Windows 10 — именно операционная система Microsoft считается самой популярной и наиболее знакомой простым пользователям из разных уголков мира. Программное обеспечение

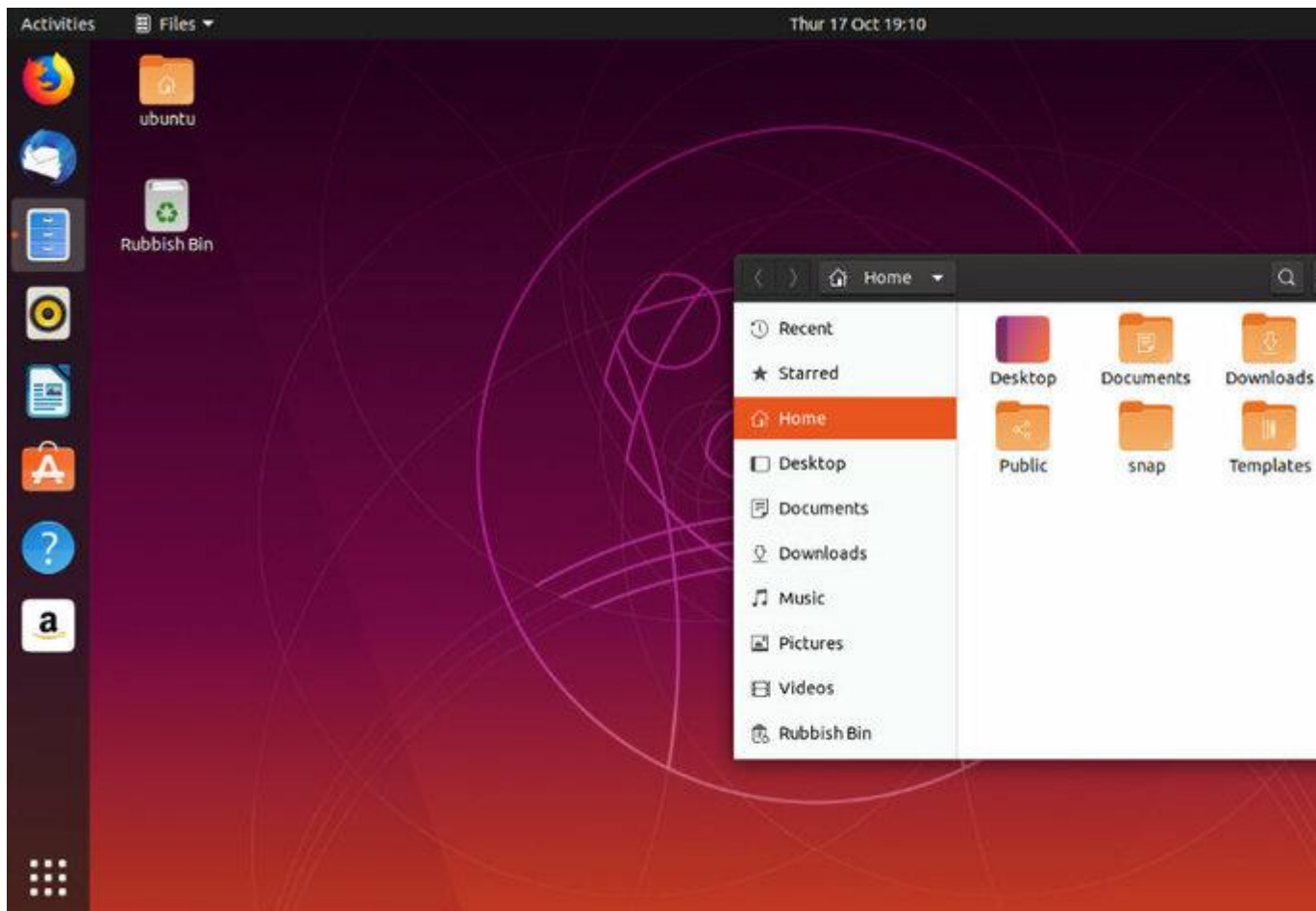
компания остается основой современных компьютеров. Его выбирают из-за простоты использования, быстрого запуска и возобновления работы. Последние версии операционной системы отличаются повышенной безопасностью и вниманием к конфиденциальности пользователей. С ней вы и ваши данные находятся под надежной защитой, и особенно переживать за свое личное пространство не приходится.

Особенности:

- удобный пользовательский интерфейс с простой навигацией, меню «Пуск», в котором собраны ключевые настройки системы, а также установленные приложения;
- простое одновременное использование нескольких рабочих пространств с открытыми окнами приложений;
- специальный режим для использования на устройствах с сенсорным экраном;
- технологии многофакторной аутентификации для повышения безопасности, распознавания отпечатков пальцев и лиц;
- автоматическое уменьшение размера файлов для уменьшения места, используемого в хранилище.

Вывод: Windows считается одной из лучших операционных систем, потому что она развивается продолжительное время. У нее современная система безопасности, а также простой пользовательский интерфейс, который подходит для самых разнообразных устройств.

2. Ubuntu



Источник: КО

- **Предназначение:** приложения, интернет-серфинг, игры.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [Ubuntu](https://ubuntu.com).

Ubuntu — операционная система на базе Linux, которая поставляется со всеми необходимыми инструментами. Она отлично подходит для разнообразных общественных организаций, школ и домашнего использования. Операционная система поддерживается Canonical — глобальным разработчиком программного обеспечения, который сыскал славу на распространении Ubuntu. Она абсолютно бесплатная, и только это должно стать мотивирующим фактором для того, чтобы самостоятельно

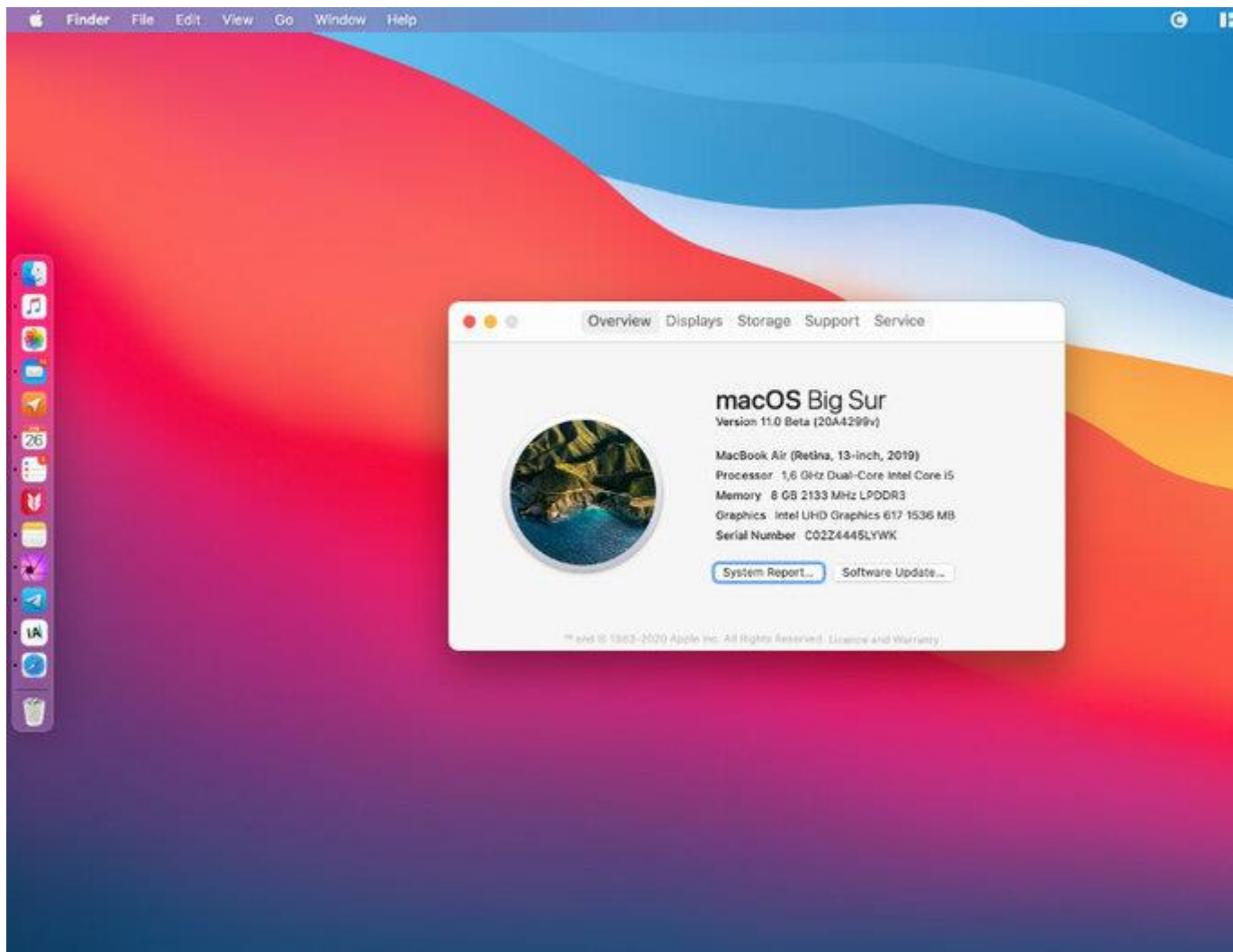
попробовать ее. Все-таки далеко не все готовы тратить на программное обеспечение в любом возможном виде.

Особенности:

- это операционная система с открытым исходным кодом, которую можно абсолютно бесплатно скачать, использовать без ограничений и свободно устанавливать любым пользователям;
- в Ubuntu встроен брандмауэр, а также программное обеспечение для защиты от вирусов — это делает ее достаточно безопасной;
- проект развивается уже несколько лет, за которые выходили регулярные обновления и необходимые исправления;
- Ubuntu полностью переведена на 50 различных языков;
- операционная система совместима со всеми новейшими ноутбуками, настольными компьютерами и устройствами с сенсорными экранами.

Вывод: Ubuntu — отличный вариант для тех, кто не хочет тратить. Это свободная операционная система с открытым кодом, что делает ее достаточно привлекательной для самых разных пользователей. Она регулярно обновляется, отличается удобным интерфейсом и высоким уровнем безопасности.

3. macOS



- **Предназначение:** эксклюзивные приложения Apple.
- **Стоимость:** бесплатно с устройствами Apple.
- **Веб-сайт:** [Apple](https://www.apple.com/macos).

Эксклюзивная операционная система Apple, у которой свой путь развития. Ее нельзя полноценно установить ни на одно другое устройство, кроме Mac. Она заточена под конкретное железо, которое использует производитель, и работает на нем максимально гладко. Для macOS подходит только программное обеспечение, которое создано конкретно для этой платформы. За операционную систему не нужно платить — ее стоимость уже вложена в цену далеко не самого дешевого аппаратного обеспечения. На [WWDC 2020](https://www.apple.com/wwdc2020/) Apple объявила о переводе macOS на

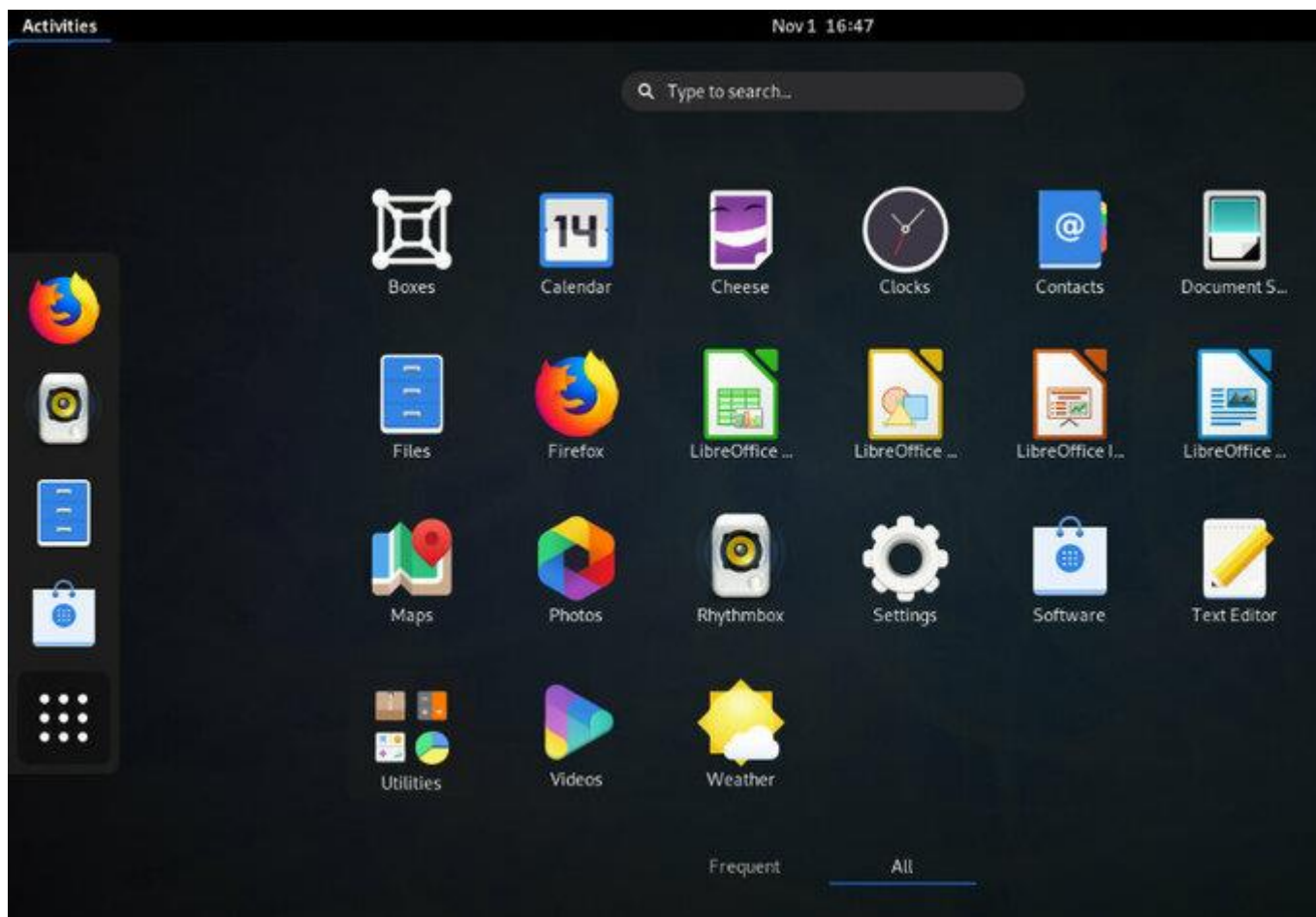
процессоры собственного производства на базе архитектуры ARM.

Особенности:

- ночная тема оформления интерфейса операционной системы, которая делает работу с ней в вечернее время суток более комфортной;
- возможности для автоматического упорядочивания файлов по типу, дате изменения и другим параметрам;
- тесная интеграция с другими устройствами Apple — к примеру, можно быстро вставить в документ фрагмент, который вы только что сфотографировали на iPhone;
- фирменный магазин приложений с программным обеспечением;
- максимальный уровень безопасности и конфиденциальности данных.

Вывод: Это — одна из наиболее привлекательных операционных систем в визуальном плане. Так как компания развивает программное и аппаратное обеспечение одновременно, она может добиться максимального уровня оптимизации. Эту систему могут полноценно использовать только владельцы Mac.

4. Fedora



Источник: Wiki

- **Предназначение:** корпоративное использование.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [Fedora](https://www.fedoraproject.org/).

Еще одна операционная система, в основу которой лег Linux. Она достаточно надежная и удобная, отлично работает как на настольных компьютерах, так и на ноутбуках. Fedora часто используют в общественных местах со свободным доступом случайных пользователей, а также для разработки.

Особенности:

- удобный пользовательский интерфейс, который не отвлекает от работы;
- полный набор инструментов с открытым исходным кодом;

- использование продвинутых инструментов для виртуализации и запуска виртуальных машин;
- использование контейнерных приложений.

Вывод: операционная система, которую чаще всего выбирают не для личного, а для корпоративного использования. Ее часто используют разработчики программного обеспечения, которые получают бесплатный доступ к полному набору инструментов для создания приложений.

5. Solaris



Источник: LegacyOS

- **Предназначение:** обработка информации, работа с базами данных.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [Solaris](#).

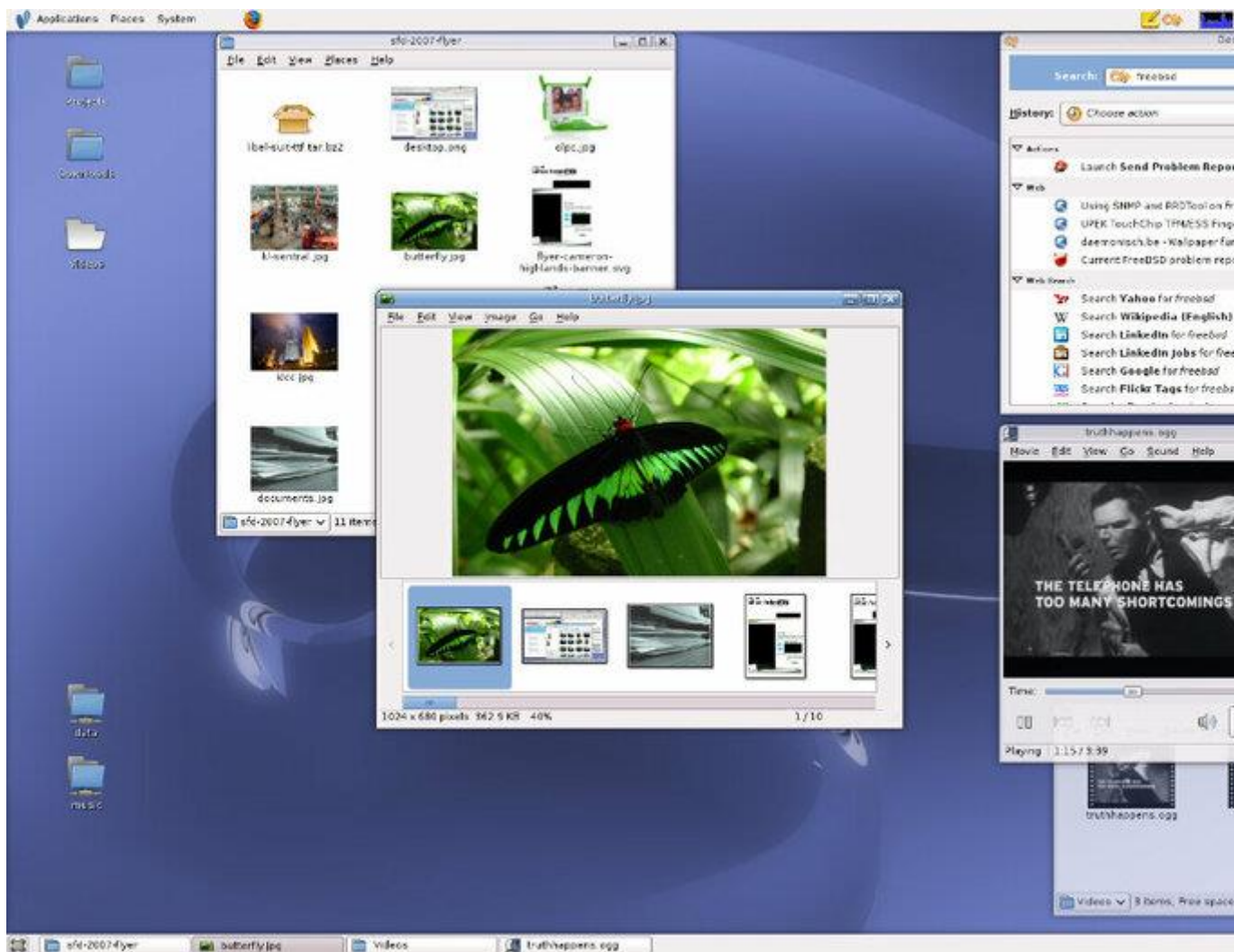
Операционная система на базе UNIX, которую в середине 90-х разрабатывала Sun Microsystems. В 2010 году, когда компанию выкупила Oracle, платформа получила название Oracle Solaris. Она хорошо известна своей масштабируемостью, а также целым набором других возможностей. К примеру, она поддерживает DTrace — фреймворк динамической трассировки, предназначенный для отладки ядра и приложений в реальном времени. Solaris работает с файловой системой ZFS, которая заточена под объемные массивы данных, а также инструментом резервного копирования TimeSlider.

Особенности:

- передовые возможности безопасности, гибкое управление правами пользователей, которое даст возможность защитить критически важные данные;
- неоспоримые преимущества в производительности для веб-служб, баз данных и сервисов Java;
- высокопроизводительная работа с сетью;
- неограниченные возможности для управления файловой системой и базами данных;
- полная совместимость для решения аппаратных и программных вопросов.

Вывод: Oracle Solaris считается одной из лучших среди бесплатных операционных систем с открытым исходным кодом. Она обеспечивает необходимую масштабируемость, функциональную совместимость, управление данными и безопасность, что высоко ценится в корпоративном секторе.

6. Free BSD



Источник: Wiki

- **Предназначение:** работа с интернетом, интранетом, серверами.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [Free BSD](http://FreeBSD.org).

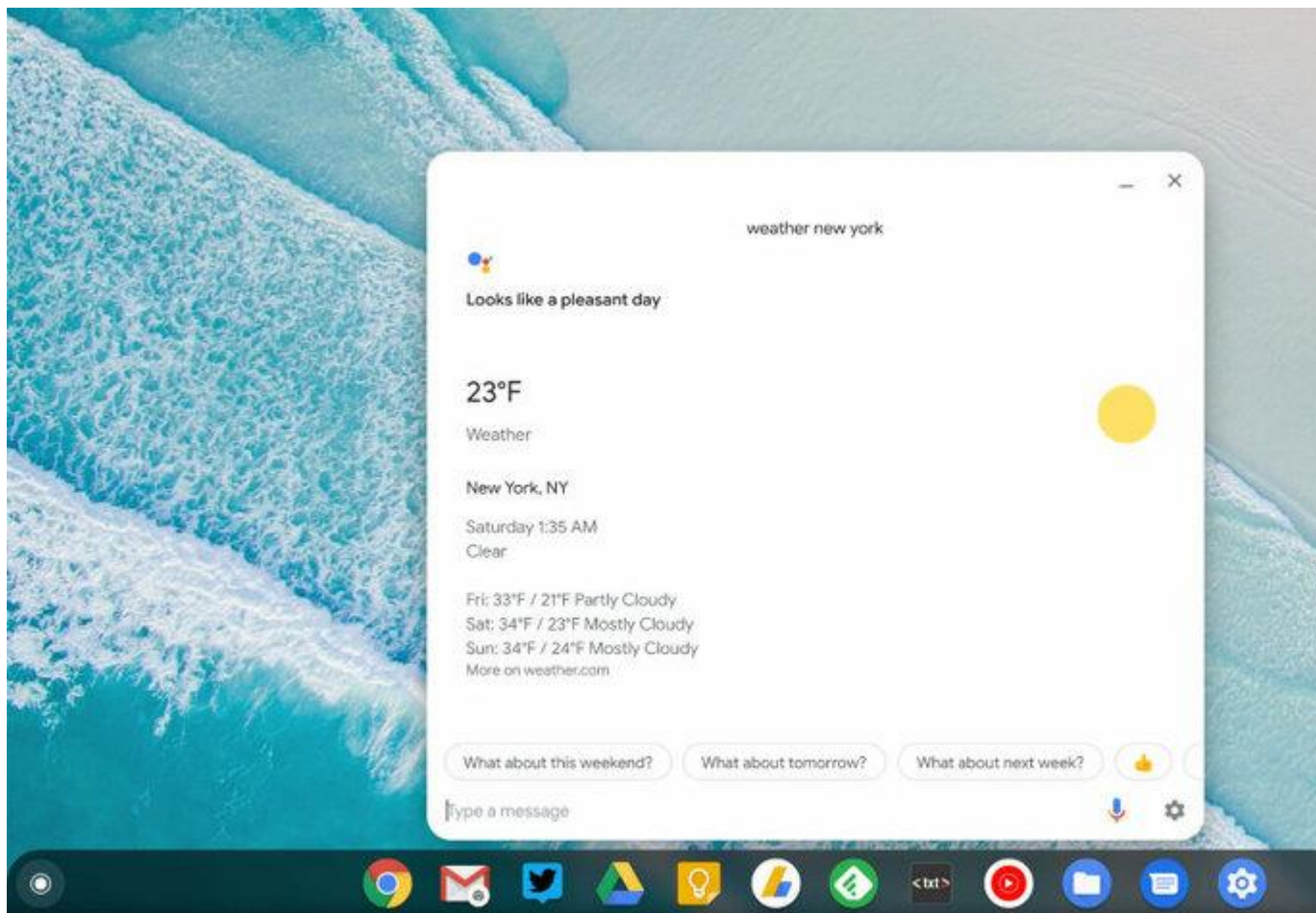
Бесплатная операционная система с открытым исходным кодом на базе UNIX. Она совместима с различными платформами, ориентирована на скорость и стабильность работы. Наиболее захватывающая часть истории платформы — ее создание общими силами сообщества в Калифорнийском университете.

Особенности:

- расширенные возможности для работы в сети, совместимости и безопасности, которые отсутствуют в других операционных системах;
- идеально подходит для услуг интернета и интрасети, может справляться с большими нагрузками и эффективно управлять памятью, чтобы поддерживать работу сразу нескольких пользователей;
- усовершенствованная платформа для работы с высокопроизводительными системами на базе процессоров Intel;
- простая и быстрая установка прямо из сети.

Вывод: надежная операционная система, построенная внушительным сообществом студентов. Она лучше всего подходит для работы в сети, отличается простотой установки и возможностью быстрого одновременного использования сразу несколькими пользователями.

7. Chrome OS



Источник: Engadget

- **Предназначение:** фирменные веб-приложения.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [Chrome OS](https://www.google.com/chromeos/).

Очередная операционная система, которая разработана на базе ядра Linux. За ее создание отвечает компания Google, которая фактически построила программную платформу вокруг своего фирменного браузера Chrome. Она заточена под работу с веб-приложениями.

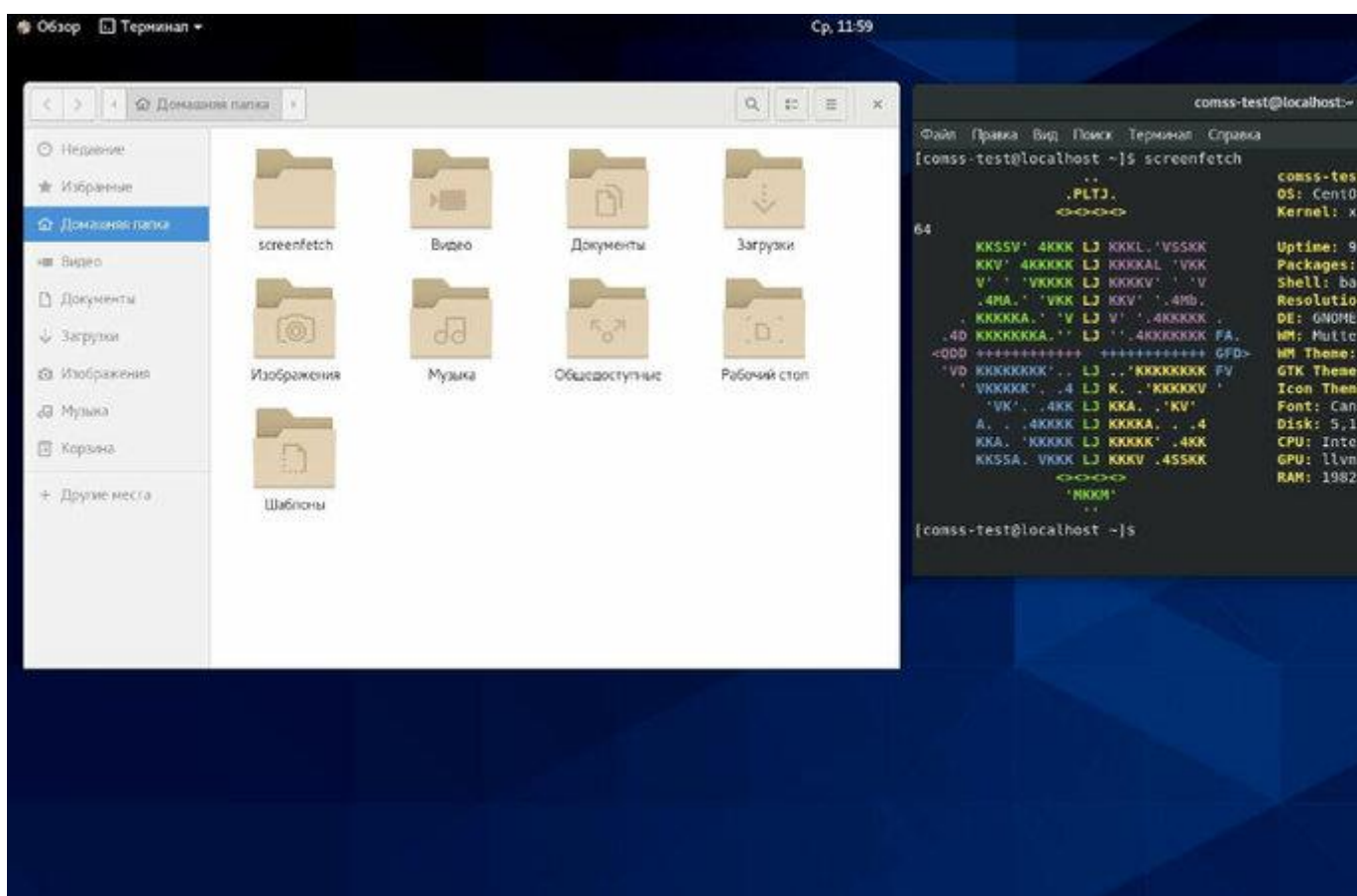
Особенности:

- встроенные инструменты для автономной работы с мультимедийными файлами;
- удаленный доступ к приложениям и виртуальному рабочему столу;

- полная совместимость со всеми приложениями, разработанными для Android;
- запуск приложений, которые созданы для Linux.

Вывод: неординарная операционная система, которая отлично работает, но все еще не имеет четкого направления для развития. На данный момент она отлично подходит для недорогих компьютеров, общественного использования, мультимедийного контента. Но ее будущее все еще под вопросом.

8. CentOS



Источник: Comms

- **Предназначение:** разработка, личное и коммерческое использование.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [CentOS](https://www.centos.org/).

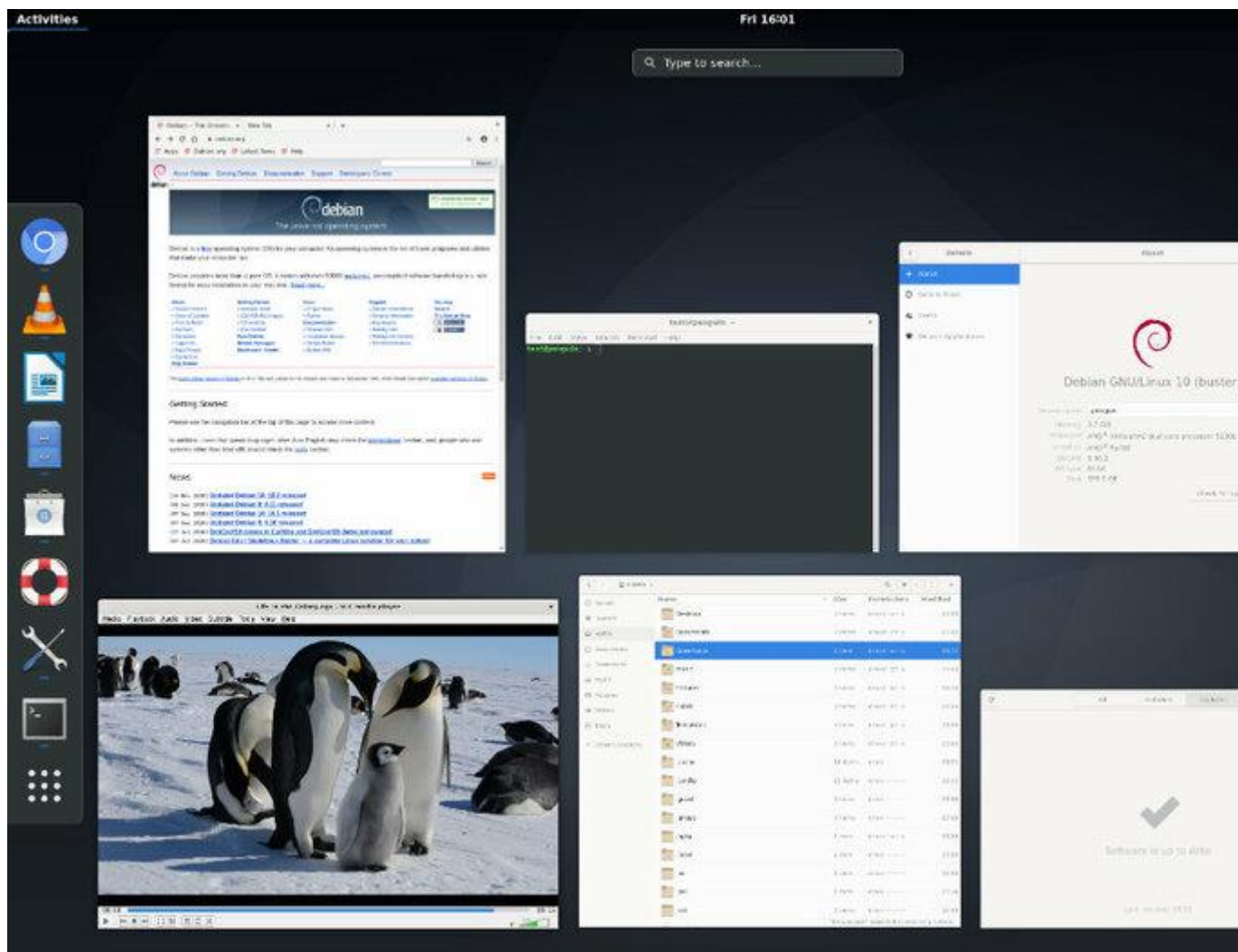
Очередная бесплатная операционная система, развитием которой занимается сообщество. Она пользуется спросом у программистов.

Особенности:

- обширный потенциал для разработчиков, желающих создавать софт и тестировать его;
- расширенные возможности для работы в сети, совместимости и безопасности, которые сейчас отсутствуют во многих других системах;
- инструменты совместимости для решения аппаратных и программных проблем;
- инструменты для тонкой настройки безопасности, управления процессами, настройки доступа к данным.

Вывод: операционную систему чаще всего выбирают разработчики программного обеспечения — для личного и домашнего использования она все же подходит не настолько хорошо. Платформа бесплатно предоставляет создателям приложений все необходимые инструменты для разработки.

9. Debian



Источник: Wiki

- **Предназначение:** разработка приложений.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [Debian](https://www.debian.org).

Еще одна бесплатная операционная система с открытым кодом, созданная на базе Linux. Она отличается максимальной легкостью для аппаратного обеспечения, простотой установки, удобным интерфейсом, а также десятками тысяч предварительно скомпилированных приложений в удобном формате.

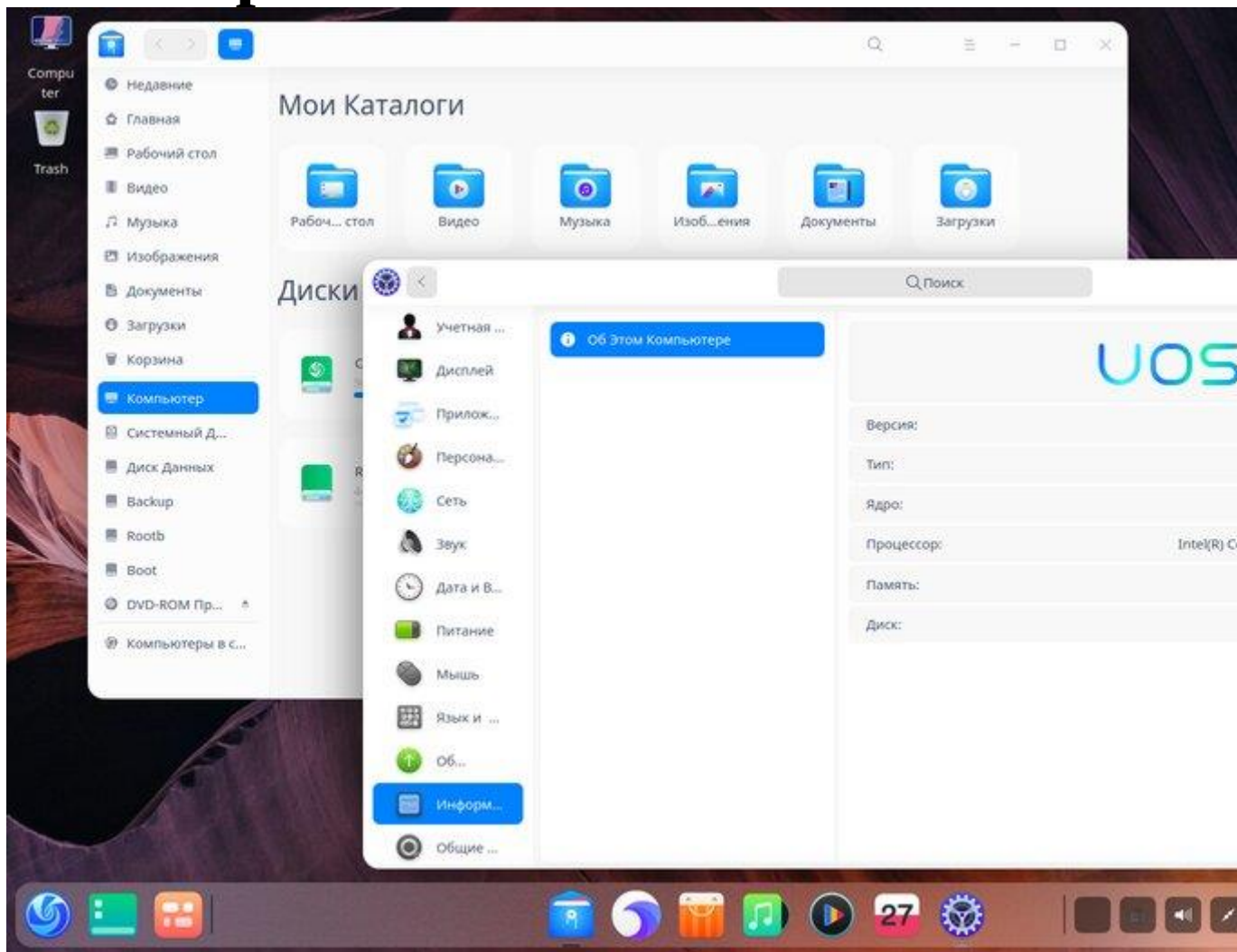
Особенности:

- высокая скорость работы вне зависимости от «железа»;
- встроенный брандмауэр для защиты данных;

- максимально простая установка;
- расширенные возможности для работы с сетью.

Вывод: от других операционных систем Debian отличается куда меньшей универсальностью, поэтому устанавливать ее нужно лишь в том случае, если у вас есть представление по поводу ее возможностей. Тем не менее, она абсолютно бесплатна, поэтому ничего не мешает попробовать.

10. Deepin



Источник: komYounity

- **Предназначение:** разработка приложений.
- **Стоимость:** бесплатно.
- **Веб-сайт:** [Deepin](https://deepin.org/en).

Операционная система с открытым исходным кодом, которая основана на стабильной ветке Debian. Она отличается инструментом DeepinDesktopEnvironment, на базе которого создан удобный пользовательский интерфейс. Ее хвалят за эстетичный и привлекательный внешний вид.

Особенности:

- удобство, надежность и привлекательный интерфейс;
- расширенные возможности безопасности;
- простая установка;
- внушительный набор встроенного программного обеспечения, который включает установщик шрифтов, файловый менеджер, скриншотер, диктофон, просмотрщик изображений и плеер для фильмов.

Вывод: бесплатная операционная система, которая исправляет большинство недостатков Debian. У нее внушительное число модификаций, которые постепенно набирают популярность. Есть вероятность, что в будущем именно эта платформа сможет в полной мере конкурировать с macOS и Windows

Подводя итоги

Для домашнего использования лучше всего подходят Windows и macOS. Первая лучше для игр, а вторая отличается эксклюзивным программным обеспечением и продвинутой работой в экосистеме. Тем не менее, когда речь заходит про скорость работы, в лидеры уверенно выбивается Linux, которому не нужны аппаратные мощности. Чтобы работать на оптимальном уровне, ему даже производительный процессор не понадобится. Операционные системы на базе Linux часто используются

в корпоративном секторе — им отдают работы с массивами информации и базами данных.

Внимания также могут заслуживать менее популярные системы, которые не попали в подборку. Syllable — бесплатная альтернатива домашним платформам. ReactOS — когда-то клон Windows 95, а сегодня отдельная полноценная операционная система.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программное обеспечение (англ. *software*) — это совокупность программ, обеспечивающих функционирование компьютеров и решение с их помощью задач предметных областей. Программное обеспечение (ПО) представляет собой неотъемлемую часть компьютерной системы, является логическим продолжением технических средств и определяет сферу применения компьютера.

ПО современных компьютеров включает множество разнообразных программ, которое можно условно разделить на три группы (рис. 3.1):

1. Системное программное обеспечение (системные программы);
2. Прикладное программное обеспечение (прикладные программы);
3. Инструментальное обеспечение (инструментальные системы).

Системное программное обеспечение (СПО) – это программы, управляющие работой компьютера и выполняющие различные вспомогательные функции, например, управление ресурсами компьютера, создание копий информации, проверка работоспособности устройств компьютера, выдача справочной информации о компьютере и др. Они предназначены для всех категорий пользователей, используются для эффективной работы компьютера и пользователя, а также эффективного выполнения прикладных программ.

Центральное место среди системных программ занимают операционные системы (англ. *operatingsystems*). **Операционная система** (ОС) – это комплекс программ, предназначенных для управления загрузкой, запуском и выполнением других пользовательских программ, а также для планирования и управления вычислительными ресурсами ЭВМ, т.е. управления работой ПЭВМ с момента включения до момента выключения питания. Она загружается автоматически при включении компьютера, ведет диалог с пользователем, осуществляет управление компьютером, его ресурсами (оперативной памятью, дисковым пространством и т.д.), запускает другие программы на выполнение и обеспечивает пользователю и программам удобный способ общения – **интерфейс** – с устройствами компьютера. Другими словами, операционная система обеспечивает функционирование и взаимосвязь всех компонентов компьютера, а также предоставляет пользователю доступ к его аппаратным возможностям.

ОС определяет производительность системы, степень защиты данных, выбор программ, с которыми можно работать на компьютере, требования к аппаратным средствам. Примерами ОС являются MS DOS, OS/2, Unix, Windows 9x, Windows XP.

Сервисные системы расширяют возможности ОС по обслуживанию системы, обеспечивают удобство работы пользователя. К этой категории относят системы технического обслуживания, программные оболочки и среды ОС, а также служебные программы.

Системы технического обслуживания – это совокупность программно-аппаратных средств ПК, которые выполняют контроль, тестирование и диагностику и используются для проверки функционирования устройств компьютера и обнаружения неисправностей в процессе работы компьютера. Они являются инструментом специалистов по эксплуатации и ремонту технических средств компьютера.

Для организации более удобного и наглядного интерфейса пользователя с компьютером используются **программные оболочки операционных систем** – программы, которые позволяют пользователю отличными от предоставляемых ОС средствами (более понятными и эффективными) осуществлять действия по управлению ресурсами компьютера. К числу наиболее популярных оболочек относятся пакеты NortonCommander (Symantec), FAR (FileandArchivemanager) (Е.Рошаль).

Служебные программы (утилиты, лат. utilitas – польза) – это вспомогательные программы, предоставляющие пользователю ряд дополнительных услуг по реализации часто выполняемых работ или же повышающие удобство и комфортность работы. К ним относятся:

- программы-упаковщики (архиваторы), которые позволяют более плотно записывать информацию на дисках, а также объединять копии нескольких файлов в один, так называемый, архивный файл (архив);
- антивирусные программы, предназначенные для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения;
- программы оптимизации и контроля качества дискового пространства;
- программы восстановления информации, форматирования, защиты данных;
- программы для записи компакт-дисков;
- драйверы – программы, расширяющие возможности операционной системы по управлению устройствами ввода/вывода, оперативной памятью и т.д. При подключении к компьютеру новых устройств необходимо установить соответствующие драйверы;
- коммуникационные программы, организующие обмен информацией между компьютерами и др.

Некоторые утилиты входят в состав операционной системы, а некоторые поставляются на рынок как самостоятельные программные продукты, например, многофункциональный пакет сервисных утилит NortonUtilities (Symantec).

Прикладное программное обеспечение (ППО) предназначено для решения задач пользователя. В его состав входят **прикладные программы пользователей** и **пакеты прикладных программ** (ППП) различного назначения.

Прикладная программа пользователя – это любая программа, способствующая решению какой-либо задачи в пределах данной проблемной области. Прикладные программы могут использоваться либо автономно, либо в составе программных комплексов или пакетов.

Пакеты прикладных программ (ППП) – это специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией. Различают следующие типы ППП:

- **ППП общего назначения** – универсальные программные продукты, предназначенные для автоматизации широкого класса задач пользователя. К ним относятся:

Текстовые редакторы (например, MS Word, WordPerfect, Лексикон);

Табличные процессоры (например, MS Excel, Lotus 1-2-3, QuattroPro);

Системы динамических презентаций (например, MS PowerPoint, Freelance Graphics, HarvardGraphics);

Системы управления базами данных (например, MS Access, Oracle, MS SQL Server, Informix);

Графические редакторы (например, CorelDraw, AdobePhotoshop);

Издательские системы (например, PageMaker, VenturePublisher);

Системы автоматизации проектирования (например, BpWin, ERWin);

Электронные словари и системы перевода (например, Prompt, Сократ, Лингво, Контекст);

Системы распознавания текста (например, FineReader, CuneiForm).

Системы общего назначения часто интегрируются в многокомпонентные пакеты для автоматизации офисной деятельности – **офисные пакеты** – MicrosoftOffice, StarOffice и др.

- *методо-ориентированные ППП*, в основе которых лежит реализация математических методов решения задач. К ним относятся, например, системы математической обработки данных (Mathematica, MathCad, Maple), системы статистической обработки данных (Statistica, Stat).;

- *проблемно-ориентированные ППП* предназначены для решения определенной задачи в конкретной предметной области. Например, информационно-правовые системы ЮрЭксперт, ЮрИнформ; пакеты бухгалтерского учета и контроля 1С: Бухгалтерия, Галактика, Анжелика; в области маркетинга – Касатка, MarketingExpert; банковская система СТБанк;

- *интегрированные ППП* представляют собой набор нескольких программных продуктов, объединенных в единый инструмент. Наиболее развитые из них включают в себя текстовый редактор, персональный менеджер (органайзер), электронную таблицу, систему управления базами данных, средства поддержки электронной почты, программу создания презентационной графики. Результаты, полученные отдельными подпрограммами, могут быть объединены в окончательный документ, содержащий табличный, графический и текстовый материал. К ним относят, например, MS Works. Интегрированные пакеты, как правило, содержат некоторое ядро, обеспечивающее возможность тесного взаимодействия между составляющими.

Обычно пакеты прикладных программ имеют средства настройки, что позволяет при эксплуатации адаптировать их к специфике предметной области.

К **инструментальному программному обеспечению** относят: **системы программирования** – для разработки новых программ, например, Паскаль, Бейсик. Обычно они включают: *редактор текстов*, обеспечивающий создание и редактирование программ на исходном языке программирования (исходных программ), *транслятор*, а также *библиотеки подпрограмм*; **инструментальные среды** для разработки приложений, например, C++, Delphi, VisualBasic, Java, которые включают средства визуального программирования; **системы моделирования**, например, система имитационного моделирования MatLab, системы моделирования бизнес-процессов BpWin и баз данных ErWin и другие.

Транслятор (англ. *translator* – переводчик) – это программа-переводчик, которая преобразует программу с языка высокого уровня в программу, состоящую из машинных команд. Трансляторы реализуются в виде *компиляторов* или *интерпретаторов*, которые существенно различаются по принципам работы.

Компилятор (англ. *compiler* – составитель, собиратель) читает всю программу *целиком*, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется. После компилирования получается исполняемая программа, при выполнении которой не нужна ни исходная программа, ни компилятор.

Интерпретатор (англ. *interpreter* – истолкователь, устный переводчик) переводит и выполняет программу *строка за строкой*. Программа, обрабатываемая интерпретатором, должна заново переводиться на машинный язык при каждом очередном ее запуске.

Откомпилированные программы работают быстрее, но интерпретируемые проще исправлять и изменять.

Прикладное программное обеспечение

Прикладное программное обеспечение (ППО) составляют программы конечного пользователя. Это самый обширный класс [программного обеспечения](#). В настоящее время в большинстве сфер человеческой деятельности разработаны и применяются прикладные программные продукты. Везде, где требуется выполнить большие математические расчеты или производится обработка больших объемов разнообразных данных, или требуется быстрый анализ ситуации с принятием управляющего решения, – компьютеры под управлением прикладного программного обеспечения с успехом заменяют человека.

Классификация прикладного программного обеспечения

Прикладное программное обеспечение можно разделить на три большие группы (рисунок 3.4). Такая классификация весьма условна, потому что некоторые типы программ (например, программы обработки текста) имеют своих представителей и в классе общего назначения (редакторы и процессоры) и в классе профессиональных программ (издательские системы).

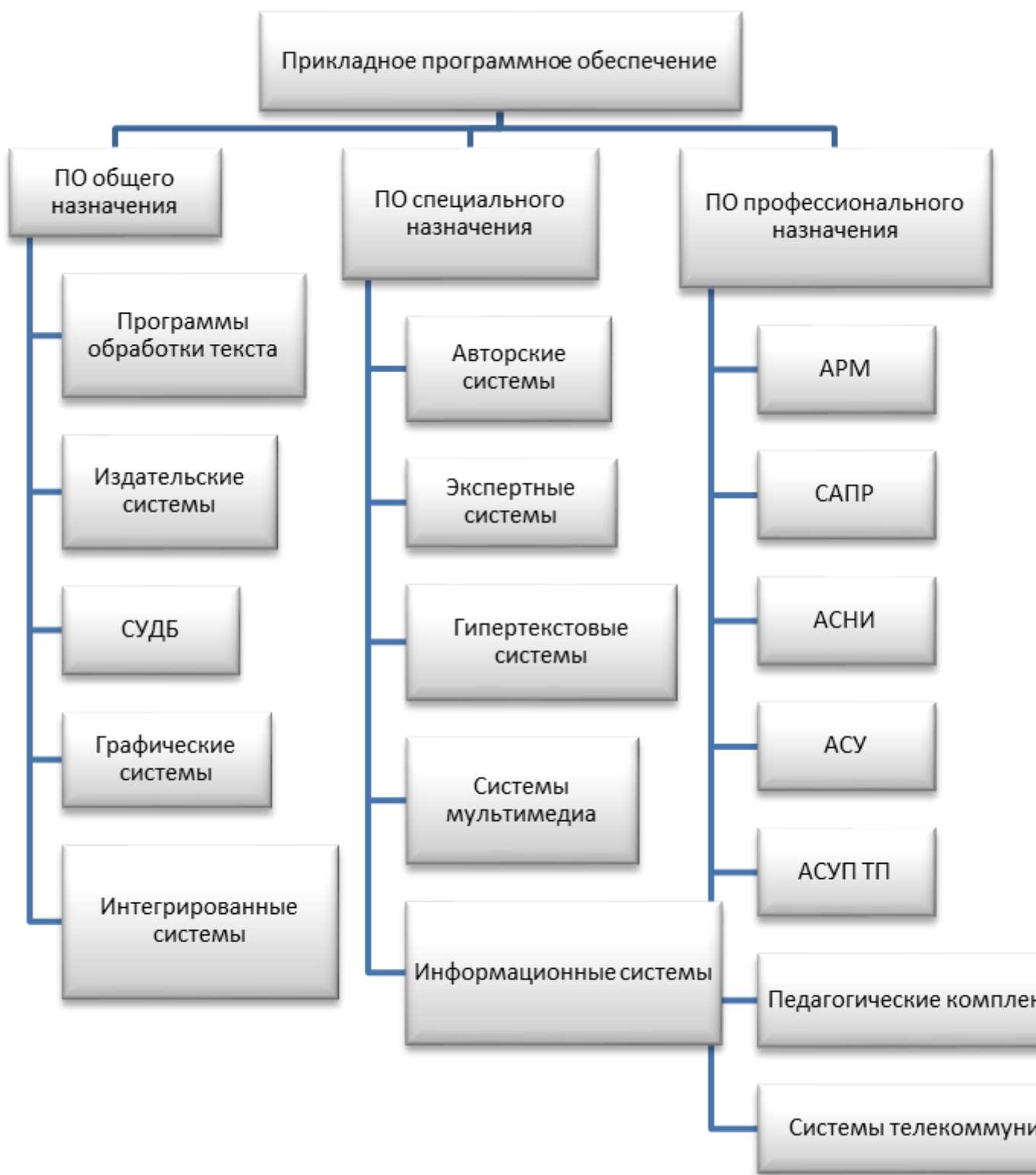


Рисунок 3.4 – Классификация прикладного программного обеспечения

Программные средства общего назначения

Прикладное программное обеспечение общего назначения используется для решения наиболее общих задач информационного характера в любой сфере человеческой деятельности. Оно объединяет в себе широко используемые программы большинством пользователей персональных компьютеров, например,

текстовые редакторы, электронные таблицы, [базы данных](#), [графические системы](#), игры, развлечения.

К **программам обработки текста** относятся текстовые редакторы, [текстовые процессоры](#). Граница между ними весьма условна. Текстовые редакторы, например *NotePad* – разработка Microsoft, способны выполнять основные функции редактирования: набор, внесение исправлений, сохранение, работа с фрагментами. Текстовые процессоры, например *Word* – разработка Microsoft, кроме того, имеют возможности разнообразного оформления, а некоторые позволяют создавать документы, предназначенные для просмотра не в бумажном виде, а на компьютере (электронные документы).

Издательские системы автоматизируют процесс верстки полиграфических изданий. Они отличаются расширенными средствами управления взаимодействия текста с параметрами страницы и графическими объектами, но имеют более слабые возможности по автоматизации ввода и редактирования текста. Их целесообразно применять к документам, которые предварительно обработаны в текстовых процессорах и графических редакторах. Наиболее известными из издательских пакетов являются: AdobePageMaker, QuarkXPress, MicrosoftPublisher, CorelVentura.

[Электронные таблицы](#). Основное назначение электронных таблиц – обработка различных типов данных табличной формы, например, планово-финансовые, бухгалтерские документы, небольшие инженерные расчеты (*Excel* — разработка Microsoft, *Lotus 1-2-3* — разработка Lotus). Основное преимущество электронных таблиц, в сравнении с текстовыми процессорами (где тоже могут вестись таблицы, производиться небольшие вычисления и сортировка), в том, что содержание одних ячеек может меняться автоматически в соответствии с изменением содержания других.

Иными словами, ячейки могут быть функционально зависимы. Кроме того, табличные процессоры имеют возможности ведения небольших баз данных и визуализации данных в виде различных таблиц, диаграмм и графиков, т.е. средства ведения таблиц, средства табличных расчетов подкрепляются возможностями создания наглядных отчетов. Они находят широкое применение в бухгалтерском учете, анализе финансовых и торговых рынков, средствах обработки результатов научных и экономических экспериментов, т.е. в автоматизации регулярно повторяемых вычислений больших объемов числовых и текстовых данных, представляющих табличные структуры.

Системы управления базами данных (СУБД). Программы этого класса (например, *Access* – разработка Microsoft) позволяют работать с большими объемами структурированных данных – базами данных (как правило, это табличные структуры). СУБД предназначены для автоматизации процедур создания, хранения и извлечения электронных данных. Многие, существующие в различных сферах человеческой деятельности, информационно-справочные

программные комплексы реализованы с использованием инструментальных средств СУБД.

С базами данных (БД) обычно работают две категории исполнителей: проектировщики (разрабатывают структуру для хранения информации) и пользователи (наполняют БД). Структура простейшей БД аналогична структуре обычной таблицы. Каждая запись соответствует строке, каждое поле – столбцу. Свойства данных в полях (столбцах) могут различаться в зависимости от их типа. Таким образом, поля в базе данных определяют не только общий вид таблицы, но и групповые свойства данных.

Это обуславливает специфические особенности при работе с программными средствами данного класса. Кроме того, работа с файлами создаваемыми СУБД, несколько отличается от работы с файлами других типов, создаваемых прочими приложениями. В частности ряд операций по преобразованию файлов осуществляется в обход требований операционной системы для обеспечения целостности данных. В случае, когда БД содержит несколько таблиц, между таблицами устанавливаются специальные связи, называемые *реляционными отношениями*. Наиболее распространенными пакетами СУБД для ПК являются MS Access, Oracle, Paradox, dBase.

Большинство современных СУБД позволяют создавать небольшие программы обработки данных на встроенных языках, имеют оформительские возможности, позволяющие на основе собранных и обработанных данных создать отчет. Множество СУБД, также как и текстовые процессоры, имеют своих представителей и в ППО общего и в ППО специального назначения. На уровне ППО общего назначения – это настольные СУБД, на уровне специальном – это большие СУБД, составляющие основу информационных систем и позволяющие работать в компьютерных сетях.

Графические системы. Это программы, предназначенные для работы с графическими изображениями. К ним относятся редакторы растровой и векторной графики, программы обработки трехмерной графики (3D-редакторы).

Растровые редакторы для представления изображений используют растры, т.е. совокупности точек, имеющих свой цвет и яркость. В них удобно обрабатывать фотографии и объекты, имеющие мягкие цветовые переходы. К сожалению, масштабирование таких картинок в любую сторону обычно ухудшает качество. При уменьшении количества точек теряются мелкие детали и деформируются надписи.

Добавление пикселей приводит к ухудшению резкости и яркости изображения, так как новым точкам приходится давать оттенки, средние между двумя и более граничащими цветами. Признанный лидер среди программ данного класса –

Adobe Photoshop. Стандартным средством обработки растровых изображений в Windows является графический редактор *Paint*. В настоящее время распространены следующие форматы растровой графики .bmp, .psx, .gif, .tif, jpg, .png и др. В Интернете графика представляется в одном из растровых форматов, понимаемых браузерами без установки дополнительных модулей – gif, jpg, png.

Векторные редакторы удобны для работы с чертежами и рисованными картинками. Своего рода стандартом в этом классе являются пакеты *CorelDraw* и *AdobeIllustrator*.

В векторной графике изображения описываются с помощью кривых линий, называемых векторами (каждая кривая аппроксимируется многочленом третьего порядка, т.е. массивом коэффициентов – многомерным вектором), а также параметров, описывающих их цвета и расположение. Например, изображение какой-либо фигуры на экране описывается точками, через которые проходит линия контура фигуры. Цвет фигуры задается цветом контура и цветом области внутри этого контура.

При редактировании элементов векторной графики можно изменять параметры линий, описывающих форму графических объектов, можно переносить их, менять размер, форму (это делается математическими преобразованиями), цвет, но это не отразится на качестве их визуального представления. Векторная графика не зависит от разрешения, т.е. может быть показана в разнообразных выходных устройствах с различным разрешением без потери качества. Очевидно, описание простых векторных графических объектов занимает значительно меньше места, чем растровых.

Еще одно преимущество – качественное масштабирование в любую сторону. Увеличение или уменьшение объектов производится увеличением или уменьшением соответствующих коэффициентов в математических формулах. Но векторный формат становится невыгодным при передаче изображений с большим количеством оттенков или мелких деталей (например, фотографий). Таким образом, выбор растрового или векторного формата зависит от целей и задач работы с изображением.

Редакторы трехмерной графики используются для создания пространственных графических композиций, позволяют проследить взаимодействия трехмерных объектов между собой и трехмерных объектов с источником света. Как правило, в таких редакторах сочетается векторный и растровый способы формирования изображений. Пакеты трехмерной графики отличаются богатыми возможностями моделирования, позволяют гибко управлять взаимодействием свойств поверхности объектов со свойствами источников освещения, включают большое число кинематографических возможностей, однако, часто, довольно требовательны к аппаратуре. Среди пакетов этого класса наиболее известны: *3D StudioMax*, *Maya*, *Softimage 3D*.

Техническим развитием графических редакторов являются специальные *системы машинной графики*, предназначенные для автоматизации проектно-конструкторских работ в машиностроении, промышленной электронике, строительстве и т.д. Подобные пакеты включают разнообразные чертежные инструменты, и средства трехмерного моделирования, облегчающие проектирование, готовые библиотеки стандартных элементов чертежей и (или) схем, предоставляют развитые средства управления изображением. Также они часто включают в свой состав встроенные макроязыки или языки программирования, средства экспорта-импорта файлов различных форматов. Стандартом среди зарубежных программ является пакет AutoCAD фирмы Autodesk. Примеры отечественных пакетов: *CADMECH, T-Flex CAD, SprutCAD, APM Graph, bCAD, АДЕМ, КОМПАС-ГРАФИК*.

Интегрированные программные средства. Отдельные программы, являясь мощным средством решения круга прикладных задач, не могут в полной мере удовлетворить пользователя. Например, выборку данных, предоставленную СУБД, бывает удобно обработать с помощью электронных таблиц, результаты, оформленные в виде наглядных таблиц, поместить в отчет, представляющий собой текстовый документ, который был составлен в текстовом процессоре.

Для совместной работы нескольких программ требуется и унификация форматов обрабатываемых файлов. Такие программные пакеты называются интегрированными программными средствами. Наиболее распространенный продукт этого класса – пакет MS Office (разработка Microsoft), который кроме текстового процессора MS Word, табличного процессора MS Excel и СУБД MS Access интегрирует в себе такие офисные программные средства, как система разработки презентаций MS PowerPoint, электронный организатор MS Outlook и др.

Объединение функций различных прикладных программ в единую систему приводит к созданию интегрированных пакетов программ, а далее к созданию автоматизированных рабочих мест (АРМ).

Программные средства для решения прикладных математических задач – ППП ориентированные на решение математических задач. При этом под математической понимается любая задача, алгоритм решения которой может быть описан в терминах того или иного раздела математики. Большинство современных систем компьютерной математики сочетают в себе возможности проведения расчетов и подготовки форматированных научно-технических документов. Наиболее популярными пакетами данного класса являются: *MathCAD, Maple, Matlab, Mathematica, Statistica*.

Специализированные системы [математического моделирования](#) – ППП, ориентированные на решение научно-прикладных задач в различных областях

инженерных знаний (задачи механики жидкости и газа, расчеты на прочность, температурный, вибрационный анализ, моделирование магнитных полей и др.). Сюда можно отнести ряд пакетов, разрабатываемых фирмами *ANSYS, Inc.*, *MSC*, *SAMTECH* и др.

Прикладное программное обеспечение специального назначения

Разработчики создают специальные программные системы целевого назначения для специалистов в некоторой предметной области. Такие программы называют авторскими инструментальными системами.

Авторская система представляет интегрированную среду с заданной интерфейсной оболочкой, которую пользователь может наполнить информационным содержанием своей предметной области.

Экспертные системы – это программы, которые ведут себя подобно эксперту в некоторой узкой прикладной области. Они предназначены для анализа данных, хранящихся в базах знаний. В отличие от СУБД, позволяющих производить операции манипуляции данными, экспертные системы производят логический анализ данных, имеют функции самообучения.

Экспертные системы решают задачи с неопределенностью и неполными исходными данными, требующие для своего решения экспертных знаний. Кроме того, эти системы должны уметь объяснять свое поведение и свое решение. Принципиальным отличием экспертных систем от других программ является их адаптивность, т.е. изменчивость в процессе самообучения.

В экспертных системах принято выделять три основных модуля:

- модуль базы знаний;
- модуль логического вывода;
- интерфейс с пользователем.

Экспертные системы, являющиеся основой искусственного интеллекта, получили широкое распространение в науке (классификация животных и растений по видам, химический анализ), в медицине (постановка диагноза, анализ электрокардиограмм, определение методов лечения), в технике (поиск неисправностей в технических устройствах, слежение за полетом космических кораблей и спутников), в политологии и социологии, юриспруденции, лингвистике и т.д.

В последнее время широкую популярность получили программы обработки гипертекстовой информации. **Гипертекст** – это форма организации текстового материала не в линейной последовательности, а в форме указаний возможных переходов (ссылок), связей между отдельными его фрагментами. В обычном тексте используется обычный линейный принцип размещения информации, и

доступ к нему осуществляется последовательно. В гипертекстовых системах информация напоминает текст энциклопедии, и доступ к любому выделенному фрагменту текста осуществляется произвольно по ссылке. Организация информации в гипертекстовой форме используется при создании справочных пособий, словарей, контекстной помощи (Help) в прикладных программах.

Расширение концепции гипертекста на графическую и звуковую информацию приводит к понятию гипермедиа. Идеи гипермедиа получили распространение в сетевых технологиях, в частности в Интернет-технологиях. Технология WWW (WorldWideWeb) позволила структурировать громадные мировые информационные ресурсы посредством гипертекстовых ссылок. Появились программные средства, позволяющие создавать подобные Web-странички. Стали развиваться механизмы поиска нужной информации в лабиринте информационных потоков.

Мультимедиа (multimedia) – это взаимодействие визуальных и аудиоэффектов под управлением интерактивного программного обеспечения. Мультимедийные игровые и обучающие системы начинают вытеснять традиционные «бумажные библиотеки». Сегодня в библиотеках CD-ROM можно «гулять» по музеям.

Важным типом ППП являются оболочки информационных систем – среда разработчика и администратора, предназначенная для разработки и сопровождения автоматизированных информационных систем (АИС). **АИС** – это комплекс автоматизированных информационных технологий, предназначенный для информационного обслуживания – организованного непрерывного технологического процесса подготовки и выдачи потребителям научной, управленческой и др. информации, используемой для принятия решений в соответствии с нуждами для поддержания эффективной деятельности.

Информационные системы предоставляют широкие возможности:

- в управлении предприятием, например, склад, документооборот офиса (1С: Предприятие);
- в бухгалтерском учете, например, системы, имеющие функции текстовых, табличных редакторов и СУБД. Предназначены для автоматизации подготовки начальных бухгалтерских документов предприятия и их учета, регулярных отчетов по итогам производственной, хозяйственной и финансовой деятельности в форме, приемлемой для налоговых органов, внебюджетных фондов и органов статистического учета ([1С](#): бухгалтерия);
- в анализе экономической и финансовой деятельности; их используют в банковских и биржевых структурах. Они позволяют контролировать и

прогнозировать ситуацию на финансовых, торговых рынках и рынках сырья, выполнять анализ текущих событий, готовить отчеты.

Проектирование информационных систем представляет собой сложную, трудоемкую и длительную работу, требующую высокой квалификации участвующих в ней специалистов.

Тенденции развития современных информационных технологий определяют постоянное возрастание сложности ПО информационных систем, что вызвало потребность в программно-технологических средствах специального класса – CASE-средствах, реализующих [CASE-технологии](#) создания и сопровождения ИС. Термин CASE (ComputerAidedSoftwareEngineering) имеет весьма широкое толкование. Первоначально значение термина CASE ограничивалось вопросами автоматизации разработки только лишь программного обеспечения, а в настоящее время оно приобрело новый смысл и охватывает процесс разработки сложных ИС в целом.

CASE-технология представляет собой совокупность методов проектирования ИС, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех стадиях разработки и сопровождения ИС и разрабатывать приложения в соответствии с информационными потребностями пользователей. Большинство существующих CASE-средств основано на методах структурного или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

Были созданы такие CASE-системы как:

- ориентированные на этапы жизненного цикла ПО, Design/IDEF (MetaSoftware), BPWin (LopicWorks);
- функционально полные, т.е. используемые на всех этапах жизненного цикла Designer (Oracle) Developer/2000 (Oracle);
- независимые от СУБД ODBC Sdesigner (SPD), ERWin (Logic- Works), Silverrun (Silverrun Technologies).

Прикладные программы профессионального уровня

Каждая прикладная программа этой группы ориентируется на достаточно узкую предметную область, но проникает в нее максимально глубоко. Так функционируют АСНИ – автоматизированные системы научных исследований, каждая из которых «привязана» к определенной области науки, САПР системы автоматизированного проектирования, каждая из которых также работает в узкой области, АСУ – ? автоматизированные системы управления.

Системы автоматизированного проектирования предназначены для выполнения проектных работ с применением математических методов и компьютерной техники. Системы САПР широко используются в архитектуре, электронике, энергетике, механике и др. В процессе автоматизированного проектирования в качестве входной информации используются технические знания специалистов, которые вводят проектные требования, уточняют результаты, проверяют полученную конструкцию, изменяют ее и т.д. Кроме того, в САПР накапливается информация, поступающая из библиотек стандартов (данные о типовых элементах конструкций, их размерах, стоимости и др.). В процессе проектирования разработчик вызывает определенные программы и выполняет их. Из САПР информация выдается в виде готовых комплектов законченной технической и проектной документации.

Автоматизированные системы научных исследований (АСНИ) предназначены для автоматизации научных экспериментов, а также для осуществления моделирования исследуемых объектов, явлений и процессов, изучение которых традиционными средствами затруднено или невозможно. В настоящее время научные исследования во многих областях знаний проводят большие коллективы ученых, инженеров и конструкторов с помощью весьма сложного и дорогого оборудования.

Большие затраты ресурсов для проведения исследований обусловили необходимость повышения эффективности всей работы. Эффективность научных исследований в значительной степени связана с уровнем использования компьютерной техники.

Компьютеры в АСНИ используются в информационно-поисковых и экспертных системах, а также решают следующие задачи:

- управление экспериментом;
- подготовка отчетов и документации;
- поддержание базы экспериментальных данных и др.

В результате применения АСНИ возникают следующие положительные моменты:

- в несколько раз сокращается время проведения исследования;
- увеличивается точность и достоверность результатов;
- усиливается контроль за ходом эксперимента;
- сокращается количество участников эксперимента;

- повышается качество и информативность эксперимента за счет увеличения числа контролируемых параметров и более тщательной обработки данных;
- результаты экспериментов выводятся оперативно в наиболее удобной форме графической или символьной (например, значения функции многих переменных выводятся средствами машинной графики в виде так называемых «горных массивов»).

На экране одного графического монитора возможно формирование целой системы приборных шкал (вольтметров, амперметров и др.), регистрирующих параметры экспериментального объекта.

Каждая из систем АСНИ и САПР, конечно, имеет свою специфику и отличается поставленными целями и методами их достижения. Однако очень часто между обоими типами систем обнаруживается тесная связь, и их роднит не только то, что они реализуются на базе компьютерной техники. Например, в процессе проектирования может потребоваться выполнение того или иного исследования и, наоборот, в ходе научного исследования может возникнуть потребность и в конструировании нового прибора и в проектировании научного эксперимента.

Такая взаимосвязь приводит к тому, что на самом деле «чистых» АСНИ и САПР не бывает: в каждой из них можно найти общие элементы. С повышением их интеллектуальности они сближаются. В конечном счете и те и другие должны представлять собой экспертную систему, ориентированную на решение задач конкретной области.

Автоматизированные системы управления.

Термин, впервые появившийся в России в 1960-е гг. в связи с применением компьютеров и информационных технологий в управлении экономическими объектами и процессами. Это дало возможность повысить эффективность производства, лучше использовать ресурсы, избавить управленцев от выполнения нетворческих рутинных операций. Предполагалось создать иерархию автоматизированных систем управления, начиная с АСУ технологическими процессами (АСУТП) и АСУ подразделения организации и кончая общегосударственной системой управления, соединенных каналами связи.

В настоящее время в мировой практике для обозначения полнофункциональных интегрированных АСУ, используемых фирмами, применяют названия система управления ресурсами (англ. managementresourceplanning, MRP) и управление ресурсами предприятия (англ. enterpriseresourceplanning, ERP). Такие системы позволяют информационно поддерживать, обеспечивать все направления управленческой деятельности предприятия.

АСУ – это человекомашинные кибернетические системы, в которых умственная деятельность людей сочетается с переработкой информации, расчетами,

логическими операциями, проводимыми с использованием вычислительной техники и современных средств хранения, передачи и обработки информации.

В составе АСУ выделяют:

- основную часть, в которую входят информационное, техническое и математическое обеспечение;
- функциональную часть, к которой относятся взаимосвязанные программы, автоматизирующие конкретные функции управления.

АСУ применяются в управлении производством, транспортом, строительством и многими другими экономическими объектами и процессами.

Следует отметить не только условность предложенной выше классификации ППО, но и наличие пересечений. Так, каждую конкретную экспертную систему вполне можно отнести к ППО профессионального уровня; принцип гипертекста реализован в ряде авторских систем и т.д.

Программное и информационное обеспечение, применяемое в эксплуатационной деятельности на железнодорожном транспорте

Постоянное внедрение все более совершенной и мощной вычислительной техники, новых системно-технических решений, прикладного программного обеспечения, а также совершенствование технологии работы пользователей информационных систем привели к тому, что в конце 1970-х-начале 1980-х годов стал появляться новый тип информационных систем - комплексные системы. Вводится понятие "модель" как способ отображения фактической работы объекта, его "жизни". Первой такой моделью стала поездная модель, отражающая формирование, движение и расформирование поездов. Параллельно появляется модель сортировочной станции - основа автоматизированной системы управления работой сортировочной станции (АСУ СС), создается первая вычислительная сеть из 15 ИВЦ и первая работающая версия отечественной системы "Экспресс". Активизируются разработки в ГВЦ, ИВЦ Октябрьской, Куйбышевской, Южно-Уральской железных дорог, ПКТБ АСУЖТ, ВНИИЖТе. На железных дорогах создаются АСУ СС, АСОУП, внедряются единые комплексы ИОДВ, ИОММ.

С развитием программно-технической среды появилась возможность создания поездных и вагонных моделей сетевого уровня. В 1980-х годах началась эксплуатация на сетевом уровне системы автоматизированного диспетчерского центра управления (АДЦУ), информационной основой которой стала автоматизированная система оперативного управления перевозками (АСОУП). Создаются информационные системы: диалоговая информационная система контроля оперативного управления перевозками (ДИСКОР), контроль сменно-суточного планирования перевозок грузов (КССП), анализ погрузки нефтеналивных грузов (АПН), информационно-справочная система внешнеторговых грузов (ИСС ВТГ) и др. Разработан сменно-суточный доклад для руководителей МПС, информация из всех систем используется в практической работе функциональных служб дорог. Объем перевозок в тот период был наибольшим и

значительно превышал сегодняшний уровень. Потребность в информационных системах была повсеместной.

Настоящая революция в идеологии создания информационных систем произошла с появлением персональных компьютеров. Они послужили идеальными элементами для построения сетей. Стало возможным двигаться вперед более быстрыми темпами. Несмотря на недостаточную мощность первых персональных компьютеров, к концу 1980-х годов на их базе началось создание автоматизированных рабочих мест. Появилась возможность подойти к новому этапу - агрегированию в более мощные комплексы разнородных данных автоматизированных систем ИОДВ, АСОУП и др., работающих на сортировочных и грузовых станциях, контейнерных площадках.

Первые разработки, позволившие объединить разные информационные системы, выполняли специалисты ВНИИЖТа. В ИВЦ Октябрьской, Куйбышевской, Южно-Уральской, Целинной железных дорог были достигнуты значительные результаты в создании автоматизированной системы организации перевозки грузов по безбумажной технологии.

Между тем в МПС происходили структурные изменения. В 1988 г. Главное управление вычислительной техники было реорганизовано и вошло в состав Главного управления сигнализации и связи в качестве Управления вычислительной техники. Был ликвидирован самостоятельный орган, централизующий, объединяющий и координирующий создание информационных систем отрасли. Именно с того времени главки, а затем департаменты и хозяйства МПС стали самостоятельно заключать договоры на разработку, приобретать технику и программное обеспечение, что противоречило идеологии централизации создания информационных систем. Возникло множество организаций-разработчиков, создававших по заказам департаментов независимо эксплуатирующиеся задачи. В результате данные дублировались, порой многократно, возникали параллельные потоки при сборе и передаче информации.

ГВЦ становится интеллектуальным центром, организующим и направляющим работы по созданию современных программно-технических комплексов, изменению структуры управления вычислительными ресурсами отрасли, разработке новых информационных технологий.

В эксплуатацию вошли новые машинные залы, центр управления производством, оснащенный новейшими системами управления вычислительными процессами ГВЦ. В начале 1990-х годов специалисты принимали активное участие в выставках, совещаниях, семинарах по новым технологиям, неоднократно проходили зарубежную стажировку по применению современных программно-технических решений. Были изучены информационные системы управления на железных дорогах США, Великобритании, Германии, Австрии, Бельгии, Швейцарии, Финляндии, Польши. Полученные знания использовались при формировании новых подходов к автоматизации управления перевозочным процессом. Было написано два учебника по информационным технологиям и по телекоммуникациям для вузов и техникумов, а в ГВЦ открыт филиал МИИТа, в котором свыше 78 сотрудников без отрыва от производства получили высшее образование.

С 1995 по 2000 г. в отрасли прошла информационно-технологическая реформа. Была осуществлена планомерная замена программно-технических средств, определены принципы новых технологий. Приступили к созданию новых информационных систем и внедрению новых информационных технологий в управление производственной деятельностью на железных дорогах. Все это вместе позволило вывести вычислительную отрасль железнодорожного транспорта на уровень мировых достижений и обеспечить дальнейшее развитие в выбранном направлении.

С начала 1995 г. в ГВЦ прошло очередное техническое переоснащение. Там появились две ЭВМ IBM 9672 R 31 - первые машины класса mainframe, имевшие самую передовую по тем временам архитектуру. Были использованы стандартные средства СУБД и инструментальные средства SAS Institute, позволившие значительно продвинуться в разработке прикладных задач. Новый программно-технический комплекс стал базой для эксплуатации современных АСУ.

Принципиально важно, что в то время в ГВЦ начали проводить единую политику программного и технического перевооружения вычислительных центров, координировать их действия в развитии ПТК, внедрении и эксплуатации новых систем. По предложению ГВЦ и Дирекции Совета по железнодорожному транспорту выполнена основополагающая и результативная работа по созданию единого информационного пространства железных дорог государств - участников Содружества, Латвийской Республики, Литовской Республики, Эстонской Республики. Организована комиссия специалистов по информатизации железнодорожного транспорта, первым председателем которой был утвержден А.П. Писарев.

Структура информатизации предусматривала формирование информационной среды и инфраструктуры. На прикладном уровне предстояло создать комплексы информационных технологий по управлению: перевозочным процессом; маркетингом, экономикой и финансами; инфраструктурой железнодорожного транспорта; персоналом и социальной сферой.

Начался новый этап в развитии информационных технологий, предоставивший колоссальные возможности для совершенствования управления производственной деятельностью. Достигнутый уровень информатизации отрасли позволил создать систему фирменного транспортного обслуживания. Начал работать и успешно функционирует Центр фирменного транспортного обслуживания. К 1998 г. была реализована современная программно-техническая среда, соответствующая мировому уровню. Произошли изменения в структуре управления информатизацией. ГВЦ становится головным центром по эксплуатации информационных систем, ему в оперативном отношении подчинены ИВЦ железных дорог. В функции ГВЦ вошли новые направления по эксплуатации СПД, обеспечению информационной безопасности, по информационной поддержке железных дорог государств - участников Содружества, Латвийской Республики, Литовской Республики, Эстонской Республики и др.

В 1999 г. в эксплуатацию вошел экономичный сервер IBM 9672 R36 ряда G5, обеспечивающий 12-кратную масштабируемость и позволяющий наращивать мощности. Устанавливается робот-хранилище Storage Tek объемом 13 Тбайт и реализуется новая стратегия хранения данных, принятая в мировой практике построения систем внешней памяти. Освоены операционные системы OS/390, СУБД ADABAS, ORACLE, DB2, новые инструментальные средства SAS Institute. Появилась возможность строить и развивать автоматизированные системы реального времени по управлению перевозочным процессом, обработке финансовых документов и др.

В 2000 г. программно-технический комплекс снова модернизируется. Осуществляется перевод производства на две машины IBM 9672 R36, что повышает надежность и увеличивает производительность системы. Вводятся в эксплуатацию крупнейшие автоматизированные системы: ДИСПАРК, ЕК АСУФР и др., внедрение которых в значительной степени изменило технологию работы функциональных служб дорог. Осуществляется промышленная эксплуатация около 200 информационных систем. Внедрение СПД и высокоскоростных каналов позволяет перейти к сетевой структуре обработки данных и реализовать систему управления вычислительными ресурсами отрасли. Инфраструктура предоставляет большие возможности для полномасштабной реализации web-технологий, технологий защиты и обеспечения достоверности отраслевых информационных ресурсов.

Сегодня АСУ РЖД состоит из более 600 интегрированных автоматизированных систем и клиентских приложений, она представляет собой распределенную информационную систему по направлениям производственной деятельности компании. С помощью информационных систем

осуществляется управление перевозочным процессом, сбытом и организацией грузовых и пассажирских перевозок, корпоративной инфраструктурой и подвижным составом, экономикой, бюджетированием, финансами и ресурсами, стратегическим развитием, инвестиционной и информационной деятельностью, информационной безопасностью, унификацией и интеграцией автоматизированных систем.

Программирование и настройка транспортного электронного оборудования.

Развитие сетей передачи данных на ж. д. транспорте прежде всего обусловлено разработкой автоматизированной системы оперативного управления грузовыми перевозками (АСОУП) и автоматизированной системы управления пассажирскими перевозками ("Экспресс").

Различие в требованиях к этим системам, этапности их разработки, методах проектирования и внедрения привели к необходимости создания на ж. д. транспорте двух относительно независимых сетей передачи данных (СПД): СПД в системе управления грузовыми перевозками и СПД системы "Экспресс". Одно из основных отличий в требованиях к этим сетям заключается в том, что СПД системы "Экспресс" должна обеспечивать диалоговое взаимодействие абонента (терминала кассира) с системой обработки в жестко регламентированные сроки (время реакции 3-5 секунд). СПД грузовыми перевозками главным образом обеспечивает обмен сообщениями между абонентами и системой обработки в менее жестком по времени режиме (десятки секунд и минут) и передачу сообщений и файлов при межмашинном обмене.

Вместе с тем автономия рассматриваемых сетей не является абсолютной, так как обе сети, например, при обмене информацией на межрегиональном уровне (ГВЦ ОАО-ИВЦ, ИВЦ-ИВЦ соседних дорог) могут использовать каналы первич-

ной сети связи, организованные в одних и тех же аналоговых или цифровых форматах передачи.

СПД линейных предприятий (СПД-ЛП) предназначена для автоматизированного съема, централизованного сбора, обработки, передачи и распределения по потребителям оперативной, в том числе диагностической, информации в реальном масштабе времени. По сети передаются данные о состоянии линейных технологических объектов, технических средств и систем автоматики, связи, энергетики; устройств контроля состояния подвижного состава на ходу поезда (ПОНАБ, ДИСК). Пользователями этой системы единой для всех служб сети являются работники службы движения (ДСП, ДНЦ, ДНЦО и т.д.), работники хозяйств Э, СЦБ и др.

Централизованный сбор, накопление и хранение первичной информации производится на общем сервере СПД-ЛП, включенном в локальную вычислительную сеть (ЛВС) центра сбора информации. В эту же ЛВС включаются АРМ пользователи СПД-ЛП, которые получают необходимую им информацию.

СПД дорожного уровня обеспечивают в пределах железной дороги обмен информацией между абонентами и системами обработки данных, решающими прикладные задачи управления перевозками и другими видами деятельности на ж. - д. транспорте. Основной

системой, требующей непрерывного обмена информацией и создающей интенсивные информационные потоки, которые должна передавать СПД дороги, является АСУ перевозками (АСУП). Для обеспечения функционирования АСУП необходимо связать большие системы обработки данных в ИВЦ дороги с многочисленными АРМ, поставляющими оперативные данные о движении, дислокации и изменении состояния поездов, вагонов и грузов. СПД дорожного уровня может производить обмен информацией с СПД-ЛП, а так же с локальными вычислительными сетями управления дороги, отделений дороги, ЦФТО и т.д.

АСУСС и АСУГУ также обмениваются информацией с ИВЦ дороги по выделенным телефонным каналам.

Отдельную группу абонентов СПД дорожного уровня представляют комплексные системы АРМ (САРМ) на станциях. В САРМ объединяются группы абонентов, использующих в своей работе общий сервер, функционально ориентированный на решение определенного круга задач управления перевозками.

Порядок выполнения работы:

1. Ознакомиться с данными методическими указаниями.
2. Ответить на контрольные вопросы.

Содержание отчета

1. Тема, цель.
2. Ответы на контрольные вопросы.
3. Вывод.

Контрольные вопросы:

1. Дайте определения основных понятий системы передачи данных.
2. Перечислите состав абонентского оборудования СПД.
3. Каково назначение и классификация СПД?
4. В чем преимущества и недостатки аналоговых СПД?
5. В чем преимущества и недостатки цифровых СПД?
6. Приведите особенности построения СПД ЛП.

7.Приведите структуру радиально-узловойСПД ЛП. 8.Приведите структуру многоточечной СПД ЛП.

Анализ сред программирования для МК

Программирование микроконтроллеров является неотъемлемой частью разработки самостоятельного электронного устройства. На данном этапе развития электроники, наиболее популярными микроконтроллерами (МК) есть: PIC, MSP, AVR, STM, ARM (процессор).

Для каждого вида МК есть узконаправленная среда программирования. Это связано с внутренней структурой МК и технического обеспечения записи программы в память МК. Если проанализировать существующие **среды программирования МК**, то найти универсальные среды программирования МК очень сложно. При анализе сред программирования было выявлено только одну универсальную среду программирования микроконтроллеров – FlowCode, функции которой позволяют программировать МК PIC, AVR, ARM.

FlowCode – графическая универсальная среда программирования МК. Программирование осуществляется благодаря построению логической структуры, т.е. блок-схем, аналогично среде HiAsm. Функция экспорта позволяет экспортировать написаний код PIC МК в программу AVR МК и наоборот. Дополнением данной среды программирование является создание HEX-кода, который может быть использовать при прошивки МК, или при проектировании схемы с поддержкой МК, например, в среде Proteus.

Большинство проанализированных **сред программирования МК** были предназначены для AVR. Среда AlgorithmBuilder – это графическая среда программирования МК AVR. Тип графической разработки в данной среде, отличается от FlowCode. Как говорит автор программы, написание кода осуществляется в 3-5 раз быстрее. Среда русифицирована и поддерживает автоматическое перекодирование строк в ANSI-кода Windows в код русифицированного буквенно-цифрового ЖКИ. Среда программирования объединяет в себе графический редактор, компилятор алгоритма, симулятор микроконтроллера, внутрисхемный программатор, функции работы с EEPROM.

Среди платных **сред программирования AVR МК** есть AVR Studio, IAR Systems, ImageCraft, WinAVR, CodeVisionAVR. CodeVisionAVR – популярная условно-бесплатная среда программирования AVR МК. Объединяет в себе Си-подобный язык программирования и ассемблер. Функции программы позволяют самостоятельно прошивать МК и устанавливать fuse-биты и ПЗУ. Конечным результатом разработки программы под МК есть создание HEX, BIN или ROM-файла, для прошивки МК с помощью программатора.

Среда IAR Systems поддерживает программирования МК AVR и MSP430, но функции программирования двух МК не объединены в одной среде. Для каждого МК были разработаны отдельные среды программирования. Аналогичным образом была разработана **среда программирования ImageCraft (ICC)**. ICC поддерживает Си-подобный синтаксис и ассемблер. IAR Systems и ImageCraft в их состав входит целенаправленные библиотеки по работе с отдельными частями МК. В состав ICC добавлена утилита для генерации кода и инициализации периферии МК, внедрен ANSI TerminalEmulator, который предоставляет возможность работать с COM-портом.

Среда CodeComposerStudio (CCS) для программирования МК MSP основана на базе стоковой универсальной среды программирования Eclipse. CCS как вышеперечисленные среды программирования, имеет Си-подобный язык программирования. Данная среда программирования поддерживает разработку для DSP (Digitalsignalprocessor) микроконтроллеров, процессоров APM семейства TMS320 и MSP430. Данная среда в сравнении с ICC или CCS использует Java-машину и требует больше системных ресурсов.

Energia – **среда программирования для МК MSP430**, которая наиболее популярной средой программирования среди начинающих. Имеет Си-подобный язык программирования, но он отличается от языков, которые используются в вышеперечисленных средах. Язык Energia (и Arduino IDE) более понятный, подобный английским словам. Energia поддерживает дополнительные библиотеки, в состав которых входят драйвера для подключение платформы LaunchPad MSP430 на базе ARM Cortex, FraunchPad и lm4f120 StallerPad. Данная среда программирования является модифицированной версией среды Arduino IDE. Работает с 1, 16МГц МК MSP430 и 80 МГц lm4f120. Внедрена функция просмотра COM-порта.

Virtualbreadboard – среда разработки с поддержкой проектирования электрической схемы на базе платформы Arduino под управлением МК ATMELE AVR. При анализе использовалась программа версии 4.2.9, которая поддерживает разработку программ под МК PIC. Синтаксис подобный среде программирования Energia и Arduino IDE. Данная среда является платной, в программу внедрены примеры, и модели электронных элементов на основе которых проектируется устройство. Для разработки программ под МК PIC существует среда MPLAB IDE. Создание программы проходит в структурированном виде, встроенный программный симулятор моделирования исполнения программ в МК с учетом состояния портов ввода-вывода, **эмулятор работы МК** в масштабе реального времени непосредственно в устройстве пользователя. Разработка проекта ведется на языке C или ассемблер. Еще одной средой программирования для МК PIC является mikroC. Среда mikroC включает в себя библиотеки, которые поддерживают следующие устройства и интерфейсы:

- встроенный аналого-цифровой преобразователь (АЦП) микроконтроллера;
- встроенную энергонезависимую память EEPROM микроконтроллера;
- встроенные широтно-импульсные модуляторы (PWM) микроконтроллера;
- внешние сменные карты памяти типа MMC, SD и CompactFlash;
- файловую систему FAT;
- алфавитно-цифровой жидкокристаллические индикаторы (LCD PKI);
- графические жидкокристаллические индикаторы (GLCD, PK-дисплей);
- интерфейсы I2C, SPI, 1-Wire, USART, RS-485, CAN, PS / 2, USB (HID) и Ethernet.

Подводя итоги, хочу сказать, что при выборе среды программирования необходимо учитывать ее основной функциональный состав, язык программирования, поддерживаемые порты и интерфейсы.

Средства программирования PIC-контроллеров

Компиляторы*Отладка*Программирование микроконтроллеров*

Введение

PIC-контроллеры остаются популярными в тех случаях, когда требуется создать недорогую компактную систему с низким энергопотреблением, не предъявляющую высоких требований по ее управлению. Эти контроллеры позволяют заменить аппаратную логику гибкими программными средствами, которые взаимодействуют с внешними устройствами через хорошие порты.

Миниатюрные PIC контроллеры хороши для построения преобразователей интерфейсов последовательной передачи данных, для реализации функций «прием – обработка – передача данных» и несложных регуляторов систем автоматического управления.

Компания Microchip распространяет MPLAB — бесплатную интегрированную среду редактирования и отладки программ, которая записывает бинарные файлы в микроконтроллеры PIC через программаторы.

Взаимодействие MPLAB и Matlab/Simulink позволяет разрабатывать программы для PIC-контроллеров в среде Simulink — графического моделирования и анализа динамических систем. В этой работе рассматриваются средства программирования PIC контроллеров: MPLAB, Matlab/Simulink и программатор PIC-KIT3 в следующих разделах.

- Характеристики миниатюрного PIC контроллера PIC12F629
- Интегрированная среда разработки MPLAB IDE
- Подключение Matlab/Simulink к MPLAB
- Подключение программатора PIC-KIT3

Характеристики миниатюрного PIC-контроллера

Семейство PIC12xxx содержит контроллеры в миниатюрном 8–выводном корпусе со встроенным тактовым генератором. Контроллеры имеют RISC–архитектуру и обеспечивают выполнение большинства команд процессора за один машинный цикл.

Для примера, ниже даны характеристики недорогого компактного 8-разрядного контроллера PIC12F629 с многофункциональными портами, малым потреблением и широким диапазоном питания [1].

- Архитектура: RISC
- Напряжение питания VDD: от 2,0В до 5,5В (< 6,5В)
- Потребление:
 - <1,0 мА @ 5,5В, 4МГц
 - 20 мкА (тип) @ 32 кГц, 2,0В
 - <1,0 мкА (тип) в режиме SLEEP@2,0В
- Рассеиваемая мощность: 0,8Вт
- Многофункциональные каналы ввода/вывода: 6/5
- Максимальный выходной ток портов GPIO: 125мА
- Ток через программируемые внутренние подтягивающие резисторы портов: ≥ 50 (250) ≤ 400 мкА @ 5,0В
- Разрядность контроллера: 8
- Тактовая частота от внешнего генератора: 20 МГц
- Длительность машинного цикла: 200 нс
- Тактовая частота от внутреннего RC генератора: 4 МГц $\pm 1\%$
- Длительность машинного цикла: 1 мкс
- FLASH память программ: 1К
- Число циклов стирание/запись: ≥ 1000
- ОЗУ память данных: 64
- EEPROM память данных: 128
- Число циклов стирание/запись: $\geq 10K$ ($-40^{\circ}C \leq T_A \leq +125^{\circ}C$)
- Аппаратные регистры специального назначения: 16
- Список команд: 35 инструкций, все команды выполняются за один машинный цикл,
- кроме команд перехода, выполняемых за 2 цикла
- Аппаратный стек: 8 уровней
- Таймер/счетчик TMR0: 8-разрядный с предделителем
- Таймер/счетчик TMR1: 16-разрядный с предделителем

Дополнительные особенности:

- Сброс по включению питания (POR)
- Таймер сброса (PWRTтаймер ожидания запуска генератора (OST
- Сброс по снижению напряжения питания (BOD)
- Сторожевой таймер WDT
- Мультиплексируемый вывод -MCLR
- Система прерываний по изменению уровня сигнала на входах
- Индивидуально программируемые для каждого входа подтягивающие резисторы
- Программируемая защита входа
- Режим пониженного энергопотребления SLEEP
- Выбор режима работы тактового генератора
- Внутрисхемное программирование ICSP с использованием двух выводов
- Четыре пользовательские ID ячейки

Предельная рабочая температура для E исполнения (расширенный диапазон) от -40оС до +125 оС;

Температура хранения от -65оС до +150 оС.

КМОП технология контроллера обеспечивает полностью статический режим работы, при котором остановка тактового генератора не приводит к потере логических состояний внутренних узлов.

Микроконтроллер PIC12F629 имеет 6-разрядный порт ввода/вывода GPIO. Один вывод GP3 порта GPIO работает только на вход, остальные выводы можно сконфигурировать для работы как на вход так и на выход. Каждый вывод GPIO имеет индивидуальный бит разрешения прерываний по изменению уровня сигнала на входах и бит включения внутреннего подтягивающего резистора.

Интегрированная среда разработки MPLAB IDE

MPLAB IDE — бесплатная интегрированная среда разработки ПО для микроконтроллеров PIC включает средства для создания, редактирования, отладки, трансляции и компоновки программ, записи машинного кода в микроконтроллеры через программаторы.

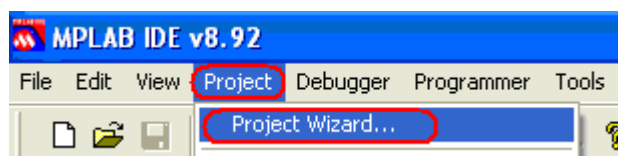
Загрузка MPLAB IDE

Бесплатные версии MPLAB (включая MPLAB 8.92) хранятся [на сайте](#) компании Microchip в разделе «DOWNLOAD ARCHIVE».

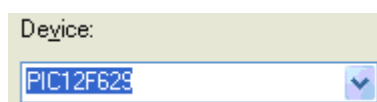
Создание проекта

Пример создания проекта программ PIC контроллера в среде MPLAB включает следующие шаги [2].

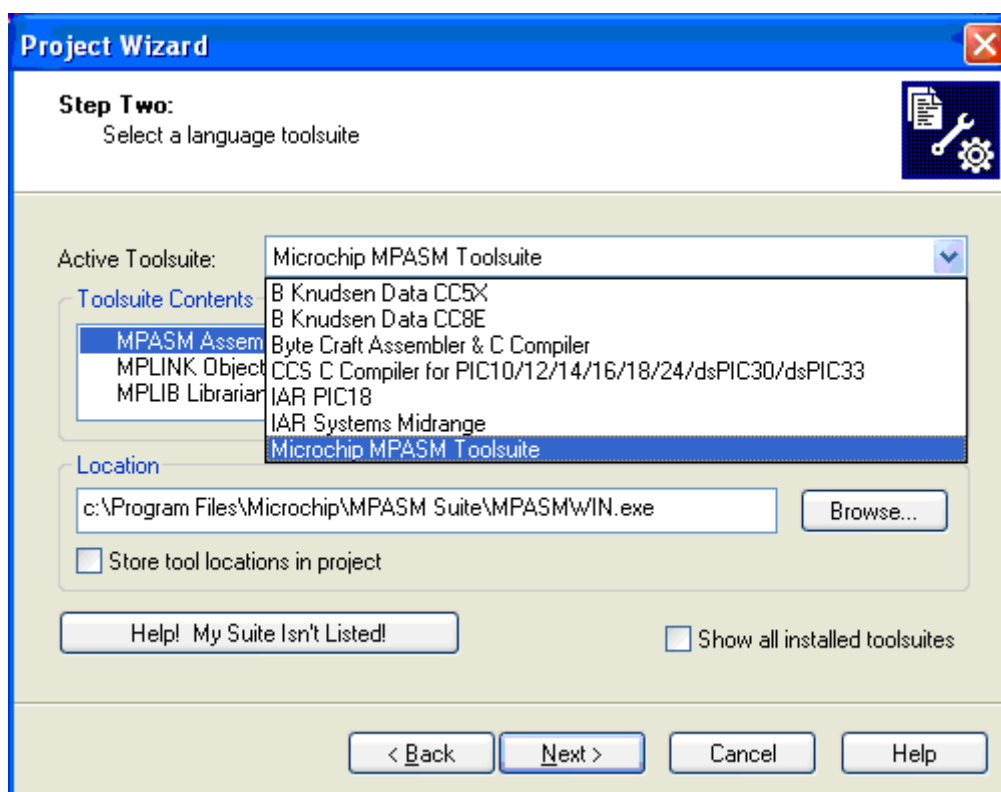
1. Вызов менеджера проекта.



2. Выбор типа PIC микроконтроллера.



3. Выбор компилятора, например, Microchip MPASM для ассемблера.



4. Выбор пути к каталогу проекта (клавиша Browse...) и ввод имени проекта.

5. Подключение файлов к проекту в окне ProjectWizard → StepFour можно не выполнять. Это можно сделать позднее, внутри активного проекта. Клавиша Next открывает следующее окно.

6. Завершение создания проекта (клавиша Finish).

В результате создания проекта FirstPrMPLAB интерфейс MPLAB принимает вид, показанный на Рис. 1.

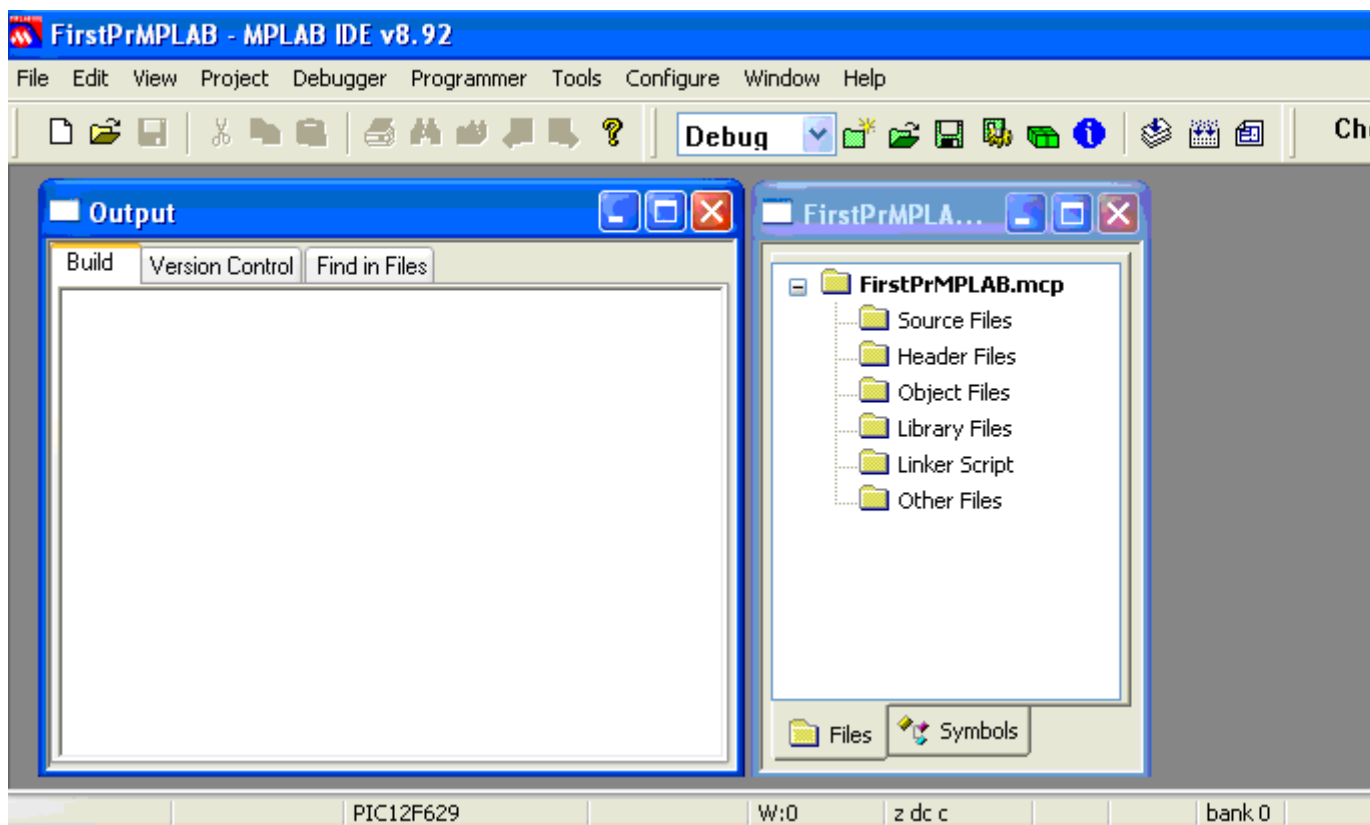


Рис. 1. Интерфейс среды MPLAB v8.92 и шаблон проекта.

Создание файла программы

Программу можно создать при помощи любого текстового редактора. В MPLAB имеется встроенный редактор, который обеспечивает ряд преимуществ, например, оперативный лексический анализ исходного текста, в результате которого в тексте цветом выделяются зарезервированные слова, константы, комментарии, имена, определенные пользователем.

Создание программы в MPLAB можно выполнить в следующей последовательности.

1. Открыть редактор программ: меню → File → New. Изначально программе присвоено имя Untitled.
2. Набрать или скопировать программу, например, на ассемблере.
3. Сохранить программу под другим именем (меню → File → SaveAs), например, FirstPrMPLAB.asm.

```

1
2          LIST          p=12F629      ; тип микроконтроллера
3          __CONFIG      03FF4H       ; бит защиты выключен, WDT выключен, внутренний RC - re
4  status  equ          03h           ; константа status равна адресу регистра STATUS
5  portb   equ          05h           ; адрес регистра GPIO в Банке 0
6  trisb   equ          05h           ; адрес регистра TRISIO в Банке 1
7  cmcon   equ          19h
8  ;optn   equ          01h           ; адрес регистра OPTION_REG в Банке 1
9          org          0             ; начало программы
10 start   bsf          status,5       ; установка 5-го бита регистра STATUS - выбор Банка 1
11 ;       bcf          optn,7         ; Разрешить использование всех подтягивающих резисторов
12         movlw        .0            ; запись константы в регистр W
13         movwf        trisb         ; копирование регистра W в рег. TRISIO, настройка портов
14         bcf          status,5       ; сброс 5-го бита регистра STATUS - выбор Банка 0
15
16         movlw        0x07          ;
17         movwf        cmcon         ; настройка GP0 и GP1 на выход
18
19 Metka    movlw        b'10101010'
20         movwf        portb         ; Копирование регистра W в порты GPIO
21         movlw        b'01010101'
22         movwf        portb         ; Копирование W в порты GPIO
23         goto         Metka         ; Переход на метку
24     end
25

```

Рис. 2. Пример простейшей программы (на ассемблере) вывода сигналов через порты контроллера GP0, GP1, GP2, GP4, GP5 на максимальной частоте.

Запись '1' в разряде регистра TRISIO переводит соответствующий выходной буфер в 3-е состояние, в этом случае порт GP может работать только на вход. Установка нуля в TRISIO настраивает работу порта GP на выход.

Примечание. По спецификации PIC12F629 порт GP3 микроконтроллера работает только на вход (соответствующий бит регистра TRISIO не сбрасывается – всегда находится в '1').

Регистры TRISIO и GPIO находятся в разных банках области памяти. Переключение банков выполняется 5-м битом регистра STATUS.

Любая программа на ассемблере начинается директивой org и заканчивается директивой end. Переход gotoMetka обеспечивает циклическое выполнение программы.

В программе (Рис. 2) используются следующие обозначения.

- Директива LIST — назначение типа контроллера
- Директива __CONFIG — установка значений битов конфигурации контроллера
- Директива equ — присвоение числового значения
- Директива org 0 — начало выполнения программы с адреса 0

Команда `bsf` — устанавливает бит указанного регистра в 1
 Команда `bcf` — сбрасывает бит указанного регистра в 0
 Команда `movlw` — записывает константу в регистр W
 Команда `movwf` — копирует содержимое регистра W в указанный регистр
 Команда `goto` — обеспечивает переход без условия на строку с меткой
 Директива `end` — конец программы

Установка требуемой конфигурации микроконтроллера

Конфигурация микроконтроллера PIC12F629 зависит от настроек слова конфигурации (2007h), которые можно задать в программе через директиву `__CONFIG`.

Непосредственно или через окно MPLAB: меню → Configure → Configuration Bits:

Configuration Bits				
☐ Configuration Bits set in code.				
Address	Value	Field	Category	
2007	3FF4	FOSC	Oscillator Selection bits	INTOSC oscillator: I
		WDTE	Watchdog Timer Enable bit	WDT disabled
		PWRT	Power-Up Timer Enable bit	PWRT disabled
		MCLRE	GP3/MCLR pin function select	GP3/MCLR pin functio
		BOREN	Brown-out Detect Enable bit	BOD enabled
		CP	Code Protection bit	Program Memory code
		CPD	Data Code Protection bit	Data memory code pro

Где:

Бит 2-0 — FOSC2:FOSC0. Выбор тактового генератора

111 — Внешний RC генератор. Подключается к выводу GP5. GP4 работает как CLKOUT

110 — Внешний RC генератор. Подключается к выводу GP5. GP4 работает как ввод/вывод

101 — Внутренний RC генератор 4МГц. GP5 работает как ввод/вывод. GP4 — как CLKOUT

100 — Внутренний RC генератор 4МГц. GP5 и GP4 работают как ввод/вывод

011 — ЕС генератор. GP4 работает как ввод/вывод. GP5 — как CLKIN

010 — НС генератор. Резонатор подключается к GP4 и GP5

001 — ХТ генератор. Резонатор подключается к GP4 и GP5

000 — LP генератор. Резонатор подключается к GP4 и GP5

Бит 3 — WDTE: настройка сторожевого таймера (WatchdogTimer)

1 — WDTE включен

0 — WDTE выключен

Сторожевой таймер предохраняет микроконтроллер от зависания – перезапускает

программу через определенный интервал времени если таймер не был сброшен. Период таймера устанавливается в регистре OPTION_REG. Обнуление сторожевого таймера вызывается командой CLRWDT.

Бит 4 — PWRT: Разрешение работы таймера включения питания:

1 — PWRT выключен

0 — PWRT включен

Таймер задерживает микроконтроллер в состоянии сброса при подаче питания VDD.

Бит 5 — MCLR: Выбор режима работы вывода GP3/-MCLR

1 — работает как -MCLR

0 — работает как порт ввода-вывода GP3

Бит 6 — BODEN: Разрешение сброса по снижению напряжения питания (как правило < 2.0В)

1 — разрешен сброс BOR

0 — запрещен сброс BOR автоматически включается таймер

При разрешении сброса BOR автоматически включается таймер PWRT

Бит 7 — .CP: Бит защиты памяти программ от чтения программатором

1 Защита выключена

0 Защита включена

При выключения защиты вся память программ стирается

Бит 8 — .CPD: Бит защиты EPROM памяти данных

1 Защита выключена

0 Защита включена

После выключения защиты вся информация будет стерта

Бит 11-9 — Не используются: Читается как '1'.

Бит 13-12 — BG1:BG0. Биты калибровки сброса по снижению питания

00 — нижний предел калибровки

11 — верхний предел калибровки

Добавление программы к проекту

Пример добавления программы к проекту показан на (Рис. 3).

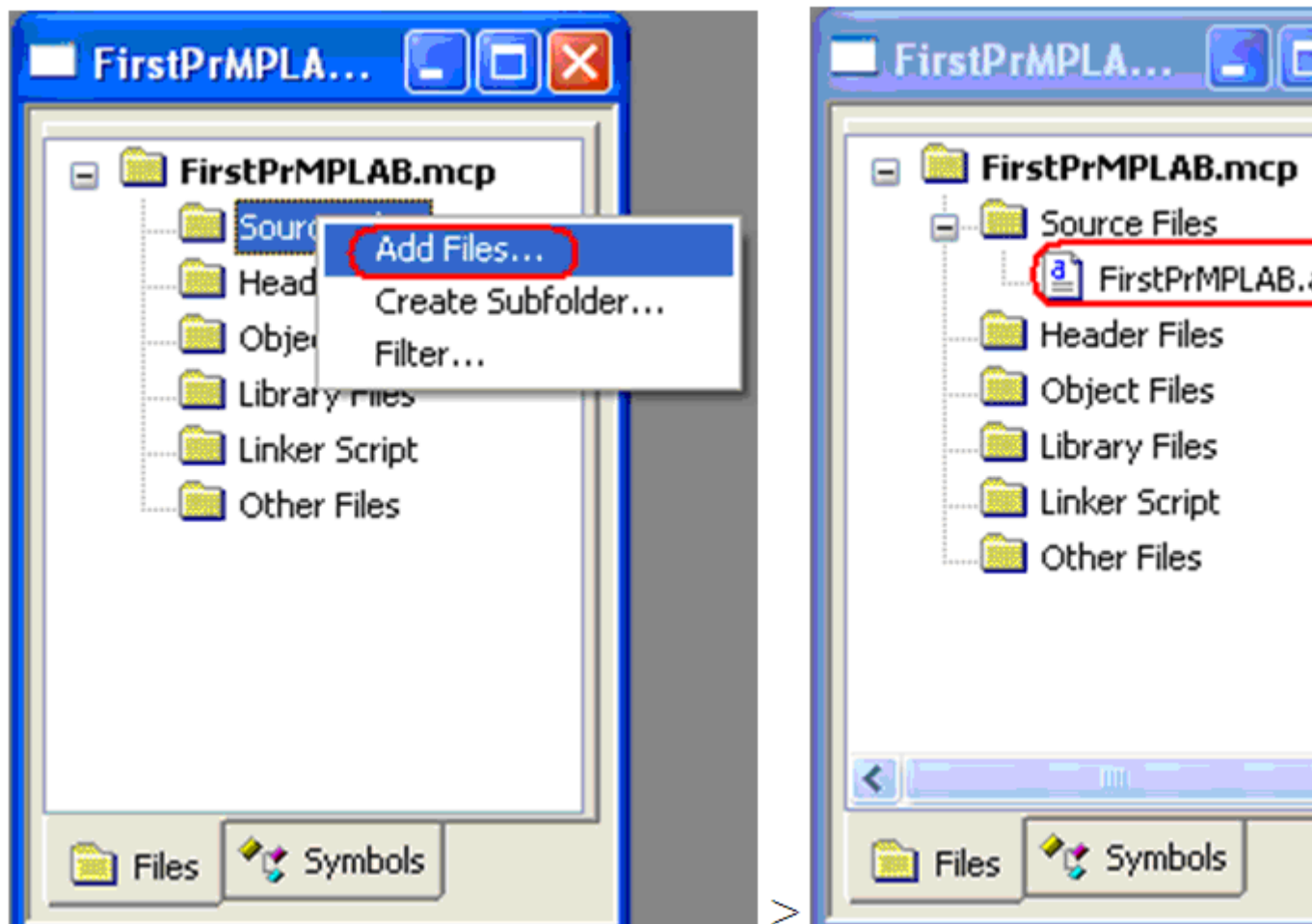


Рис. 3. Добавление программы FirstPrMPLAB.asm к проекту FirstPrMPLAB.mcp

Сохранить материалы проекта можно командой: меню → File → SaveWorkspace.

Компиляция

Чтобы создать бинарный файл с расширением hex для прошивки микроконтроллера необходимо откомпилировать проект. Запуск компиляции выполняется командой меню → Project → BuildAll. Результаты компиляции можно увидеть в окне Output (Рис. 1). Если в программе нет ошибок, то компилятор выдаёт сообщение об успешной компиляции: BUILD SUCCEEDED, загрузочный HEX файл можно найти в рабочем каталоге:

FirstPrMPLAB	cof	781
FirstPrMPLAB	map	1,815
FirstPrMPLAB	HEX	109
FirstPrMPLAB	o	741
FirstPrMPLAB	err	0
FirstPrMPLAB	lst	3,245
FirstPrMPLAB	mcp	1,005
FirstPrMPLAB	asm	932
FirstPrMPLAB	mcw	32,768

Отладка программы

Отладку программы в среде MPLAB IDE можно выполнить при помощи аппаратного эмулятора MPLAB REAL ICE или программного симулятора MPLAB SIM. Запуск последнего выполняется как показано на Рис. 4.

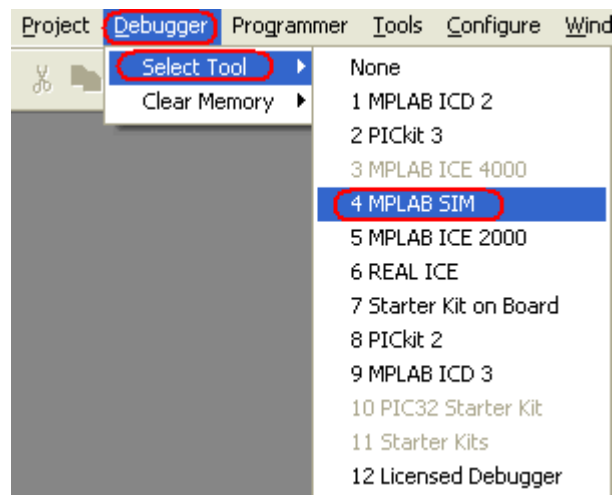


Рис. 4. Подключение к симулятору MPLAB SIM для отладки программы.

После запуска отладчика в окне Output (Рис. 1) появляется закладка MPLAB SIM, куда MPLAB выводит текущую информацию отладчика. Команды отладчика (Рис. 5) после запуска становятся активными.

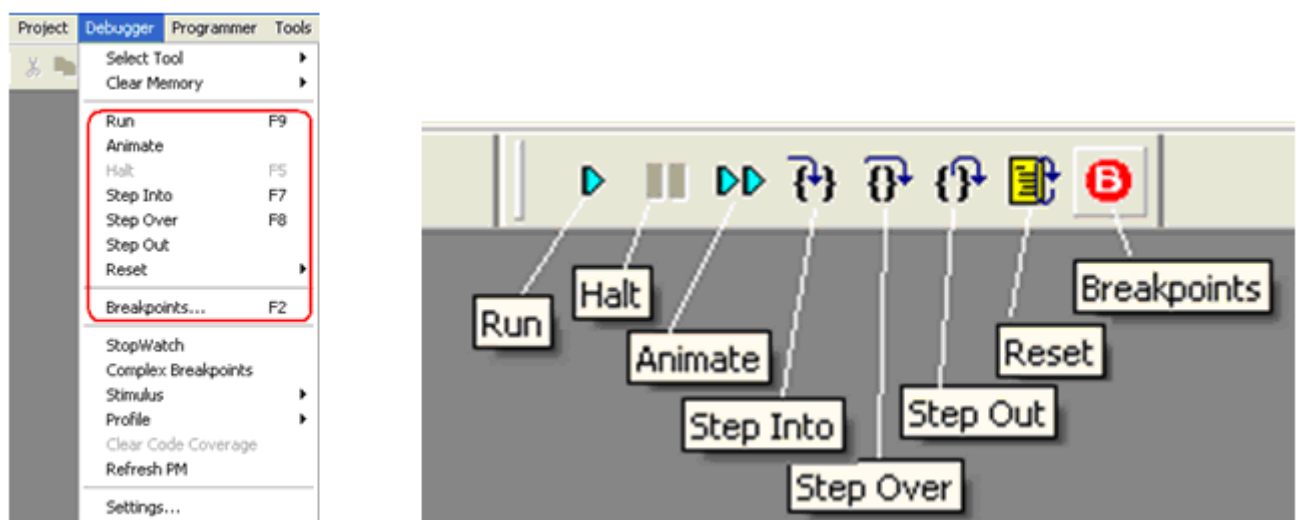


Рис. 5. Команды отладчика.

Команды отладчика:

- Run — Непрерывное выполнение программы до точки останова (Breakpoint) если таковая установлена.

- Halt — Остановка программы на текущем шаге выполнения.
- Animate — Анимация непрерывного выполнения программы.
- StepInto — Выполнение по шагам (вызовы Call выполняются за один шаг).
- StepOver — Выполнение по шагам включая команды вызовов Call.
- Reset — Начальная установка программы. Переход указателя на первую команду.
- Breakpoints — Отображение списка точек останова. Обработка списка.

При выполнении программы по шагам текущий шаг выделяется стрелкой (Рис. 6). Непрерывное выполнение программы останавливается командой Halt или достижением программой точки останова. Точка останова устанавливается/снимается в строке программы двойным щелчком.

Пример программы на ассемблере, которая с максимальной скоростью меняет состояние портов контроллера показан на Рис. 6 (справа). Программа передаёт в регистр портов GPIO данные b'10101010' и b'01010101'. Поскольку в регистре GPIO передачу данных в порты контроллера выполняют не все разряды, а только 0,1,2,4 и 5, то состояние регистра GPIO (Рис. 6, слева) отличается значениями: b'00100010' и b'00010101'.

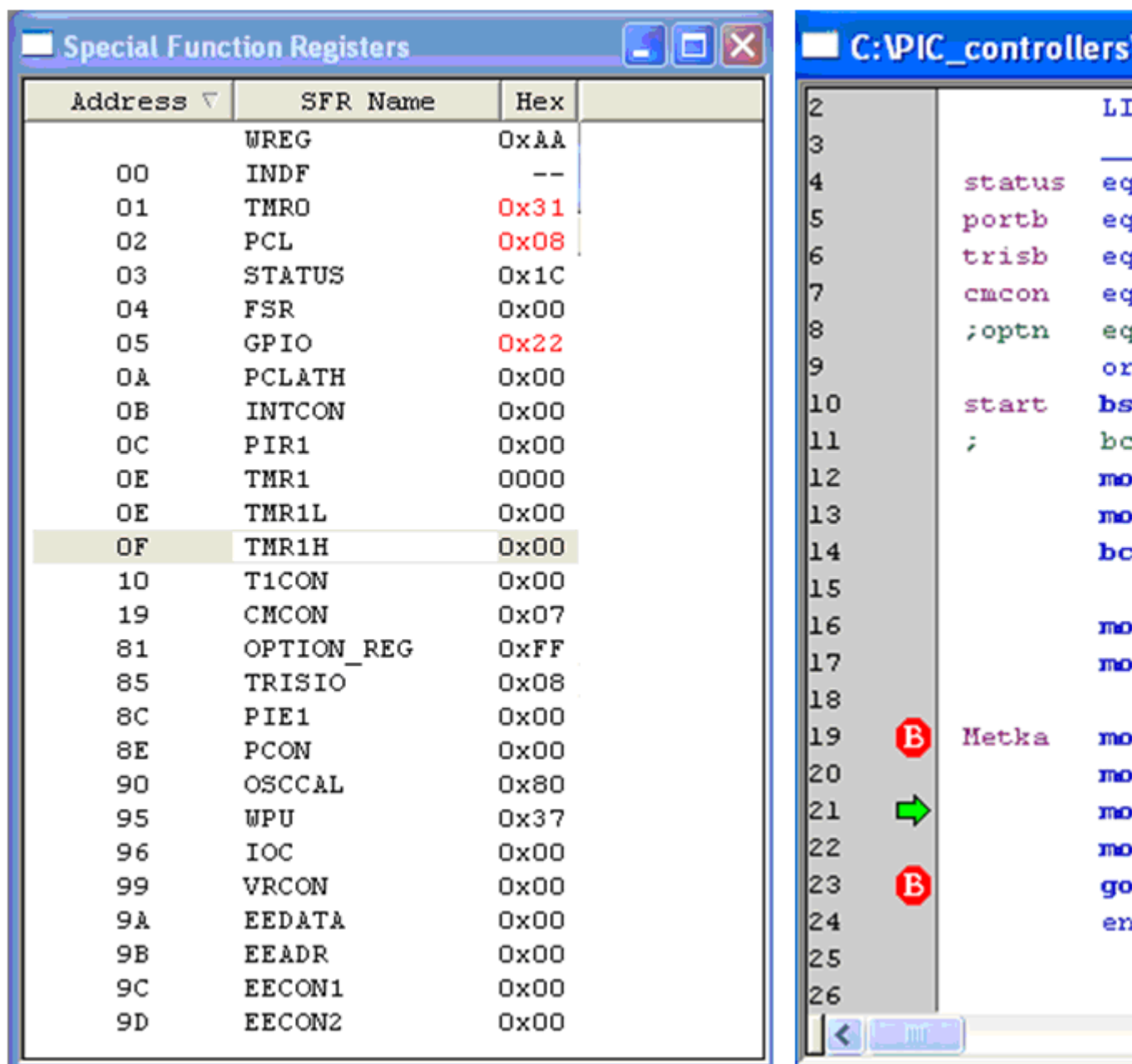


Рис. 6. Состояние регистров специального назначения контроллера на момент выполнения программы (слева) и выполняемая по шагам программа (справа).

В процессе отладки можно наблюдать за состоянием регистров, переменных, памяти в соответствующих окнах, открываемых в разделе View основного меню. В процессе отладки можно вносить изменения в код программы, содержимое регистров, памяти, изменять значения переменных. После изменения кода необходимо перекомпилировать программу. Изменение содержимого регистров, памяти и значения переменных (окна раздела View: SpecialFunctionRegister, FileRegister, EEPROM, Watch) не требует перекомпиляции.

Входные сигналы портов модели микроконтроллера можно задать в разделе Debugger → Stimulus. Устанавливаемые состояния сигналов портов привязываются к времени (тактам) отладки.

Иногда результаты выполнения программы в режиме отладки не соответствуют выполнению этой же программы в реальном контроллере, так, например, отладчик программы (Рис. 6) без инструкций `movlw 0x07` и `movwf cmcon` показывает, что выходы GP0 и GP1 регистра GPIO не изменяются — находятся в нулевом состоянии, содержимое регистра GPIO попеременно равно 0x14 и 0x20. Однако, контроллер, выполняющий программу без указанных инструкций, показывает на осциллографе циклическую работу всех пяти выходов: 0x15 и 0x22, включая GP0 и GP1 (см. Рис. 7).

Осциллограммы контроллера, выполняющего циклы программы Рис. 6 (Метка... gotoMetka) показаны на Рис. 7.

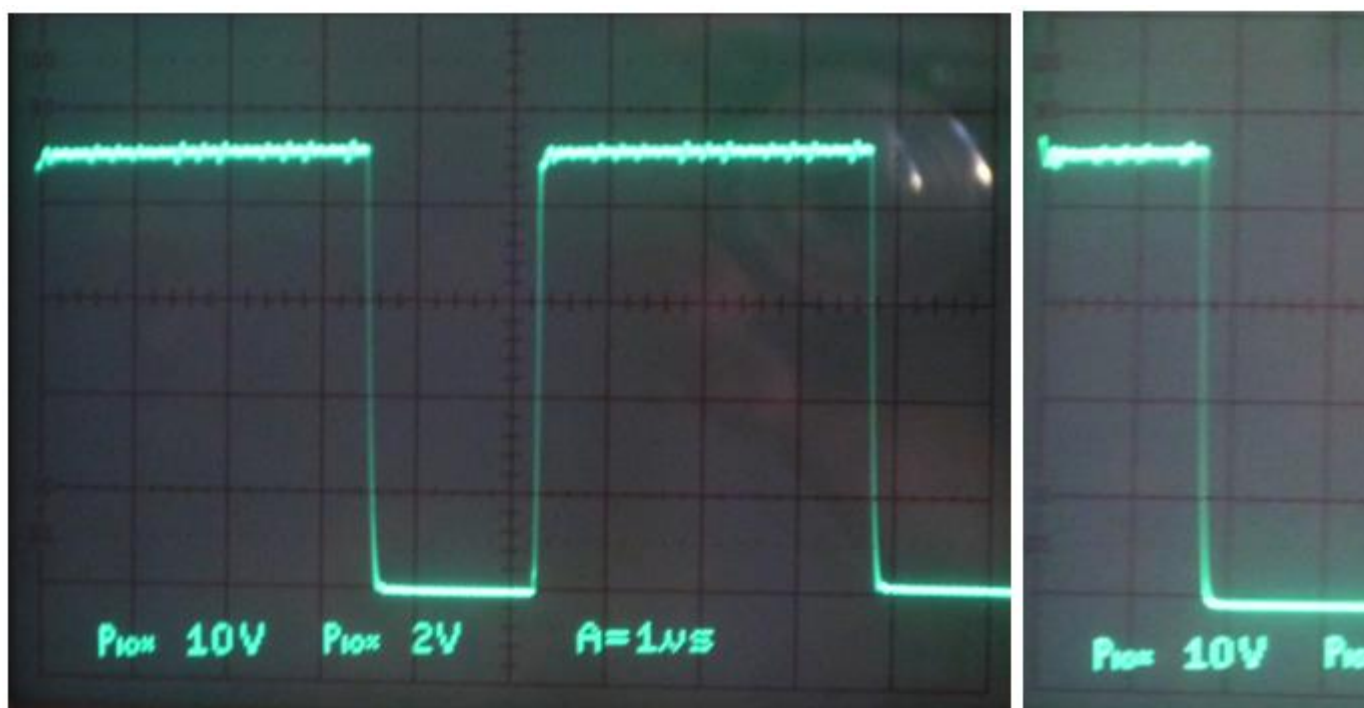


Рис. 7. Осциллограммы выхода GP0 (слева) и GP1 (справа) микроконтроллера PIC12F629, работающего от внутреннего 4МГц RC генератора. Программа (Рис. 6) формирует сигналы максимальной частоты на всех выходах контроллера. За период сигналов 5.3 мкс выполняется 5 команд (6 машинных циклов), амплитуда GP0 сигнала на осциллограмме равна 4.6В, измеренное программатором питание контроллера 4.75В.

Прошивка микроконтроллера

Для записи программы в микроконтроллер (прошивки контроллера) необходимо микроконтроллер подключить к интегрированной среде MPLAB IDE через программатор. Организация подключения показана ниже в разделе «Подключение программатора PIC-KIT3».

Примечание. В контроллер PIC12F629 записана заводская калибровочная константа настройки частоты внутреннего тактового генератора. При необходимости её можно прочитать и восстановить средствами MPLAB с использованием программатора.

Команды для работы с программатором и изменения его настроек находятся в меню MPLAB Programmer. Тип программатора в MPLAB выбирается в разделе: меню → Programmer → SelectProgrammer.

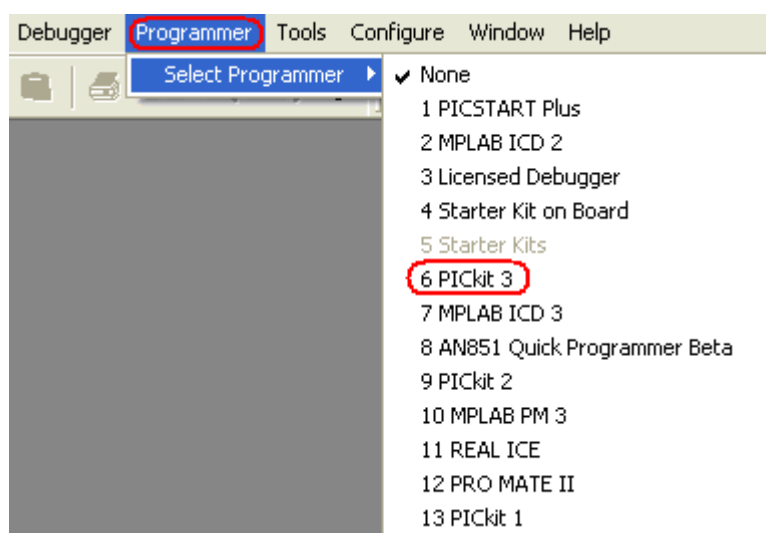


Рис. 8. Выбор программатора для подключения к среде MPLAB.

Прошивка микроконтроллера через программатор запускается командой: меню → Programmer → Program. Сообщение об успешной прошивке показано на Рис. 9.

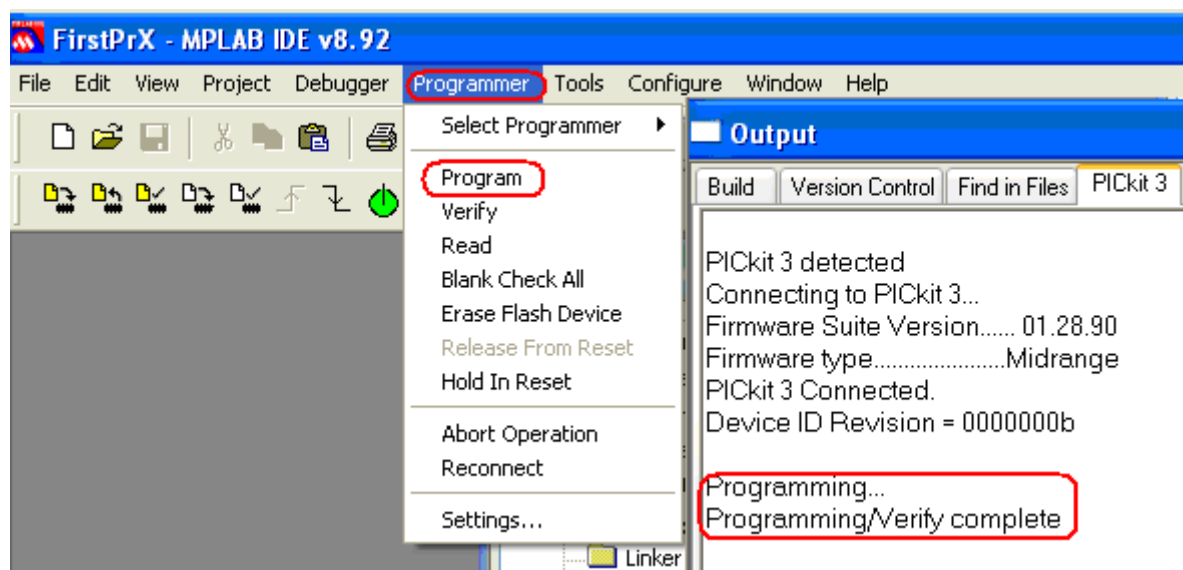


Рис. 9. Запуск прошивки микроконтроллера и вид сообщения об успешной прошивке.

Примечание: Во время прошивки микроконтроллера у программатора PIC-KIT3

мигает желтый светодиод.

Подключение MATLAB/SIMULINK к MPLAB

В системе моделирования динамических систем Simulink (приложение к Matlab) на языке графического программирования [7] можно разрабатывать программы для семейства PIC контроллеров имеющих АЦП/ЦАП, счетчики, таймеры, ШИМ, DMA, интерфейсы UART, SPI, CAN, I2C и др.

Пример Simulink программы PIC контроллера показан на Рис. 10.

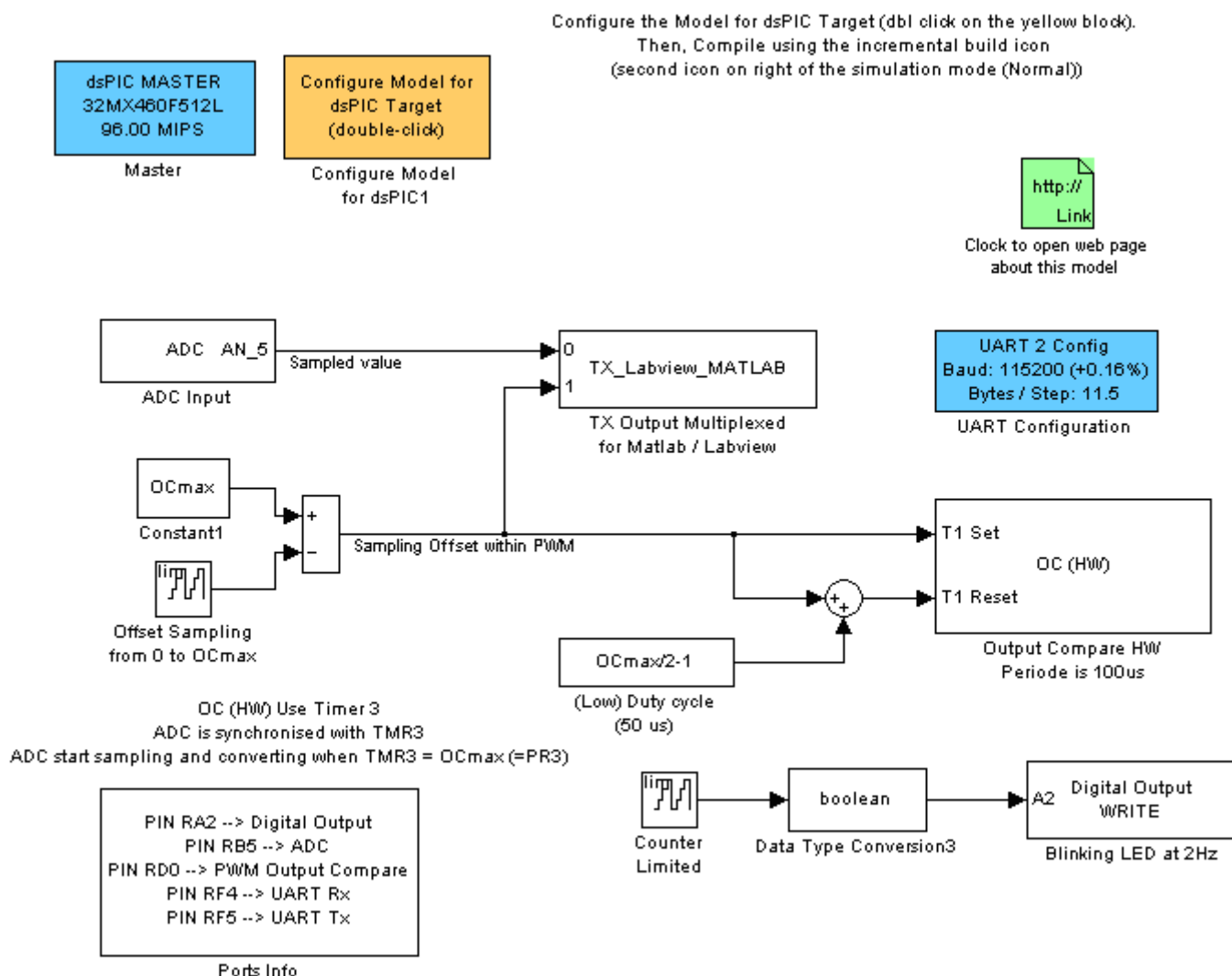


Рис. 10. Пример программы на языке графического программирования для PIC контроллера выполненной в среде моделирования динамических систем Simulink.

Взаимодействие средств разработки и компиляции программ для PIC контроллеров в Simulink показано на Рис. 11 [6].



Рис. 11. Структура средств построения адекватной модели PIC контроллера на языке графического программирования.

Для построения среды разработки необходимы следующие компоненты Matlab:

- Simulink
- Real-TimeWorkshopEmbeddedCoder
- Real-TimeWorkshop

И Си компилятор компании Microchip:

- C30 для контроллеров PIC24, dsPIC30 и PIC33
- или C32 для контроллеров серии PIC32

Установка компонентов Matlab

На [сайте](#) имеются Simulink библиотеки (dsPICToolbox) для PIC контроллеров и версий Matlab с R2006a по R2012a:

Для скачивания библиотеки необходимо зарегистрироваться. Программы поддерживают работу 100 микроконтроллеров из серий PIC 16MC, 24F, 30F, 32MC, 33F, 56GP, 64MC, 128MC, 128GP.

Бесплатные версии работают с Simulink моделями PIC контроллеров имеющих до 7 портов ввода-вывода.

Для установки dsPICToolbox — библиотеки блоков PIC контроллеров для Matlab/Simulink необходимо [4]:

- Скачать dsPICToolbox для требуемой версии Matlab.
- Распаковать zip файл в папке, в которой будут установлены Simulink блоки.
- Запустить Matlab.
- Настроить текущий каталог Matlab на папку с распакованным файлом.
- Открыть и запустить файл install_dsPIC_R2012a.m, например, кнопкой меню или клавишей клавиатуры.

Библиотеки dsPIC и примеры Simulink моделей устанавливаются в текущую папку Matlab (Рис. 12). Установленные блоки для моделирования PIC контроллеров доступны в разделе EmbeddedTargetforMicrochipdsPIC библиотеки Simulink (Рис. 13).

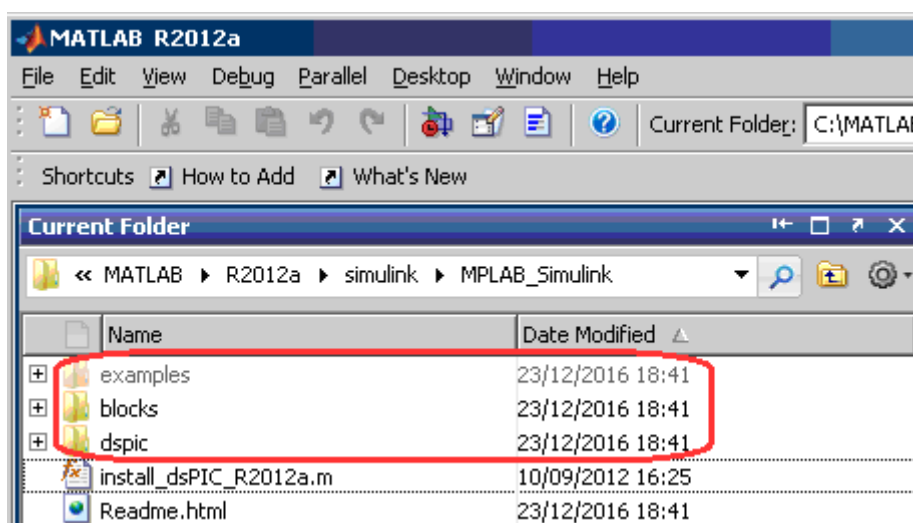


Рис. 12. Содержимое текущего каталога после выполнения install_dsPIC_R2012a.m.

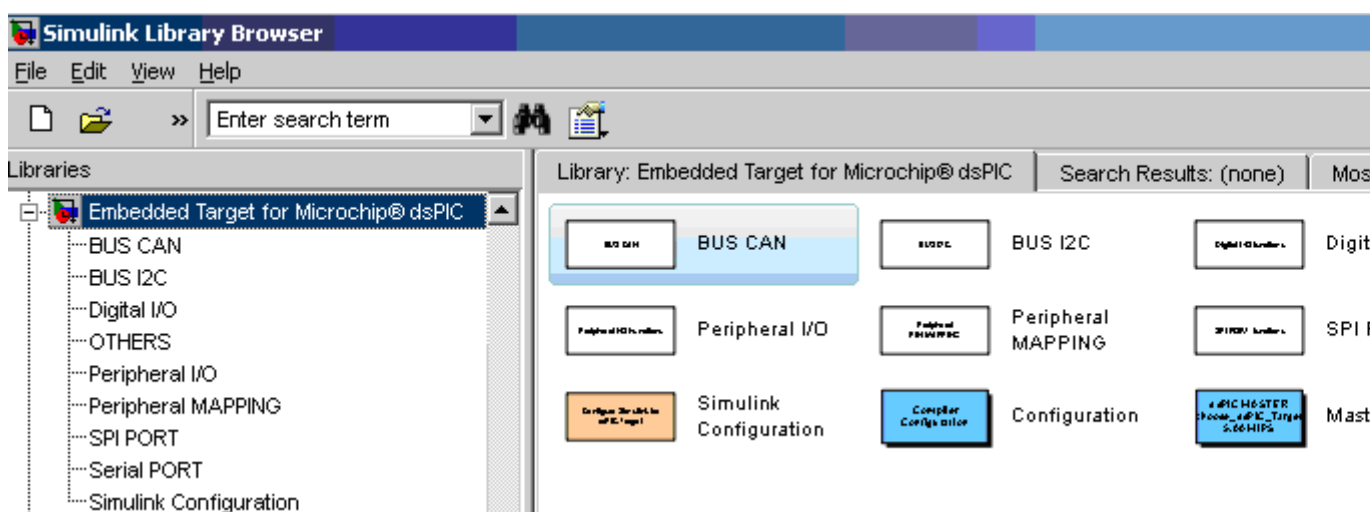


Рис. 13. Блоки, установленной библиотеки «EmbeddedTargetforMicrochipdsPIC».

Для совместной компиляции Simulink модели средствами Matlab и MPLAB необходимо прописать в переменной окружения pathMatlab с высшим

приоритетом путь к каталогу MPLAB с файлами MplabOpenModel.m, MplabGetBuildinfo.m и getHardwareConfigs.m:

```
>> path('c:\Program Files (x86)\Microchip\MPLAB IDE\Tools\MATLAB\',path)
```

Установка Си компилятора MPLAB

Компиляторы MPLAB находятся [на сайте Microchip](#) (Download Archive → MPLAB C Compiler for PIC24 and dsPIC DSCs). Для установки демонстрационной версии компилятора C30 необходимо его скачать по ссылке PIC24/dsPIC v3.25 (Рис. 14) и запустить принятый файл mplabc30-v3.25-comboUpgrade.exe.



Рис. 14. Версии Си компилятора (слева) и режимы его установки (справа).

Примечание. Работа выполнена с версией v3.25 компилятора C30 для PIC24/dsPIC. Проверка показала, что следующая версия v3.30 не поддерживает совместную компиляцию моделей Matlab R2012a (dsPICToolbox) без ошибок.

Установочный exe файл создаёт в разделе c:\Program Files (x86)\Microchip\ новый каталог mplabc30 с файлами:

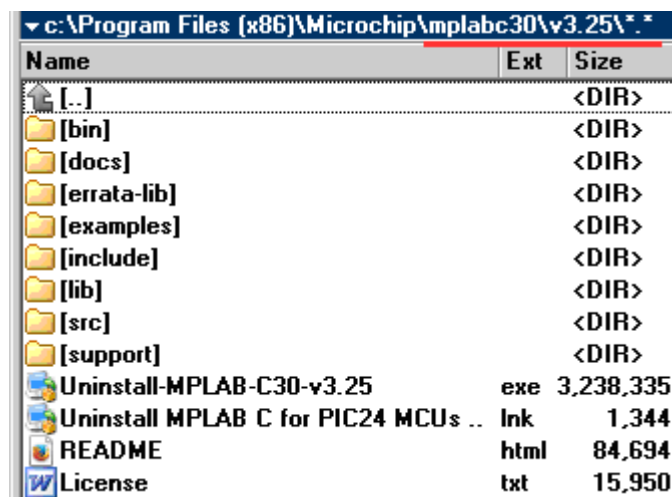


Рис. 15. Каталоги компилятора C30 MPLAB.

Последовательность Simulink программирования для PIC контроллеров

1. Создайте рабочий каталог и скопируйте в него *.mdl примеры из раздела example (см. Рис. 12).
2. Загрузите Matlab. Настройте его на рабочий каталог.
3. Включите в переменную окружения pathMatlab с высшим приоритетом путь к MPLAB — каталогу c:\Program Files (x86)\Microchip\MPLAB IDE\Tools\MATLAB\:

```
>> path('c:\Program Files (x86)\Microchip\MPLAB IDE\Tools\MATLAB\',path)
```

Примечание: Использование команды >>path без аргументов приводит к отображению списка путей переменной path в окне команд (CommandWindow). Удалить путь из переменной path можно командой rmpath, например:

```
>>rmpath(' c:\Program Files\Microchip\MPLAB IDE\Tools\MATLAB\')
```

4. Создайте Simulink модель для PIC контроллера, используя блоки библиотеки «EmbeddedTargetforMicrochipdsPIC» (Рис. 13), или загрузите готовую модель, например, Servo_ADC.mdl.

Тип контроллера, для которого разрабатывается Simulink модель, выбирается из списка в блоке Master> PIC (Рис. 16, Рис. 10), который должен быть включен в состав модели.

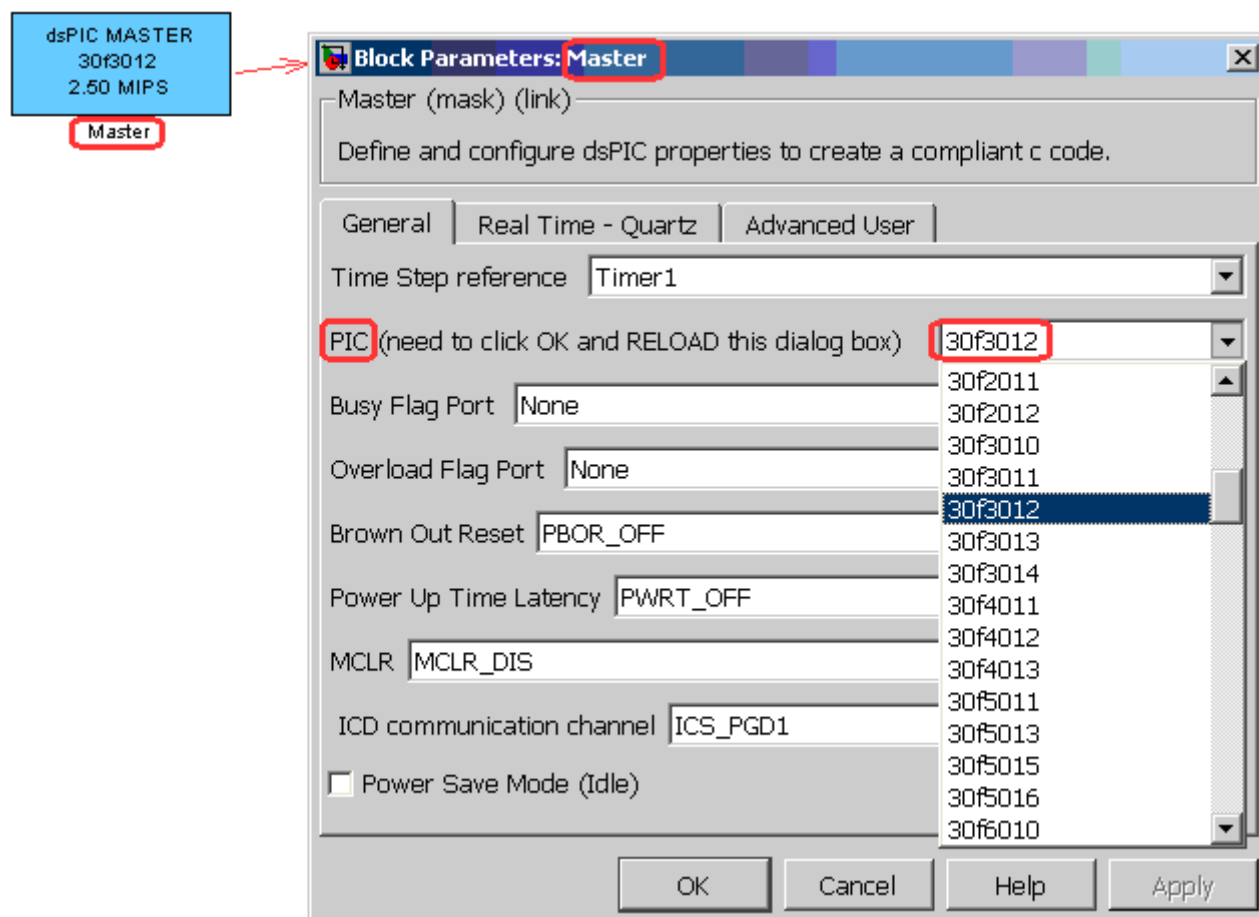


Рис. 16. Выбор типа контроллера в блоке Master модели.

5. Проверьте настройки конфигурации модели: Меню → Simulation → ConfigurationParameters<Ctrl+E>. В строке ввода Systemtargetfile раздела CodeGeneration должен быть указан компилятор S-функций dspic.tlc (Рис. 17). Выбор dspic.tlc настраивает все остальные параметры конфигурации модели, включая шаг и метод интегрирования.

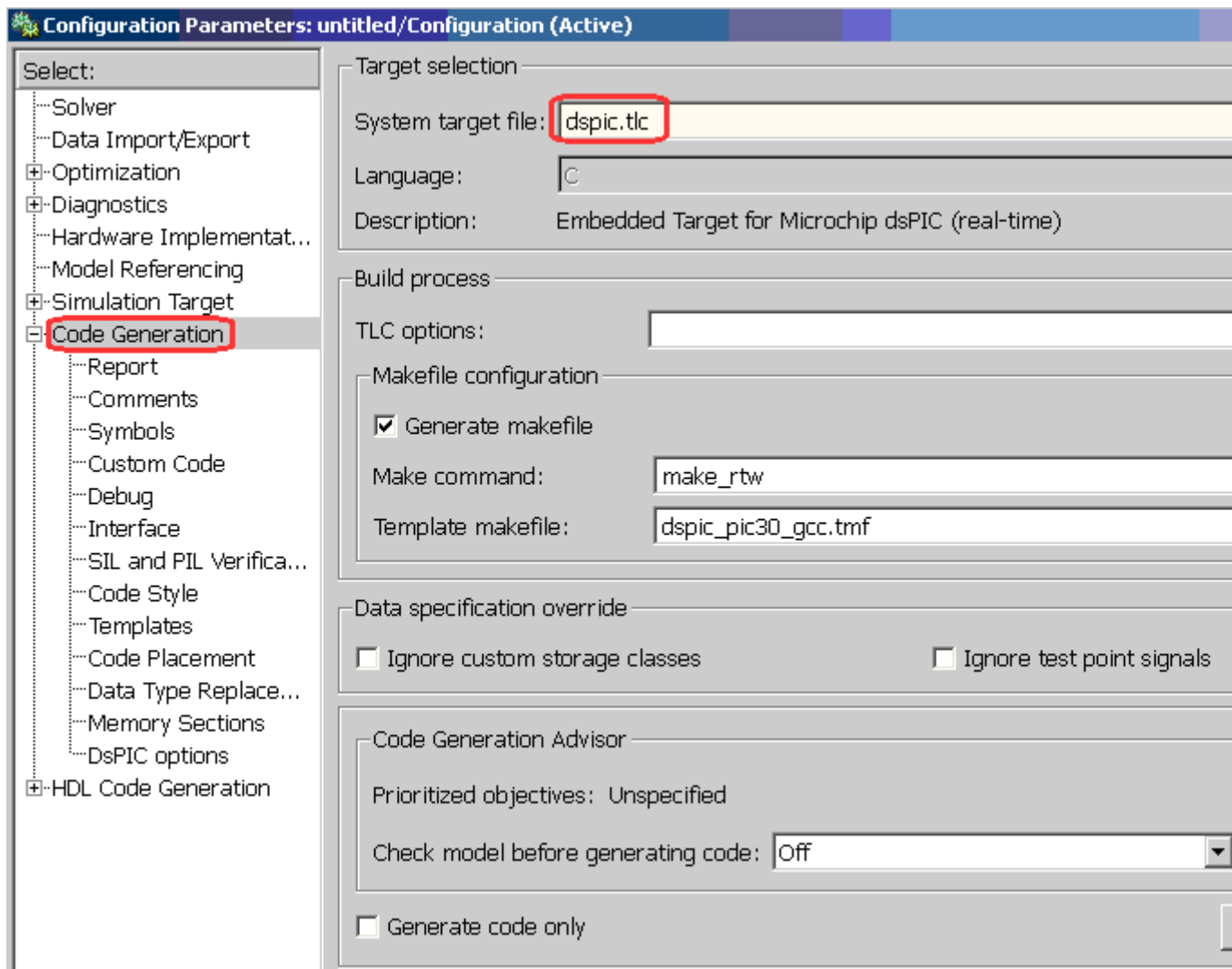


Рис. 17. Выбор компилятора S-функций *dspic.tlc* для моделей PIC-контроллеров в разделе «основное меню → Simulation → ConfigurationParameters → CodeGeneration».

6. Откомпилируйте модель *tmp_Servo_ADC.mdl*. Запуск компилятора показан на Рис. 18.

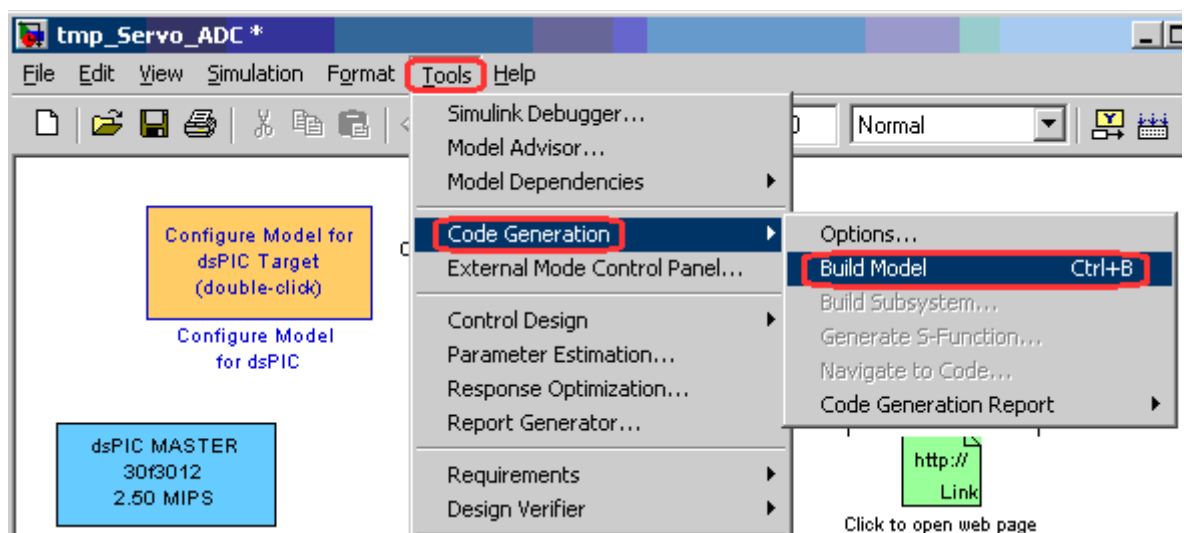


Рис. 18. Запуск компилятора Simulink модели.

В результате успешной компиляции (сообщение: ### Successfulcompletionofbuildprocedureformodel: Servo_ADC) в текущем каталоге создаются HEX файл для прошивки PIC контроллера и MCP проект среды MPLAB (Рис. 19).

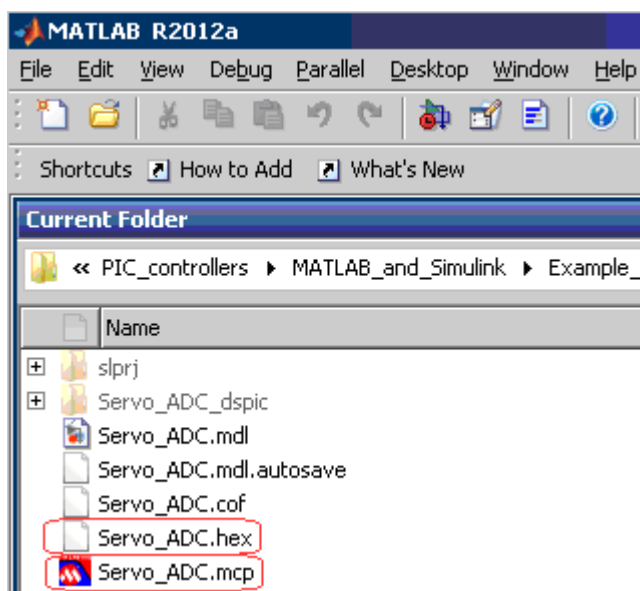


Рис. 19. Результаты компиляции модели.

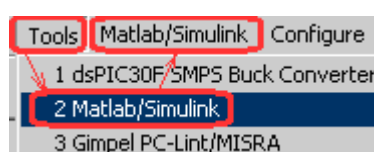
Запуск модели в Matlab/Simulink выполняется в окне модели кнопкой, условное время моделирования устанавливается в строке:



Управление компиляцией Simulink моделей из среды MPLAB

Управление компиляцией Simulink модели можно выполнять командами раздела Matlab/Simulink среды MPLAB, например, в следующем порядке.

1. Разработайте модель PIC контроллера в Matlab/Simulink. Сохраните модель.
2. Запустите MPLAB.
3. Выберите MPLAB меню → Tools → Matlab/Simulink и новый раздел появится в составе меню.



4. В разделе Matlab/Simulink откройте Simulink модель, например, Servo_ADC,

командой «Matlab/Simulink → SpecifySimulinkModelName → Open → Filename → Servo_ADC.mdl → Open». Команда Open запускает Matlab и открывает модель.

5. Откомпилируйте модель и создайте MCP проект командами GenerateCodes или GenerateCodesandImportFiles. Перевод MDL модели в MCP проект выполняется TLC компилятором Matlab.

В результате создаётся проект MPLAB:



со скриптами модели на языке Си.

6. Откройте проект: меню → Project → Open → Servo_ADC.mcp (Рис. 20).

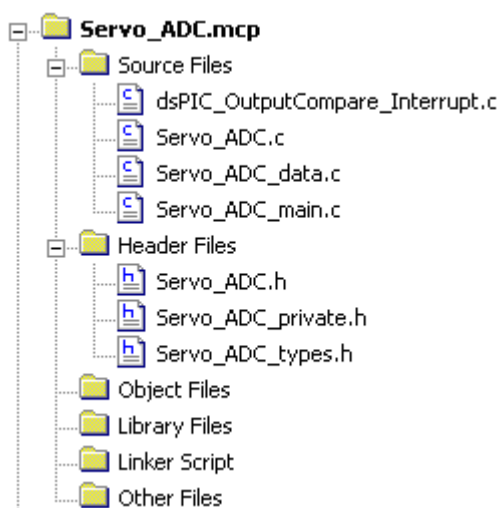


Рис. 20. Структура MCP проекта Simulink модели Servo_ADC.mdl в среде MPLAB. Проект Simulink модели готов для редактирования, отладки и компиляции в машинные коды контроллера средствами MPLAB.

Подключение программатора PIC-KIT3

Узнать какие программаторы записывают бинарный код в конкретный микроконтроллер можно в разделе меню → Configure → SelectDevice среды MPLAB 8.92. Например, программатор PIC-KIT3 не поддерживает контроллер PIC12C508A (Рис. 21, левый рисунок), но работает с контроллером PIC12F629 (Рис. 21, правый рисунок).

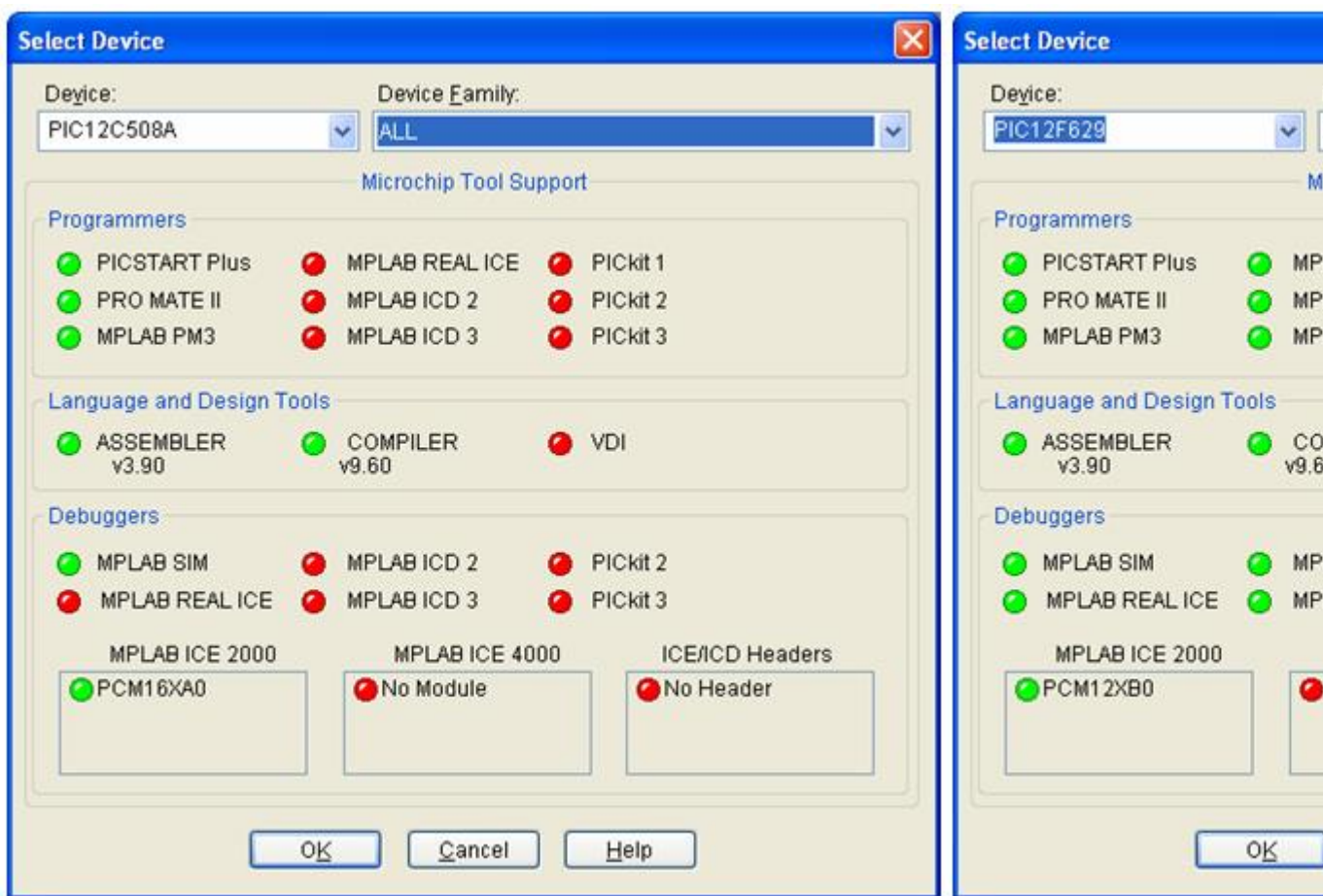


Рис. 21. Перечень программаторов для прошивки микроконтроллера.

Информацию об установленном драйвере программатора PIC-KIT3 можно запросить у менеджера устройств ОС Windows (Рис. 22).

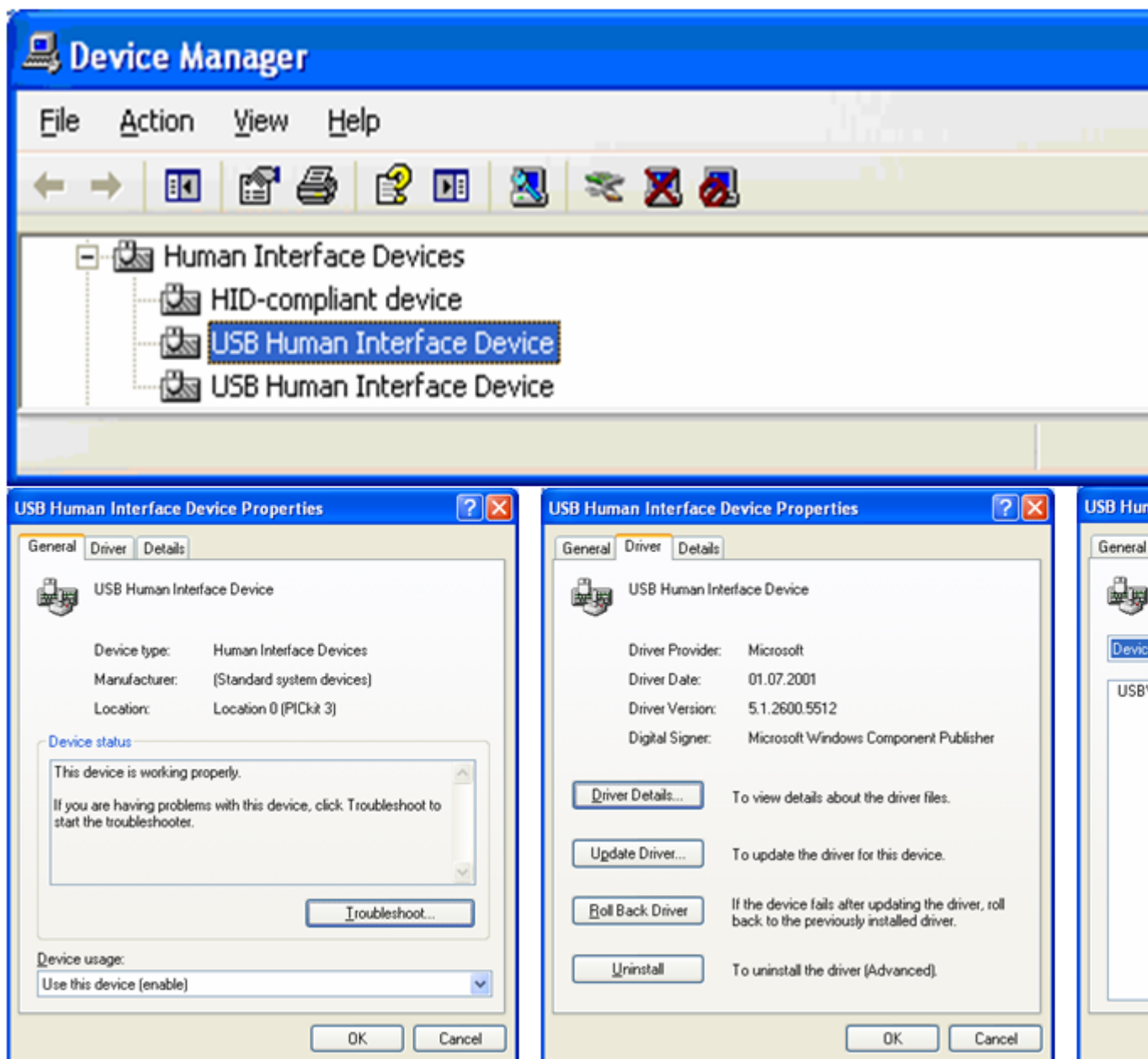


Рис. 22. Информация об установленном драйвере программатора PIC-KIT3.

Схема подключения микроконтроллера PIC12F629 к программатору PIC-KIT3 показана на Рис. 23.

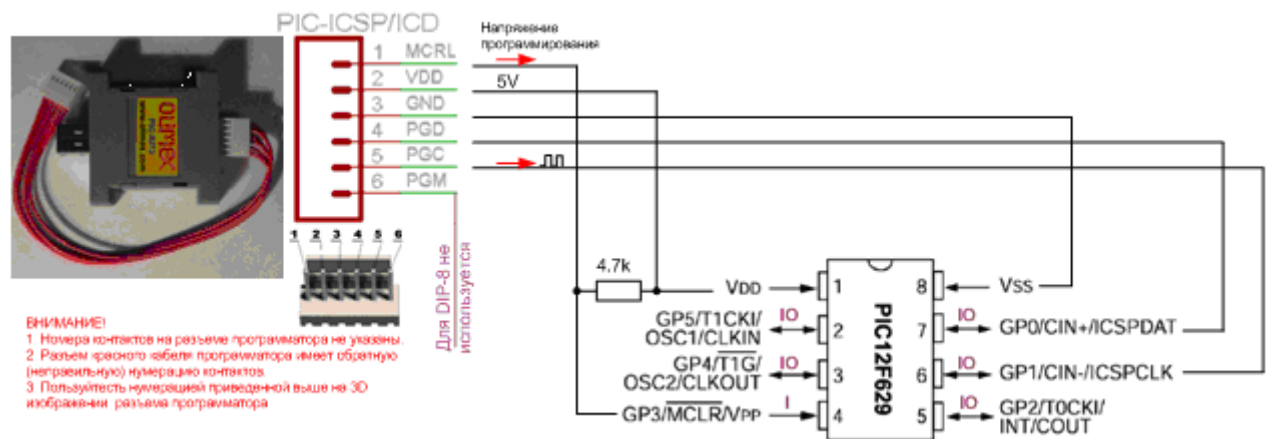


Рис. 23. Схема подключения микроконтроллера PIC12F629 к программатору PIC-KIT3.

Выход PGM программатора для прошивки контроллеров PIC12F629 не используется. Наличие вывода PGM для разных типов PIC контроллеров показано на Рис. 24. Вывод PGM рекомендуется «притягивать» к общему проводу (GND), через резистор, номиналом 1К [3].

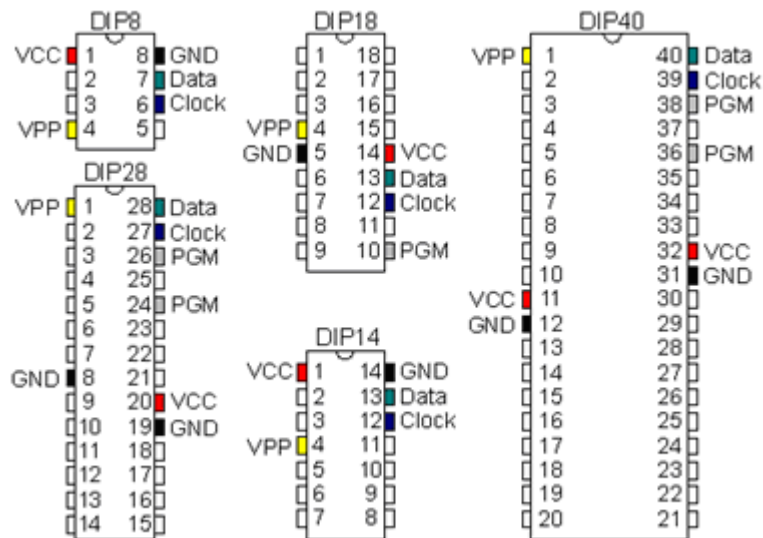


Рис. 24. Выводы PGM PIC контроллеров.

Индикация светодиодов программатора Olimex PIC-KIT3 показана в ниже:

Желтый — Красный — Состояние программатора

Вкл — Выкл — Подключен к USB линии

Вкл — Вкл — Взаимодействие с MPLAB

Мигает — Включен постоянно — Прошивка микроконтроллера

Не следует подключать питание микроконтроллера VDD (Рис. 23) к программатору, если контроллер запитывается от своего источника питания.

При питании микроконтроллера от программатора на линии VDD необходимо установить рабочее напряжение, например, 5В программой MPLAB (Menu → Programmer → Settings → Power), как показано на Рис. 25.

Примечание. При отсутствии напряжения на линии VDD MPLAB IDE выдает сообщение об ошибке: PK3Err0045: You must connect to a target device to use

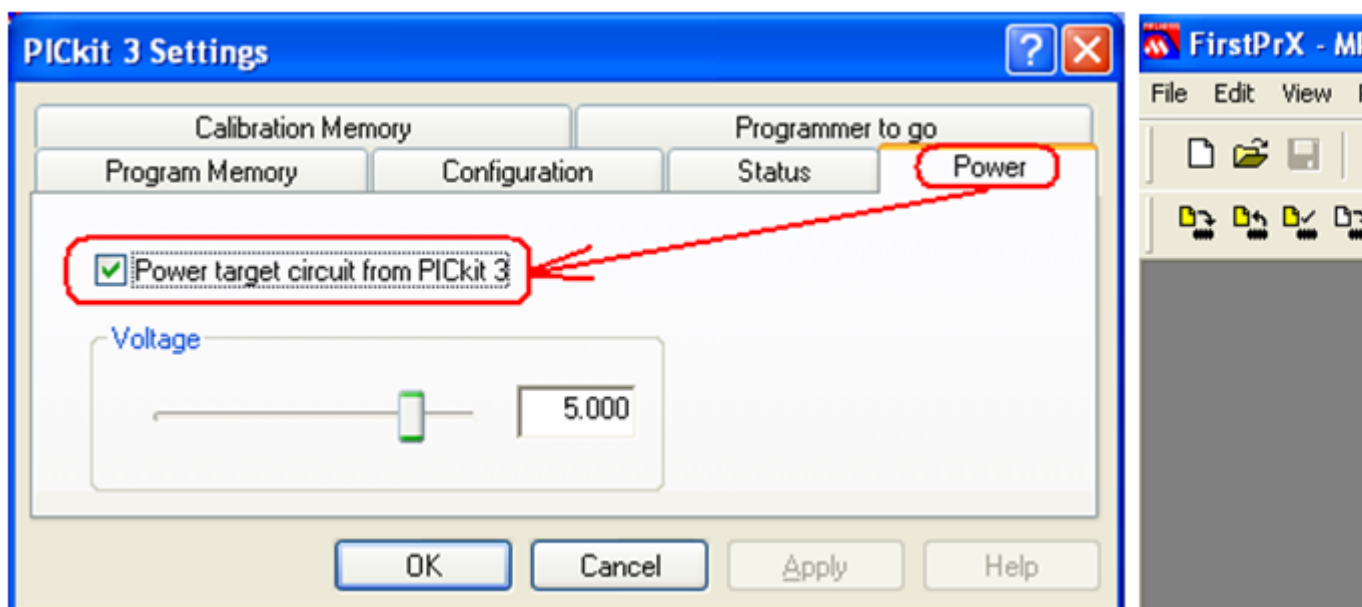


Рис. 25. Установка напряжения VDD на программаторе PIC-KIT3 программой MPLAB IDE v8.92.

Если программатор не может установить требуемое напряжение, например, 5В при его питании от USB, в которой напряжение меньше 5В, MPLAB IDE выдает сообщение об ошибке: PK3Err0035: Failed to get Device ID. В этом случае, сначала необходимо измерить напряжение программатора — считать его в закладке меню → Programmer → Settings → Status, а затем установить напряжение (не больше измеренного) в закладке меню → Programmer → Settings → Power.

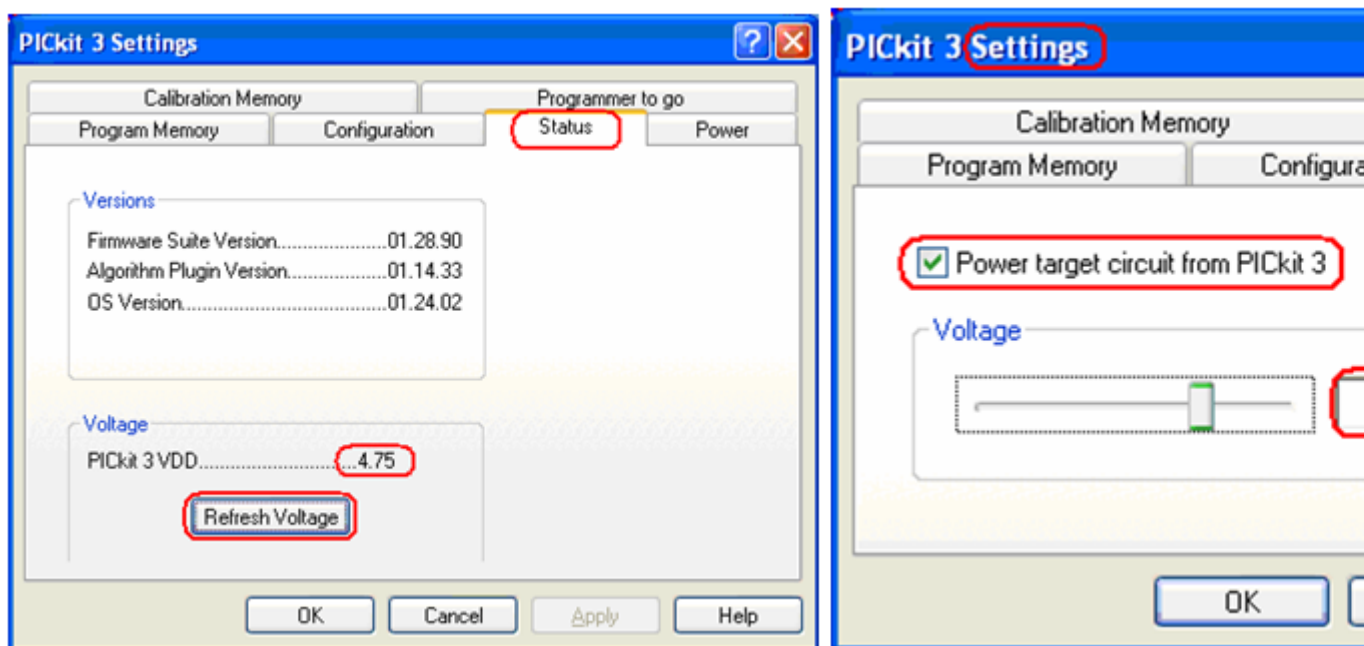


Рис. 26. Измерение (слева) и установка (справа) VDD напряжения программатора PIC-KIT3 программой MPLAB IDE v8.92.

Пример MPLAB сообщения успешного подключения микроконтроллера к программатору по команде меню → Programmer → Reconnect показан на Рис. 27.

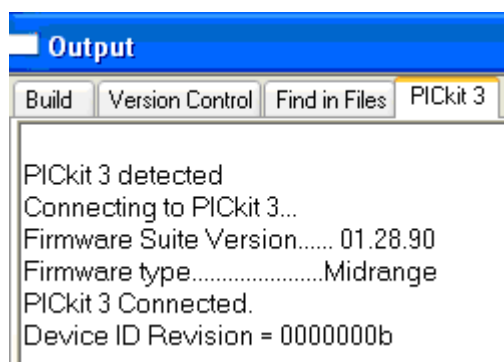


Рис. 27. Сообщение MPLAB об успешном подключении микроконтроллера к программатору.

Можно программировать не только отдельный PIC контроллер, но и контроллер, находящийся в составе рабочего устройства. Для программирования PIC контроллера в составе устройства необходимо предусмотреть установку перемычек и токоограничивающих резисторов как показано на Рис. 28 [3].

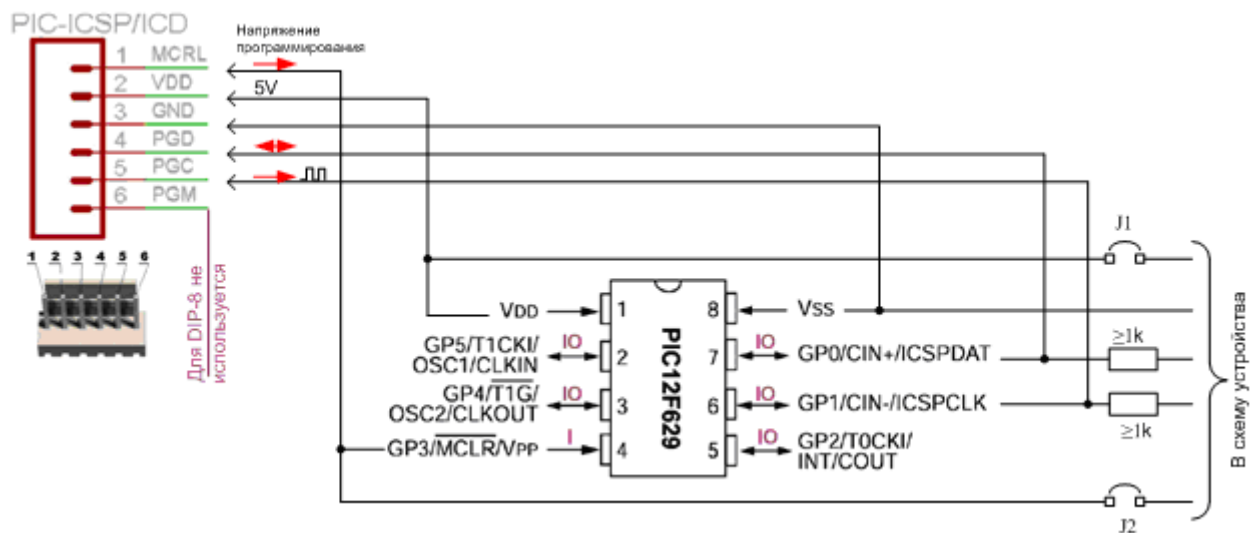


Рис. 28. Подключение микроконтроллера в составе электронного устройства к программатору.

Закключение

Малоразрядные PIC-контроллеры имеют широкий диапазон питания, низкое потребление и малые габариты. Они программируются на языках низкого уровня. Разработка программ на языке графического программирования Simulink с использованием многочисленных библиотек значительно сокращает время разработки и отладки в сравнении с программированием на уровне ассемблера. Разработанные для PIC-контроллеров Simulink структуры можно использовать и для компьютерного моделирования динамических систем с участием контроллеров. Однако, из-за избыточности кода такой подход применим только для семейств PIC контроллеров с достаточными ресурсами.

Программно-аппаратная платформа

Платформа, её составляющие, основные характеристики

В соответствии с определением информационных технологий в их основе заложены средства компьютерной техники, реализующие вычислительные процессы в программной среде под управлением соответствующей операционной системы. Разнообразие как технических, так и программных средств привело к использованию понятия «программно-аппаратная платформа» или «платформа ИТ».

Платформа ИТ определяется как комплекс аппаратных (технических) средств, реализованных на соответствующем типе процессора и соответствующей операционной системы.

Программно-аппаратная платформа состоит из взаимосвязанной совокупности следующих основных элементов:

- — комплекс технических средств (КТС), на базе которого проектируются ИС;
- — базовое программное обеспечение, обеспечивающее интеграцию КТС в программно-технический комплекс, конфигурирование систем и реализующее другие универсальные функции ИС;
- — средства автоматизации проектирования, верификации и валидации ИС;
- — комплект документации, регламентирующий процесс разработки ИС на базе данной платформы.

Комплекс технических средств (техническая платформа) задает тип оборудования, на котором можно установить программное обеспечение, реализующее заданный набор ИТ. Комплекс технических средств имеет сложную структуру. Основным компонентом является электронная вычислительная машина (ЭВМ) — комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач, тип которого определяется типом процессора: Macintosh, Atari, Sincler, Intel, J2EE и т.д. Многие современные технические платформы используют дополнительное оборудование. Например, ИТ, реализуемые в сетях, зависят от сетевого оборудования: модемов, адаптеров, каналов связи и т.д. Для использования технологий мультимедиа необходимы приводы DVD, видео-, звуковые карты. Добавочное оборудование также входит в состав технической платформы.

Вычислительные машины могут быть классифицированы по ряду признаков, например: принцип действия; этапы создания и элементная база; назначение; способ организации вычислительного процесса; размер, вычислительная мощность; функциональные возможности; способность к параллельному выполнению программ.

Функциональные возможности компьютеров обусловлены такими важнейшими технико-эксплуатационными характеристиками, как:

- — быстродействие, измеряемое усредненным количеством операций, выполняемых машиной за единицу времени;
- — разрядность и формы представления чисел, с которыми оперирует компьютер;
- — виды, емкость и быстродействие всех запоминающих устройств;
- — виды и технико-экономические характеристики внешних устройств хранения, обмена и ввода-вывода информации;
- — тип внутри машинного интерфейса, т.е. типы и пропускная способность устройств связи и сопряжения узлов компьютера между собой;
- — многопрограммность, т.е. способность компьютера одновременно работать с несколькими пользователями и выполнять одновременно несколько программ;
- — типы и технико-эксплуатационные характеристики операционных систем, используемых в машине;
- — программная совместимость с другими типами компьютеров, т.е. способность выполнять программы, написанные для других типов компьютеров;
- — система и структура машинных команд;
- — наличие и функциональные возможности программного обеспечения;
- — возможность подключения к каналам связи и вычислительной сети;
- — эксплуатационная надежность компьютера и другие параметры и характеристики.

Дадим краткую характеристику основным классам компьютеров, разделенных по критерию «размер и вычислительная мощность».

Большие компьютеры за рубежом называют *мэйнфреймами* (mainframe). К ним относят, как правило, компьютеры, имеющие производительность не менее 100 MIPS, основную память

емкостью от 512 до 10 000 Мбайт, внешнюю память не менее 100 Гбайт, многопользовательский режим работы при одновременном обслуживают от 16 до 1000 пользователей.

Компьютеры класса «mainframe» нашли широкое применение при решении научнотехнических задач, используются в качестве платформы в вычислительных системах с пакетной обработкой информации, в работе с большими базами данных, в управлении вычислительными сетями и их ресурсами, в качестве больших серверов вычислительных сетей. По данным экспертов, на мэйнфреймах находится около 70 % «компьютерной» информации.

Многопользовательские микрокомпьютеры — это мощные микрокомпьютеры, оборудованные несколькими видеотерминалами и работающие в режиме разделения времени, что позволяет эффективно работать на них сразу нескольким пользователям.

Микрокомпьютеры имеют широкую номенклатуру.

Персональный компьютер (ПК) — однопользовательские микрокомпьютеры, удовлетворяющие требованиям общедоступности и универсальности применения при следующих технико-экономических характеристиках:

- - малая стоимость ПК, находящаяся в пределах доступности для индивидуального покупателя;
- - автономность эксплуатации без специальных требований к условиям окружающей среды;
- - дружелюбность операционной системы и приложений;
- - гибкость архитектуры, обеспечивающая адаптируемость к разнообразным применениям в сфере управления, науки, образования, в быту;
- - обуславливающая возможность работы с ней пользователя без специальной профессиональной подготовки;
- - высокая надежность работы (более 5000 часов наработки на отказ).

Рабочие станции (workstation) представляют собой однопользовательские микрокомпьютеры, часто специализированные для выполнения определенного вида работ, таких как инженерные, издательские и т.д.

Серверы (server) — многопользовательские мощные микрокомпьютеры в вычислительных сетях, выделенные для обработки запросов от всех рабочих станций сети.

Сетевые компьютеры (networkcomputer) — упрощенные микрокомпьютеры, обеспечивающие работу в сети и доступ к сетевым ресурсам, часто специализированные на выполнении определенного вида работ, таких как защита сети от несанкционированного доступа, организация просмотра сетевых ресурсов, электронной почты и т.д.

В современных условиях происходит своеобразное разделение труда в среде компьютеров. Решение пользовательских задач, поддержание GUI (graphicaluserinterface), предварительная обработка информации осуществляются с помощью ПК с небольшой мощностью. А поддержание больших БД, сложные расчеты и вычисления реализуются с помощью рабочих станций.

Основным компонентом базового программного обеспечения (программной платформы) является операционная система, обеспечивающая работоспособность прикладного программного обеспечения на том или ином процессоре. Для обеспечения работоспособности добавочного оборудования разрабатываются специальные программные средства (драйверы). Многие из них входят в состав операционных систем.

Средства автоматизации проектирования (Computer-Aided Design — CAD) — комплекс программных, технических, технологических, информационных средств, включающих в себя и проектно-конструкторскую документацию, а также персонал системы, предназначенный для автоматизации процессов проектирования, в том числе подготовку проектно-конструкторской документации различных технических объектов. Такие средства широко используются при создании ИС. Технология создания крупных ИС предъявляет особые требования к методикам реализации и программным инструментальным средствам. Реализацию крупных проектов принято разбивать на стадии анализа, проектирования, непосредственного кодирования, тестирования и сопровождения. Известно, что исправление ошибок, допущенных на предыдущей стадии, обходится примерно в 10 раз дороже, чем на текущей, откуда следует, что наиболее критичными являются первые стадии проекта. В связи с этим крайне важно иметь эффективные средства автоматизации ранних этапов реализации проекта.

Крупный проект невозможно реализовать в одиночку. Коллективная работа существенно отличается от индивидуальной, поэтому при реализации крупных проектов необходимо иметь средства координации и управления коллективом разработчиков. Жизненный цикл создания сложной ИС сопоставим с ожидаемым временем ее эксплуатации.

Другими словами, в современных условиях компании перестраивают свои бизнес-процессы примерно один раз в два года, столько же требуется (если работать в традиционной технологии) для создания ИС. Может оказаться, что к моменту сдачи ИС она уже никому не нужна, поскольку компания, ее заказавшая, вынуждена перейти на новую технологию работы. Значит, для создания крупной ИС жизненно необходим инструмент значительно (в несколько раз) уменьшающий время разработки ИС. Вследствие значительного жизненного цикла может оказаться, что в процессе создания системы внешние условия изменились.

Обычно внесение изменений в проект на поздних этапах создания ИС — весьма трудоемкий и дорогостоящий процесс. В связи с этим для успешной реализации крупного проекта необходимо, чтобы инструментальные средства, на которых он реализуется, были достаточно гибкими к изменяющимся требованиям. На современном рынке средств разработки ИС достаточно много систем, в той или иной степени удовлетворяющих перечисленным требованиям, например, технология разработки, основывающаяся на решениях фирм «LogicWorks» и «RationalSoftware», которая является одной из лучших на сегодняшний день по критерию «стоимость — эффективность».

Комплект документации включает в себя полное описание программы и необходимый состав сведений для ее распространения (в том числе продажи) и использования. Состав и содержание документации программного обеспечения зависят от характеристик проектирования, разработки и модификации программных средств, а также от требований к их качеству и особенностям технологической среды. В связи с этим необходимый комплект документов для каждого предприятия или проекта следует выбирать и адаптировать применительно к этим характеристикам. Оцениваемыми показателями являются наличие соответствующих документов и практическое выполнение требований определенного уровня модели зрелости CapabilityMaturityModelIntegrated (CMMI) или адаптированного профиля стандартов на базе ISO 9000:2000, а также созданных на их основе должностных инструкций специалистами предприятия-разработчика.

Ориентировочный комплект основных документов при сертификации состоит из трех групп:

- 1) базовые нормативные документы систем качества в соответствии с номенклатурой и содержанием профиля стандартов на базе ISO 9000:2000 или модели зрелости CMMI, а также подготовленные разработчиками на их основе программа, руководство и инструкции;

- 2) исходные документы, характеризующие конкретное предприятие или проект, а также жизненный цикл программного средства, подготавливаемые руководством проекта;
- 3) отчетные документы испытателей, отражающие результаты проверки (сертификации) программного продукта.

Перечень и приблизительное содержание групп этих документов ориентированы на общий случай проверки их качества, обеспечивающих жизненный цикл крупных программных продуктов. Комплект документов может сокращаться и адаптироваться. Некоторые документы могут объединяться в интегрированные отчеты с четкой ответственностью определенных специалистов за их выполнение.

Классификацию программно-аппаратных платформ можно провести по нескольким признакам (характеристикам). Если в качестве классификационного признака взять используемый сервер баз данных, то можно выделить следующие классы программно-аппаратных платформ. В однопользовательской, или настольной, платформе, не используется сервер базы данных. Такая платформа позволяет работать как одному сотруднику, так и небольшой группе. В корпоративной платформе почти всегда оперируют с одним или несколькими серверами баз данных. Такая платформа предназначена для рабочей группы или компании. Интернет-платформа позволяет применять internet- или intranet-приложения, которые используют Web-сервер.

Рассмотрим основные характеристики, которыми должна обладать платформа:

1) **необходимая производительность.** В зависимости от задач, стоящих перед организацией, платформа должна обеспечивать необходимую производительность. С ростом организации её информационные потребности могут значительно вырасти, поэтому также необходимо обеспечить возможность роста производительности платформы;

- 2) **высокая надежность и живучесть.** Информационные системы, реализуемые на базе платформы, не могут простаивать, так как при этом останавливается работа организации. Поэтому необходимы, с одной стороны, надежность и устойчивость системы к неполадкам, а с другой стороны, возможность устранения неисправности без остановки общей работы;
- 3) **защита от несанкционированного доступа.** В большинстве информационных систем хранится конфиденциальная информация, поэтому необходимы средства защиты информации от доступа посторонних лиц.
- 1.6.2 **Операционные системы в обеспечении информационных технологий**

Операционная система (ОС) является обязательной частью системного программного обеспечения компьютера. В функции операционной системы входит организация выполнения программ и взаимодействия пользователя и внешних устройств с компьютером, обеспечение эффективного функционирования ПК в различных режимах.

С технической точки зрения операционная система представляет комплекс программ, обеспечивающий управление ресурсами компьютера, процессами обработки информации, использующими эти ресурсы, и данными.

К программным ресурсам относятся все доступные пользователю программные средства управления вычислительными процессами и данными.

Управление процессами обработки информации заключается в организации и реализации эффективных режимов функционирования компьютера:

- - однопользовательский и многопользовательский режимы (совместная работа с компьютером одновременно нескольких пользователей через отдельные терминалы);
- - однопрограммный (однозадачный) и многопрограммный (многозадачный) режимы работы;
- - режим формирования виртуальных машин (каждому пользователю в рамках основной конфигурации компьютера выделяется как бы отдельная машина меньшей производительности, возможно, со своей операционной системой);
- - работа в однопроцессорных, многопроцессорных, многомашинных, в том числе и сетевых, вычислительных системах.

Управление данными имеет целью обеспечить идентификацию, организацию и хранение данных, обрабатываемых в компьютере. Организация данных связана с созданием библиотек и баз данных, их актуализацией, обеспечением эффективного доступа к данным и их выборки.

Дружественность операционной системы обеспечивает необходимый сервис пользователю в процессе выполнения прикладных программ, ком-

фортные условия программисту для разработки и отладки программ, а также для хранения, преобразования, отображения и копирования информации.

Отличительными чертами операционной системы Windows NT являются:

- - встроенная сетевая поддержка, обеспечивающая совместное сетевое использование файлов, устройств и объектов;
- - приоритетная многозадачность, позволяющая приложениям с более высоким приоритетом вытеснять менее приоритетные приложения;
- - наличие достаточно мощных средств защиты программ и файлов различных пользователей от несанкционированного доступа;
- - наличие многоуровневого доступа к ресурсам с назначением пользователям уровня доступа в соответствии с их компетенцией;
- - поддержка нескольких файловых систем.

LUNIX — представляет собой одну из альтернатив семейству операционных систем Windows. Система UNIX — это не только многозадачная, но и многопользовательская операционная система, которая позволяет нескольким пользователям разделять вычислительные ресурсы одного компьютера. Система обладает простым пользовательским интерфейсом, поскольку она написана на языке высокого уровня, ее легко понимать, изменять, переносить на другие аппаратные платформы. Файловая система UNIX обеспечивает единый интерфейс доступа к данным и к периферийным устройствам.

LUNIX — является полной многозадачной и многопользовательской операционной системой, поддерживающей национальные и стандартные клавиатуры, разные типы файловых систем для хранения данных, обеспечивающая полный набор протоколов TCP/IP для работы в сети. Система компактна, отличается высоким уровнем надежности.

Таким образом, операционные системы позволяют автоматизировать стандартные процедуры управления аппаратными и программными средствами.

Контрольные вопросы

- 1. Что такое платформа ИТ?
- 2. Что понимается под технологическим процессом (цепочкой)?
- 3. Что такое технология?
- 4. Какова цель применения информационной технологии?

- 5. Какова основная роль ИТ в развитии экономики?
- 6. Что такое информатизация общества?
- 7. Что произвело вторую информационную революцию в обществе?
- 8. Что произвело первую информационную революцию?
- 9. Что включает информационная технология решения экономических задач?
- 10. Что такое информационная технология?

- 11. Что относят к средствам проектирования ИС?
- 12. Что определяет прикладное программное обеспечение?
- 13. Для чего используются системы программирования?
- 14. Что входит в состав общего программного обеспечения?
- 15. Что понимается под информационными ресурсами?
- 16. Что обеспечивает дружелюбность операционной системы?
- 17. Что является отличительными чертами операционной системы Windows NT?
- 18. Какова цель управления данными?
- 19. Что представляют собой серверы (server)?
- 20. Что представляет собой операционная система с технической точки зрения?
- 21. Что относится к программным ресурсам?
- 22. В чем заключается управление процессами обработки информации?
- 23. К какому виду программного обеспечения компьютера относится операционная система?
- 24. Что входит в функции операционной системы?
- 25. Что представляют собой рабочие станции?

Прикладные программные комплексы

Под прикладным программным комплексом понимается совокупность работающих совместно прикладных программных продуктов. Прикладной программный комплекс, может включать следующую совокупность прикладных программных продуктов:

- • базу данных;
- • табличный процессор;
- • текстовый редактор;
- • графический редактор;
- • бухгалтерские и другие программы.

Структура прикладных программных комплексов, как правило, носит динамичный характер. По мере необходимости, в прикладной программный комплекс подключаются новые прикладные продукты. В том случае, если какой-либо программный продукт морально устарел или в нем отпала необходимость, то данный программный продукт может быть исключен из прикладного программного комплекса.

Программные продукты, входящие в прикладной программный комплекс, как правило, пользуются единой информационной базой. Информационная база прикладного программного комплекса может быть представлена в виде сетевой модели. Допустимо, если при передаче информации в какой-либо, прикладной программный продукт выполняется ее преобразование.

Информационная взаимосвязь между различными компонентами прикладного

программного комплекса, может обеспечиваться с использованием объектно-ориентированных технологий.

Информационный обмен между прикладными программными продуктами

Информационный обмен является одним из важнейших средств повышения эффективности применения прикладного программного обеспечения. Применяя технологии передачи информации, пользователь может использовать на различных этапах обработки информации те прикладные программные продукты, которые наиболее эффективны для данного вида обработки.

Можно выделить следующие виды передачи информации:

- передача информации через стандартный текстовый файл;
- импорт и экспорт информации с использованием, файлов других прикладных систем;
- использование технологии OLE (Object Linking and Embedding).

Использование *файловой обработки* позволяет обеспечить широкие возможности обмена информации с другими прикладными программными продуктами.

Текстовые файлы предназначены для хранения информации в виде последовательности символов, представленных в формате ASCII (AmericanStandardCodeofInformationInterchange) или ANSI (AmericanNationalStandardInstitute).

Преобразование файлов, при импорте и экспорте информации, как правило, эффективно для тех прикладных продуктов, которые получили достаточно широкое распространение. В этом случае, другие прикладные системы имеют встроенные средства для преобразования в формат данной системы. В соответствии с международными стандартами прикладные продукты должны иметь средства преобразования в форматы других систем как, для интерактивной, так и для командной обработки.

Технология OLE - связывание и встраивание объектов (связь и внедрение), представляет собой один из механизмов обмена данными. Данная технология поддерживается почти во всех прикладных системах, работающих под управлением операционной системы Windows. В частности, технология OLE поддерживается в таких программных продуктах как VisualBasic, VisualFoxPro, Delphi, пакет Statistica.

Объект OLE представляет собой программно обрабатываемый элемент данных. Например, объектом могут быть следующие элементы: диаграмма, рисунок, текст, таблица, звук.

Объект связан с прикладной программой, с помощью которой он создавался. Это обеспечивает возможность обращения к прикладной программе при помощи

объекта. Если необходимо внести изменения в объект, то возможность использования инструмента, обеспечивающего изменения, предоставляется при обращении к объекту. Например, в табличный файл базы данных встроен рисунок, созданный с помощью графического редактора. Если необходимо, вновь задействовать инструмент графического редактора, то выполняется двойной щелчок мыши по встроенному рисунку.

OLE реализуется в том случае, когда несколько прикладных продуктов Windows одновременно функционируют в многозадачном режиме.

Прикладные программы, поддерживающие связь и внедрение объектов, делятся на две категории:

Во-первых, программы, объекты которых встраиваются или связываются с другими прикладными программами. Данные программы называются серверами (Server).

Во-вторых, программы, позволяющие принимать встраиваемые или связываемые объекты. Эти программы называются **клиентами** Client.

Например, если в таблицу (базу данных) VisualFoxPro встраивается картина, нарисованная с помощью средств графического редактора, то графический редактор будет сервером, а VisualFoxPro будет клиентом.

При реализации OLE сервер передает клиенту специальный документ, содержащий информацию об объекте. Этот документ называется **документ клиента** (ClientDocument). Документ клиента сохраняется в специально отведенном для него месте, которое называется **контейнер** (Container). В VisualFoxPro для запоминания информации об объекте используется поле базы данных, имеющее тип GEN (GeneralField). Таким образом, поле табличного файла, объявленное с типом GeneralField является контейнером.

Прикладные программные продукты могут быть использованы или как серверы, или как клиенты, или и как серверы, и как клиенты.

Такие прикладные продукты, как Graph могут быть использованы только как сервер, но не могут быть клиентами.

Такие прикладные продукты, как Excel и Word могут быть как серверами, так и клиентами.

VisualFoxPro может быть использован только как клиент. Это значит, что в таблицы базы данных VisualFoxPro можно встроить и связать объект из другого прикладного программного продукта. Но в VisualFoxPro нельзя создать объект, который можно было бы встроить в другую прикладную программу.

Связывание и встраивание объектов различаются следующим образом.

При использовании «связывания» (Linking) объекта с VisualFoxPro он сохраняется в отдельном файле. Формат файла определяется родителем для данного объекта приложением (сервером). При открытии файла, с которым связаны некоторые внешние объекты, выполняется актуализация, и в файл принимается свежая информация из связанных объектов. Один объект можно связать с несколькими файлами. Внесенные в этот объект изменения будут

отражены во всех файлах, с которыми связан этот объект.

При «встраивании» (Embedding) объекта VisualFoxPro сохраняет объект в своем файле базы данных. Встроенный объект является составной частью базы данных, но этот объект нельзя создать непосредственно в VisualFoxPro.

Процедуры связывания и встраивания объектов, как правило, выполняются в диалоговом режиме.

Контрольные вопросы

Расскажите о программном обеспечении, его классификации.

- 1. Расскажите о системном и прикладном программном обеспечении.
- 2. Дайте понятие программного изделия.
- 3. Каковы основные требования, предъявляемые к программному изделию?
- 4. Какова классификация прикладных программных средств по характеру применения?
- 5. В чем особенности применения программных средств коммерческого использования (программных продуктов)?
- 6. Каковы особенности применения индивидуально разрабатываемых программных средств?
- 7. Дайте характеристику основных классов программных средств, реализующих типовые процедур (системы управления базами данных, табличные процессоры, текстовые редакторы, бухгалтерские программы, статистические программы, графические редакторы).
- 8. Расскажите об информационном обмене между различными прикладными системами.
- 9. Какое прикладное программное обеспечение эффективно использовать в данном случае?
- 10. Какое прикладное программное обеспечение необходимо индивидуально разработать в данном случае?
- 11. Каким образом обеспечить комплексную обработку экономической информации по определенной задаче?

Тесты

- 1. *Прикладное программное обеспечение по характеру применения делится на прикладные программные средства коммерческого использования (программные продукты) и*
 - 1)
 - 2)
 - 3)
 - 4)

операционные системы;

системы программирования;

архиваторы;

индивидуально разрабатываемые программные средства (не распространяемые

программы);

- 5) драйверы.
- 2. *Запросы на обработку информации в базах данных, как правило, делаются с помощью языка*
- 1) Delphi;
- 2) VisualC;
- 3) SQL;
- 4) VBA;
- 5) PERL.
- 3. *Укажите соответствие между названием и видом прикладной системы:*
- 1)
- 2)
- 3)
- 4)
- 5)
- 1) статистическая программа;
- 2) бухгалтерская программа;
- 3) система управления базами данных;
- 4) табличный процессор;
- 5) текстовый редактор.

Oracle

Excel

Word

1С: Бухгалтерия

Statistica

- 4. *Программные продукты, входящие в прикладной программный комплекс:*
- 1) не пользуются единой информационной базой;
- 2) пользуются единой информационной базой.
- 5. *Информационная взаимосвязь между различными компонентами прикладного программного комплекса:*
- 1) может обеспечиваться с использованием объектно-ориентированных технологий;
- 2) не может обеспечиваться с использованием объектно-ориентированных технологий.
- 6. *Можно выделить следующие виды передачи информации:*
- 1) передача информации через стандартный текстовый файл;
- 2) импорт и экспорт информации с использованием, файлов других прикладных систем;
- 3) использование технологии структурного программирования;
- 4) использование технологии OLE;
- 5) через операторы цикла.
- 7. *Программы, объекты которых встраиваются или связываются с другими прикладными программами называются ...:*
- 1) операционные системы;

- 2) системы программирования;
- 3) архиваторы;
- 4) серверы (Server);
- 5) драйверы.
- 8. *Программы, позволяющие принимать встраиваемые или связываемые объекты называются ...:*
 - 1) операционные системы;
 - 2) системы программирования;
 - 3) архиваторы;
 - 4) клиенты (Client);
 - 5) драйверы.
- 9. *При реализации OLE сервер передает клиенту специальный документ, содержащий информацию об объекте, который называется...:*
 - 1) Application;
 - 2) ProjectManajer;
 - 3) ClientDocument;
 - 4) DatabaseDesigner;
 - 5) TableDesigner.

Автоматизация процесса программирования. CASE-технологии

Автоматизация технологических процессов и производств

Аббревиатура CASE (Computer-Aided Software Engineering – автоматизированная разработка программного обеспечения) обозначает специальный тип программного обеспечения, предназначенного для поддержки таких процессов создания ПО, как разработка требований, проектирование, кодирование и тестирование программ. Поэтому к CASE-средствам относятся редакторы проектов, словари данных, компиляторы, отладчики, средства построения систем и т.п. CASE-средства предлагают поддержку процесса создания ПО путем автоматизации некоторых этапов разработки, а также создания и представления информации, необходимой для разработки.

В настоящее время подходящие CASE-технологии существуют для большинства процессов, выполняемых в ходе разработки ПО. Это ведет к улучшению качества создаваемых программ и повышению производительности труда разработчиков программного обеспечения. Уже сложилась развитая индустрия CASE-средств, а круг поставщиков и разработчиков этих программных продуктов очень широк.

Классификация CASE-средств

Существует несколько различных классификаций CASE-средств, и каждая предлагает свой взгляд на эти программные продукты. Рассмотрим классификацию по категориям, где CASE-средства классифицируются по степени

интеграции программных модулей, поддерживающих различные процессы разработки ПО. Эта классификация содержит три основные категории:

1. *Вспомогательные программы (tools)* поддерживают отдельные процессы разработки ПО, такие как проверка непротиворечивости архитектуры системы, компиляция программ, сравнение результатов тестов и т.д. Вспомогательные программы могут быть универсальными функционально законченными средствами или могут входить в состав инструментальных средств.

2. *Инструментальные средства (workbenches)* поддерживают определенные процессы разработки ПО, такие как создание спецификации, проектирования. Обычно они представляют собой набор интегрированных вспомогательных программ.

3. *Рабочие среды разработчиков (environments)* поддерживают все или большинство процессов разработки ПО и включают в себя несколько различных интегрированных инструментальных средств.

Схематично классификация по категориям с примерами CASE-средств различных категорий представлена на рис. 2.



Рис. 2. Классификация CASE-средств по категориям

Вспомогательные программы выбираются разработчиками ПО обычно по своему усмотрению. Инструментальные средства поддерживают определенные методы разработки на основе некоторой модели разработки ПО, наборов правил и нормативных указаний. Интегрированные рабочие среды представляют

инфраструктуру поддержки для данных, управления и интеграции системных представлений. Экспертные рабочие среды более интеллектуальны. Они включают базу знаний о процессах создания ПО и механизм, предлагающий разработчику те или иные вспомогательные или инструментальные средства.

Инструментальные средства обычно объединяют через общий репозиторий.

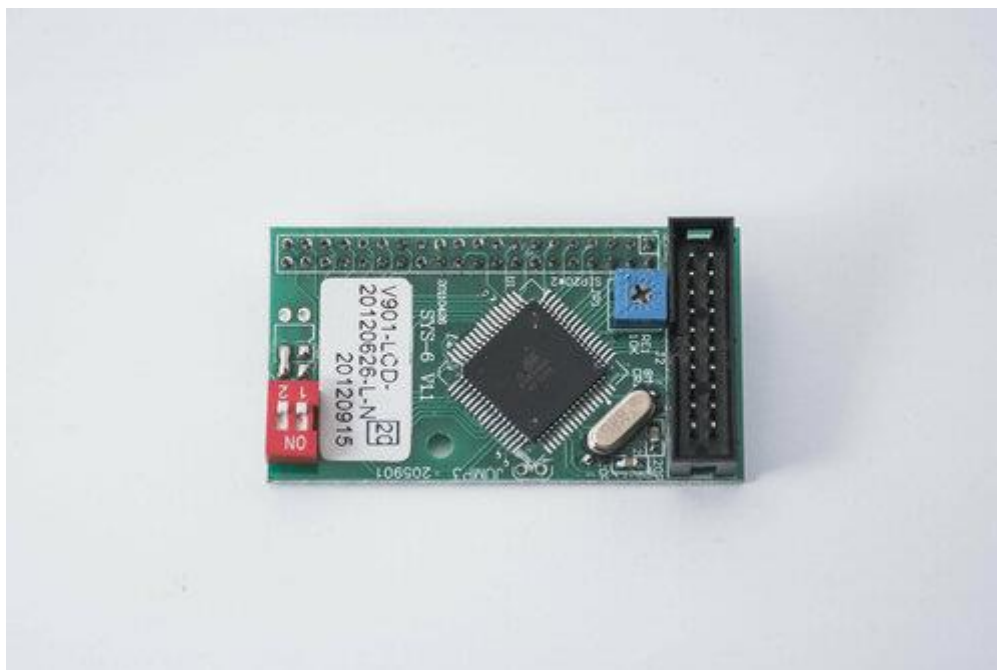
Структура пакета инструментальных средств показана на рис. 3.



Рис. 3. Пакет инструментальных CASE-средств

Такой пакет образуют следующие средства.

1. Редакторы диаграмм предназначены для создания диаграмм потоков данных, иерархий объектов, диаграмм «сущность-связь». Эти редакторы не только имеют средства рисования, но и поддерживают различные типы объектов, используемых в диаграммах.
2. Средства проектирования, анализа и проверки выполняют проектирование ПО и создают отчет об ошибках и дефектах в системной архитектуре. Они могут работать совместно с системой редактирования, поэтому обнаруженные ошибки можно устранить на ранней стадии проектирования.
3. Центральный репозиторий позволяет проектировщику найти нужный проект и соответствующую ему проектную информацию.



Автоматизация технологических процессов и производств

4. Словарь данных хранит информацию об объектах, которые используются в структуре системы.
5. Средства генерирования отчетов на основе информации из центрального репозитория автоматически генерируют системную документацию.
6. Средства создания форм определяют форматы экранных форм и документов.
7. Средства импортирования и экспортирования позволяют обмениваться информацией из центрального репозитория различными инструментальным средствам.
8. Генераторы программного кода автоматически генерируют программы на основе проектов, хранящихся в центральном репозитории.

Различают два класса инструментальных систем технологии программирования: инструментальные системы поддержки проекта и языково-зависимые инструментальные системы.

Инструментальная система поддержки проекта - это открытая система, способная поддерживать разработку ПС на разных языках программирования после соответствующего ее расширения программными инструментами, ориентированными на выбранный язык. Набор инструментов такой системы поддерживает разработкой ПС, а также содержит независимые от языка

программирования инструменты, поддерживающие разработку ПС (текстовые и графические редакторы, генераторы отчетов и т.п.). Кроме того, он содержит инструменты расширения системы. Ядро такой системы обеспечивает, в частности, доступ к репозиторию.

Языково-зависимая инструментальная система - это система поддержки разработки ПС на каком-либо одном языке программирования, существенно использующая в организации своей работы специфику этого языка. Эта специфика может сказываться и на возможностях ядра (в том числе и на структуре репозитория), и на требованиях к оболочке и инструментам. Примером такой системы является среда поддержки программирования на языке Ада.

На сегодняшний день наиболее развитым из всех поставляемых в России комплексов такого рода является комплекс технологий и инструментальных средств создания информационных систем, основанный на методологии и технологии DATARUN. В состав комплекса входят следующие инструментальные средства:

- CASE-средство Silverrun;
- средство разработки приложений JAM;
- мост Silverrun-RDM JAM;
- комплекс средств тестирования QA;
- менеджер транзакций Tuxedo;
- комплекс средств планирования и управления проектом SE Companion;
- комплекс средств конфигурационного управления PVCS;
- объектно-ориентированное CASE-средство RationalRose;
- средство документирования SoDA.

Понятие жизненного цикла ПО. Процессы жизненного цикла

Понятие жизненного цикла программного обеспечения (ЖЦ ПО) является одним из базовых в программной инженерии. *Жизненный цикл программного обеспечения* определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации

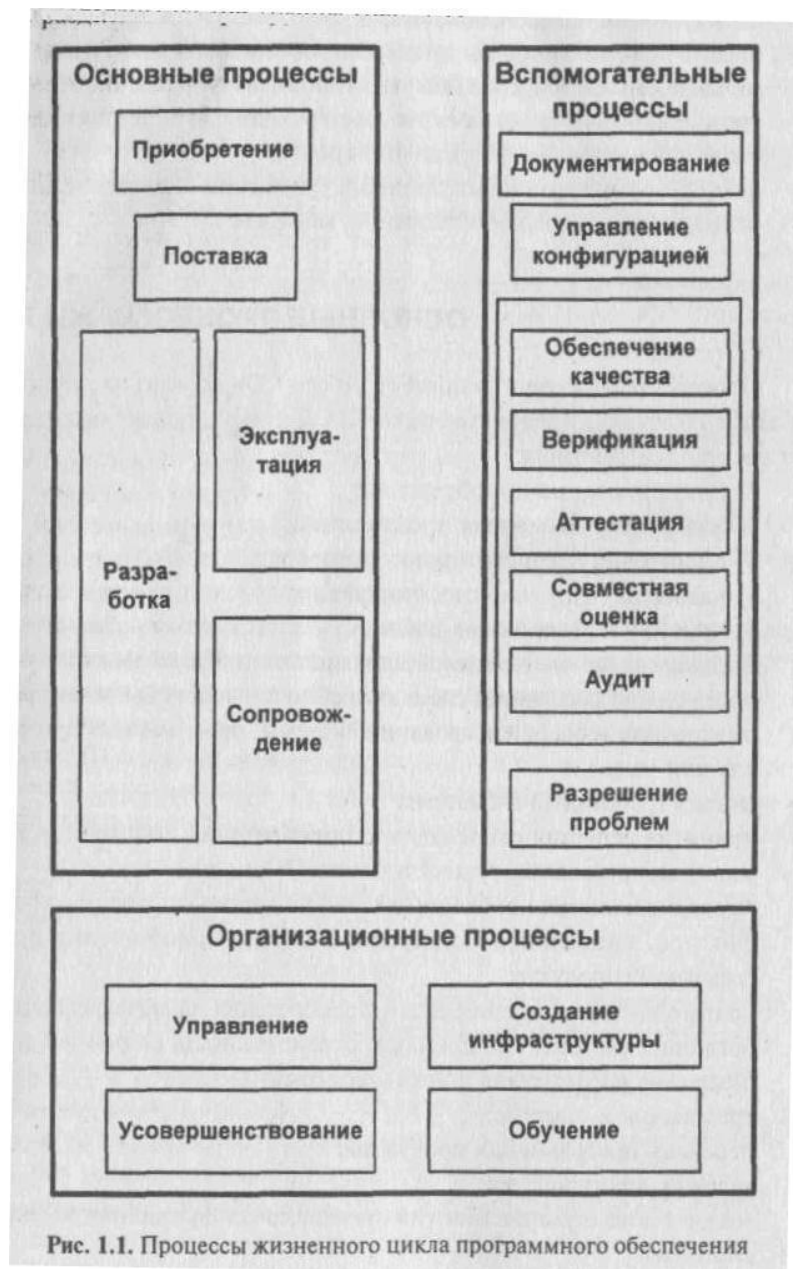
Основным нормативным документом, регламентирующим состав процессов ЖЦ ПО, является международный стандарт ISO/IEC 12207: 1995 "Information Technology -Software Life Cycle Processes" (ISO — International Organization for Standardization - Международная организация по стандартизации, IEC — International Electrotechnical Commission — Международная комиссия по электротехнике). Он определяет структуру ЖЦ, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания ПО. В данном стандарте ПО (или программный продукт) определяется как набор компьютерных программ, процедур и,

возможно, связанной с ними документации и данных. Процесс определяется как совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные. Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными от других процессов, и результатами.

Каждый процесс разделен на набор действий, каждое действие — на набор задач. Каждый процесс, действие или задача инициируется и выполняется другим процессом по мере необходимости, причем не существует заранее определенных последовательностей выполнения (естественно, при сохранении связей по входным данным).

Следует отметить, что в России создание ПО первоначально, в 70-е гг., регламентировалось стандартами ГОСТ ЕСПД (Единой системы программной документации — серия ГОСТ 19.XXX), которые были ориентированы на класс относительно простых программ небольшого объема, создаваемых отдельными программистами. В настоящее время эти стандарты устарели концептуально и по форме, их сроки действия закончились и использование нецелесообразно. Процессы создания автоматизированных систем (АС), в состав которых входит и ПО, регламентированы стандартами ГОСТ 34.601-90 "Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания", ГОСТ 34.602-89 "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы" и ГОСТ 34.603-92 "Информационная технология. Виды испытаний автоматизированных систем". Однако процессы создания ПО для современных распределенных ЭИС, функционирующих в неоднородной среде, в этих стандартах отражены недостаточно, а отдельные их положения явно устарели. В результате для каждого серьезного проекта ЭИС приходится создавать комплекты нормативных и методических документов, регламентирующих процессы создания конкретного прикладного ПО, поэтому в отечественных разработках целесообразно использовать современные международные стандарты.

В соответствии со стандартом ISO/IEC 12207 все процессы ЖЦ ПО разделены на три группы (рис. 1.1):



- пять основных процессов (приобретение, поставка, разработка, эксплуатация, сопровождение);
- восемь вспомогательных процессов, обеспечивающих выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем);
- четыре организационных процесса (управление, создание инфраструктуры, усовершенствование, обучение).

Основные процессы жц ПО

Процесс приобретения (acquisition process). Он состоит из действий и задач заказчика, приобретающего ПО. Данный процесс охватывает следующие действия:

1. инициирование приобретения;
2. подготовку заявочных предложений;
3. подготовку и корректировку договора;
4. надзор за деятельностью поставщика;
5. приемку и завершение работ.

Инициирование приобретения включает следующие задачи:

- определение заказчиком своих потребностей в приобретении, разработке или усовершенствовании системы, программных продуктов или услуг;
- анализ требований к системе;
- принятие решения относительно приобретения, разработки или усовершенствования существующего ПО;
- проверку наличия необходимой документации, гарантий, сертификатов, лицензий и поддержки в случае приобретения программного продукта;
- подготовку и утверждение плана приобретения, включающего требования к системе, тип договора, ответственность сторон и т. д. *Заявочные предложения* должны содержать:
- требования к системе;
- перечень программных продуктов;
- условия и соглашения;
- технические ограничения (например, среда функционирования системы).

Заявочные предложения направляются выбранному поставщику (или нескольким поставщикам в случае проведения тендера). *Поставщик* -это организация, которая заключает договор с заказчиком на поставку системы, ПО или программной услуги на условиях, оговоренных в

договоре.

Подготовка и корректировка договора включают следующие задачи:

- определение заказчиком процедуры выбора поставщика, включающей критерии оценки предложений возможных поставщиков;
- выбор конкретного поставщика на основе анализа предложений;
- подготовку и заключение договора с поставщиком;
- внесение изменений (при необходимости) в договор в процессе его выполнения.

Надзор за деятельностью поставщика осуществляется в соответствии с действиями, предусмотренными в процессах *совместной оценки и аудита*.

В процессе *приемки* подготавливаются и выполняются необходимые тесты. *Завершение работ* по договору осуществляется в случае удовлетворения всех условий приемки.

Процесс поставки (supply process). Он охватывает действия и задачи, выполняемые поставщиком, который снабжает заказчика программным продуктом или услугой. Данный процесс включает следующие действия:

1. инициирование поставки;
2. подготовку ответа на заявочные предложения;
3. подготовку договора;
4. планирование;
5. выполнение и контроль;
6. проверку и оценку;
7. поставку и завершение работ.

Инициирование поставки заключается в рассмотрении поставщиком заявочных предложений и принятии решения согласиться с выставленными требованиями и условиями или предложить свои.

Планирование включает следующие задачи:

- принятие решения поставщиком относительно выполнения работ своими силами или с привлечением субподрядчика;
- разработку поставщиком плана управления проектом, содержащего организационную структуру проекта, разграничение ответ-

ственности, технические требования к среде разработки и ресурсам, управление субподрядчиками и др.

Процесс разработки (development process). Он предусматривает действия и задачи, выполняемые разработчиком, и охватывает работы по созданию ПО и его компонентов в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, подготовку материалов, необходимых для проверки работоспособности и соответствующего качества программных продуктов, материалов, необходимых для организации обучения персонала, и т. д. Процесс разработки включает следующие действия:

1. подготовительную работу;
2. анализ требований к системе;
3. проектирование архитектуры системы;
4. анализ требований к ПО;
5. проектирование архитектуры ПО;
6. детальное проектирование ПО;
7. кодирование и тестирование ПО;
8. интеграцию ПО;
9. квалификационное тестирование ПО;
10. интеграцию системы;
11. квалификационное тестирование системы;
12. установку ПО;
13. приемку ПО.

Подготовительная работа начинается с выбора модели ЖЦ ПО, соответствующей масштабу, значимости и сложности проекта (см. разд. 1.2). Действия и задачи процесса разработки должны соответствовать выбранной модели. Разработчик должен выбрать, адаптировать к условиям проекта и использовать согласованные с заказчиком стандарты, методы и средства разработки, а также составить план выполнения работ.

Анализ требований к системе подразумевает определение ее функциональных возможностей, пользовательских требований, требований к надежности и безопасности, требований к внешним интерфейсам и т. д. Требования к системе оцениваются исходя из критериев реализуемости и возможности проверки при тестировании.

Проектирование архитектуры системы на высоком уровне заключается в определении компонентов ее оборудования, ПО и операций, выполняемых эксплуатирующим систему персоналом. Архитектура системы должна соответствовать требованиям, предъявляемым к системе, а также принятым проектным стандартам и методам.

Анализ требований к ПО предполагает определение следующих характеристик для каждого компонента ПО:

- функциональных возможностей, включая характеристики производительности и среды функционирования компонента;
- внешних интерфейсов;
- спецификаций надежности и безопасности;
- эргономических требований;
- требований к используемым данным;
- требований к установке и приемке;
- требований к пользовательской документации;
- требований к эксплуатации и сопровождению.

Требования к ПО оцениваются исходя из критериев соответствия требованиям к системе, реализуемости и возможности проверки при тестировании.

Проектирование архитектуры ПО включает следующие задачи (для каждого компонента ПО):

- трансформацию требований к ПО в архитектуру, определяющую на высоком уровне структуру ПО и состав его компонентов;
- разработку и документирование программных интерфейсов ПО и баз данных;
- разработку предварительной версии пользовательской документации;
- разработку и документирование предварительных требований к тестам и плана интеграции ПО.

Архитектура компонентов. ПО должна соответствовать требованиям, предъявляемым к ним, а также принятым проектным стандартам и методам.

Детальное проектирование ПО включает следующие задачи:

- описание компонентов ПО и интерфейсов между ними на более низком уровне, достаточном для их последующего самостоятельного кодирования и тестирования;
- разработку и документирование детального проекта базы данных;
- обновление (при необходимости) пользовательской документации;
- разработку и документирование требований к тестам и плана тестирования компонентов ПО;
- обновление плана интеграции ПО.
- Кодирование и тестирование ПО охватывают следующие задачи:
- разработку (кодирование) и документирование каждого компонента ПО и базы данных, а также совокупности тестовых процедур и данных для их тестирования;
- тестирование каждого компонента ПО и базы данных на соответствие предъявляемым к ним требованиям. Результаты тестирования компонентов должны быть документированы;
- обновление (при необходимости) пользовательской документации;
- обновление плана интеграции ПО.

Интеграция ПО предусматривает сборку разработанных компонентов ПО в соответствии с планом интеграции и тестирование агрегированных компонентов. Для каждого из агрегированных компонентов разрабатываются наборы тестов и тестовые процедуры, предназначенные для проверки каждого из квалификационных требований при последующем квалификационном тестировании. *Квалификационное требование* — это набор критериев или условий, которые необходимо выполнить, чтобы квалифицировать программный продукт как соответствующий своим спецификациям и готовый к использованию в условиях эксплуатации.

Квалификационное тестирование ПО проводится разработчиком в присутствии заказчика (по возможности) для демонстрации того, что ПО удовлетворяет своим спецификациям и готово к использованию в условиях эксплуатации. Квалификационное тестирование выполняется для каждого компонента ПО по всем разделам требований при широком варьировании тестов. При этом также проверяются полнота технической и пользовательской документации и ее адекватность самим компонентам ПО.

Интеграция системы заключается в сборке всех ее компонентов, включая ПО и оборудование. После интеграции система, в свою очередь, подвергается *квалификационному тестированию* на соответствие совокупности требований к ней. При этом также производится оформление и проверка полного комплекта документации на систему.

Установка ПО осуществляется разработчиком в соответствии с планом в той среде и на том оборудовании, которые предусмотрены договором. В процессе установки проверяется работоспособность ПО и баз данных. Если устанавливаемое ПО заменяет существующую систему, разработчик должен обеспечить их параллельное функционирование в соответствии с договором.

Приемка ПО предусматривает оценку результатов квалификационного тестирования ПО и системы и документирование результатов оценки, которые проводятся заказчиком с помощью разработчика. Разработчик выполняет окончательную передачу ПО заказчику в соответствии с договором, обеспечивая при этом необходимое обучение и поддержку.

Процесс эксплуатации (operation process). Он охватывает действия и задачи оператора — организации, эксплуатирующей систему. Данный процесс включает следующие действия:

1. подготовительную работу;
2. эксплуатационное тестирование;
3. эксплуатацию системы;
4. поддержку пользователей.

Подготовительная работа включает проведение оператором следующих задач:

- планирование действий и работ, выполняемых в процессе эксплуатации, и установку эксплуатационных стандартов;
- определение процедур локализации и разрешения проблем, возникающих в процессе эксплуатации.

Эксплуатационное тестирование осуществляется для каждой очередной редакции программного продукта, после чего она передается в эксплуатацию.

Эксплуатация системы выполняется в предназначенной для этого среде в соответствии с пользовательской документацией.

Поддержка пользователей заключается в оказании помощи и консультаций при обнаружении ошибок в процессе эксплуатации ПО.

Процесс сопровождения (maintenance process). Он предусматривает действия и задачи, выполняемые сопровождающей организацией (службой сопровождения). Данный процесс активизируется **при** изменениях (модификациях) программного продукта и соответствующей документации, вызванных возникшими проблемами или потребностями в модернизации либо адаптации ПО. В соответствии со стандартом IEEE-90 под *сопровождением* понимается внесение изменений в ПО в целях исправления ошибок, повышения производительности или адаптации к изменившимся условиям работы или требованиям.

Изменения, вносимые в существующее ПО, не должны нарушать его целостность. Процесс сопровождения включает перенос ПО в другую среду (миграцию) и заканчивается снятием ПО с эксплуатации.

Процесс сопровождения охватывает следующие действия:

1. подготовительную работу;
2. анализ проблем и запросов на модификацию ПО;
3. модификацию ПО;
4. проверку и приемку;
5. перенос ПО в другую среду;
6. снятие ПО с эксплуатации.

Подготовительная работа службы сопровождения включает следующие задачи:

- планирование действий и работ, выполняемых в процессе сопровождения;
- определение процедур локализации и разрешения проблем, возникающих в процессе сопровождения.

Анализ проблем и запросов на модификацию ПО, выполняемый службой сопровождения, включает следующие задачи:

- анализ сообщения о возникшей проблеме или запроса на модификацию ПО относительно его влияния на организацию, существующую систему и интерфейсы с другими системами. При этом определяются следующие характеристики возможной модификации: тип (корректирующая, улучшающая, профилактическая или адаптирующая к новой среде); масштаб (размеры модификации, стоимость и время ее реализации); критичность (воздействие на производительность, надежность или безопасность);
- оценка целесообразности проведения модификации и возможных вариантов ее проведения;
- утверждение выбранного варианта модификации. *Модификация ПО* предусматривает определение компонентов ПО,

их версий и документации, подлежащих модификации, и внесение необходимых изменений в соответствии с правилами *процесса разработки*. Подготовленные изменения тестируются и проверяются по критериям, определенным в документации. При подтверждении корректности изменений в программах производится корректировка документации.

Проверка и приемка заключаются в проверке целостности модифицированной системы и утверждении внесенных изменений.

При *переносе ПО в другую среду* используются имеющиеся или разрабатываются новые средства переноса, затем выполняется конвертирование программ и данных в новую среду. С целью облегчить переход предусматривается параллельная эксплуатация ПО в старой и новой среде в течение некоторого периода, когда проводится необходимое обучение пользователей работе в новой среде.

Снятие ПО с эксплуатации осуществляется по решению заказчика при участии эксплуатирующей организации, службы сопровождения и пользователей. При этом программные продукты и соответствующая документация подлежат архивированию в соответствии с договором. Аналогично переносу ПО в другую среду с целью облегчить переход к новой системе предусматривается параллельная эксплуатация старого и нового ПО в течение некоторого периода, когда выполняется необходимое обучение пользователей работе с новой системой.

Основы сетевых технологий. Эталонная модель взаимосвязи открытых систем.

Развитие средств вычислительной техники, а особенно появление персональных компьютеров привело к созданию нового типа информационно-вычислительных систем под названием *локальная вычислительная сеть* (ЛВС).

ЛВС нашли широкое применение в системах автоматизированного проектирования и технологической подготовки производства, системах управления производством и технологическими комплексами, в конторских системах, бортовых системах управления и т.д. ЛВС является эффективным способом построения сложных систем управления различными производственными подразделениями. ЛВС интенсивно внедряются в медицину, сельское хозяйство, образование, науку и др.

Локальная сеть - (LAN - LocalAreaNetwork), данное название соответствует объединению компьютеров, расположенных на сравнительно небольшой территории (одного предприятия, офиса, одной комнаты). Существующие стандарты для ЛВС обеспечивают связь между компьютерами на расстоянии от 2,5 км до 6 км (Ethernet и ARCNET, соответственно).

ЛВС - набор аппаратных средств и алгоритмов, обеспечивающих соединение компьютеров, других периферийных устройств (принтеров, дисковых контроллеров и т.п.) и позволяющих им совместно использовать общую дисковую память, периферийные устройства, обмениваться данными.

В настоящее время информационно-вычислительные системы принято делить на 3 основных типа:

- **LAN** (LocalAreaNetwork) - локальная сеть в пределах предприятия, учреждения, одной организации;

- **MAN** (MetropolitanAreaNetwork) - городская или региональная сеть, т.е. сеть в пределах города, области и т.п.;
- **WAN** (WideAreaNetwork) - глобальная сеть, соединяющая абонентов страны, континента, всего мира.

Информационные системы, в которых средства передачи данных принадлежат одной компании и используются только для нужд этой компании, принято называть *Сеть Масштаба Предприятия* или *Корпоративная Сеть* (*EnterpriseNetwork*). Для автоматизации работы производственных предприятий часто используются системы на базе протоколов **MAP/TOP**:

MAP (ManufacturingAutomationProtocol) - сеть для производственных предприятий, заводов (выполняется автоматизация работы конструкторских отделов и производственных, технологических цехов). MAP позволяет создать единую технологическую цепочку от конструктора, разработавшего деталь, до оборудования, на котором изготавливают эту деталь.

TOP (TechnicalandOfficeProtocol) - протокол автоматизации технического и административного учреждения.

MAP/TOP системы, полностью автоматизирующие работу производственного предприятия.

Основное назначение ЛВС - в распределении ресурсов ЭВМ: программ, совместимости периферийных устройств, терминалов, памяти. Следовательно, ЛВС должна иметь надежную и быструю систему передачи данных, стоимость которой должна быть меньше по сравнению со стоимостью подключаемых рабочих станций. Иными словами, стоимость передаваемой единицы информации должна быть значительно ниже стоимости обработки информации в рабочих станциях. Исходя из этого ЛВС, как система распределенных ресурсов, должна основываться на следующих принципах:

- единой передающей среды;
- единого метода управления;
- единых протоколов;
- гибкой модульной организации;
- информационной и программной совместимости.

Международная организация по стандартизации (ISO), основываясь на опыте многомашинных систем, который был накоплен в разных странах, выдвинула концепцию архитектуры открытых систем - эталонную модель, используемую при разработке международных стандартов.

На основе этой модели вычислительная сеть предстает как распределенная вычислительная среда, включающая в себя большое число разнообразных аппаратных и программных средств.

По вертикали данная среда представляется рядом логических уровней, на каждый из которых возложена одна из задач сети.

По горизонтали информационно-вычислительная среда делится на локальные части (открытые системы), отвечающие требованиям и стандартам структуры открытых систем.

Часть открытой системы, выполняющая некоторую функцию и входящая в состав того или иного уровня, называется объектом.

Правила, по которым осуществляется взаимодействие объектов одного и того же уровня, называются протоколом (методика связи).

Протоколы определяют порядок обмена информацией между сетевыми объектами. Они позволяют взаимодействующим рабочим станциям посылать друг другу вызовы, интерпретировать данные, обрабатывать ошибочные ситуации и выполнять множество других различных функций. Суть протоколов заключается в регламентированных обменах точно специфицированными командами и ответами на них (например, назначение физического уровня связи - передача блоков данных между двумя устройствами, подключенными к одной физической среде).

Каждый уровень подразделяется на две части:

- спецификацию услуг;
- спецификацию протокола.

Спецификация услуг определяет, что делает уровень, а спецификация протокола - как он это делает.

Причем, каждый конкретный уровень может иметь более одного протокола. Большое число уровней, используемых в модели, обеспечивает декомпозицию информационно-вычислительного процесса на простые составляющие. В свою очередь, увеличение числа уровней вызывает необходимость включения

дополнительных связей в соответствии с дополнительными протоколами и интерфейсами. Интерфейсы (макрокоманды, программы) зависят от возможностей используемой ОС.

Международная организация по стандартизации предложила семиуровневую модель, которой соответствует и программная структура.

Рассмотрим функции, выполняемые каждым уровнем программного обеспечения:

1. Физический - осуществляет как соединения с физическим каналом, так и расторжение, управление каналом, а также определяется скорость передачи данных и топология сети.

2. Канальный - осуществляет обрамление передаваемых массивов информации вспомогательными символами и контроль передаваемых данных. В ЛВС передаваемая информация разбивается на несколько пакетов или кадров. Каждый пакет содержит адреса источника и места назначения, а также средства обнаружения ошибок.

3. Сетевой - определяет маршрут передачи информации между сетями (ПЭВМ), обеспечивает обработку ошибок, а так же управление потоками данных. Основная задача сетевого уровня - маршрутизация данных (передача данных между сетями). Специальные устройства - *Маршрутизаторы (Router)* определяют для какой сети предназначено то или другое сообщение, и направляет эту посылку в заданную сеть. Для определения абонента внутри сети используется *Адрес Узла (NodeAddress)*. Для определения пути передачи данных между сетями на маршрутизаторах строятся *Таблицы Маршрутов (RoutingTables)*, содержащие последовательность передачи данных через маршрутизаторы. Каждый маршрут содержит адрес конечной сети, адрес следующего маршрутизатора и стоимость передачи данных по этому маршруту. При оценке стоимости могут учитываться количество промежуточных маршрутизаторов, время, необходимое на передачу данных, просто денежная стоимость передачи данных по линии связи. Для построения таблиц маршрутов наиболее часто используют либо *Метод Векторов* либо *Статический Метод*. При выборе оптимального маршрута применяют динамические или статические методы. На сетевом уровне возможно применение одной из двух процедур передачи пакетов:

- датаграмм - т.е., когда часть сообщения или пакет независимо доставляется адресату по различным маршрутам, определяемым сложившейся динамикой в

сети. При этом каждый пакет включает в себя полный заголовок с адресом получателя. Процедуры управления передачей таких пакетов по сети называются датаграммой службой;

· виртуальных соединений - когда установление маршрута передачи всего сообщения от отправителя до получателя осуществляется с помощью специального служебного пакета - запроса на соединение. В таком случае для этого пакета выбирается маршрут и, при положительном ответе получателя на соединение закрепляется для всего последующего трафика (потока сообщений в сети передачи данных) и получает номер соответствующего виртуального канала (соединения) для дальнейшего использования его другими пакетами того же сообщения. Пакеты, которые передаются по одному виртуальному каналу, не являются независимыми и поэтому включают сокращенный заголовок, включающий порядковый номер пакета, принадлежащему одному сообщению.

Недостатки: значительная по сравнению с датаграммой сложность в реализации, увеличение накладных расходов, вызванных установлением и разъединением сообщений.

ВЫВОД. Датаграммный режим предпочтительнее использовать для сетей сложной конфигурации, где значительное число ЭВМ в сети, иерархическая структура сети, надежность, достоверность передачи данных по каналам связи, длина пакета более 512 байт.

4. Транспортный - связывает нижние уровни (физический, канальный, сетевой) с верхними уровнями, которые реализуются программными средствами. Этот уровень как бы разделяет средства формирования данных в сети от средств их передачи. Здесь осуществляется разделение информации по определенной длине и уточняется адрес назначения. Транспортный уровень позволяет мультиплексировать передаваемые сообщения или соединения.

Мультиплексирование сообщений позволяет передавать сообщения одновременно по нескольким линиям связи, а мультиплексирование соединений - передает в одной посылке несколько сообщений для различных соединений.

5. Сеансовый - на данном уровне осуществляется управление сеансами связи между двумя взаимодействующими пользователями (определяет начало и окончание сеанса связи: нормальное или аварийное; определяет время, длительность и режим сеанса связи; определяет точки синхронизации для промежуточного контроля и восстановления при передаче данных;

восстанавливает соединение после ошибок во время сеанса связи без потери данных.

6.Представительский - управляет представлением данных в необходимой для программы пользователя форме, генерацию и интерпретацию взаимодействия процессов, кодирование/декодирование данных, в том числе компрессию и декомпрессию данных. На рабочих станциях могут использоваться различные операционные системы : DOS, UNIX, OS/2. Каждая из них имеет свою файловую систему, свои форматы хранения и обработки данных. Задачей данного уровня является преобразование данных при передаче информации в формат, который используется в информационной системе. При приеме данных данный уровень представления данных выполняет обратное преобразование. Таким образом появляется возможность организовать обмен данными между станциями, на которых используются различные операционные системы. Форматы представления данных могут различаться по следующим признакам:

- порядок следования битов и размерность символа в битах;
- порядок следования байтов;
- представление и кодировка символов;
- структура и синтаксис файлов.

Компрессия или упаковка данных сокращает время передачи данных.

Кодирование передаваемой информации обеспечивает защиту ее от перехвата.

7.Прикладной - в его ведении находятся прикладные сетевые программы, обслуживающие файлы, а также выполняет вычислительные, информационно-поисковые работы, логические преобразования информации, передачу почтовых сообщений и т.п. Главная задача этого уровня - обеспечить удобный интерфейс для пользователя. На разных уровнях обмен происходит различными единицами информации: биты, кадры, пакеты, сеансовые сообщения, пользовательские сообщения.

Вопросы:

1. Что такое локальная вычислительная сеть?
2. На какие типы принято делить информационно-вычислительные системы?
3. Что такое корпоративная сеть?

4. Какие протоколы используются для автоматизации работы производственных предприятий?
5. Концепция архитектуры открытых систем?
6. Опишите функции, выполняемые каждым уровнем программного обеспечения ЛВС?

Стандартные сетевые протоколы

Протоколы – это набор правил и процедур, регулирующих порядок осуществления связи. Компьютеры, участвующие в обмене, должны работать по одним и тем же протоколам, чтобы в результате передачи вся информация восстанавливалась в первоначальном виде.

О протоколах нижних уровней (физического и канального), относящихся к аппаратуре, уже упоминалось в предыдущих разделах. В частности, к ним относятся методы кодирования и декодирования, а также управления обменом в сети. Подробнее некоторые из них будут изложены в главах книги, посвященных стандартным сетям. А сейчас следует остановиться на особенностях протоколов более высоких уровней, реализуемых программно.

Связь сетевого адаптера с сетевым программным обеспечением осуществляют драйверы сетевых адаптеров. Именно благодаря драйверу компьютер может не знать никаких аппаратных особенностей адаптера (его адресов, правил обмена с ним, его характеристик). Драйвер унифицирует, делает единообразным взаимодействие программных средств высокого уровня с любым адаптером данного класса. Сетевые драйверы, поставляемые вместе с сетевыми адаптерами, позволяют сетевым программам одинаково работать с платами разных поставщиков и даже с платами разных локальных сетей (Ethernet, Arcnet, Token-Ring и т.д.). Если говорить о стандартной модели OSI, то драйверы, как правило, выполняют функции канального уровня, хотя иногда они реализуют и часть функций сетевого уровня (рис. 6.1). Например, драйверы формируют передаваемый пакет в буферной памяти адаптера, читают из этой памяти пришедший по сети пакет, дают команду на передачу, информируют компьютер о приеме пакета.

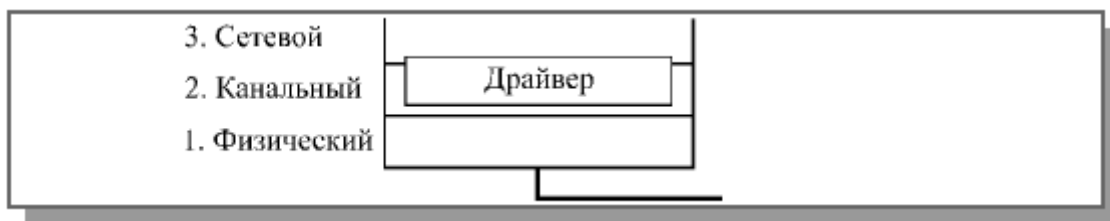


Рис. 6.1. Функции драйвера сетевого адаптера в модели OSI

Качество написания программы драйвера во многом определяет эффективность работы сети в целом. Даже при самых лучших характеристиках сетевого адаптера некачественный драйвер может резко ухудшить обмен по сети.

Прежде чем приобрести плату адаптера, необходимо ознакомиться со списком совместимого оборудования (HardwareCompatibilityList, HCL), который публикуют все производители сетевых операционных систем. Выбор там довольно велик (например, для Microsoft Windows Server список включает более сотни драйверов сетевых адаптеров). Если в перечень HCL не входит адаптер какого-то типа, лучше его не покупать.

Протоколы высоких уровней.

Существует несколько стандартных наборов (или, как их еще называют, стеков) протоколов, получивших сейчас широкое распространение:

- набор протоколов ISO/OSI;
- IBM System Network Architecture (SNA);
- DigitalDECnet;
- NovellNetWare;
- AppleAppleTalk;
- набор протоколов глобальной сети Интернет, TCP/IP.

Включение в этот список протоколов глобальной сети вполне объяснимо, ведь, как уже отмечалось, модель OSI используется для любой открытой системы: на базе как локальной, так и глобальной сети или комбинации локальной и глобальной сетей.

Протоколы перечисленных наборов делятся на три основных типа:

- Прикладные протоколы (выполняющие функции трех верхних уровней модели OSI – прикладного, представительского и сеансового);
- Транспортные протоколы (реализующие функции средних уровней модели OSI – транспортного и сеансового);
- Сетевые протоколы (осуществляющие функции трех нижних уровней модели OSI).

Прикладные протоколы обеспечивают взаимодействие приложений и обмен данными между ними. Наиболее популярны:

- FTAM (File Transfer Access and Management) – протокол OSI доступа к файлам;
- X.400 – протокол CCITT для международного обмена электронной почтой;

- X.500 – протокол CCITT служб файлов и каталогов на нескольких системах;
- SMTP (SimpleMailTransferProtocol) – протокол глобальной сети Интернет для обмена электронной почтой;
- FTP (FileTransferProtocol) – протокол глобальной сети Интернет для передачи файлов;
- SNMP (SimpleNetworkManagementProtocol) – протокол для мониторинга сети, контроля за работой сетевых компонентов и управления ими;
- Telnet – протокол глобальной сети Интернет для регистрации на удаленных серверах и обработки данных на них;
- MicrosoftSMBs (ServerMessageBlocks, блоки сообщений сервера) и клиентские оболочки или редиректоры фирмы Microsoft;
- NCP (NovellNetWareCoreProtocol) и клиентские оболочки или редиректоры фирмы Novell.

Транспортные протоколы поддерживают сеансы связи между компьютерами и гарантируют надежный обмен данными между ними. Наиболее популярные из них следующие:

- TCP (TransmissionControlProtocol) – часть набора протоколов TCP/IP для гарантированной доставки данных, разбитых на последовательность фрагментов;
- SPX – часть набора протоколов IPX/SPX (InternetworkPacketExchange/SequentialPacketExchange) для гарантированной доставки данных, разбитых на последовательность фрагментов, предложенных компанией Novell;
- NWLink – реализация протокола IPX/SPX компании Microsoft;
- NetBEUI – (NetBIOSExtendedUserInterface, расширенный интерфейс NetBIOS) – устанавливает сеансы связи между компьютерами (NetBIOS) и предоставляет верхним уровням транспортные услуги (NetBEUI).

Сетевые протоколы управляют адресацией, маршрутизацией, проверкой ошибок и запросами на повторную передачу. Широко распространены следующие из них:

- IP (InternetProtocol) – TCP/IP-протокол для негарантированной передачи пакетов без установления соединений;
- IPX (InternetworkPacketExchange) – протокол компании NetWare для негарантированной передачи пакетов и маршрутизации пакетов;
- NWLink – реализация протокола IPX/SPX компании Microsoft;
- NetBEUI – транспортный протокол, обеспечивающий услуги транспортировки данных для сеансов и приложений NetBIOS.

Все перечисленные протоколы могут быть поставлены в соответствие тем или иным уровням эталонной модели OSI. Но при этом надо учитывать, что

разработчики протоколов не слишком строго придерживаются этих уровней. Например, некоторые протоколы выполняют функции, относящиеся сразу к нескольким уровням модели OSI, а другие – только часть функций одного из уровней. Это приводит к тому, что протоколы разных компаний часто оказываются несовместимы между собой. Кроме того, протоколы могут быть успешно использованы исключительно в составе своего набора протоколов (стека протоколов), который выполняет более или менее законченную группу функций. Как раз это и делает сетевую операционную систему "фирменной", то есть, по сути, несовместимой со стандартной моделью открытой системы OSI.

В качестве примера на рис. 6.2, рис. 6.3 и рис. 6.4 схематически показано соотношение протоколов, используемых популярными фирменными сетевыми операционными системами, и уровней стандартной модели OSI. Как видно из рисунков, практически ни на одном уровне нет четкого соответствия реального протокола какому-нибудь уровню идеальной модели. Выстраивание подобных соотношений довольно условно, так как трудно четко разграничить функции всех частей программного обеспечения. К тому же компании-производители программных средств далеко не всегда подробно описывают внутреннюю структуру продуктов.

Теперь следует подробнее рассмотреть некоторые наиболее распространенные протоколы.

Модель OSI допускает два основных метода взаимодействия абонентов в сети:

- Метод взаимодействия без логического соединения (или метод дейтаграмм).
- Метод взаимодействия с логическим соединением.

Метод дейтаграмм – это простейший метод, в котором каждый пакет рассматривается как самостоятельный объект (рис. 6.5).

Пакет при этом методе передается без установления логического канала, то есть без предварительного обмена служебными пакетами для выяснения готовности приемника, а также без ликвидации логического канала, то есть без пакета подтверждения окончания передачи. Дойдет пакет до приемника или нет – неизвестно (проверка факта получения переносится на более высокие уровни).

Метод дейтаграмм предъявляет повышенные требования к аппаратуре (так как приемник всегда должен быть готов к приему пакета). Достоинства метода в том, что передатчик и приемник работают независимо друг от друга, к тому же пакеты могут накапливаться в буфере и затем передаваться вместе, можно также использовать широковещательную передачу, то есть адресовать пакет всем абонентам одновременно. Недостатки метода – это возможность потери пакетов, а также бесполезной загрузки сети пакетами в случае отсутствия или неготовности приемника.

Метод с логическим соединением (рис. 6.6, рис. 4.5) разработан позднее, чем метод дейтаграмм, и отличается усложненным порядком взаимодействия.

При этом методе пакет передается только после того, как будет установлено логическое соединение (канал) между приемником и передатчиком. Каждому информационному пакету сопутствует один или несколько служебных пакетов (установка соединения, подтверждение получения, запрос повторной передачи, разрыв соединения). Логический канал может устанавливаться на время передачи одного или нескольких пакетов.



Рис. 6.2. Соотношение уровней модели OSI и протоколов сети Интернет

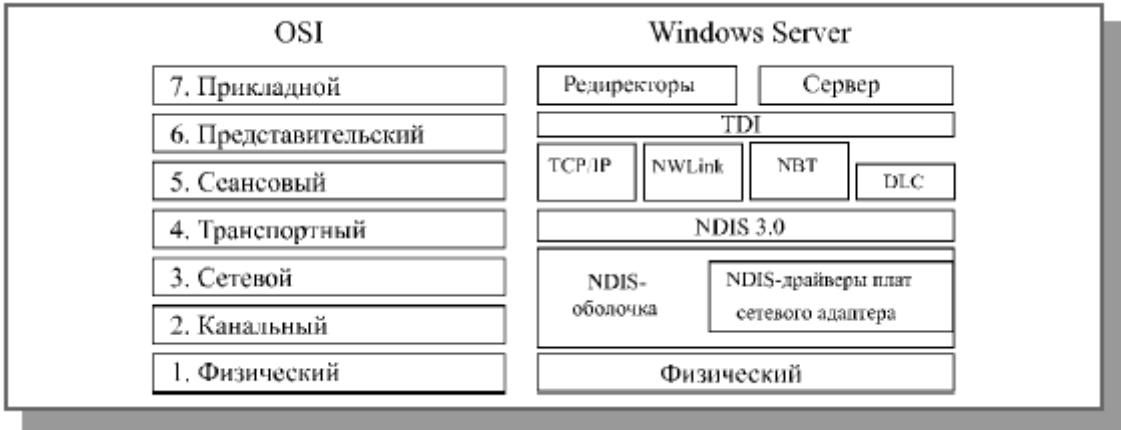


Рис. 6.3. Соотношение уровней модели OSI и протоколов операционной системы Windows Server

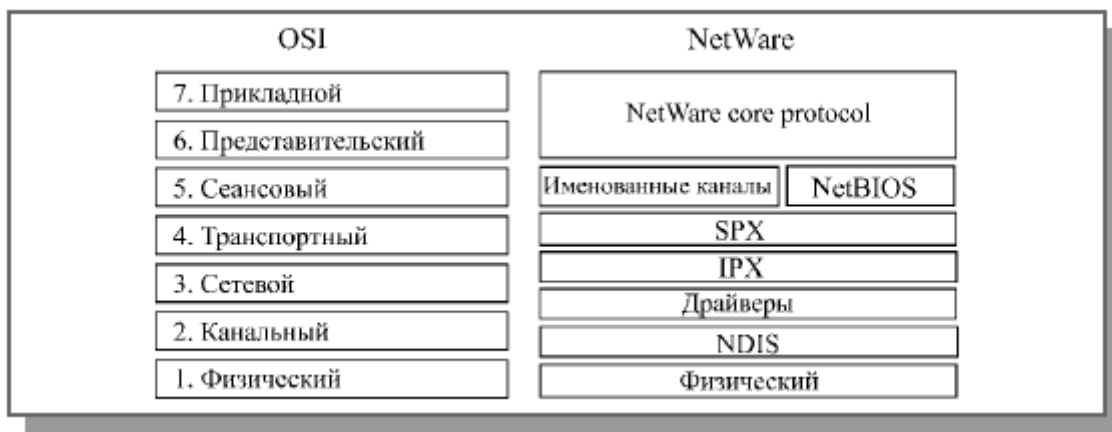


Рис. 6.4. Соотношение уровней модели OSI и протоколов операционной системы NetWare



Рис. 6.5. Метод дейтаграмм

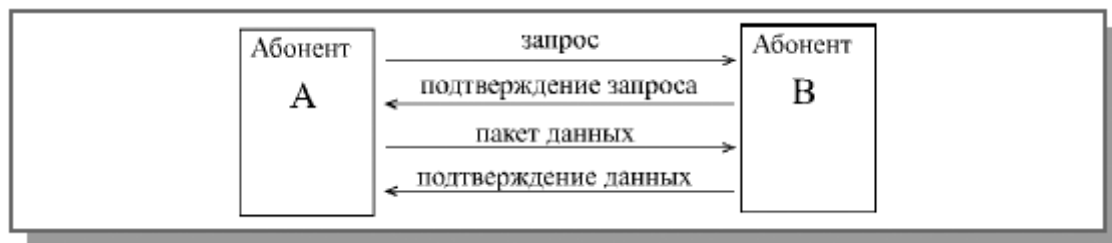


Рис. 6.6. Метод с логическим соединением

Метод с логическим соединением, как уже говорилось, более сложен, чем метод дейтаграмм, но гораздо надежнее, поскольку к моменту ликвидации логического канала передатчик уверен, что все его пакеты дошли до места назначения, причем дошли успешно. Не бывает при данном методе и перегрузки сети из-за бесполезных пакетов. Недостаток метода с логическим соединением состоит в том, что довольно сложно разрешить ситуацию, когда принимающий абонент по тем или иным причинам не готов к обмену, например, из-за обрыва кабеля, отключения питания, неисправности сетевого оборудования, сбоя в компьютере. При этом требуется алгоритм обмена с повторением неподтвержденного пакета заданное количество раз, причем важен и тип неподтвержденного пакета. Не может этот метод передавать широковещательные пакеты (то есть адресованные

всем абонентам), так как нельзя организовать логические каналы сразу со всеми абонентами.

Примеры протоколов, работающих по методу дейтаграмм— это протоколы IP и IPX.

Примеры протоколов, работающих по методу с логическим соединением — это TCP и SPX.

Именно для того, чтобы объединить достоинства обоих методов, эти протоколы используются в виде связанных наборов: TCP/IP и IPX/SPX, в которых протокол более высокого уровня (TCP, SPX), работающий на базе протокола более низкого уровня (IP, IPX), гарантирует правильную доставку пакетов в требуемом порядке.

Протоколы IPX/SPX, разработанные компанией Novell, образуют набор (стек), используемый в сетевых программных средствах довольно широко распространенных локальных сетей Novell (NetWare). Это сравнительно небольшой и быстрый протокол, поддерживающий маршрутизацию. Прикладные программы могут обращаться непосредственно к уровню IPX, например, для отправки широковещательных сообщений, но значительно чаще работают с уровнем SPX, гарантирующим быструю и надежную доставку пакетов. Если скорость не слишком важна, то прикладные программы применяют еще более высокий уровень, например, протокол NetBIOS, предоставляющий удобный сервис. Компанией Microsoft предложена своя реализация протокола IPX/SPX, называемая NWLink. Протоколы IPX/SPX и NWLink поддерживаются операционными системами NetWare и Windows. Выбор этих протоколов обеспечивает совместимость по сети любых абонентов с данными операционными системами.

Набор (стек) протоколов TCP/IP был специально разработан для глобальных сетей и для межсетевого взаимодействия. Он изначально ориентирован на низкое качество каналов связи, на большую вероятность ошибок и разрывов связей. Этот протокол принят во всемирной компьютерной сети Интернет, значительная часть абонентов которой подключается по коммутируемым линиям (то есть обычным телефонным линиям). Как и протокол IPX/SPX, протокол TCP/IP также поддерживает маршрутизацию. На его основе работают протоколы высоких уровней, такие как SMTP, FTP, SNMP. Недостаток протокола TCP/IP — более низкая скорость работы, чем у IPX/SPX. Однако сейчас протокол TCP/IP используется и в локальных сетях, чтобы упростить согласование протоколов локальных и глобальных сетей. В настоящее время он считается основным в самых распространенных операционных системах.

В стек протоколов TCP/IP часто включают и протоколы всех верхних уровней (рис. 6.7). И тогда уже можно говорить о функциональной полноте стека TCP/IP.

Как протокол IPX, так и протокол IP являются самыми низкоуровневыми протоколами, поэтому они непосредственно инкапсулируют свою информацию, называемую дейтаграммой, в поле данных передаваемого по сети пакета (см. рис. 4.6). При этом в заголовок дейтаграммы входят адреса абонентов (отправителя и получателя) более высокого уровня, чем MAC-адреса, – это IPX-адреса для протокола IPX или IP-адреса для протокола IP. Эти адреса включают номера сети и узла, хоста (индивидуальный идентификатор абонента). При этом IPX-адреса (рис. 6.8) более простые, имеют всего один формат, а в IP-адрес (рис. 6.9) могут входить три формата (класса А, В и С), различающиеся значениями трех начальных битов.



Рис. 6.7. Соотношение уровней модели OSI и стека протоколов TCP/IP

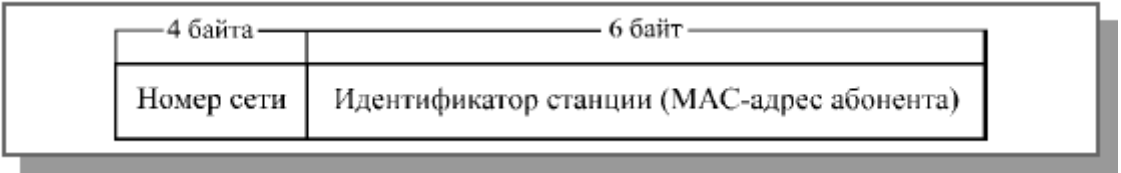


Рис. 6.8. Формат IPX- адреса

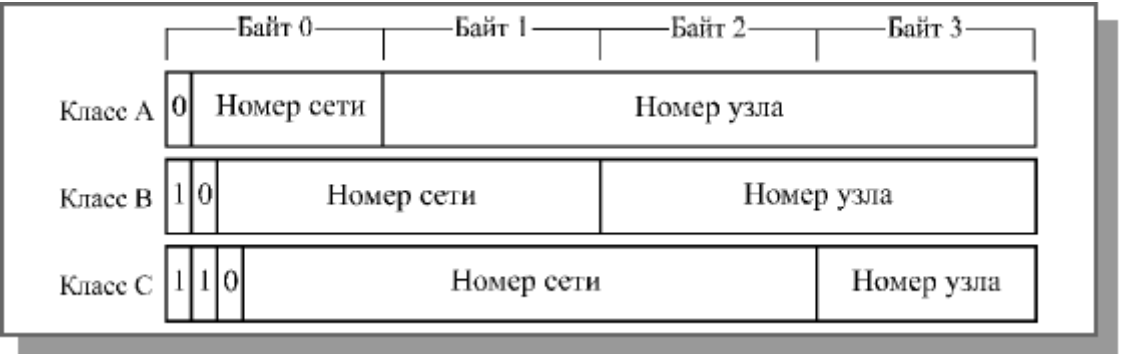


Рис. 6.9. Форматы IP-адреса

Интересно, что IP-адрес не имеет никакой связи с MAC-адресами абонентов. Номер узла в нем присваивается абоненту независимо от его MAC-адреса. В качестве идентификатора станции IPX-адрес включает в себя полный MAC-адрес абонента.

Номер сети – это код, присвоенный каждой конкретной сети, то есть каждой ширококвещательной области общей, единой сети. Под ширококвещательной областью понимается часть сети, которая прозрачна для ширококвещательных пакетов, пропускает их беспрепятственно.

Протокол NetBIOS (сетевая базовая система ввода/вывода) был разработан компанией IBM для сетей IBM PC Network и IBM Token-Ring по образцу системы BIOS персонального компьютера. С тех пор этот протокол стал фактическим стандартом (официально он не стандартизован), и многие сетевые операционные системы содержат в себе эмулятор NetBIOS для обеспечения совместимости. Первоначально NetBIOS реализовывал сеансовый, транспортный и сетевой уровни, однако в последующих сетях на более низких уровнях используются стандартные протоколы (например, IPX/SPX), а на долю эмулятора NetBIOS остается только сеансовый уровень. NetBIOS обеспечивает более высокий уровень сервиса, чем IPX/SPX, но работает медленнее.

На основе протокола NetBIOS был разработан протокол NetBEUI, который представляет собой развитие протокола NetBIOS до транспортного уровня. Однако недостаток NetBEUI состоит в том, что он не поддерживает межсетевое взаимодействие и не обеспечивает маршрутизацию. Поэтому данный протокол используется только в простых сетях, не рассчитанных на подключение к Интернет. Сложные сети ориентируются на более универсальные протоколы TCP/IP и IPX/SPX. Протокол NetBEUI в настоящее время считается устаревшим, хотя даже в операционной системе Windows XP предусмотрена его поддержка, правда, только как дополнительная опция.

Наконец, упоминавшийся уже набор протоколов OSI – это полный набор (стек) протоколов, где каждый протокол точно соответствует определенному уровню стандартной модели OSI. Набор содержит маршрутизируемые и транспортные протоколы, серии протоколов IEEE 802, протокол сеансового уровня, представительского уровня и несколько протоколов прикладного уровня. Пока широкого распространения этот набор протоколов не получил, хотя он и полностью соответствует эталонной модели OSI.

Конфигурация сети

Все сети имеют некоторые общие компоненты, функции и характеристики. В их числе:

- **серверы**, компьютеры, предоставляющие свои ресурсы сетевым пользователям. Серверы могут быть *выделенными* или *невыделенными*. *Выделенный* сервер - это компьютер, который используется только как сервер. *Невыделенный* сервер может работать одновременно и как сервер, и как рабочая станция;
- **клиенты** (рабочие станции) – компьютеры, осуществляющие доступ к сетевым ресурсам, предоставляемым серверами. Назначение рабочей станции - выполнять программы, полученные по сети, а назначение сервера - доставлять эти программы рабочей станции. Такую схему работы часто называют *распределенной обработкой*. Существуют программы, распределяющие нагрузку между рабочими станциями и сервером, более равномерно. Такую схему обработки называют системой *клиент - сервер*. В ней рабочая станция (клиент) и сервер делят между собой задачи обработки с целью повышения производительности.
- **среда передачи** – сетевая коммуникационная система. Наиболее часто в локальных сетях используются следующие типы линий связи или, как часто называют, сред передачи информации: электрический кабель (витая пара проводов), оптоволоконный (он же волоконно-оптический), радиоканал и инфракрасный канал (последние два не требуют проводов);
- **совместно используемые данные** – файлы, предоставляемые серверами по сети;
- **совместно используемые периферийные устройства**, например принтеры, библиотеки CD-ROM и т.д. – ресурсы, предоставляемые серверами;
- **ресурсы** – файлы, периферийные устройства и другие элементы, используемые в сети.

Выбор типа сети зависит от следующих факторов:

- размера предприятия;
- необходимой степени безопасности;
- вида деятельности;
- доступности административной поддержки;
- объема сетевого трафика;
- потребностей сетевых пользователей;
- уровня финансирования.

Большинство сетей работают на основе выделенного сервера. Выделенный сервер предназначен для быстрой обработки запросов от сетевых клиентов и для повышения защищенности файлов и каталогов. Сеть на основе сервера показана на рис.1.5.

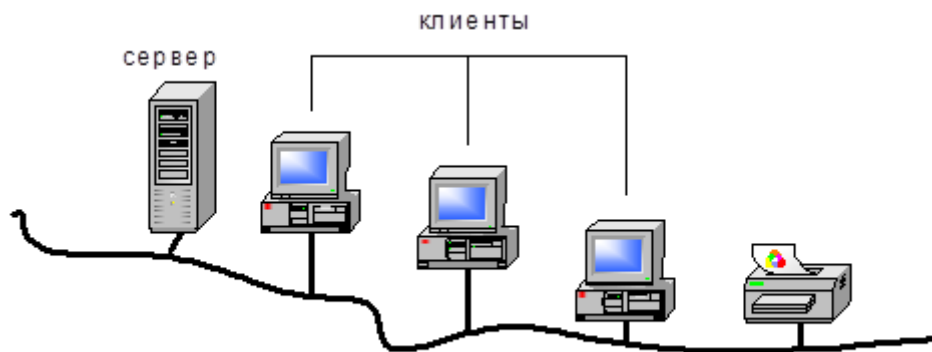


Рисунок 1.5. Сеть на основе сервера

При увеличении размера сети и объема сетевого трафика увеличивают количество серверов. В одной сети могут работать один или несколько серверов, например, сервер файлов и печати, сервер приложений, почтовый сервер и коммуникационный сервер.

Для нормальной работы сети требуется специальная сетевая операционная система, например Novell NetWare или Windows Server. Эта операционная система предназначена для выполнения двух основных задач: для управления работой файлового сервера и для совместной работы с операционной системой рабочей станции. Кроме этого сетевая операционная система управляет защитой файлового сервера от несанкционированного доступа, а также управляет правами доступа пользователя к сети.

Операционные системы для сетей на основе сервера в этом случае отвечают за совместное использование основных приложений и данных. Компьютеры-клиенты используют ресурсы выделенных серверов и в то же время – по мере необходимости – предоставляют в совместное использование свои жесткие диски.

Топология сети

При объединении в сеть большого числа компьютеров в первую очередь необходимо выбрать способ организации физических связей, то есть, топологию.

Под топологией вычислительной сети понимается конфигурация графа, вершинам которого соответствуют компьютеры сети (иногда и другое оборудование, например коммутаторы), а ребрам – физические связи между ними. Характеристики сети зависят от типа устанавливаемой топологии. В частности, выбор той или иной топологии влияет на:

- состав необходимого сетевого оборудования;
- возможности сетевого оборудования;
- возможности расширения сети;
- способ управления сетью.

Например, наличие резервных связей повышает надежность сети, простота присоединения новых узлов делает сеть легко расширяемой и т.д.

Чтобы совместно использовать ресурсы сети или выполнять другие сетевые задачи, компьютеры должны быть подключены друг к другу. Для этих целей в большинстве сетей применяется кабель. Однако просто подключить компьютер к кабелю, соединяющему другие компьютеры, недостаточно. Различные типы кабелей в сочетании с различными сетевыми платами, сетевыми ОС и другими компонентами требуют и различных методов реализации.

Топология также определяет и способ взаимодействия компьютеров в сети. Различным видам топологий соответствуют различные методы взаимодействия, влияющие на работу сети.

Рассмотрим некоторые наиболее встречающиеся топологии

Базовые топологии

Все сети строятся на основе трех базовых топологий, известных как:

- шина (bus);
- звезда (star);
- кольцо (ring);
- ячеистая.

Если компьютеры подключены вдоль одного кабеля (сегмента), топология называется **«шиной»**. В том случае, когда компьютеры подключены к сегментам кабеля, исходящим из одной точки (коммутатора), топология называется **«звездой»**. Если кабель, к которому подключены компьютеры, замкнут в кольцо, такая топология носит название **«кольцо»**, а если все компьютеры соединены друг с другом отдельными кабелями, топология называется **ячеистой**.

Сами по себе базовые топологии несложны, однако на практике часто встречаются довольно сложные их комбинации, сочетающие свойства и характеристики нескольких топологий.

Топология «**шина**» часто называемая «**общая шина**» (рис.1.6) является наиболее простой и распространенной топологией для локальных сетей. В ней используется один кабель, именуемый магистралью, к которому подключены все компьютеры сети. Концы кабеля не соединяются друг с другом, а заканчиваются заглушками (терминаторами).

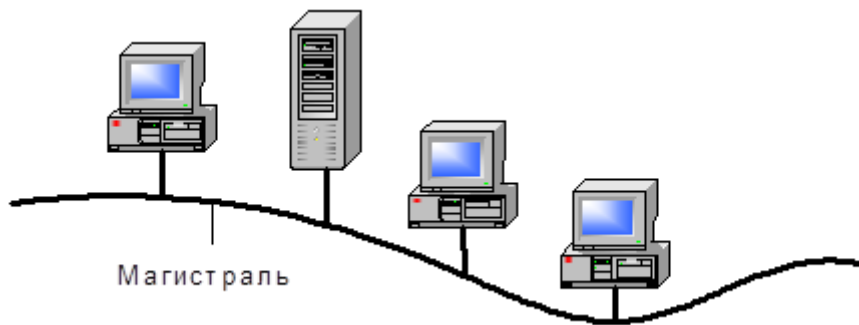


Рисунок 1.6. Сеть с топологией «шина»

В сетях, построенных по топологии «шина», информация может распространяться в обе стороны. Компьютеры адресуют данные конкретному компьютеру, передавая их по кабелю в виде электрических сигналов. Данные передаются всем компьютерам сети; однако информацию принимает только тот компьютер, чей адрес совпадает с адресом получателя. В каждый момент времени данные в сеть передаются лишь одним компьютером, поэтому производительность сети зависит от количества компьютеров, подключенных к шине. Чем больше компьютеров, тем большее их число ожидает передачи, и тем медленнее сеть. Кроме этого на быстродействие сети влияет и множество других факторов, например:

- тип аппаратного обеспечения сетевых компьютеров;
- частота, с которой компьютеры передают данные;
- тип работающих сетевых приложений;
- тип сетевого кабеля;
- расстояние между компьютерами в сети.

Компьютеры в сети с топологией «шина» только «слушают» передаваемые по сети данные, но не перемещают их от отправителя к получателю. Поэтому, если какой-либо компьютер выйдет из строя, это не скажется на работе сети. При

разрыве сетевого кабеля или отсоединении одного из его концов компьютеры не смогут взаимодействовать друг с другом, хотя и будут работоспособными.

Для расширения сети с топологией «шина» кабель обычно удлиняют двумя способами.

1. Для соединения двух отрезков кабеля используют баррел-коннектор. Однако лучше использовать один длинный кабель, чем соединять несколько коротких: при большом количестве «стыковок» происходит искажение сигнала.

2. Для соединения двух отрезков кабеля используют повторитель. В отличие от коннектора, повторитель усиливает сигнал перед подачей его в следующий сегмент. Поэтому его лучше использовать, чем баррел-коннектор или даже длинный кабель: сигналы на большие расстояния передаются без искажения. В настоящее время для построения сетей эта топология не используется.

При топологии «звезда» (рис.1.7) все компьютеры с помощью сегментов кабеля подключаются к центральному устройству, называемому соединительным модулем коммутатором. Сигналы от передающего компьютера поступают через коммутатор к одному или ко всем остальным компьютерам сети.

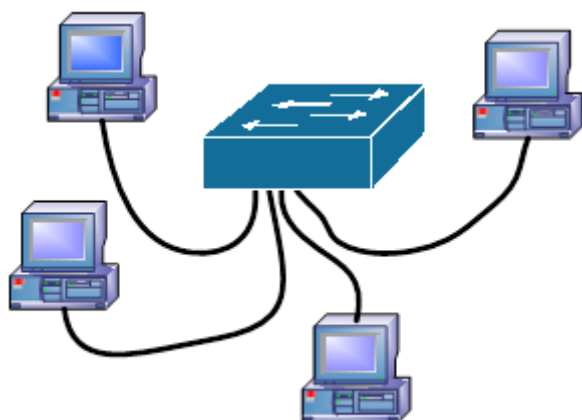


Рисунок 1.7. Сеть с топологией «звезда»

Главное преимущество такой топологии перед «шиной» – большая надежность. При выходе из строя одного из компьютеров или кабеля, соединяющего его с коммутатором, оставшаяся часть сети продолжает работать. Только отказ коммутатора выводит из строя всю сеть.

К недостаткам топологии относится более высокая стоимость сетевого оборудования из-за необходимости приобретения коммутаторов. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов коммутатора. Поэтому, имеет смысл, строить сеть с использованием нескольких коммутаторов, соединенных между собой связями типа звезда (рис.1.8). В настоящее время такая топология является самой распространенной как в локальных, так и в глобальных сетях.

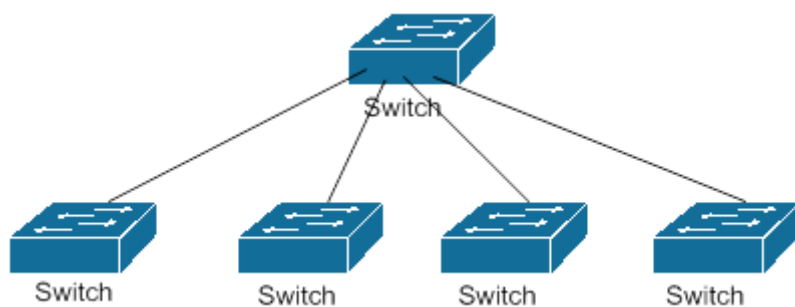


Рисунок 1.8. Соединение коммутаторов связями типа «звезда»

При топологии «кольцо» (рис.1.9) компьютеры подключаются к общему кабелю, концы которого соединены друг с другом в виде кольца. Сигналы передаются по кольцу в одном направлении и проходят через каждый компьютер. При такой схеме соединения каждый компьютер выступает в роли повторителя, усиливая сигналы и передавая их следующему компьютеру. Поэтому, если выйдет из строя один компьютер, прекращает функционировать вся сеть.

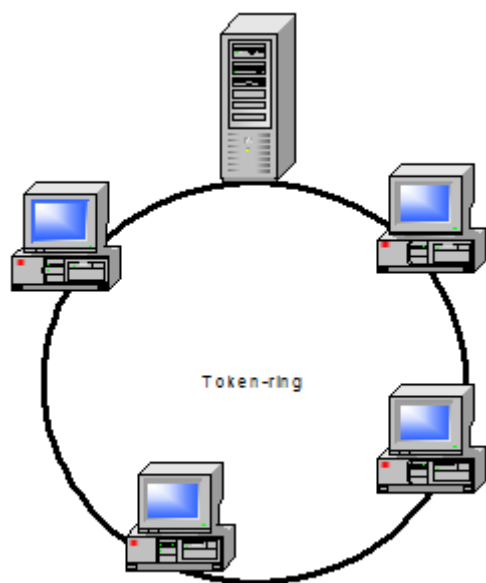


Рисунок 1.9. Сеть с топологией «кольцо»

Один из способов передачи данных по кольцевой сети называется передачей маркера. Маркер – это специальная последовательность бит, передающаяся по сети; в каждой сети существует только один маркер. Маркер последовательно, от одного компьютера к другому, передается до тех пор, пока его не получит тот компьютер, который «хочет» послать данные. Передающий компьютер видоизменяет маркер, добавляет к нему данные и адрес получателя и отправляет его дальше по «кольцу».

Данные проходят через каждый компьютер до тех пор, пока не попадают к тому, чей адрес совпадает с адресом получателя.

После этого принимающий компьютер посылает передающему сообщение, в котором подтверждает факт приема данных. Получив подтверждение, передающий компьютер создает новый маркер и возвращает его в сеть.

Сеть с «ячеистой» топологией обладает высокой избыточностью и надежностью, так как каждый компьютер такой сети соединен с любым другим отдельным кабелем (рис.1.10). Сигнал от компьютера-отправителя до компьютера-получателя может проходить по разным маршрутам, поэтому разрыв кабеля не сказывается на работоспособности сети. Основной недостаток такой топологии – большие затраты на прокладку кабеля, что компенсируется высокой надежностью и простотой обслуживания. Ячеистая топология применяется, как правило, в комбинации с другими топологиями при построении относительно больших сетей, таких как FrameRelay или ATM, где требуется большая производительность и надежность.

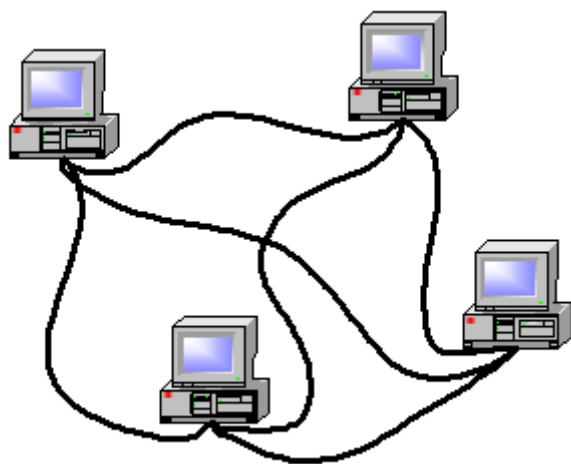


Рисунок 1.10. Сеть с ячеистой топологией

Маршрутная политика

Содержанием политики маршрутизации являются правила обмена маршрутной информацией между автономными системами (RIPE-181.txt). Не следует путать "маршрутную политику" и просто "политику", между ними такое же различие, как между "милостивым государем" и "государем". Способы их описания разнятся столь же значительно. При описании обычной политики одной из главных задач является сокрытие истинных намерений, а одним из средств - многословие. При описании же маршрутной политики важна лаконичность и четкость. В Интернет для решения этой задачи выработан стандарт, краткое изложение которого на

конкретных примерах будет приведено ниже. Объектами маршрутной политики являются автономные системы (AUT-NUM) и маршруты (route). Существует два акта маршрутной политики:

оповещение (announce)

и

восприятие (ассепт).

Эти акты определяют взаимодействие с ближайшими соседями. Совокупность информации, выданной всеми маршрутизаторами региональной сети, описывает ее граф. Следует иметь в виду, что в пределах автономной системы (AS) может работать только один внутренний протокол маршрутизации (IGP), а обмен маршрутной информацией между автономными системами происходит в соответствии с внешним протоколом маршрутизации (EGP). Эта идея продемонстрирована на рис. 4.4.11.4.1. ЭВМ (или узлы) A1,B1,C1,D1 и маршрутизатор G-1 составляют одну автономную систему, а A2,B2,C2,D2,E2 и маршрутизатор G-2 - вторую.

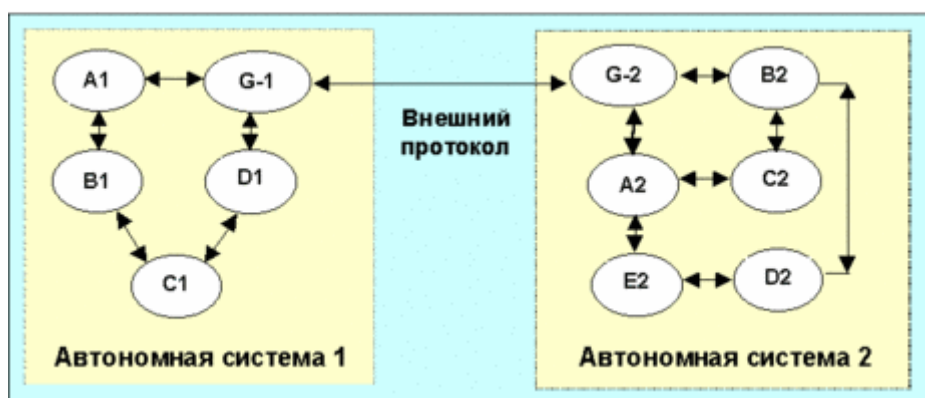


Рис. 4.4.11.4.1. Схема связи автономных систем

Пусть имеются две автономные системы ASX и ASY, обменивающиеся маршрутной информацией. ASX знает, как можно достичь сети Net1, которая может и не принадлежать ASX. Аналогично ASY знает, как послать пакет для сети Net2.

net1 ASX <----> ASY Net2

Для того чтобы пакеты попали из Net1 в Net2 через ASX и ASY, автономная система ASX должна анонсировать сеть Net1 автономной системе ASY, используя один из внешних протоколов маршрутизации (EGP или BGP), а ASY должна воспринять эту информацию и использовать. Предметом маршрутной политики в этом случае является решение ASX послать маршрутную информацию ASY, а также решение ASY эту информацию принять и использовать. Не существует никаких правил, которые бы вынуждали ASX и ASY к принятию таких решений. Таким образом, протокол маршрутизации определяет формат маршрутной информации, способ ее пересылки и хранения, но решения о ее посылке той или

иной AS, а также решение об использовании маршрутной информации, поступающей извне остаются в руках администратора AS.

Так как все существующие протоколы маршрутизации используют при работе с пакетами только адрес места назначения, разделить поток пакетов кроме как по этому параметру не возможно. Если пакеты с одним и тем же адресом места назначения попали в общий маршрутизатор, AS или канал связи, они обречены далее двигаться вместе.

Особый случай составляет топология, при которой две as имеют много возможных маршрутов связи с различными политиками маршрутизации.

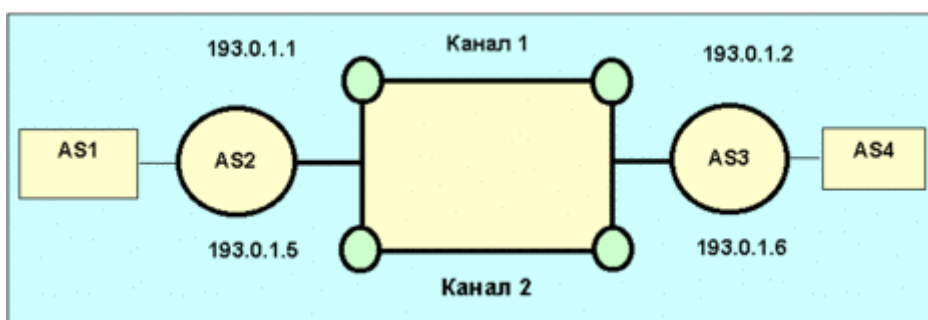


Рис. 4.4.11.4.2

Под каналом в данном случае подразумевается любая среда коммуникации - Ethernet, FDDI и т.д.. Может так случиться, что AS2 предпочитает использовать канал 2 только для обмена с AS4. А канал 1 используется для связи с AS3 и в качестве резервного маршрута (back-up) к AS4 в случае выхода из строя канала 2. Для описания маршрутной политики используются атрибуты *interas-in* и *interas-out*. Эти атрибуты позволяют описать локальные решения AS, основанные на ее предпочтениях, так как это делается в протоколах BGP-4 или IGRP. Пример такого описания представлен ниже:

aut-num: AS2 - (номер автономной системы, формат: AS<целое положительное число в интервале 1-65535>)

as-in: from AS3 10 accept AS3 AS4

(описание воспринимаемой маршрутной информации от других AS-партнеров. Формат описания:

from <aut-num><cost> accept >выражение маршрутной политики <. здесь . <aut-num> - относится к AS-соседям; <cost> - положительное целое число, характеризующее относительную ценность маршрутов, чем меньше cost, тем предпочтительнее маршрут; ключевые слова from (от) и accept (воспринимает) могут отсутствовать.)

Маршрутная политика (<выражение маршрутной политики>) может иметь следующие форматы.

1. Список из одного или нескольких кодов AS, AS-макро или список маршрутов. Список маршрутов записывается в префиксной форме, в качестве разделителей используются запятые и фигурные скобки.
2. Список ключевых слов. В настоящее время определено ключевое слово ANY, которое говорит о том, что речь может идти о любых соседних AS.
3. Логическое выражение, включающее в себя объекты типа 1 или 2, объединенные операторами AND, OR, NOT, которые в данном случае, строго говоря, не являются Булевыми. Так например, AS1 OR AS2 означает все маршруты AS1 или AS2. Или AS1 AND HepNet означает маршрут AS1, принадлежащий объединению HepNet. NOT AS3 означает любой маршрут кроме маршрутов AS3.

Приоритеты операторов распределены в следующем порядке: для оператора () слева направо; для NOT - справа на лево; для AND и OR - слева направо.

В отсутствии логических операторов элементы списка (AS, AS-макро, объединения и списки маршрутов) предполагаются объединенными оператором OR.

as-out: to AS3 announce AS1 AS2

(описание сформированной маршрутной информации, рассылаемой другим AS-партнерам). Формат описания:

to . <aut-num> announce . <выражение маршрутной политики>
 ключевые слова to {указатель адресата} и announce
 {указатель списка доступных AS} могут и отсутствовать.)
 относится к AS-соседям.

```

interas-in: from AS3 193.0.1.1/32 193.0.1.2/32 (pref= 5) accept AS3
interas-in: from AS3 193.0.1.1/32 193.0.1.2/32 (pref=15) accept AS4
interas-in: from AS3 193.0.1.5/32 193.0.1.6/32 (pref=10) accept AS4
...

```

interas-in: описывает локальные предпочтения для соединений с другими AS. Это описание имеет формат:

from accept <выражение маршрутной политики>

Ключевые слова from (от) и accept (воспринимает) могут отсутствовать. <aut-num> - автономная система, описанная в as-in. <местный-rid> - (идентификатор местного маршрутизатора) содержит IP-адрес пограничного маршрутизатора, политика которого описывается. <соседний-rid> содержит IP-адрес соседнего маршрутизатора, от которого воспринимается маршрутная информация, описанная в <выражении маршрутной политики> . IP-адреса имеют префиксный формат (описание префиксного формата см. выше в конце раздела 1.4.11.4 [бесклассовая интердоменная маршрутизация - CIDR]).

Предпочтение описывается, как <pref-type> = <значение>; ключевое слово <pref-type> должно обязательно присутствовать. В настоящее время стандарт поддерживает только один вид <pref-type> - "pref". <значение> может принимать один из следующих видов:

<стоимость> - положительное число, служащее выражением относительной ценности исследуемых маршрутов. Чем меньше <стоимость> - тем предпочтительнее маршрут. <стоимость> имеет смысл при сравнении атрибутов interas-in и совершенно не применима для сравнения с атрибутами as-in.

Любой маршрут, описанный в interas-in и неупомянутый в AS-IN, предполагается отвергнутым. Система диагностики выдаст при этом сигнал ошибки. Атрибуты interas-out сходны с interas-in, также как as-out и as-in.

Если мы рассмотрим соответствующие атрибуты interas-out для AS3, то увидим следующее:

aut-num:						AS3
as-in:	from	AS2	10	accept	AS1	A2
as-out:	to	AS2		announce	AS3	AS4
interas-out:	to	AS2	193.0.1.2/32	193.0.1.1/32	(metric-out=5)	announce AS3
interas-out:	to	AS2	193.0.1.2/32	193.0.1.1/32	(metric-out=15)	announce AS4
interas-out:	to	AS2	193.0.1.6/32	193.0.1.5/32	(metric-out=10)	announce AS4
...						

Оператор interas-out: имеет формат:

```
to      aut-num      <    местный-rid><    соседний-rid>      [<метрика>      ]
announce <выражение маршрутной политики>
```

Ключевые слова IN и announce могут и отсутствовать. Остальные фрагменты оператора идентичны описанным выше.

< метрика> является необязательным параметром и описывается как:

(< metric-type> =< величина>), следует заметить, что в данном случае наличие скобок "()" и ключевого слова <metric-type> (тип метрики) обязательно. В настоящее время поддерживается только один тип метрики "metric-out". Параметр <величина> может иметь следующий вид:

< num-metric> - метрика для оценки выходных маршрутов, чем ниже ее значение, тем предпочтительнее маршрут. Именно эта оценка маршрута передается соседним маршрутизаторам. IGP - этот тип метрики указывает на то, что она отражает оценку внутренних маршрутов AS. Следует избегать использования <num-metric> и IGP для одних и тех же точек назначения.

Атрибут as-exclude отмечает список транзитных AS, маршруты через которые должны быть исключены из рассмотрения. Формат использования:

exclude <aut-num> to <exclude-route-keyword> , ключевые слова exclude и to не являются обязательными. <aut-num> - описывает транзитные AS. В качестве <exclude-route-keyword> можно использовать одно из:

<aut-num> , AS макро, ANY (любой) или community.

Атрибут default указывает на маршрут по умолчанию. Формат атрибута:

<aut-num><relative cost><default-expression> ,

где <aut-num> указывает на AS-партнера, путь к которому и предлагается в качестве маршрута по умолчанию. Атрибут <relative cost> - положительное целое число, характеризующее уровень приоритета предлагаемого маршрута. AS-макро является удобным средством группировать автономные системы. AS-макро могут использоваться в <выражениях маршрутной политики> для атрибутов as-in и as-out. В качестве примера можно привести:

```
aut-num: AS786
as-in: from AS1755 100 accept AS-EBONE AND NOT AS1104
as-out: to AS1755 announce AS786
.....
```

Здесь присутствует as-макро с именем AS-EBONE, описание которого может выглядеть как:

```
as-macro: AS-EBONE (имя AS-макро)
descr: ASes routed by EBONE (AS, маршрутизируемые EBONE)
as-list: AS2121 AS1104 AS2600 AS2122 (список AS)
as-list: AS1103 AS1755 AS2043
guardian: guardian@ebone.net (администратор EBONE)
.....
```

В результате описание маршрутной политики будет выглядеть как:

```
aut-num: AS786
as-in: from AS1755 100 accept (AS2121 OR AS1104 OR AS2600 OR AS2122
as-in: from AS1755 100 accept AS1103 OR AS1755 OR AS2043)AND NOT AS1104
.....
```

Community - это группа маршрутов, которые не могут быть представлены одной AS или группой AS. Это может быть полезно при описании доступа к супер-ЭВМ, это может быть группа маршрутов, используемых для специальных целей, возможно объединение в группу для получения сетевой статистики. Такие группы не обмениваются маршрутной информацией. Группа Community может использоваться в качестве объекта маршрутной политики автономных систем. Примерами объектов типа community могут служить NERUNET (объединение сетей

для физики высоких энергий), NSFNET (Национальная Научная сеть США), опорная московская оптоволоконная сеть.

Существует еще несколько атрибутов, которые помогают получить вспомогательную информацию. Более детальное описание форматов и алгоритмов решения проблем реализации различных политик маршрутизации читатель может найти в ссылке в начале раздела.

Интернет-технологии, способы и скоростные характеристики подключения, провайдер

Интернет-технологии - это коммуникационные, информационные и иные технологии и сервисы, основываясь на которые осуществляется деятельность в Интернете или с помощью него. В первую очередь, это, конечно, сайты, а также: чаты, почта, Интернет-магазины, форумы и т.д.

С 1962 года министерство обороны США активно вкладывало деньги в разработки технологий взаимодействия между компьютерами, Благодаря этим исследованиям, в 1969 году, американскими военными был создана локальная сеть - предшественница Интернета. С 1972 года функционирует всеми нами любимая электронная почта.

Переломным этапом в развитии интернет-технологий стали 90-е годы. С изобретением в 1993 году первого браузера, предполагаемые возможности Интернета стали приобретать глобальный характер. Появились многочисленные телекоммуникационные операторы, компании электронной торговли, разработчики программного обеспечения и т.п.

В наши дни, интернет - технологии нашли свое применение во всех сферах жизни современного общества и, в первую очередь, конечно же, - в информационной сфере.

Логические компоненты Интернет-технологий

Интернет - сервисы

- * WorldWideWeb - Всемирная паутина
- * Электронная почта. Системы телеконференций
- * Передача файлов (FTP).
- * Интерактивный чат (chat).

- * Передача мгновенных сообщений (ICQ).

- * Аудио- и Видеоконференции.

- * Голосовое общение (IP-телефония).

Работа в Интернете

- * Браузеры,

- * Поисковые системы. Навигация в Интернете

- * Просмотр Web-страницы в браузере.

Информационные ресурсы в Интернете

- * Web-страницы и Web-узлы, порталы, Web - пространство.

- * Адресация, URL и протоколы передачи данных,

- * Создание Web-страниц. Языки Web-публикаций.

- * Публикации в Интернете. Представительство.

Провайдер – организация, предоставляющая услуги доступа к Интернету и иные связанные с Интернетом услуги.

В число предоставляемых интернет-провайдером услуг могут входить:

- § доступ в Интернет по коммутируемым и выделенным каналам;

- § выделение дискового пространства для хранения и обеспечения работы сайтов (хостинг);

- § поддержка работы почтовых ящиков или виртуального почтового сервера;

- § резервирование данных и другие.

Способы подключения к Интернет

Самыми распространенными способами подключения к Интернету являются:

- 1.Модемное соединение (коммутируемый доступ) – Dial-Up, ADSL
- 2.Соединение по выделенной линии (оптоволокно и т.д.)
- 3.GPRS-доступ (через сотовый телефон)

4. Радиодоступ

5. Спутниковый интернет

Все они отличаются друг от друга принципом работы, скоростью передачи данных, надежностью, сложностью настройки оборудования и, конечно же, ценой.

Скорость передачи данных – это количество информации, передаваемой пользователю за единицу времени, обычно измеряется в килобайтах/сек или килобитах/сек. Для высокоскоростных каналов измерение скорости уже идет в мегабитах или мегабайтах в секунду.

1. Модемное соединение (Dial-Up) – самый старый способ подключения, сейчас пользуется популярностью только в городах, где нет широкого выбора провайдеров.

Плюсами такого соединения являются: простота настройки и установки оборудования (требуется только аналоговый модем), низкая цена оборудования.

Недостатков у модемного соединения гораздо больше: занятость телефонной линии абонента, низкая скорость передачи данных, низкое качество соединения и передачи данных из-за изношенности телефонных линий.

При скорости, обеспечиваемой модемным соединением, практически невозможно скачать из Интернета большие файлы – видео, дистрибутивы больших программ и т.д.

Более перспективной по сравнению с Dial-Up в настоящее время является технология **ADSL (Asymmetric Digital Subscriber Line)**. Это технология, позволяющая предоставлять по обычным аналоговым телефонным линиям высокоскоростной широкополосный доступ в Интернет. К тому же данная технология не загружает телефонную линию абонента, благодаря разделению диапазонов сигналов в телефонной линии. Абоненту также не нужно дозваниваться до провайдера. Минусы – достаточно высокая стоимость трафика.

Соединение по выделенной линии

Провайдер проводит до компьютера абонента выделенную линию (витая пара или оптоволокно) и выдает диапазон IP-адресов для выхода в Интернет. При этом пользователь получает свободную телефонную линию, постоянную связь с сетью Интернет, высокое качество соединения и передачи данных, высокую скорость (до 100 Мбит/с).

Однако стоимость установки и настройки такого соединения прямо зависит от расстояния компьютера до точки подключения провайдера, и в любом случае, достаточно велика по сравнению с другими способами соединения. А при переносе компьютера в другое место к нему снова придется прокладывать кабель.

Требуемое оборудование

В данном случае пользователю из оборудования необходима только сетевая карта, а в настоящее время она имеется практически на любой из материнских плат.

GPRS-доступ

Практически все провайдеры на сегодняшний день предлагают услугу использования сотового телефона вместо традиционного модема. Для пользователя такое подключение к Интернет

хорошо тем, что при поддержке телефоном GPRS из дополнительного оборудования требуется только устройство для связи мобильного телефона с компьютером (USB-кабель, ИК-порт или

Bluetooth). Плюсом также является мобильность данного соединения.

Скорость передачи данных зависит от оператора сотовой связи и применяемого оборудования, но в целом невысока – всего в два раза выше модемной. Определяющим недостатком является высокая стоимость трафика.

4. Радиодоступ – беспроводной способ подключения к Интернет. У провайдера и абонента устанавливается все необходимое оборудование (специальный радиомодем, антенна), с

помощью которого осуществляется обмен информацией между пользователем и Интернет.

Минусы радиодоступа заключаются в необходимости покупки дорогостоящего оборудования и высокой абонентской платы.

5. Спутниковое соединение – бывает односторонним (асинхронным) и двусторонним. Второе по причине дороговизны оборудования (счет идет на десятки тысяч) рассматривать не будем.

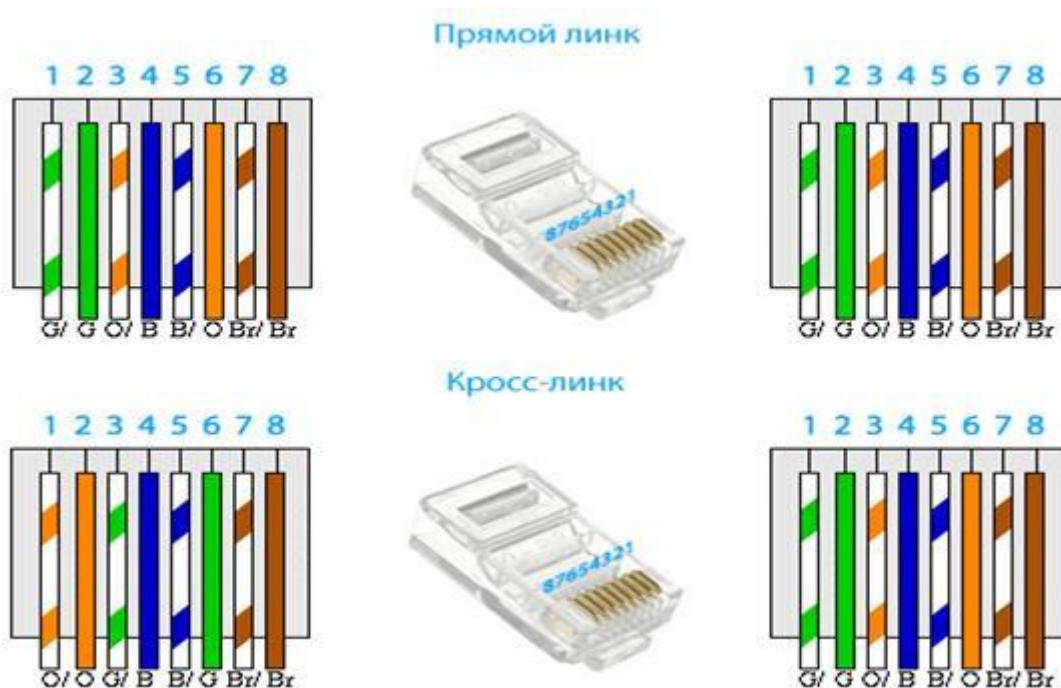
Чаще всего спутниковым Интернет называют асинхронный (или совмещенный) способ доступа – данные к пользователю поступают через спутниковую тарелку, а запросы (трафик) от пользователя передаются любым другим соединением – GPRS или по наземным каналам (ADSL, dial-up). Главное требование к запросному каналу – надежность соединения. В большинстве случаев лучшим выбором для него является ADSL подключение с бесплатным исходящим трафиком.

Преимущества спутникового подключения к Интернет – в первую очередь, это очень низкая стоимость трафика – от 10 до 100 копеек за 1 мегабайт. Стоимость комплекта оборудования и подключения доступна практически для всех и составляет приблизительно 200-300 \$.

Скорость передачи данных значительно варьируется в зависимости от провайдера и тарифного плана, выбранного пользователем. Провайдеры спутникового Интернет предлагают очень широкий выбор тарифных планов, в том числе и безлимитных. Очень приятным бонусом является также возможность бесплатного приема спутникового телевидения.

Минусом спутникового подключения к Интернет является необходимость наличия канала для исходящего трафика – телефонной линии или телефона с поддержкой GPRS.

Стоит заметить, что в крупных городах России из-за растущей конкуренции провайдеров и развития локальных сетей доступ в Интернет становится все дешевле. Зачастую можно установить безлимитный доступ в сеть с приличной скоростью и абонентской платой 500-1000 руб./месяц. В провинции и небольших городах с этим все намного сложнее и цены выше на порядок. Самым оптимальным выбором в этой ситуации будет использование спутникового Интернет.



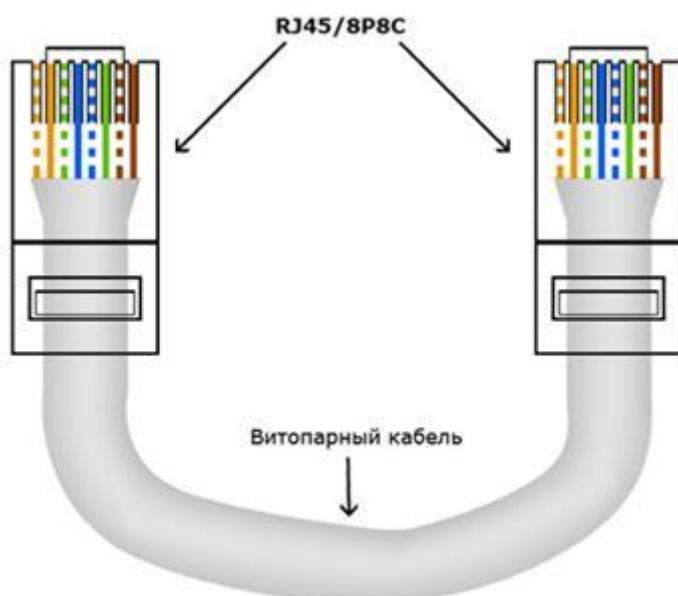
РЕКЛАМА

Новая книга Коллектива авторов — «Незабываемые»
Читайте и слушайте новую книгу на ЛитРес

ЛитРес:

litres.ru

[Подробнее](#)



EIA/TIA-568B

"нуль-хабный" кабель		
одна сторона	цвет провода	другая сторона
1	бело/зеленый	3
2	зеленый	6
3	бело/оранж	1
4	синий	7
5	бело/синий	8
6	оранжевый	2
7	бело/коричн.	4
8	коричневый	5

Или, например, другой вариант.

"нуль-хабный" кабель		
одна сторона	цвет провода	другая сторона
1	бело/зеленый	3
2	зеленый	6
3	бело/оранж	1
4	синий	4
5	бело/синий	5
6	оранжевый	2
7	бело/коричн.	7
8	коричневый	8

Тестирование по разделу «Телекоммуникационные технологии»

1. Услуга по размещению и хранению файлов клиента на сервере организации, предоставляющей подобную услугу - это ...

- a) Хостинг
- b) Провйдер
- c) WEB-сайт
- d) Социальные сети

2. Какой протокол является базовым протоколом Интернета?

- a) FTP
- b) TCP/IP
- c) URL
- d) DNS

3. ivanov@rambler.ru

Что в этом адресе электронной почты означает имя сервера?

- a) rambler
- b) ivanov
- c) rambler.ru
- d) ivanov@rambler.ru
- e) ivanov@

4. На каком языке записываются Web-страницы?

- a) Pascal

- b) C++
- c) HTML
- d) VisualBasic

5. www.klyaksa.net Что является доменом верхнего уровня в этом адресе?

- a) net
- b) klyaksa.net
- c) www.klyaksa.net
- d) www.klyaksa
- e) www

6. Как называются программы, позволяющие просматривать Web- страницы:

- a) Адаптеры
- b) Операционные системы
- c) Браузеры
- d) Трансляторы

7. Провайдер Интернета – это:

- a) техническое устройство;
- b) антивирусная программа;
- c) организация – поставщик услуг Интернета;
- d) средство просмотра Web-страниц.

8. Электронная почта позволяет передавать:

- a) только сообщения;
- b) только файлы;
- c) сообщения и приложенные файлы;

d) видеоизображение.

9. Гипертекст – это:

a) очень большой текст;

b) текст, в котором могут осуществляться переходы по выделенным ссылкам;

c) текст, набранный на компьютере;

d) текст, в котором используется шрифт большого размера

10. Теги языка HTML- это...

a) названия элементов страницы

b) набор символов

c) окно браузера

Локальные и глобальные информационные сети

Информационной (компьютерной) сетью называется группа компьютеров, соединенных между собой с помощью специальной аппаратуры, обеспечивающей обмен данными. Компьютеры, расположенные в пределах одного или нескольких рядом стоящих зданий и объединенные с помощью высокоскоростного сетевого оборудования, называют *локальной сетью*. Для подключения компьютера к локальной сети необходимо устройство, называемое *сетевым адаптером*. Современные сетевые адаптеры обеспечивают передачу информации со скоростью 10—100 Мбит (миллионов бит) в секунду. При объединении компьютеров, расположенных на более значительном расстоянии друг от друга (в разных городах, странах), говорят о *глобальной сети*. Основным (наиболее массовым в настоящее время, но далеко не единственным) каналом передачи данных в этом случае является телефонный канал. Устройство, необходимое для подключения компьютера к телефонной линии, называется *модемом*. Скорость передачи данных здесь меньше, чем в локальных сетях и находится в

существенной зависимости от качества канала связи и типа модема. В настоящее время наблюдается тенденция к подключению локальных сетей к мировой глобальной сети *Internet*.

Существует несколько способов объединения компьютеров в локальную сеть. Наиболее широко используются топологии «звезда», «общая шина», «кольцо». Топология «звезда» предполагает, что каждый компьютер подключен с помощью отдельного кабеля к объединяющему устройству. Топология «общая шина» предполагает использование одного кабеля, к которому подключены все компьютеры сети. В топологии «кольцо» данные передаются от одного компьютера к другому как бы по эстафете. Несколько локальных сетей, выполненных с помощью различных топологий, можно объединить в единую сеть.

Передача данных по сети регламентируется определенными правилами. Набор правил взаимодействия между компьютерами сети называют *протоколами передачи данных*, или *сетевыми протоколами*. Протоколы определяют формат, способ синхронизации, порядок следования, методы обработки ошибок при передаче данных. Передача данных между компьютерами требует выполнения многих шагов. Например, для передачи файла с одного компьютера на другой файл должен быть разбит на части, эти части должны быть определенным образом сгруппированы.

Таким образом, компьютер, принимающий файл, должен получить дополнительную информацию о том, каким образом связаны между собой образованные группы, а также информацию о способе синхронизации, информацию, позволяющую корректировать ошибки, связанные с передачей данных, и т. д. Учитывая сложность осуществления коммуникаций между компьютерами, этот процесс обычно разбивается на шаги. Каждый такой шаг выполняется в соответствии со своими правилами, т. е. в соответствии со своим протоколом. Работая в локальной сети, пользователи могут посылать друг другу текстовые сообщения, получать доступ к файлам, находящимся на локальных дисках других компьютеров сети, использовать другие устройства (ресурсы) сети. Примером может служить использование принтера, подключенного к другому компьютеру сети.

Понятие «глобальная сеть» в настоящее время является синонимом понятия *Internet*. Сеть *Internet* — это множество серверов и локальных сетей, созданных на базе компьютеров различной мощности, от небольших с операционными системами *UNIX* или *Microsoft Windows NT* до мини-ЭВМ и крупных компьютеров. Эти компьютеры служат хранилищами данных и принадлежат различным организациям, коммерческим и некоммерческим, университетам, исследовательским институтам, национальным библиотекам, отдельным лицам и т. д.

Серверы объединяются между собой различными линиями связи: спутниковыми, волоконно-оптическими, а также телефонными. Серверы глобальной сети могут быть включены в состав локальных сетей. В настоящее время

сеть *Internet* обеспечивает пользователей такими видами сервиса, как электронная почта, передача файлов, просмотр и получение информации, организованной в виде гипертекстовых файлов (сервис *WorldWideWeb*), электронные конференции, *Telnet*. Рассмотрим эти виды сервиса более подробно.

Электронная почта

Этот сервис был реализован первым. Но и в настоящее время к его услугам прибегают практически все пользователи глобальной сети. Электронная почта представляет собой наиболее удобный и быстрый способ доставки сообщений в любую точку планеты. Для того чтобы послать электронное письмо, вы должны знать электронный адрес своего корреспондента. Помимо текстового сообщения вы можете передать через электронную почту произвольный файл. В сети *Internet* существуют так называемые *серверы рассылки*, способные по предварительной заявке автоматически рассылать сообщения. С помощью такого сервера вы можете подписаться, например, на электронный вариант газеты или получать любую другую периодически обновляемую информацию в виде писем.

Передача файлов

Для передачи файлов используется протокол *FTP* (*FileTransferProtocol*), позволяющий переписывать файлы с дисков удаленного сервера на локальный диск вашего компьютера. Разработаны удобные программы, напоминающие широко известную оболочку *NortonCommander*, упрощающие процесс передачи файлов. Кроме того, для получения файлов из серверов *FTP* вы можете использовать *программы-навигаторы*, предназначенные для работы с серверами *WWW*. Последние становятся наиболее популярным средством хранения и представления информации в сети *Internet*,

Серверы *WorldWideWeb*

Электронная почта и передача файлов - это то, с чего начиналось становление глобальных сетей. В последнее время во всем мире наблюдается лавинообразный рост числа серверов WWW (WorldWideWeb), которые могут быть использованы для представления мультимедийной информации, имеющей отношение к самым разным сферам человеческой деятельности. Серверы WWW хранят информацию в виде гипертекстовых файлов, подготовленных специальным образом. Эти файлы ссылаются на другие такие же файлы, на файлы, содержащие изображение, звук и т. п. Примечательно то, что ссылки могут указывать на файлы, расположенные не только на том же самом сервере WWW, но и на любом другом сервере, подключенном к сети Internet.

Электронные конференции

В глобальной сети имеются серверы *электронных конференций*, которые хранят статьи (в виде текстовых документов), объединенные в группы. Имея доступ к такому серверу, пользователи сети могут посылать в выбранные ими группы свои статьи, а также просматривать и получать статьи, записанные туда другими пользователями.

Группы создаются по интересам, причем существует большое число таких групп и появляются все новые и новые. Посылая статьи в конференцию, вы можете участвовать в обсуждении любой проблемы, имеющей отношение к теме конференции. Большинство конференций доступны во всей сети *Internet* (глобальные конференции). Существуют и локальные конференции, доступные ограниченному кругу пользователей сети *Internet*. Существует возможность подписки на интересующие вас конференции. При этом вы будете получать новые статьи из выбранной вами конференции.

Средство *Telnet*

В сети Internet имеется немало серверов, предоставляющих пользователям доступ к своей консоли (экрану и клавиатуре). Обычно такие серверы работают под управлением операционной системы UNIX. Для получения доступа к консоли компьютера вы должны подключиться к сети Internet и запустить программу с названием TELNET. Версии этой программы имеются практически для любой операционной системы. В частности, программа TELNET есть в операционных системах Microsoft Windows 95 и Microsoft Windows NT. По своему назначению удаленная консоль компьютера ничем не отличается от локальной. Поэтому вы можете управлять операционной системой с символьного терминала, подключенного непосредственно к компьютеру.

Тесты *

** Для выполнения тестов выберите правильный ответ из предложенных вариантов.*

5.1. *С чем связано появление консалтинговых компаний:*

- а) с тем, что руководство предприятий не способно самостоятельно справиться с возникшими проблемами;*
- б) с переходом к рыночным отношениям;*
- в) с развитием информационных технологий?*

5.2. В процессе разработки консалтинговых проектов преследуются следующие цели:

- а) представление деятельности предприятия и принятых в нем технологий в виде иерархии диаграмм;*
- б) формирование организационной структуры управления на основе анализа предложений по реорганизации;*
- в) упорядочение информационных потоков, в том числе документооборота;*
- г) выработка рекомендаций по построению рациональных технологий работы подразделений предприятия и его взаимодействию с внешней средой;*
- д) анализ требований и проектирование спецификаций корпоративных информационных систем;*
- е) выработка рекомендаций и предложений по применимости существующих систем управления;*
- ж) цели со второй по шестую ($\bar{б} + в + г + д + е$);*
- з) все шесть целей ($а + \bar{б} + в + г + д + е$);*
- и) первая, вторая и шестая цели ($а + \bar{б} + е$).*

5.3. Назовите этапы разработки консалтинговых проектов, выполняемых консалтинговыми организациями:

- а) анализ первичных требований и планирование работ;*
- б) проведение обследования деятельности предприятия;*
- в) построение и анализ моделей деятельности предприятия;*
- г) разработка системного проекта (модели требований к будущей системе);*
- д) разработка технического проекта;*
- е) разработка рабочего проекта;*
- ж) ввод в опытную и промышленную эксплуатацию;*
- з) сопровождение и реинжиниринг;*
- и) все восемь этапов ($а + \bar{б} + в + г + д + е + ж + з$);*
- к) первые четыре этапа ($а + \bar{б} + в + г$).*

5.4. При обследовании предприятия целесообразно применять следующие методы:

- а) анкетирование;*
- б) сбор документов;*
- в) интервьюирование;*
- г) личное участие;*
- д) все четыре метода (а + б + в + г).*

5.5. Что представляет собой модель «как есть»:

- а) «снимок» положения дел на предприятии на момент обследования;*
- б) перспективные предложения руководства и сотрудников предприятия, экспертов и системных аналитиков по совершенствованию деятельности предприятия?*

ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ (ЛВС)

Интернет или, по-другому, всемирная сеть – это, конечно, хорошо. Но возможности соединения по ней ограничены "черепашьей" скоростью вашего модема и толщиной вашего кошелька.

Для объединения близко расположенных компьютеров без использования телефонных линий применяются ЛВС.

В свою очередь уже объединенные в ЛВС компьютеры могут быть подключены к Интернету.

Компьютерные сети можно классифицировать по многим признакам, например по удаленности сетевых узлов.

Расстояние между узлами Масштаб сети Обозначение

0,1 км	Здание	Локальная сеть (Local Area Network)
1-10 км	Город	Городская сеть (Metropolitan Area Network)

10-1000 км	Страна	Глобальная сеть (Wide Area Network)
Более 1000 км	Планета	Internet

Локальная вычислительная сеть (ЛВС) – это совокупность компьютеров и других средств вычислительной техники (активного сетевого оборудования, принтеров, сканеров и т.п.), объединенных с помощью кабелей и сетевых адаптеров и работающих под управлением сетевой операционной системы. Вычислительные сети создаются для того, чтобы группа пользователей могла совместно задействовать одни и те же ресурсы: файлы, принтеры, модемы, процессоры и т.п. Каждый компьютер в сети оснащается сетевым адаптером, адаптеры соединяются с помощью сетевых кабелей и тем самым связывают компьютеры в единую сеть. Компьютер, подключенный к вычислительной сети, называется рабочей станцией или сервером, в зависимости от выполняемых им функций. Эффективно эксплуатировать мощности ЛВС позволяет применение технологии «клиент/сервер». В этом случае приложение делится на две части: клиентскую и серверную. Один или несколько наиболее мощных компьютеров сети конфигурируются как серверы приложений: на них выполняются серверные части приложений. Клиентские части выполняются на рабочих станциях; именно на рабочих станциях формируются запросы к серверам приложений и обрабатываются полученные результаты.

Различают сети с одним или несколькими выделенными серверами и сети без выделенных серверов, называемые одноранговыми сетями. Рассмотрим сначала локальные сети с выделенным сервером. В сетях с выделенным сервером именно ресурсы сервера, чаще всего дисковая память, доступны всем пользователям. Серверы, разделяемым ресурсом которых является дисковая память, называются файл-серверами. Можно сказать, что сервер обслуживает все рабочие станции. Файловый сервер обычно используется только администратором сети и не предназначен для решения прикладных задач. Поэтому он может быть оснащен недорогим, даже монохромным дисплеем. Однако файловые серверы почти всегда содержат несколько быстродействующих накопителей. Сервер должен быть высоконадежным, поскольку выход его из строя приведет к остановке работы всей сети. На файловом сервере, как правило, устанавливается сетевая операционная система.

На рабочих станциях, как правило, устанавливается обычная операционная система, например: DOS, Windows или Windows NT. Рабочая станция – это индивидуальное рабочее место пользователя. Полноправным владельцем всех ресурсов рабочей станции является пользователь. В то же время ресурсы файл-сервера разделяются всеми пользователями. В качестве рабочей станции может использоваться компьютер практически любой конфигурации. Но, в конечном счете, все зависит от тех приложений, которые этот компьютер используют. Существует несколько признаков, по которым можно узнать, работает компьютер в составе сети или автономно. Если компьютер является сетевой рабочей станцией, то, во-первых, после его включения появляются соответствующие

сообщения, во-вторых, для входа в сеть необходимо пройти процедуру регистрации и, в-третьих, после регистрации в нашем распоряжении оказываются новые дисковые накопители, принадлежащие файловому серверу.

Отметим еще одну важную функцию файлового сервера – управление работой сетевой принтера. Сетевой принтер подключается к файл-серверу, но пользоваться им можно с любой рабочей станции. То есть каждый пользователь может отправить на сетевой принтер материалы, предназначенные для печати. Регулировать очередность доступа к сетевому принтеру будет файловый сервер.

ЛВС с выделенным сервером

При выборе компьютера на роль файлового сервера необходимо учитывать следующие факторы:

- быстродействие процессора;
- скорость доступа к файлам, размещенным на жестком диске;
- емкость жесткого диска;
- объем оперативной памяти;
- уровень надежности сервера;
- степень защищенности данных.

Возникает вопрос, зачем файл-серверу высокое быстродействие, если прикладные программы выполняются на рабочих станциях? Во время работы большой ЛВС файловый сервер обрабатывает огромное количество запросов на обслуживание файлов, а на это затрачивается значительное процессорное время. Для того чтобы ускорить обслуживание запросов и создать у пользователя впечатление, что именно он является единственным клиентом сети, необходим быстродействующий процессор.

Но все же наиболее важным компонентом файлового сервера является дисковый накопитель. На нем хранятся все файлы пользователей сети. Быстрота доступа, емкость и надежность накопителя во многом определяют, насколько эффективным будет использование сети.

Сетевые ОС с выделенным файл-сервером обычно имеют более высокую производительность, поскольку они оптимизированы именно под выполнение операций с файлами. В принципе, никаких более важных действий на выделенном файл-сервере не выполняется. Значительного повышения производительности работы сервера можно добиться, увеличивая его оперативную память. В одноранговой сети нескольких мегабайт памяти может быть вполне достаточно, в то время как для крупной сети с выделенным файл-сервером желательна память объемом 128 и более мегабайт. Если файловый сервер снабжен оперативной

памятью достаточного объема, то он имеет возможность именно в оперативной памяти хранить те области дискового пространства, к которым обращаются наиболее часто. Такой метод хорошо известен, часто применяется для ускорения доступа к данным на обычных ПК и называется методом кэширования. Ведь если идет обращение к файлу, данные которого в данный момент находятся в кэше, сервер может передать искомую информацию, не обращаясь к диску. В результате этого будет достигнут значительный временной выигрыш.

Сетевой адаптер, установленный на файловом сервере – это такое устройство, через которое проходят практически все данные, функционирующие в локальной сети. В связи с этим необходимо, чтобы этот адаптер работал быстро. Сетевой адаптер становится более быстродействующим в результате, во-первых, повышения его разрядности и, во-вторых, увеличения объема его собственного ОЗУ. На файл-сервере должен быть установлен сетевой адаптер для шины PCI, что позволяет поддерживать высокую скорость передачи данных.

Одноранговые ЛВС

В одноранговых сетях любой компьютер может быть и файловым сервером, и рабочей станцией одновременно. Преимущество одноранговых сетей заключается в том, что нет необходимости копировать все используемые сразу несколькими пользователями файлы на сервер. В принципе любой пользователь сети имеет возможность использовать все данные, хранящиеся на других компьютерах сети, и устройства, подключенные к ним. Основной недостаток работы одноранговой сети заключается в значительном увеличении времени решения прикладных задач. Это связано с тем, что каждый компьютер сети обрабатывает все запросы, идущие к нему со стороны других пользователей. Следовательно, в одноранговых сетях каждый компьютер работает значительно интенсивнее, чем в автономном режиме.

Затраты на организацию одноранговых вычислительных сетей относительно небольшие. Однако при увеличении числа рабочих станций эффективность их использования резко уменьшается. Пороговое значение числа рабочих станций составляет, по оценкам фирмы Novell, – 25-30. Поэтому одноранговые сети используются только для относительно небольших рабочих групп.

Архитектура ЛВС

Различают три наиболее распространенные сетевые архитектуры, которые используют и для одноранговых сетей и для сетей с выделенным файл-сервером. Это так называемые шинная, кольцевая и звездообразная структуры.

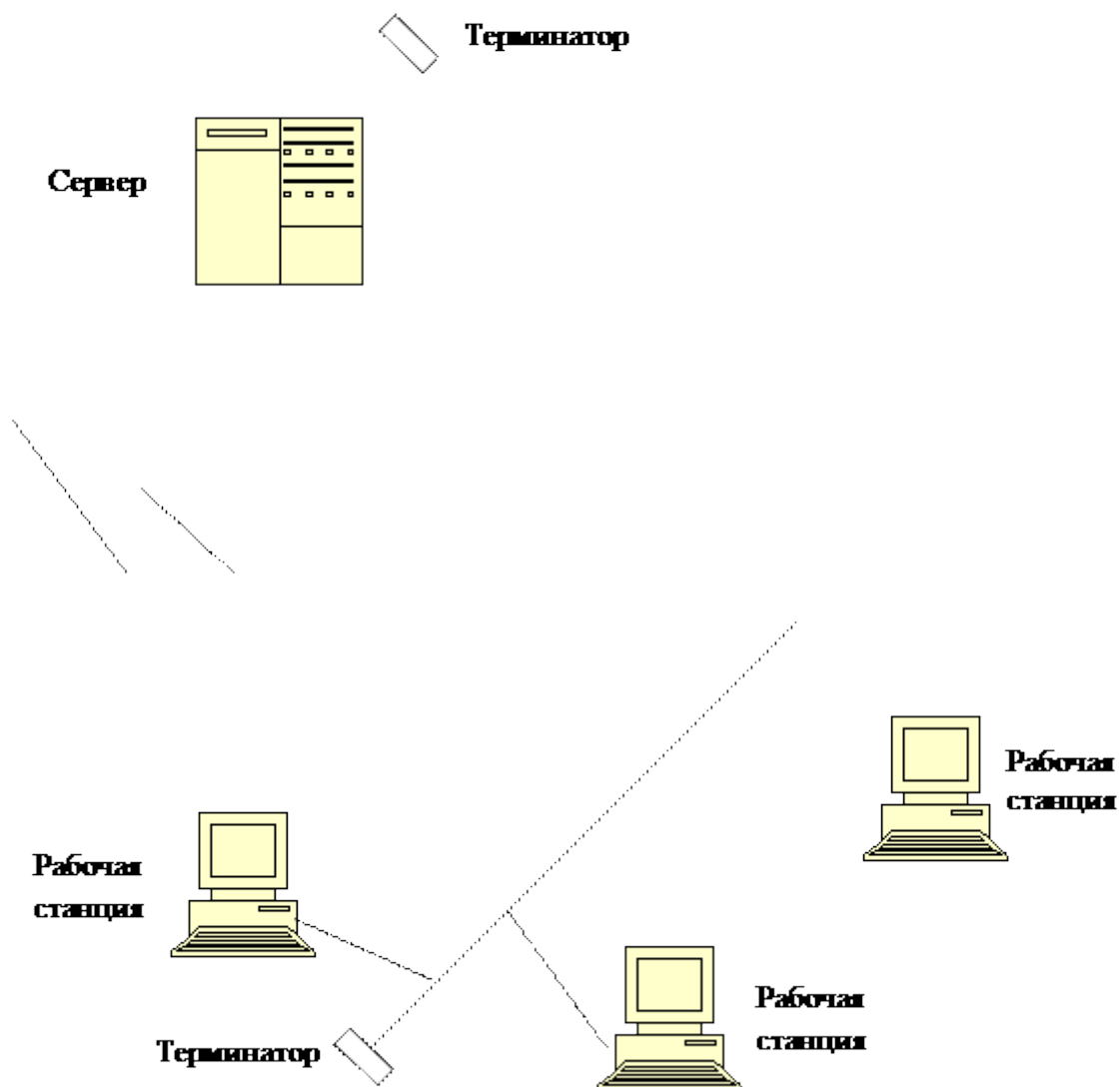


Рис. 81. Шинная структура одноранговой сети

В случае реализации шинной структуры все компьютеры связываются в цепочку. Причем на ее концах надо разместить так называемые терминаторы, служащие для гашения сигнала. Если же хотя бы один из компьютеров сети с шинной структурой оказывается неисправным, вся сеть в целом становится неработоспособной. В сетях с шинной архитектурой для объединения компьютеров используется тонкий и толстый кабель. Максимальная теоретически возможная пропускная способность таких сетей составляет 10 Мбит/с. Такой пропускной способности для современных приложений, использующих видео- и мультимедийные данные, явно недостаточно. Поэтому почти повсеместно применяются сети со звездообразной архитектурой.

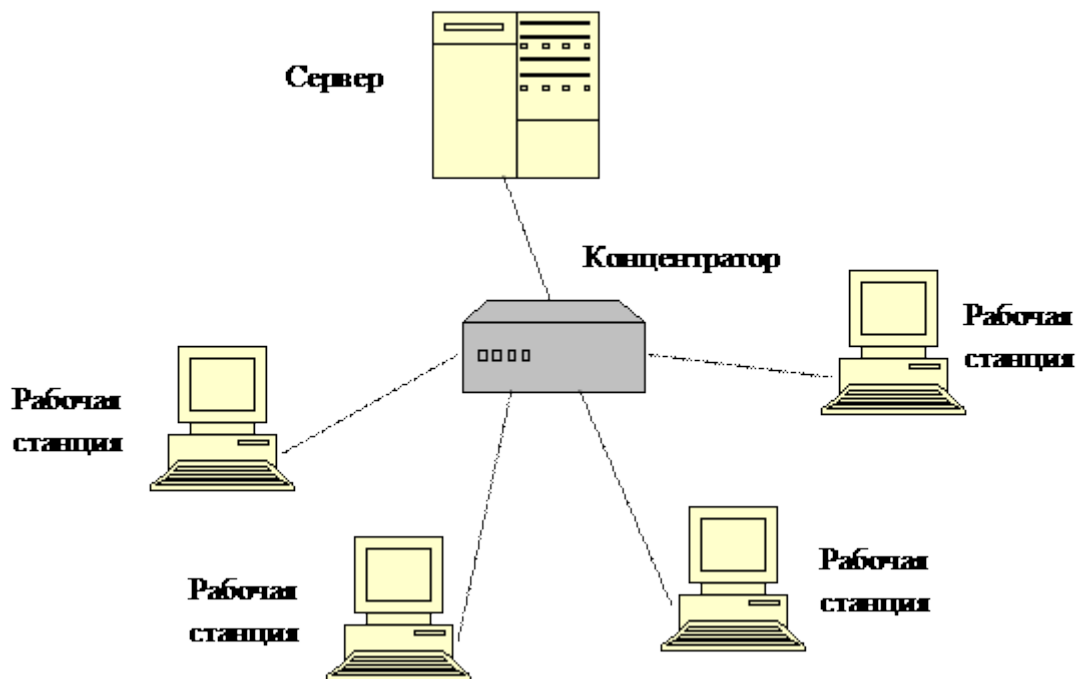


Рис. 82. Звездообразная структура одноранговой сети

Для построения сети с звездообразной архитектурой в центре сети необходимо разместить концентратор. Его основная функция – обеспечение связи между компьютерами, входящими в сеть. То есть все компьютеры, включая файл-сервер, не связываются непосредственно друг с другом, а присоединяются к концентратору. Такая структура надежнее, поскольку в случае выхода из строя одной из рабочих станций все остальные сохраняют работоспособность. В сетях же с шинной топологией в случае повреждения кабеля хотя бы в одном месте происходит разрыв единственного физического канала, необходимого для движения сигнала. Кроме того, сети со звездообразной топологией поддерживают технологии FastEthernet и GigabitEthernet, что позволяет увеличить пропускную способность сети в десятки и даже сотни раз (разумеется, при использовании соответствующих сетевых адаптеров и кабелей).

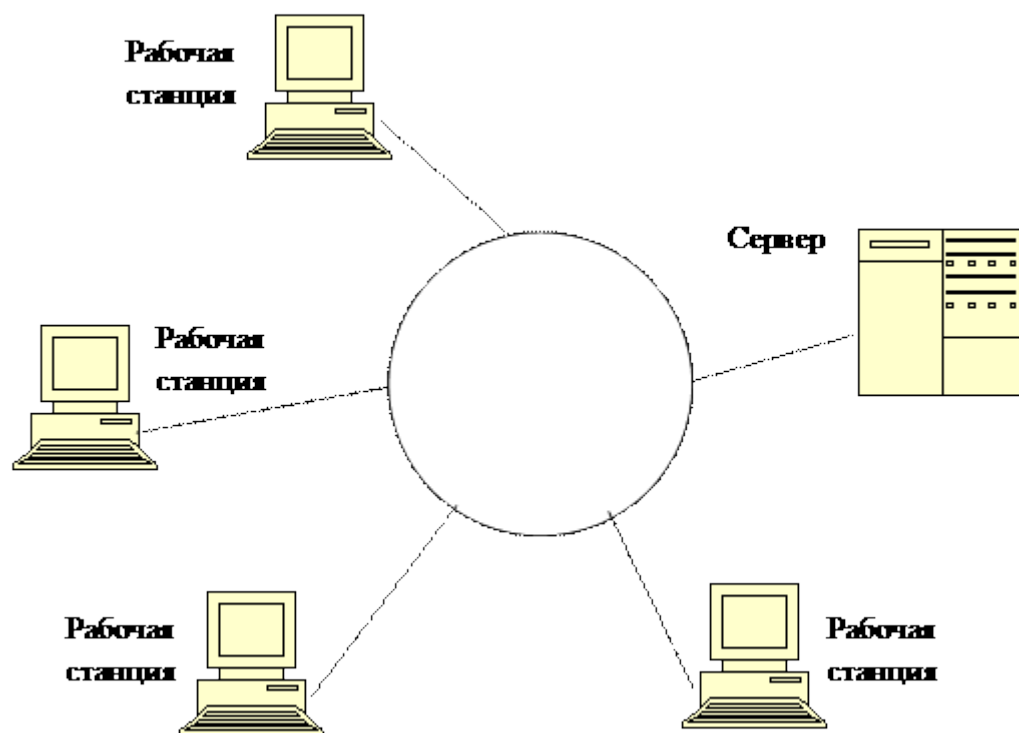


Рис. 83. Кольцевая структура одноранговой сети

Кольцевая структура используется в основном в сетях TokenRing и мало чем отличается от шинной. Также в случае неисправности одного из сегментов сети вся сеть выходит из строя. Правда, отпадает необходимость в использовании терминаторов.

В сети любой структуры в каждый момент времени обмен данными может происходить только между двумя компьютерами одного сегмента. В случае ЛВС с выделенным файл-сервером – это файл-сервер и произвольная рабочая станция; в случае одноранговой ЛВС – это любые две рабочие станции, одна из которых выполняет функции файл-сервера. Упрощенно диалог между файл-сервером и рабочей станцией выглядит так: открыть файл – подтвердить открытие файла; передать данные файла – пересылка данных; закрыть файл – подтверждение закрытия файла. Управляет диалогом сетевая операционная система, клиентские части которой должны быть установлены на рабочих станциях.

Остановимся подробнее на принципах работы сетевого адаптера. Связь между компьютерами ЛВС физически осуществляется на основе одной из двух схем – обнаружения коллизий и передачи маркера. Метод обнаружения коллизий используется стандартами Ethernet, FastEthernet и GigabitEthernet, а передачи маркера – стандартом TokenRing. В сетях Ethernet адаптеры непрерывно находятся в состоянии прослушивания сети. Для передачи данных сервер или рабочая станция должны дождаться освобождения ЛВС и только после этого приступить к передаче. Однако не исключено, что передача может начаться несколькими узлами одного сегмента сети одновременно, что приведет к коллизии. В случае возникновения коллизии, узлы должны повторить свои сообщения. Повторная

передача производится адаптером самостоятельно без вмешательства процессора компьютера. Время, затрачиваемое на преодоление коллизии, обычно не превышает одной микросекунды. Передача сообщений в сетях Ethernet производится пакетами со скоростью 10, 100 и 1000 Мбит/с. Естественно, реальная нагрузка сети меньше, поскольку требуется время на подготовку пакетов. Все узлы сегмента сети принимают сообщение, передаваемое компьютером этого сегмента, но только тот узел, которому оно адресовано, посылает подтверждение о приеме. Основными поставщиками оборудования для сетей Ethernet являются фирмы 3Com, BayNetworks (недавно компания IMortel купила BayNetworks), CNet.

В ЛВС с передачей маркера сообщения передаются последовательно от одного узла к другому вне зависимости от того, какую архитектуру имеет сеть – кольцевую или звездообразную. Каждый узел сети получает пакет от соседнего. Если данный узел не является адресатом, то он передает тот же самый пакет следующему узлу. Передаваемый пакет может содержать либо данные, направляемые от одного узла другому, либо маркер. Маркер – это короткое сообщение, являющееся признаком занятости сети. В том случае, когда рабочей станции необходимо передать сообщение, ее сетевой адаптер дожидается поступления маркера, а затем формирует пакет, содержащий данные, и передает этот пакет в сеть. Пакет распространяется по ЛВС от одного сетевого адаптера к другому до тех пор, пока не дойдет до компьютера-адресата, который произведет в нем стандартные изменения. Эти изменения являются подтверждением того, что данные достигли адресата. После этого пакет продолжает движение дальше по ЛВС, пока не возвратится в тот узел, который его сформировал. Узел-источник убеждается в правильности передачи пакета и возвращает в сеть маркер. Важно отметить, что в ЛВС с передачей маркера функционирование сети организовано так, что коллизии возникнуть не могут. Пропускная способность сетей TokenRing равна 16 Мбит/с. Оборудование для сетей TokenRing производят IBM, 3Com и некоторые другие фирмы.

Системы управления сетью

Классификация средств мониторинга и анализа

Инструменты мониторинга и анализа сети

Виды администрирования

Сетевая операционная среда

Операционная среда - набор функций и сервисов ОС и правила обращения к ним. Так же – набор интерфейсов для обращения к функциям и сервисам ОС.

По аналогии: сетевая операционная среда – набор функций и сервисов сети и правила обращения к ним.

Централизованное – наиболее привычное нам, яркие примеры: administrator в Windows и root в Linux. Один или несколько пользователей обладают всей полнотой власти по предоставлению доступа другим пользователям к ресурсам сети.

Распределенное – старшие пользователи наделяют других пользователей правами частичного администрирования, например предоставления доступа к какому-либо одному ресурсу. Характерно для систем с Интернет-архитектурой, как правило использует усложненные системы аутентификации (напр. один и тот же пользователь имеет различные права в зависимости от того, находится он в локальной сети или подсоединяется извне).

Информационная среда – ИС, являющаяся проекцией некоторой области реальности, обладающая обратной связью.

Все многообразие средств, применяемых для мониторинга и анализа вычислительных сетей, можно разделить на несколько крупных классов:

1. *Системы управления сетью (NetworkManagementSystems)* - централизованные программные системы, которые собирают данные о состоянии узлов и коммуникационных устройств сети, а также данные о трафике, циркулирующем в сети. Эти системы не только осуществляют мониторинг и анализ сети, но и выполняют в автоматическом или полуавтоматическом режиме действия по управлению сетью - включение и отключение портов устройств, изменение параметров мостов адресных таблиц мостов, коммутаторов и маршрутизаторов и т.п.

2. *Средства управления системой (SystemManagement)*. Средства управления системой часто выполняют функции, аналогичные функциям систем управления, но по отношению к другим объектам. В первом случае объектом управления является программное и аппаратное обеспечение компьютеров сети, а во втором - коммуникационное оборудование. Вместе с тем, некоторые функции этих двух видов систем управления могут дублироваться, например, средства управления системой могут выполнять простейший анализ сетевого трафика.

3. *Встроенные системы диагностики и управления (Embeddedsystems)*. Эти системы выполняются в виде программно-аппаратных модулей, устанавливаемых в коммуникационное оборудование, а также в виде программных модулей, встроенных в операционные системы. Они выполняют функции диагностики и

управления только одним устройством, и в этом их основное отличие от централизованных систем управления. Как правило, встроенные модули управления "по совместительству" выполняют роль SNMP-агентов, поставляющих данные о состоянии устройства для систем управления.

4. *Анализаторы протоколов (Protocol analyzers)*. Представляют собой программные или аппаратно-программные системы, которые ограничиваются в отличие от систем управления лишь функциями мониторинга и анализа трафика в сетях. Хороший анализатор протоколов может захватывать и декодировать пакеты большого количества протоколов, применяемых в сетях - обычно несколько десятков. Анализаторы протоколов позволяют установить некоторые логические условия для захвата отдельных пакетов и выполняют полное декодирование захваченных пакетов, то есть показывают в удобной для специалиста форме вложенность пакетов протоколов разных уровней друг в друга с расшифровкой содержания отдельных полей каждого пакета.

5. *Оборудование для диагностики и сертификации кабельных систем*. Условно это оборудование можно поделить на четыре основные группы: сетевые мониторы, приборы для сертификации кабельных систем, кабельные сканеры и тестеры (мультиметры).

- Сетевые мониторы (называемые также сетевыми анализаторами) предназначены для тестирования кабелей различных категорий. Следует различать сетевые мониторы и анализаторы протоколов. Сетевые мониторы собирают данные только о статистических показателях трафика - средней интенсивности общего трафика сети, средней интенсивности потока пакетов с определенным типом ошибки и т.п.
- Назначение устройств для сертификации кабельных систем, непосредственно следует из их названия. Сертификация выполняется в соответствии с требованиями одного из международных стандартов на кабельные системы.
- Кабельные сканеры используются для диагностики медных кабельных систем.
- Тестеры предназначены для проверки кабелей на отсутствие физического разрыва.

6. *Экспертные системы*. Этот вид систем аккумулирует человеческие знания о выявлении причин аномальной работы сетей и возможных способах приведения сети в работоспособное состояние. Экспертные системы часто реализуются в виде отдельных подсистем различных средств мониторинга и анализа сетей: систем управления сетями, анализаторов протоколов, сетевых анализаторов. Простейшим вариантом экспертной системы является контекстно-зависимая help-система. Более сложные экспертные системы представляют собой так называемые базы знаний, обладающие элементами искусственного интеллекта.

7. *Многофункциональные устройства анализа и диагностики*. Портативные приборы, совмещающие функции анализаторов протоколов, кабельных сканеров и

некоторые возможности ПО сетевого управления. Получили распространение в задачах диагностики и проектирования больших локальных сетей.

В соответствии с рекомендациями ISO можно выделить следующие функции средств управления сетью:

- Управление конфигурацией сети и именованием - состоит в конфигурировании компонентов сети, включая их местоположение, сетевые адреса и идентификаторы, управление параметрами сетевых операционных систем, поддержание схемы сети; также эти функции используются для именования объектов.
- Обработка ошибок - это выявление, определение и устранение последствий сбоев и отказов в работе сети.
- Анализ производительности - помогает на основе накопленной статистической информации оценивать время ответа системы и величину трафика, а также планировать развитие сети.
- Управление безопасностью - включает в себя контроль доступа и сохранение целостности данных. В функции входит процедура аутентификации, проверки привилегий, поддержка ключей шифрования, управления полномочиями. К этой же группе можно отнести важные механизмы управления паролями, внешним доступом, соединения с другими сетями.
- Учет работы сети - включает регистрацию и управление используемыми ресурсами и устройствами. Эта функция оперирует такими понятиями как время использования и плата за ресурсы.

Из приведенного списка видно, что системы управления выполняют не только функции мониторинга и анализа работы сети, необходимые для получения исходных данных для настройки сети, но и включают функции активного воздействия на сеть - управления конфигурацией и безопасностью, которые нужны для отработки выработанного плана настройки и оптимизации сети. Сам этап создания плана настройки сети обычно остается за пределами функций системы управления, хотя некоторые системы управления имеют в своем составе экспертные подсистемы, помогающие администратору или интегратору определить необходимые меры по настройке сети.

Применение электронных и сетевых средств коммуникаций

В настоящее время деловая и личная переписка в абсолютном своем большинстве осуществляется благодаря разнообразным средствам электронной связи. Это удобно, комфортно, быстро, безопасно и т.д. Современные электронные коммуникации поддерживаются посредством компьютера, подключенного к сети. Также в ней происходит обмен файлами и сообщениями, непосредственная виртуальная связь с коллегами при помощи сетевой системы электронной почты (для чего используется одна или несколько информационных служб, обеспечивающих контакты с партнерами) и для факсимильных сообщений для связи с людьми и фирмами, которые не имеют подключения к электронной почте.

Электронная почта (e-mail) базируется на сетевом использовании компьютеров, она предоставляет различные возможности в зависимости от используемого программного обеспечения, позволяет пользователю получать, хранить и отправлять сообщения своим партнерам. Сеть может быть использована как однонаправленная связь, так и двухсторонняя (многократный процесс отправки и принятия сообщений по электронной почте). Посылаемое сообщение, например, чтобы стало доступно всем пользователям электронной почты, может быть размещено на компьютерной доске объявлений. Также можно отправить сообщение с уведомлением о его получении адресатом. Потенциал современной электронной почты многообразен и не ограничивается указанными спецификами.

Компьютерные сетевые конференции основаны на применении компьютерных систем для обмена информацией между сотрудниками, решающими определенную проблему. Преимущество данной технологии заключается в том, что количество участников компьютерной конференции может быть во много раз больше, чем во время проведения аудио- или видеоконференций. Кроме того, у пользователей сети есть возможность одновременного доступа к практически неограниченным объемам накопленной информации. Проведение подобных мероприятий целесообразно тогда, когда нет необходимости в конкретном количестве и составе людей, принимающих в нем участие. Это может быть, например, некая разновидность референдума или опроса мнений и пр.

Видеотекст как вид технологически обработанной информационной коммуникации, основан на использовании компьютера для получения отображения сменяемых друг друга текстовых и графических данных на экране монитора. Это своего рода электронный аналог телеграфной информационной ленты, применяемой в различных телеграфных и информационных агентствах прошлого.

Для компаний, принимающих решение об использовании видеотекста, имеются три возможности получения информации в форме видеотекста:

- создание файлов видеотекста на собственных компьютерах;
- заключение договора со специализированной компанией на получение доступа к разработанным ею файлам видеотекста (файлы, специально предназначенные для продажи, могут храниться на серверах компании, осуществляющей подобные услуги, или поставляться клиенту на магнитных или оптических дисках);
- заключение договора с другими компаниями на получение доступа к их файлам видеотекста.

Когда различные представители бизнеса рассматривают друг друга не столько как конкурентов-соперников, а как партнеров, в деловом мире популярным становится обмен каталогами и ценниками (прайс-листами) продукции. В условиях хорошо налаженной системы информационного обеспечения между участниками-партнерами какого-либо производственного процесса обмен значительного объема специализированной оперативной информации целесообразно проводить в форме видеотекста. Услуги компаний, работающих в сфере разработки и продажи технологических систем трансляции видеотекста, начинают

конкурировать с печатной продукцией. Так, актуальным становится заказ текущих сводок биржевой информации, газеты или журнала в форме видеотекста.

Хранение изображений. В любой фирме со временем накапливается большое количество документов. Их бывает так много, что хранение даже в форме файлов вызывает серьезные проблемы. Поэтому эффективным представляется архивирование не самого документа, а его образа (изображения) в цифровой форме.

Хранение изображений (imaging) - перспективная офисная технология, основанная на использовании специального устройства - оптического распознавателя образов, позволяющего преобразовывать изображение документа или фильма в цифровой вид для дальнейшего хранения во внешней памяти компьютера. Сохраненное в цифровом формате изображение в любой момент может быть выведено в его реальном виде на экран или принтер. Для хранения изображений используются оптические диски, обладающие огромными емкостями. Идея хранения изображений не нова, так, раньше для этого использовались микрофильмы и микрофиши. Данная технология появилась благодаря созданию оптического диска в комбинации с цифровой записью изображения.

Аудиопочта представляет собой систему компьютерной связи для передачи сообщений голосом. Она напоминает электронную почту, где вместо набора сообщения на клавиатуре компьютера происходит его передача посредством телефона, таким же образом осуществляется и получение информации. Данная система включает в себя специальное устройство для преобразования аудиосигналов в цифровой код и обратно, а также компьютер для хранения аудиосообщений в цифровой форме. Реализация аудиопочты происходит также в сети.

Почта для передачи аудиосообщений эффективно используется в различных областях человеческой деятельности, где необходима оперативность и однозначность передаваемого блока информации. Она часто используется журналистами или административно-управленческим аппаратом для проведения совещаний и коллегиального решения проблем в организации. Главное преимущество аудиопочты по сравнению с электронной в ее простоте - нет необходимости ввода данных с клавиатуры при ее использовании.

Факсимильная связь основана на использовании факс-аппарата, способного читать документ на одном конце коммуникационного канала и воспроизводить его изображение на другом. В настоящее время очень часто можно слышать утверждение о том, что этот вид коммуникации уже отжил свое. На смену факсу пришло более совершенное оборудование, а документооборот на бумажных носителях стал архаизмом. Однако давайте не будем забывать о юридических и правовых аспектах любого бизнеса.

Легитимность любой деятельности легче подтверждается тогда, когда она сопровождается документооборотом с высокой степенью защищенности от возможных несанкционированных изменений или двояких толкований. А это условие на сегодняшний день гарантируется именно использованием только традиционных носителей в делопроизводстве. Поэтому факс-аппаратура еще долго будет стоять на вооружении деловых кругов. Факсимильная связь полезна: в архивном деле; в аналитике; в научной деятельности, связанной с различными историческими исследованиями; в избирательном процессе всевозможных выборов и т.д.

То есть тогда, когда есть необходимость с высокой скоростью и на значительное расстояние переместить документальную информацию, имеющую высокую степень надежности. Факсимильная связь, например, вносит свой вклад в принятие решений за счет быстрой и легкой рассылки документов участникам группы, решающей определенную проблему, независимо от их физического местонахождения.

Аудиоконференция представляет собой голосовое общение нескольких лиц, которые находятся в различных географических точках, посредством использования системы коммуникаций, компьютерных технологий или селекторно-телефонной связи. Оперативное коллегиальное обсуждение экстренно возникших ситуаций и вопросов, требующих участия не только лиц, уполномоченных принимать решения, без этого вида коммуникации. Техническим средством реализации аудиоконференции является телефонная связь, оснащенная дополнительными устройствами, что позволяет взаимодействовать более чем двум участникам, при этом не требуется обязательного наличия компьютера. Использование аудиоконференции облегчает принятие тех или иных решений, оно довольно дешево и удобно.

Аудиоконференции позволяют избежать или значительно сократить ненужные затраты на командировки. Людям приходится ездить в командировки по множеству причин: чтобы встретиться с клиентом, заключить крупную сделку (и отпраздновать ее потом), решить сложную проблему при личном участии, пообщаться с коллегами на выставке или съезде. Однако очень часто без этих поездок можно и обойтись, например, когда необходимо привлечь внимание клиента (иногда, кажется, что другого способа просто нет), переговорить с одним или двумя людьми по поводу проекта, провести стандартную презентацию для коллег, пообщаться напрямую, в деловой среде, тем более что программы работы на дому и локальные небольшие офисы получают все большее распространение ввиду их преимуществ с точки зрения затрат, продуктивности и стиля жизни.

Видеоконференция - система с двухсторонней передачей видеоизображений. Данная система связи используется коммуникантами, удаленными друг от друга на значительное расстояние, но при обязательном условии участия всех членов коммуникации в визуальном контакте. Известен целый ряд специфических видов деловых связей, когда важное значение имеет не только что говорят (о чем, в соответствии с чем, на основании чего и т.д.), но и как выглядит говорящий. Невербальное поведение участников коммуникации часто содержит в себе значительную долю информации, необходимой для оценки истинности намерений собеседников. Кроме этого, с применением видеоконференции (телемост, теледебаты, открытые селекторные видеосовещания административно-управленческих подразделений государства и т.д.) можно добиться у потенциального зрителя эффекта присутствия или сопричастности, что очень часто бывает необходимым в рамках различных политехнологических мероприятий. Проведение видеоконференций осуществляется без применения компьютера. Посредством видеоаппаратуры одновременно с телевизионным изображением передается звуковое сопровождение. Видеоконференции позволяют сократить транспортные и командировочные расходы, вместе с тем данный тип связи позволяет привлечь к решению проблем максимальное количество менеджеров и других сотрудников, территориально удаленных от главного офиса. Наиболее популярны три конфигурации построения видеоконференций:

А) односторонняя видео- и аудиосвязь (сигналы идут только в одном направлении, например, от руководителя проекта к исполнителям);

Б) односторонняя видео- и двухсторонняя аудиосвязь (дает возможность участникам конференции, принимающим видеоизображение, обмениваться аудиоинформацией с передающим видеосигнал участником);

В) двухсторонняя видео- и аудиосвязь (видео- и аудио-связь между всеми участниками конференции).

Заключение

В целом, в 21 веке электронные коммуникации занимают больше позитивных, положительных аспектов. Электронные средства выступают вещественным, материальным компонентом коммуникативного процесса. Изучение таких средств важно, так как в последнее время мы

замечаем их быстрое развитие. Сегодня Интернет является самым популярным средством передачи информации и в ближайшем будущем эта популярность не спадет, а вырастет. Скорее всего, новые технологии не будут автоматически заменять старые, такие как газеты и журналы, радио и телевидение. Эти сферы вещания приспособятся к новым экономическим условиям.

Средства электронной коммуникации активно участвуют в процессе становления общества сегодня. Электронное общение прочно вошло в деловую жизнь, но, тем не менее, никогда не сможет заменить общение в реальном мире. Информационное общество представляет собой новый этап развития человеческой цивилизации, характеризующейся, в первую очередь, высокой скоростью коммуникационных процессов, которая обеспечивается наукоемкими, высокотехнологичными средствами - микропроцессорными технологиями и сетью Интернет. Распространение Интернет-технологий на все сферы жизнедеятельности индивидов и социальных групп становится одним из определяющих факторов социально-экономического развития общества.

Характеристика автоматизированного рабочего места

Автоматизированное рабочее место (АРМ) — это рабочее место специалиста, оснащенное персональным компьютером, программным обеспечением и совокупностью информационных ресурсов индивидуального или коллективного пользования, которые позволяют ему вести обработку данных с целью получения информации, обеспечивающей поддержку принимаемых им решений при выполнении профессиональных функций.

Создание АРМ предполагает, что основные операции по накоплению, хранению и переработке информации возлагаются на вычислительную технику, а экономист выполняет часть ручных операций и операций, требующих творческого подхода при подготовке управленческих решений.

Персональная техника применяется пользователем для контроля производственно - хозяйственной деятельности, изменения, значений отдельных параметров в ходе решения задачи, а также ввода исходных данных в АИС для решения текущих задач и анализа функций управления. Анализируя сущность АРМ, специалисты определяют их чаще всего как профессионально - ориентированные малые вычислительные системы, расположенные непосредственно на рабочих местах специалистов и предназначенные для автоматизации их работ. Структура АРМ-а специалиста, включает в себя пять основных компонентов:

персональный компьютер;

комплекс программ для обработки информации;

обучающую систему (гипертекстовую систему документации для пользователя; интегрированную систему подсказок; систему закладок, указателей и справок; систему примеров; систему контроля и обнаружения ошибок);

средства настройки АРМ (алгоритмов расчетов, аналитических и технологических параметров; устройств: принтера, сканера, модема; эргономики экранных форм и т. д.);

средства эксплуатации АРМ (классификаторы, генератор отчетных форм, инструментарий приема/передачи данных по каналам связи, копирования и сохранности данных, администратор баз данных, мониторинг работы конкретных пользователей).

Кроме этого, АРМ комплектуется документацией и методическими материалами по применению программ, а также регламентами выполнения работ по обработке информации. Конкретная насыщенность каждой из компонент определяется решаемыми задачами. АРМ могут функционировать автономно или в составе компьютерной сети. При автономном режиме работы АРМ создаются для решения отдельных функциональных задач и не могут оперативно использовать всю информационную базу экономического объекта, а обмен информацией между различными АРМ выполняется с помощью машинных носителей. Работа на базе компьютерных сетей позволяет организовать обмен данными между АРМ по каналам связи, объединить информационное пространство объекта управления и организовать доступ к нему любого работника в пределах его полномочий.

Каждое АРМ рассматривается как самостоятельная подсистема, а вместе они составляют единое целое. При этом начальник отдела имеет возможность руководить процессом решения функциональных задач и интегрировать результаты работы отдельных специалистов, оперативно получая обработанную информацию для принятия решений. В то же время сохраняется возможность автономной работы каждого специалиста. Как правило, АРМ организуются в соответствии с существующим распределением работ. В зависимости от объема работ и общего числа компьютеров на одном рабочем месте могут решаться различные задачи. Возможен и другой вариант, когда одна задача будет распределена между несколькими рабочими местами.

Классификация автоматизированных рабочих мест.

1. По степени автоматизации:

- ручные рабочие места — имеющиеся в распоряжении работника специальная мебель (стол, стул, шкафы, телефон, линейки, таблицы и другие подсобные средства);

- механизированные рабочие места, кроме того, содержат простейшие или программируемые калькуляторы;

- автоматизированные рабочие места обязательно используют ПК с соответствующим программным обеспечением.

2. По количеству сотрудников, использующих АРМ, и выполняемых ими функций:

- индивидуальные АРМ, которые характерны для руководителей различных рангов;

- групповые АРМ, используемые лицами, которые готовят информацию с целью ее дальнейшего использования и принятия управленческих решений руководителями (АРМ бухгалтеров, финансистов, делопроизводителей и др.).

3. По типизации решаемых функциональных задач:

- уникальные АРМ, узкоспециализированные для решения набора нестандартных задач;

- массовые АРМ, создаваемые для решения типичных задач в различных отраслях.

4. По специализации: АРМ руководителя характеризуется функциональной замкнутостью, полностью обеспечивающей автономную работу руководителя. АРМ специалиста должно предоставить ему возможность решать любые стоящие перед ним функциональные задачи, максимально используя всю необходимую информацию. АРМ технического работника должно избавить его от каждодневного выполняемой рутинной работы, требующей определенных профессиональных навыков.

5. По технической базе создания АРМ: АРМ на базе больших (универсальных) ЭВМ, обеспечивающих специалистам возможность работать с большими массивами данных при технической и программной поддержке, осуществляемой силами работников собственного информационного центра (ИВЦ). АРМ на базе персональных компьютеров являются наиболее простым и распространенным вариантом создания автоматизированных рабочих мест, так как избавляют от всех недостатков АРМ на базе больших ЭВМ.

Общие принципы создания АРМ остаются неизменными, к ним относят: системность; гибкость; устойчивость; эффективность. Под принципом

системности понимается следующее: автоматизированное рабочее место должно представлять собой систему взаимосвязанных компонентов. При этом структура АРМ должна четко соответствовать тем функциям, для выполнения которых создается данное автоматизированное рабочее место.

Принцип гибкости имеет огромное значение при создании современных и эффективно работающих автоматизированных рабочих мест. Данный принцип означает возможность приспособления АРМ к предполагаемой модернизации как программного обеспечения, так и технических средств. В настоящее время, когда скорость устаревания программных и технических средств постоянно растет, соблюдение данного принципа становится одним из важнейших условий при создании АРМ. Для обеспечения принципа гибкости в реально работающих автоматизированных рабочих местах все подсистемы отдельно взятого АРМ выполняются в виде отдельных, легко заменяемых модулей. Чтобы при замене не возникало проблем несовместимости, все элементы должны быть стандартизированы.

Большое значение имеет принцип устойчивости. Он заключается в выполнении заложенных в АРМ функций, независимо от воздействия как внутренних, так и внешних факторов. При возникновении сбоев работоспособность системы должна быстро восстанавливаться, неполадки отдельных элементов должны легко устраняться.

Принцип эффективности подразумевает, что затраты на создание и эксплуатацию системы не должны превышать экономическую выгоду от ее реализации. Кроме того, при создании АРМ надо учитывать, что его эффективность будет во многом определяться правильным распределением функций и нагрузки между работником и машинными средствами обработки информации, ядром которых является ПЭВМ. Только при соблюдении этих условий АРМ становится средством повышения не только производительности труда и эффективности управления, но и социальной комфортности специалистов.

Необходимо разработать автоматизированное рабочее место специалиста кафедры. Программное обеспечение должно иметь интуитивный понятный интерфейс, быть простым в использовании, удобным, информативным, гибким и многофункциональным. Суть автоматизированного рабочего места специалиста кафедры заключается в следующем: формирования индивидуальной нагрузки; создания графиков контрольных, самостоятельных, консультационных работ, графиков защиты курсовых работ. Графики создаются для удобства в Microsoft Word.

Использование возможностей современной вычислительной техники для автоматизации процесса обработки информации позволяет увеличить

производительность труда, повысить эффективность работы с документами и ускорить обмен управленческой информацией. В настоящее время большое распространение получила концепция распределенных автоматизированных систем управления, направленных на локальную обработку информации. Это позволяет организовать разделение труда управленческого персонала и автоматизировать выполнение им своих функций. Для реализации данной идеи необходимо создание для каждого уровня управления и каждой предметной области автоматизированных рабочих мест на базе персональных электронно-вычислительных машин (ПЭВМ).

В настоящее время большое распространение получила концепция распределенных автоматизированных систем управления, направленных на локальную обработку информации. Это позволяет организовать разделение труда управленческого персонала и автоматизировать выполнение им своих функций. Для реализации данной идеи необходимо создание для каждого уровня управления и каждой предметной области автоматизированных рабочих мест на базе персональных электронно-вычислительных машин (ПЭВМ).

Распределенных систем управления позволяет выделить следующие требования к эффективно и полноценно функционирующему автоматизированному рабочему месту:

- своевременное удовлетворение информационных потребности пользователя;
- минимальное время ответа на запросы пользователя;
- адаптация к уровню подготовки пользователя и специфике выполняемых им функций;
- возможность быстрого обучения пользователя основным приемам работы;
- надежность и простота обслуживания;
- дружественный интерфейс;
- возможность работы в составе вычислительной сети.

АРМ состоит из технических и программных средств (рис.10) вычислительной техники, а также необходимой методической документации, позволяющей пользователю эффективно взаимодействовать с данными средствами.

Для процветания фирм необходимо качественное и своевременное управление их деятельностью. Главным условием правильности принимаемых решений является исчерпывающая осведомленность.

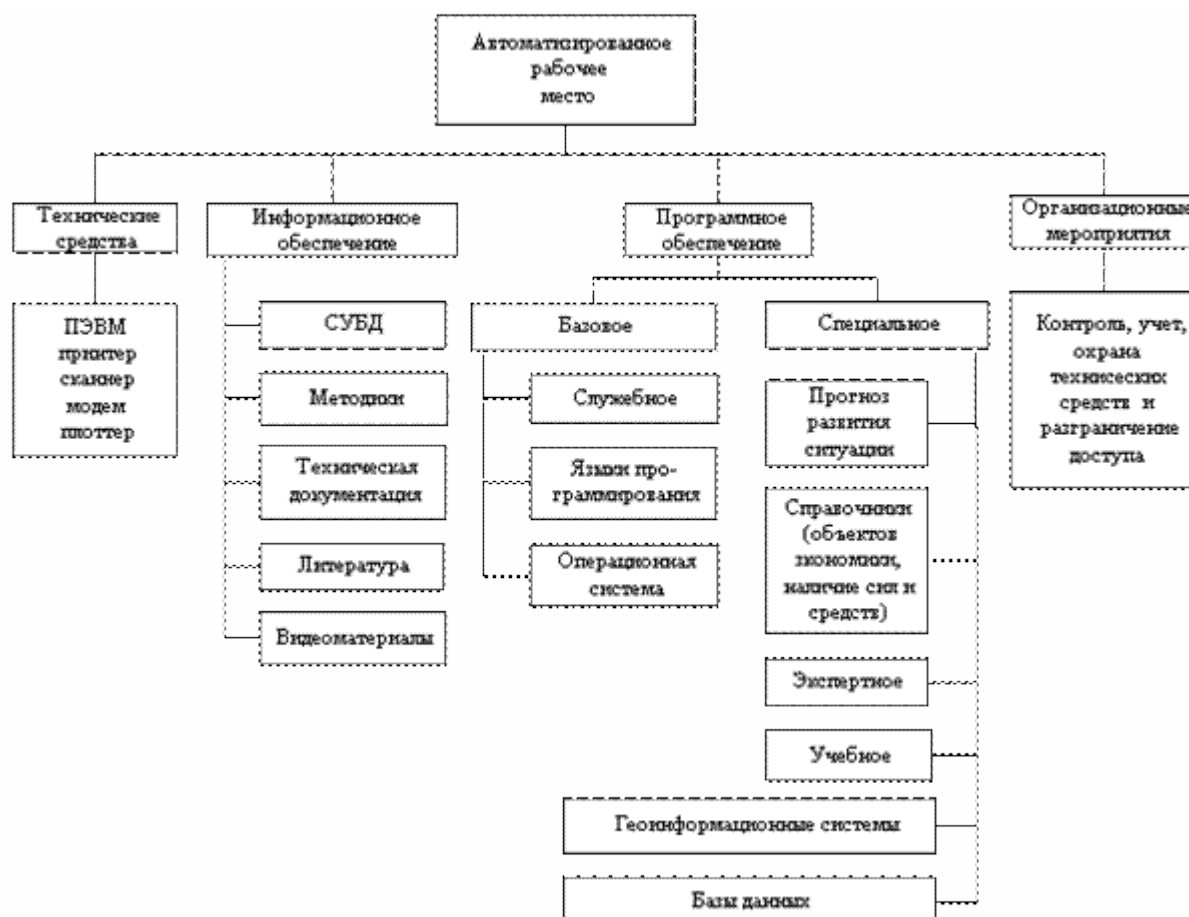


Рис. 10. Состав АРМ

Для процветания фирм необходимо качественное и своевременное управление их деятельностью. Главным условием правильности принимаемых решений является исчерпывающая осведомленность. Выполнение этого условия в настоящее время возможно только на базе современных информационных технологий. Поэтому сейчас все фирмы, от организаций малого и среднего бизнеса до крупных промышленных комплексов, разрабатывают и внедряют автоматизированные информационные системы поддержки своей деятельности. Внедрение подобных систем повышает оперативность и достоверность получаемой информации, сокращается время на текущий сбор и обработку информации, а это приводит к высвобождению рабочего времени, как руководителей и специалистов для решения перспективных проблем, так и рядовых сотрудников для более качественного и продуктивного выполнения своих функций.

Программное обеспечение автоматизированного рабочего места

Программное обеспечение АРМ включает следующие компоненты:

- операционные системы;
- языки и системы программирования;
- прикладное программное обеспечение (ППП): общесистемные (базовые) ППП и проблемно-ориентированные ППП профессионального назначения.

Операционная система представляет собой группу программ, которые обеспечивают управление ресурсами компьютера, поддержку работы всех программ, их взаимодействия с аппаратными средствами и предоставляют пользователю возможность общего управления компьютером. Операционная система управляет компьютером, периферийным оборудованием, запускает программы, обеспечивает защиту данных, выполняет различные сервисные функции по запросам пользователя и программ. Каждая программа пользуется услугами ОС, а потому может работать только под управлением той ОС, которая обеспечивает для нее эти услуги. Таким образом, выбор ОС для ЭВМ в составе АРМ чрезвычайно важен, так как он определяет, какие программы будут работать на АРМ, какие аппаратные средства будут включены в АРМ, какова будет степень защиты данных, и насколько комфортными и безопасными будут условия работы для специалиста АРМ. Наиболее широко в настоящее время на компьютерах типа IBM PC применяются ОС, разработанные фирмой Microsoft, это существующие версии Windows.

Современная концепция АРМ предъявляет к ОС жесткие требования, направленные на соблюдение безопасности и комфорта (эргономичность) в работе на АРМ, на повышение производительности АРМ в целом, на расширение парка обслуживаемого периферийного оборудования, на возможность синхронизации выполняемых операций и процедур.

Общее программное обеспечение (ПО) обеспечивает функционирование вычислительной техники, разработку и подключение новых программ. Сюда входят операционные системы, системы программирования и обслуживающие программы.

Профессиональная ориентация АРМ определяется функциональной частью ПО (ФПО). Именно здесь закладывается ориентация на конкретного специалиста, обеспечивается решение задач определенных предметных областей.

При разработке ФПО очень большое внимание уделяется вопросам организации взаимодействия “человек-машина”. Пользователю интересно и увлекательно работать на ЭВМ только в том случае, когда он чувствует, что он занимается полезным, серьезным делом. В противном случае его ждут неприятные ощущения. Непрофессионал может почувствовать себя обойденным и даже в чем-то ущемленным только потому, что он не знает неких “мистических” команд, набора символов, вследствие чего у него может возникнуть глубокая досада на все программное обеспечение или служителей культа ЭВМ.

Анализ диалоговых систем с точки зрения организации этого диалога показал, что их можно разделить (по принципу взаимодействия пользователя и машины) на:

- системы с командным языком;
- “человек в мире объектов”;
- диалог в форме “меню”.

Применение командного языка в прикладных системах это перенос идей построения интерпретаторов команд для мини- и микро ЭВМ. Основное его преимущество - простота построения и реализации, а недостаток - продолжение их достоинств: необходимость запоминания команд и их параметров, повторение ошибочного ввода, разграничение

доступности команд на различных уровнях и пр. Таким образом в системах с командным языком пользователь должен изучать язык взаимодействия.

Внешне противоположный подход “человек в мире объектов” - отсутствуют команды и человек в процессе работы “движется” по своему объекту с помощью клавиш управления курсором, специальных указывающих устройств (мышь, перо), функциональных комбинаций клавиш. Диалог в форме меню “меню” представляет пользователю множества альтернативных действий, из которых он выбирает нужные. В настоящее время наиболее широкое распространение получил пользовательский интерфейс, сочетающий в себе свойства двух последних. В нем все рабочее пространство экрана делится на три части (объекта). Первая (обычно располагающаяся вверху) называется строкой или полосой меню. С ее помощью пользователь может задействовать различные меню, составляющие “скелет” программы, с их помощью производится доступ к другим объектам (в т.ч. управляющим). Вторая часть (обычно располагается внизу или в небольших программах может вообще отсутствовать) называется строкой состояния. С ее помощью могут быстро вызываться наиболее часто используемые объекты или же отображаться какая-либо текущая информация. Третья часть называется рабочей поверхностью (поверхностью стола) - самая большая. На ней отображаются все те объекты, которые вызываются из меню или строки состояния. Такая форма организации диалога человека и машины наиболее удобна (по крайней мере на сегодняшний день ничего лучшего не придумано) и все современные программы в той или иной мере используют ее. В любом случае она должна соответствовать стандарту CUA (Common User Access) фирмы IBM.

Рассмотрим теперь два подхода к разработке АРМ. Первый подход - функциональный представляет собой автоматизацию наиболее типичных функций.

Посмотрим, как адаптируется функциональное ПО (ФПО) к конкретным условиям применения. Отметим программные средства, которые являются базовыми при АРМ для различных профессий, связанных с обработкой деловой информации и принятием управленческих решений.

Первыми появились программные средства для автоматизации труда технического персонала, что обусловлено, вероятно, большой формализацией выполняемых ими функций. Наиболее типичным примером являются текстовые редакторы (процессоры). Они позволяют быстро вводить информацию, редактировать ее, сами осуществляют поиск ошибок, помогают подготовить текст к распечатке.

Применение текстовых редакторов позволяют значительно повысить производительность труда машинисток.

Специалистам часто приходится работать с большими объемами данных, с тем чтобы найти требуемые сведения для подготовки различных документов. Для облегчения такого рода работ были созданы системы управления базами данных (СУБД: DBASE, RBASE, ORACLE и др.). СУБД позволяют хранить большие объемы информации, и, что самое главное, быстро находить нужные данные. Так, например при работе с картотекой постоянно нужно прерывать большие архивы данных для поиска нужной информации, особенно если карточки отсортированы не по нужному признаку. СУБД справится с этой задачей за считанные секунды.

Большое число специалистов связано также с обработкой различных таблиц, так как в большинстве случаев экономическая информация представляется в виде табличных документов. КЭТ (крупноформатные электронные таблицы) помогают создавать подобные документы. Они очень удобны, так как сами пересчитывают все итоговые и промежуточные данные при изменении исходных. Поэтому они широко используются, например при прогнозировании объемов сбыта и доходов.

Достаточно большой популярностью в учреждениях пользуются программные средства АРМ для контроля и координации деятельности организации, где вся управленческая деятельность описывается как совокупность процессов, каждый из которых имеет даты начала, конца и ответственных исполнителей. При этом деятельность каждого работника увязывается с остальными. Таким образом, создается план-график работ. Пакет может автоматически при наступлении срока формировать задания исполнителям, напоминать о сроке завершения работы и накапливать данные об исполнительской деятельности сотрудников.[4]

Важную роль в учрежденческой деятельности играет оперативный обмен данными, который занимает до 95% времени руководителя и до 53% времени специалистов. В связи с этим получили распространение и программные средства типа “электронная почта”. Их использование позволяет осуществлять рассылку документов внутри учреждения, отправлять, получать и обрабатывать сообщения с различных рабочих мест и даже проводить совещания специалистов, находящихся на значительном расстоянии друг от друга.

Проблема обмена данными тесно связана с организацией работы АРМ в составе вычислительной сети.

В настоящее время наблюдается тенденция к созданию так называемых интегрированных пакетов, которые вмещают в себя возможности и текстовых редакторов, и таблиц, и графических редакторов. Наличие большого числа различных программ для выполнения в сущности одинаковых операций - создания и обработки данных обусловлено наличием трех различных основных видов информации: числовой, текстовой и графической. Для хранения информации чаще всего используются СУБД, которые позволяют соединять все эти типы данных в единое целое. Сейчас идет бурное развитие двух других видов информации: звуковой и видеоинформации. Для них уже созданы свои редакторы и не исключено что в скором времени эти виды информации станут неотъемлемой частью большинства баз данных.

Хотя современное ФПО отвечает почти всем требованиям, налагаемым на него работниками различных профессий, чего-то все равно всегда не хватает. Поэтому большим плюсом такого ПО является возможность его доработки и изменения. Что же касается разработки новых программных средств в АРМ, то она ведется по двум направлениям: создание нового ПО для новых профессий и специализация ПО для существующих профессий. В настоящее время наблюдается тенденция перехода к созданию АРМ профессионального назначения[2].

Оно выражается в следующем:

- учет решаемых задач;
- взаимодействие с другими сотрудниками;
- учет профессиональных привычек и склонностей;
- разработка не только ФПО, но и специальных технических средств (мышшь, сеть, автоматический набор телефонных номеров и пр.).

Оснащение специалистов такими АРМ позволяет повысить производительность труда учрежденческих работников, сократить их численность и при этом повесить скорость обработки экономической информации и ее достоверность, что необходимо для эффективного планирования и управления.

В нашей стране созданы и успешно применяются замечательные программные продукты, реализующие различные функции и предназначенные для автоматизации традиционного делопроизводства на основе новейших технологий с использованием электронных сетей.

Современные офисные системы разрабатываются российскими компаниями на основе популярных пакетов офисных систем - Lotus Notes, DOCS Open, Microsoft Office и др.

Наиболее известными офисными программами на сегодняшний день являются:

- . Система автоматизации делопроизводства и документооборота «ДЕЛО», разработанная компанией «Электронные офисные системы»;
- . Система автоматизации конфиденциального документооборота и организации потоков работ «ОПТИМА-Work Flow», разработанная фирмой «Оптима»;
- . Система электронного документооборота и автоматизации делопроизводства компании «ИнтерТраст»;
- . Системы автоматизации делопроизводства, документооборота и управления деловыми процессами крупных организаций и предприятий на базе DOCS Open и WorkRoute, выпускаемые компанией «Весть-Метатехнология».

Системы КонсультантПлюс стали неотъемлемой частью правового обеспечения современного предприятия. Без справочных правовых систем сегодня немыслима работа руководителей, бухгалтеров и юристов. Охватывая полный спектр областей современного законодательства, системы КонсультантПлюс позволяют специалистам успешно решать весь комплекс правовых вопросов, связанных с их профессиональной деятельностью.

Пакетные поставки систем занимают особое место в практике обеспечения предприятий правовой информацией. Фактически они позволяют реализовать комплексный подход к задаче правовой информатизации как для предприятия в целом, так и для отдельных специалистов. Гибкость пакетных поставок обеспечивает максимальный учет интересов конкретного пользователя. При установке пакетов учитываются такие факторы, как сфера деятельности предприятия, его масштаб, структура, территориальное расположение, международные связи. Основой для формирования пакета являются системы по федеральному и региональному законодательству. Технология КонсультантПлюс позволяет учесть уровень используемой компьютерной техники и установить пакеты систем, эффективно работающие практически на любых компьютерах, любых программных платформах и в сетях любой конфигурации.

Существует множество программ для составления и редактирования служебных документов. Такие программы называются текстовыми редакторами или текстовыми процессорами (Word и др.)

При использовании этих программ для составления документов текст редактируемого документа выводится на экран дисплея и в него можно вносить необходимые изменения, переносить фрагменты из одного места документа в другое, объединять документы, использовать различные шрифты для выделения отдельных участков текста, легко работать с таблицами и списками, многократно распечатывать текст на принтере и т.д. программы включают словарь синонимов делового языка, проверку орфографии и грамматики, современное стилевое оформление документа, возможность составления документа на нескольких языках и многое другое.

Составление документа в текстовом редакторе можно производить с использованием «заготовок» документов разных видов, заранее введенных в память ПК.

Большую популярность у секретарей приобрели текстовые процессоры Word 2003, Word 2007.

В программный пакет Word включены шаблоны документов, которые помогают создавать конкретный документ - стандартное, изысканное и современное письмо, а также резюме,

факс. Можно выбрать интересующий вариант документа, отредактировать его, заполнить изменяемые реквизиты: дату, номер документа, подпись, адресат, фамилию и телефон исполнителя и т.д. В окне предварительного просмотра можно увидеть общий вид созданного документа и расположение текста в нем. В новом текстовом редакторе Word 2007 можно не только обнаружить ошибки, но и узнать почему необходимо внести исправление.

Таким образом, новый редактор одновременно является учебником орфографии и пунктуации.

Созданные в текстовом редакторе документы запоминаются, хранятся и пересылаются в виде файлов или выводятся на печать на бумажный носитель.

Автоматизированное рабочее место (АРМ). Автоматизация основных функций.

Автоматизированное рабочее место (АРМ) - это рабочее место специалиста в предметной области, оборудованное компьютером и специальным программным обеспечением, помогающее решать задачи в рамках деятельности этого специалиста или другими словами это программно-технический комплекс, предназначенный для автоматизации деятельности определенного вида (например АРМ бухгалтера, АРМ конструктора, АРМ технолога и т.п.).

Принципы Автоматизированного рабочего места

Существуют общие принципы создания любых автоматизированных рабочих мест

- *Системность* - согласно данному принципу, автоматизированные рабочие места следует рассматривать как системы, структура которых определяется функциональным назначением.
- *Гибкость* - данный принцип означает приспособленность системы к возможным перестройкам, благодаря модульности построения всех подсистем и стандартизации их элементов.
- *Устойчивость* - заключается в том, что система автоматизированного рабочего места должна выполнять функции не зависимо от воздействия на нее внутренних или внешних факторов. Это также означает, что при неполадках в отдельных частях системы должны быть легко устранимы, а работоспособность легко восстанавливаемая.
- *Эффективность* - согласно данному принципу эффективность использования АРМ следует рассматривать как интегральный показатель всех уровней

реализации приведенных выше принципов, по отношению к затратам на создание и эксплуатацию системы.

Функционирование автоматизированного рабочего места дает желаемый эффект при условии правильного распределения функций и нагрузки между человеком и машинными средствами обработки информации

АРМ могут быть: *индивидуальными, групповыми, коллективными.*

Применительно к групповым и коллективным АРМ в целях эффективного функционирования системы ЭВМ - специалистам (коллективу) необходимо ужесточить требования к организации работы АРМ и четко определить функции администрирования в такой системе. Система АРМ, являющаяся человеком-машиной, должна быть открытой, гибкой, приспособленной к постоянному развитию и совершенствованию. В такой системе должны быть обеспечены:

Задачи решаемые на автоматизированных рабочих местах делятся на:

Информационные- к таким задачам относятся кодирование, классификация, сбор, структурная организация, корректировка, хранение, поиск и выдача информации. Часто информационные задачи включают несложные вычислительные и логические процедуры арифметического и текстового характера и отношения (связи). Информационные задачи являются, как правило, наиболее трудоемкими и занимают большую часть рабочего времени специалистов.

Вычислительные - эти задачи являются как формализуемыми, так и не полностью формализуемыми. Формализуемые задачи решаются на базе формальных алгоритмов и делятся на две группы: задачи прямого счета и задачи на основе математических моделей. Задачи прямого счета решаются с помощью простейших алгоритмов. Для более сложных задач требуется применять различные математические модели.

Структура типового автоматизированного рабочего места

Автоматизированное рабочее место представляет собой место пользователя-специалиста какой-либо профессии, оборудованное средствами автоматизации для выполнения им определенных функций, поэтому структура автоматизированного рабочего места представляет собой программно-аппаратный комплекс.

При этом главную роль комплекса играет программная составляющая, которая и будет предъявлять определенные требования по техническим характеристикам (в основном по производительности) к аппаратным составляющим.

Программное обеспечение состоит из системного программного обеспечения и прикладного. Основой системного обеспечения является операционная система. Системные программы обеспечивают рациональную технологию обработки информации. Так называемые сервисные программы, которыми АРМ комплектуется в зависимости от потребности в них, расширяют возможности операционной системы.

Прикладное программное обеспечение составляют программы пользователей и пакеты прикладных программ разного назначения. Стандартные программы пользователей представляют собой программные решения определённых задач. Почти все прикладное программное обеспечение выполнено по модульному принципу и ориентированы на решение определенного класса задач. Прикладное программное обеспечение является основным видом пользовательского программного обеспечения. Оно в основном и позволяет автоматизировать деятельность определенного специалиста.

При этом прикладное программное обеспечение может быть как покупным, т.е. сторонних производителей программного обеспечения, или может быть создано собственными разработками. Следует отметить, что разработка программного обеспечения - это процесс очень сложный, дорогостоящий и доступный специалистам высокой квалификации.

Техническое обеспечение представляет собой комплекс технических аппаратных средств, основой которого служит профессиональный персональный компьютер, предусматривающий работу специалиста без посредников (программистов, операторов и др.). У групповых автоматизированных рабочих мест таким компьютером могут пользоваться несколько человек. В комплект типового персонального компьютера входит: системный блок (состоящий из материнской платы, процессора, оперативной памяти, жесткого диска, видео-контроллера, аудио-контроллера и сетевого-контроллера), монитор, клавиатура, мышка, при необходимости периферийные устройства (сканеры, принтеры).

К комплексу технических средств следует отнести и средства коммуникаций, локальная компьютерная информационная сеть (Intranet - это внутренняя частная сеть предприятия) для связи различных АРМ между собой, а также средства телефонной связи.

Так как АРМы предназначены для автоматизации мест пользователей-специалистов всех проектных и офисных профессий, поэтому прежде всего необходимо выбрать базовое прикладное программное обеспечение для каждой наиболее значимой профессии.

Программы призваны брать на себя задачи, которые человек делает медленнее или для выполнения которых требуются специальные знания.

Информационные системы

Информационная система (ИС) - это совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией.

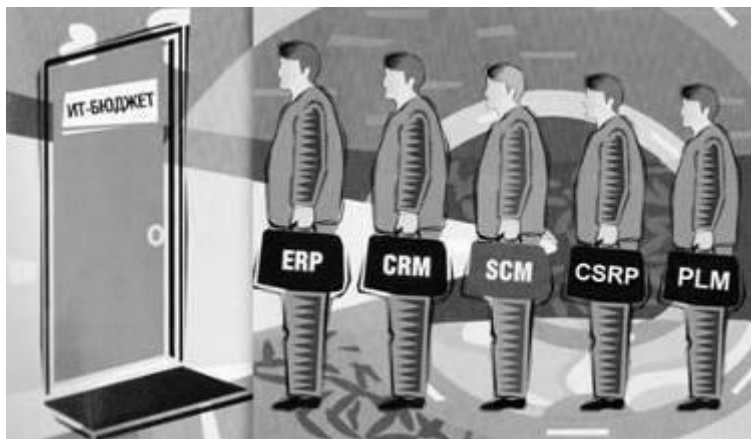


Рисунок 4. Информационные системы предприятия

В настоящее время на отечественном рынке преобладают две основные тенденции разработки и внедрения автоматизированных систем управления.

Первая заключается в том, что предприятие пытается постепенно внедрить системы автоматизации лишь на отдельных участках своей деятельности, предполагая в дальнейшем объединить их в общую систему либо довольствуясь «кусочной» («лоскутной») автоматизацией. Несмотря на то, что этот путь на первый взгляд кажется менее затратным, опыт внедрения таких систем показывает, что минимальные затраты в подобных проектах чаще всего оборачиваются и их минимальной отдачей, а то и вовсе не приносят желаемого результата. К тому же сопровождение и развитие таких систем чрезвычайно затруднено и затратно.

Вторая тенденция - комплексное внедрение систем автоматизации, что позволяет охватить все звенья системы менеджмента - от низового уровня производственных подразделений до верхнего управленческого уровня. В этом случае такая система включает в себя:

- автоматизацию многих направлений деятельности предприятия (бухгалтерский учет, управление персоналом, сбыт, снабжение и т.д.);
- автоматизацию основных технологических процессов предприятия;
- автоматизацию собственно управленческих процессов, процессов анализа и стратегического планирования.

В настоящее время в мировой практике для обозначения полнофункциональных интегрированных АСУ, используемых предприятиями, применяют следующие названия (таблица 2.)

Таблица 2.

Принципы создания	Сокр (англ)	Наименование (рус)	Сокр (рус)	Описание
Manufacturing Resource Planning	MRP	Планирование производственных ресурсов	система планирования потребностей в материалах, одна из наиболее популярных в мире логистических концепций, на основе которой разработано и функционирует большое число микрологистических систем.	
Enterprise Resource Planning	ERP	Планирование ресурсов предприятия	организационная стратегия интеграции производства и операций, управления трудовыми ресурсами, финансового менеджмента и управления активами, ориентированная на	

			непрерывную балансировку и оптимизацию ресурсов предприятия посредством специализиро ванного интегрирован ного пакета прикладного программног о обеспечения, обеспечиваю щего общую модель данных и процессов для всех сфер деятельности	
Customer Synchronized Relationship Planning	ERP-II и CSRP	Планировани е ресурсов, синхронизир ованное с покупателем		
Computer Aided Design / Computer Aided Manufacturing	CAD/CAM	системы автоматизиро ванного проектирован ия / изготовления	САПР	автоматизиро ванная система, реализующая информацион ную технологию выполнения функций проектирован ия, представляет собой организацион но- техническую

				систему, предназначенную для автоматизации и процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации и его деятельности.
EnterpriseResourcePlanning	ERP	комплексная автоматизированная система управления предприятием	АСУП	организационная стратегия интеграции производства и операций, управления трудовыми ресурсами, финансового менеджмента и управления активами, ориентированная на непрерывную балансировку и оптимизацию ресурсов предприятия посредством специализированного интегрированного пакета прикладного

				программного обеспечения, обеспечивающего общую модель данных и процессов для всех сфер деятельности
WorkFlow	WF	потоки работ	графическое представление потока работ в процессе и связанных с ним подпроцессов, включая специфические работы, информационные зависимости и последовательность решений и работ	
CustomerRelationship Management	CRM	управление отношениями с клиентами	прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности,	

			для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживани я клиентов путём сохранения информации о клиентах и истории взаимоотнош ений с ними, установления и улучшения бизнес- процедур и последующег о анализа результатов.	
DecisionSupportSyste m	DSS	поддержка принятия управленческ их решений	СППР	компьютерна я автоматизиро ванная система, целью которой является помощь людям, принимающи м решение в сложных условиях для полного и объективного анализа предметной деятельности .

MIS	управляющая информационная система, (АРМ) руководителя			
SupplyChainManagement	SCM	управление цепями поставок	предназначены для автоматизации и управления всеми этапами снабжения предприятия и для контроля всего товародвижения на предприятии	
HR	«Управление персоналом»; можно рассматривать и как самостоятельную задачу, и как входящую в состав ERP (что и отображено на рисунке в виде двух связей)	совокупность специализированных структурных подразделений в сфере управления предприятием вместе с занятыми в них должностными лицами		
LocalAreaNet	LAN	локальные вычислитель	ЛВС	компьютерная сеть,

		ные сети		покрывающа я обычно относительно небольшую территорию или небольшую группу зданий
WideAreaNet	WAN	глобальные (внешние) сети и телекоммуни кации	компьютерна я сеть, охватывающа я большие территории и включающая в себя большое число компьютеров	

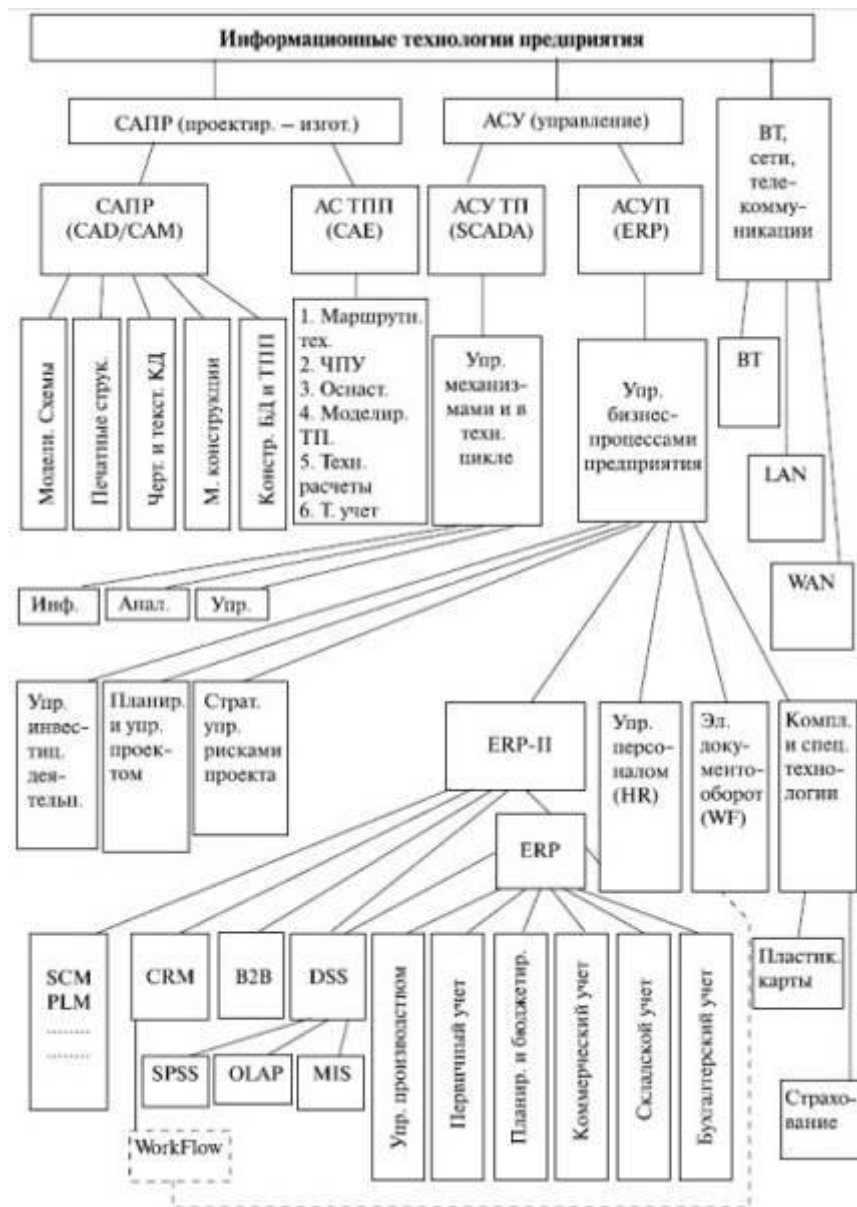


Рисунок 5. Обобщенная структура информационных технологий предприятия

Какой-либо однозначной и общепринятой общей классификации ИТ - предприятий не существует. Возможный вариант обобщенной структуры современных информационных технологий, внедряемых на промышленных производствах различного типа, приведен на рис. 5.

Архитектура сетей арм

1. Основные принципы работы **АРМ** в сети.
Протоколы **Интернета**.
2. Варианты построения архитектуры сетей **АРМ**.

2.1. Одиночные АРМ.

2.2. Групповые системы (сети) АРМ.

2.3. Корпоративные информационные системы (КИС).

3. Архитектура распределенной обработки данных.

3.1. Архитектура *Файл-Сервер*.

3.2. Архитектура *двухуровневый Клиент—Сервер*.

3.3. *Многоуровневый Клиент—Сервер*.

4. Современный подход к функционированию АРМ аналитика, финансиста и бухгалтера в системах распределенной обработки информации.

В прошлой лекции были даны основные понятия построения вычислительных сетей. Для правильного понимания структурно-информационных принципов взаимодействия многочисленных АРМ в таких сетях необходимо четко представлять базовые понятия, на которых строятся информационные технологии обмена информацией.

1. Основные принципы работы арм в сети. Протоколы Интернета.

Сеть *Интернет* объединяет десятки миллионов разнотипных компьютеров - от ПК разных моделей до огромных больших и сверхбольших компьютеров — *мэйнфреймов*, работающих под управлением различных ОС. Обеспечение бесконфликтного их взаимодействия в *Интернет* - весьма сложная задача. *Международная организация по стандартизации (ISO)*, основываясь на опыте многомашинных систем, который был накоплен в разных странах, выдвинула концепцию *архитектуры открытых систем* – т.н. эталонную *семиуровневую модель OSI (Open System Interconnection)*, используемую при разработке

т.н. *сетевых протоколов* общения компьютеров, т.е. международных стандартов правил взаимодействия в сети.

Другими словами, работа компьютерных сетей основана на использовании *сетевых протоколов* - системы правил взаимодействия между собой различных технических и программных компонентов сети.

Основу этой системы составляет *стек протоколов TCP/IP*. В техническом понимании **TCP/IP** — два протокола, лежащих на разных уровнях:

- **TCP (*Transmission Control Protocol*)** — выполняет функции *транспортного уровня* модели взаимодействия открытых систем **OSI**. Он управляет передачей данных.
- **IP (*Internet Protocol*)** - межсетевой протокол, выполняет функции *сетевого уровня* модели **OSI**. **IP - адресный протокол**. Он определяет, куда происходит передача данных.

На основе *протоколов TCP/IP* разработаны многие сетевые прикладные сервисные протоколы:

- **FTP (*File Transfer Protocol*)** — протокол передачи файлов;
- **Telnet** — протокол удаленного доступа, то есть дистанционного исполнения команд на удаленном компьютере;
- **SMTP (*Simple Mail Transfer Protocol*)** — простой протокол пересылки электронной почты;
- **HTTP (*HyperText Transfer Protocol*)** — протокол передачи гипертекста (используется при передаче сообщений в **World Wide Web**);
- **NNTP (*Network News Transfer Protocol*)** — протокол передачи новостей (телеконференций).

Использование протоколов TCP/IP происходит следующим образом Информация для передачи упаковывается средствами *прикладной программы* в пронумерованные блоки

определенного формата, чтобы при получении можно было правильно собрать информацию. Текст информационного сообщения запаковывается в **кодový конверт**, формируемый из **специальных символов начала и конца** и **ТСР** - **заголовка** сообщения, в котором указываются **адреса отправителя и получателя** (так называемые **IP-адреса**), а также содержится информация об исправлении ошибок и о **последовательности** передачи пакетов. **Кодový конверт** обеспечивает целостность сообщения и служит его проводником в сети.

Затем **ТСР - пакеты** разделяются на еще более мелкие **IP - пакеты**.

Т.о., пакеты состоят из трех различных уровней, каждый из которых содержит:

- данные приложения,
- информацию **ТСР**,
- информацию **IP**.

Принимающая программа объединяет пакеты **IP** в пакеты **ТСР**, из которых реконструируются исходные данные.

Протоколы **ТСР / IP** обеспечивают передачу информации между компьютерами. Все остальные протоколы с их помощью реализуют самые разные услуги **Интернет**.

- У каждого участника сети должен быть свой **уникальный IP-адрес**. Без этого нельзя говорить о точной доставке **ТСР - пакетов** на нужное место.
- Структура **IP-адреса** организована так, что каждый компьютер, через который проходит какой-либо **ТСР - пакет**, может по этому адресу определить, кому из ближайших компьютеров сети надо переслать пакет, чтобы он оказался как бы *ближе* к получателю.
- В результате конечного числа перебросок **ТСР - пакет** достигает адресата.

Термин «*близость*» не является географическим. В расчет принимаются условия связи и пропускная способность линии. Два компьютера, находящиеся на разных континентах, но связанные высокопроизводительной линией космической связи, считаются более близкими друг к другу, чем два компьютера из соседних посёлков, связанные простым телефонным проводом.

Оптимальный путь определяют *компьютеры - маршрутизаторы*, объединяющие отдельные участки сети между собой. Роль маршрутизаторов в сети обычно выполняют специализированные компьютеры, но это могут быть и специальные программы, работающие на узловых серверах сети

2. Варианты построения архитектуры сетей арм.

Системы, содержащие в себе функционирующие **АРМы**, могут быть условно разделены на три группы по такому важнейшему классификационному признаку, как *масштаб* и *степень интеграции* АРМ (Слайд 4):

- *Одиночные* или *локальные системы - локальные АРМ* (РАССМАТРИВАЛИСЬ РАНЕЕ);
- *Групповые системы* - малые интегрированные или средние интегрированные системы;
- *Корпоративные системы.*

2.1. Одиночные арм.

(Слайд 5)

Реализуются, как правило, на автономном **ПК** (сеть не используется). Такой **АРМ** может содержать несколько простых приложений, связанных общим информационным фондом, и рассчитан на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место. Подобные приложения создаются с помощью так называемых *настольных* или *локальных* систем управления базами данных (**СУБД**). Среди локальных **СУБД** наиболее известными являются: *Clipper, FoxPro, Paradox, dBase* и *Microsoft Access*.

ПОВТОРНОЕ ОПРЕДЕЛЕНИЕ: Локальный АРМ

(автоматизированное рабочее место) — программно-технический комплекс, реализующий управленческие функции на отдельном рабочем месте, информационно и функционально не связан с другими ИС (АРМ).

На одиночном АРМ производится централизованная обработка данных (Слайд 6):

На компьютере такого АРМ функционируют:

- программные средства пользовательского интерфейса, обеспечивающие интерактивный режим работы пользователя;
- программные средства приложений, выполняющие обработку данных;
- сама БД.

Развитие данной системы построения АРМ ограничено:

- техническими параметрами ПК: объем оперативной памяти, объем дисковой памяти для БД, надежность работы компьютера и ПО;
- ресурсами ПК, влияющими на своевременность обработки всех приложений.

2.2. Групповые системы (сети) арм.

(Слайд 7)

Групповые системы (сети) АРМ ориентированы на коллективное использование информации членами рабочей группы и чаще всего строятся на базе **локальной вычислительной сети**.

АРМ, работающий в составе групповой информационной системы (ИС) — программно-технический комплекс, реализующий управленческие функции на отдельном рабочем месте, но при этом информационно и функционально он может быть связан с другими АРМ не только данной ИС, но и с другими ИС.

Здесь в качестве инструмента построения БД используются т.н. **серверы баз данных** (называемые также **SQL-серверами**) для

рабочих групп. Наиболее известные серверы баз данных: **Oracle, DB2, Microsoft SQL Server** и др.

Такие системы определяются, как:

1. Комплекс информационно и функционально связанных арм, реализующих в полном объеме функции управления, или
2. Компьютерная сеть арм на единой информационной базе, обеспечивающая интеграцию функций управления в масштабе предприятия.

Как и рассматриваемые ниже корпоративные системы (**КИС**), использующие для обмена информацией среду **Интернет**, современные локальные сети также строятся на технологиях, применяемых при построении глобальной сети, и имеют свое специальное название – сети **Ethernet**.

2.3. Корпоративные информационные системы (кис).

Корпоративные системы являются развитием **групповых систем (систем для рабочих групп)**, они ориентированы на крупные компании и могут поддерживать территориально разнесённые узлы сети. В основном они имеют многоуровневую иерархическую структуру и состоят из большого количества **АРМ**, объединенных в локальные сети, которые, в свою очередь, объединены в региональные сети с выходом в глобальную сеть.

Для таких систем характерна архитектура **Клиент-Сервер** или же **многоуровневая архитектура**. При разработке таких систем могут использоваться те же **серверы баз данных**, что и при разработке групповых систем. В **ИС** крупнейших фирм мира наибольшее распространение получили продукты немецкой фирмы **SAP AG**. В **ИС** крупнейших фирм мира наибольшее распространение получили продукты немецкой фирмы **SAP AG**. Используется также корпоративное **ПО** таких фирм, как **Microsoft, Oracle, SAS Institute** и **BAAS**. В **России** базовым отечественным **ПО** для таких систем является **1С:Предприятие** последних версий.

3. Архитектура распределенной обработки данных.

При построении больших систем, имеющих в своем составе большое количество функционирующих АРМ, актуальна проблема создания *распределённых систем обработки данных* на основе интеграции неоднородных аппаратно-программных платформ. Многоуровневая архитектура соединения АРМ в сети обеспечивает изоляцию параллельно работающих процессов, в результате ошибки в работе одной программы не влияют на работу других программ либо операционной системы.

Такие системы имеют различную информационно-технологическую архитектуру, зависящую от используемых технических средств и ПО, типа сетей и организации БД.

Виды архитектур распределенной обработки данных:

- централизованная обработка данных - т.е. *одинокные АРМ* рассматривалась выше в п.2.1.;
- архитектура *Файл-Сервер*;
- двухуровневый *Клиент—Сервер*;
- многоуровневый *Клиент—Сервер*.

3.1. Архитектура Файл-Сервер.

Система АРМ с распределенной обработкой данных *Файл-Сервер* (Слайд 8) использует компьютерные сети, как правило, локального типа. Компьютеры в сети делятся на *рабочие станции* (основа построения АРМ) и *серверы*. В ряде случаев один и тот же ПК может выступать и как *сервер* для других компьютеров, и как *рабочая станция*, пользующаяся услугами *сервера*. Это так называемые *одноранговые сети без выделенного сервера*. В сетях с *выделенным сервером* компьютер является либо *сервером*, либо *рабочей станцией*.

На *рабочей станции* установлено ПО *пользовательского интерфейса, программные средства приложений*, выполняющие содержательную обработку данных.

На *файловом сервере* находится БД.

Достоинство архитектуры *Файл - Сервер* — обеспечение высокого уровня защиты данных от несанкционированного доступа.

Недостатки архитектуры *Файл - Сервер*:

- *обмен на уровне файлов*, доступ к которым в режиме корректировки блокируется для других пользователей;
- перегрузка трафика сети;
- высокие требования к техническому оснащению АРМ.

3.2. Архитектура двухуровневый Клиент—Сервер.

В отличие от архитектуры *Файл - Сервер*, распределенная обработка данных типа *Двухуровневый Клиент—Сервер* (Слайд 9) предполагает, что на сервере находится БД под управлением СУБД в архитектуре *Клиент—Сервер*.

Все *рабочие станции (клиенты)* - основа построения АРМ - посылают запросы на данные к *серверу*, который осуществляет извлечение и предварительную обработку данных. Единицей обмена по сети является *запрос* и выполненная по запросу *выборка данных* из БД. Существенно уменьшается трафик сети, снимаются ограничения на доступность данных БД различным приложениям.

Рабочая станция должна иметь достаточно высокие технические параметры для выполнения сложных приложений.

Недостатком архитектуры является наличие очень высоких требований к техническому комплексу *сервера* БД, который становится центральным звеном всей ИС и определяет ее надежность.

3.3. Многоуровневый Клиент—Сервер

(Слайд 10)

На *рабочей станции* установлены только программные средства, поддерживающие интерфейс с БД. На сервере БД находятся БД под управлением СУБД, архитектура сети — *Клиент—Сервер*, В архитектуре ИС выделен *сервер приложений*, на котором находятся *программные средства общего пользования*.

Эти *серверы* выполняют всю содержательную обработку данных.

В отличие от двухуровневой архитектуры, данная архитектура обеспечивает эффективное использование приложений общего пользования многими клиентами, при этом снижаются требования к оборудованию рабочих станций. Если серверов приложений и БД в сети несколько, архитектура ИС становится *Многоуровневой Клиент—Серверной* архитектурой. Наличие самостоятельных уровней в информационно-технологической архитектуре ИС дает возможность варьировать аппаратными и программными средствами — т.е. выбирать:

- операционные системы,
- СУБД,
- интерфейсы конечных пользователей,
- типы серверов,
- типы рабочих станций.

4. Современный подход к функционированию арм аналитика, финансиста и бухгалтера в системах распределенной обработки информации.

В зависимости от структуры функциональных компонентов, сложности алгоритмов обработки информации и соответствующих информационных технологий такие системы определяются как:

1. *Системы оперативной обработки данных — OLTP (On - Line Transaction Processing) (Слайд 11) -* традиционные ИС учета регистрации первичной информации (бухгалтерские, складские системы, системы учета выпуска готовой продукции и т. п.) построенные как система из АРМ бухгалтера, АРМ кассира и пр.

Выполняется сбор и регистрация больших объемов первичной информации, используются достаточно простые алгоритмы расчетов и запросов к БД, структура которой стабильна в течение длительного времени (5-7 лет) для эффективного функционирования прикладного ПО.

В OLTP-системах большое значение имеет защита БД от несанкционированного доступа, аппаратных и программных сбоев в работе АРМ. Формы входных и выходных документов, схемы документооборота жестко регламентированы. Для повышения эффективности функционирования используются компьютерные сети АРМ с архитектурой *Клиент—Сервер*.

2. *Системы поддержки и принятия решений — DSS (Decision Support Systems) - (Слайд 12)* ориентированы на реализацию сложных *бизнес - процессов*, требующих аналитической обработки информации, *формирование новых знаний*.

Анализ информации имеет определённую целевую ориентацию, например финансовый анализ предприятия, аудит бухучета и реализуется на базе **АРМ финансиста, либо АРМ аналитика** - в зависимости от предметной области ИС.

Отличительной особенностью таких систем являются:

- создание *хранилищ данных большой ёмкости DW (Data Warehouse)* путем интеграции разнородных источников, находящихся в системах оперативной обработки данных (OLTP);
- использование методов и средств аналитической обработки данных -**OLAP**-технологии (*On - Line Analytical Processing*);
- интеллектуальный анализ данных, обеспечивающий формирование новых знаний – **DM** - технологии (*Data Mining*).

Определение *Хранилище данных* — это предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных для поддержки процесса принятия управляющих решений.

На основе *Хранилищ данных* создаются *подмножества данных* — т.н. **OLAP-кубы** - *многомерные иерархические структуры данных*, содержащие множество признаков (Слайд 13):

- дата/время (период времени, к которому относятся данные);
- уровень управления (структурное подразделение), которому соответствуют данные;
- сфера деятельности (бизнес-сфера, результат), к которой относятся данные;
- субъект управления (лицо принимающее решение);
- вид ресурса и другие.

По этим признакам с рабочего места **АРМ аналитика** можно производить выборку данных в разных ракурсах путем их произвольного сочетания и вычисления статистических оценок. В результате анализа информации создаются различные отчеты, служащие основанием для принятия управленческих решений.

Основные современные инструментальные средства OLAP-технологий (Слайд 14):

1. **Многомерные СУБД (MDDDB — *Multi Disentail Data Base*)**, сложная аналитическая обработка которых обеспечивается специальным программным обеспечением. (**SAS System** компании **SAS Institute**, **Plato OLAP Server beta 3** фирмы **Microsoft**).
2. **Оперативная аналитическая обработка т.н. реляционных БД (Relation OLAP)**. Данные реляционных БД (например – **Access**) отображаются в иерархические структуры или, например, в виде так называемых **OLAP-кубов (см. выше)**. Обеспечены гибкость алгоритмов сбора исходных данных для формирования данных для анализа, универсальность методов проведения анализа данных, возможность представления результатов анализа в различных форматах. (Модуль **Decision Cube Borland Delphi 4.0**, **OLAP-кубы Excel 2007**).
3. **Инструментальные средства генерации запросов и отчетов**, дополненные функциями **OLAP** - создание аналитических отчетов и отчетов иерархической структуры данных большой глубины, вычисление статистических итоговых функций и др. (**Business Objects**, **Power Play** фирмы **Cognos**).
4. **Специализированные модули OLAP-технологий** в составе систем управления крупными предприятиями типа **Enterprise Resources Planning (ERP)** как отдельные программные модули или самостоятельные **ИС**, имеющие доступ к корпоративному хранилищу данных (**SAP R / 3**).

•

Системы управления базами данных

Цель занятия: усвоить знания основных понятий СУБД, признаков классификации, основных возможностей СУБД для организации работы пользователя с базой данных.

План

1. Основные понятия. Классификация СУБД.
2. Функциональные возможности СУБД
3. Режимы работы пользователя с СУБД. Направления развития СУБД

1. Основные понятия. Классификация субд

Системой управления базами данных называют **комплекс программ**, предназначенный для создания и использования базы данных для решения множества задач. СУБД служат для поддержания базы данных в актуальном состоянии, обеспечивают эффективный доступ пользователей к содержащимся в ней данным в рамках предоставленных пользователям полномочий.

В архитектуре СУБД можно выделить средства проектирования, подсистему обработки, и ядро СУБД. Все современные СУБД имеют средства проектирования объектов базы данных, редактирования данных, сортировки, защиты данных. Управляющим компонентом многих СУБД является ядро, выполняющее следующие функции:

- управление данными во внешней памяти;
- управление буферами оперативной памяти (рабочими областями, в которые осуществляется подкачка данных из базы для повышения скорости работы);
- управление транзакциями.

Транзакция — это последовательность операций над БД, рассматриваемая СУБД как единое целое. При выполнении транзакция может быть либо успешно завершена и зафиксирует произведенные изменения во внешней памяти, либо, например, при аппаратной части ПК, ни одно из изменений не отразится в БД.

Язык современной СУБДП включает подмножество команд, относящихся к следующим специализированным языкам:

- *язык описания данных* — высокоуровневый непроцедурный язык декларативного типа, предназначенный для описания логической структуры данных;
- *язык манипулирования данными* — командный язык СУБД, обеспечивающий выполнение основных операций по работе с данными — ввод, модификацию и вывод данных по запросам;
- *структурированный язык запросов* (Structured Query Language, SQL) — обеспечивает манипулирование данными и определение схемы реляционной БДП, является стандартным средством доступа к серверу БД.

По степени **универсальности** различают два класса СУБД:

- системы общего назначения;
- специализированные системы.

СУБД общего назначения не ориентированы на какую-либо предметную область или на информационные потребности какой-либо группы пользователей. Каждая система такого рода реализуется как программный продукт, способный функционировать на некоторой модели ЭВМ в определенной операционной системе и поставляется многим пользователям как коммерческое изделие. Такие СУБД обладают средствами настройки на работу с конкретной базой данных. Использование СУБД общего назначения в качестве инструментального средства для создания автоматизированных информационных систем, основанных на технологии баз данных, позволяет существенно сокращать сроки разработки, экономить трудовые ресурсы. Этим СУБД присущи развитые функциональные возможности и даже определенная функциональная избыточность.

Специализированные СУБД создаются в редких случаях при невозможности или нецелесообразности использования СУБД общего назначения.

СУБД общего назначения — это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.

По поддерживаемой модели данных — реляционные, сетевые, иерархические, объектно-ориентированные СУБД. Большинство СУБД для персональных ЭВМ поддерживают реляционную модель данных (dBase, FoxBase, Paradox, MS Access, Clipper). Такие СУБД эффективны для создания небольших баз данных (до 40 Мбайт) с несложной структурой данных и несложными запросами. Надо отметить, что выпускаемые в последнее время СУБД поддерживают несколько моделей данных.

По поддерживаемому **режиму работы** с базой данных (по уровню использования) выделяют следующие СУБД, поддерживающие следующие режимы:

- локальные (для настольных систем, то есть однопользовательские — dBase, FoxBase, Paradox, MS Access),
- серверные (для работы в компьютерных сетях архитектуры клиент-сервер — MS Sql Server, Informix, Sybase, Oracle, эти СУБД называют многопользовательскими),
- многопроцессорные (для параллельной обработки данных).

По способу организации базы данных различают следующие виды СУБД:

- Однофайловые – самые простые в использовании, не требуют больших ресурсов ЭВМ. Применяют для телефонных справочников, карточек в библиотеке и т.д. (Professional Report, Formula)
- Многофайловые непрограммируемые – позволяют хранить записи разных типов, предоставляют большие возможности создания баз данных (Magic PC, MS Works)
- Многофайловые программируемые – предоставляют пользователю возможности написания прикладных программ. Имеют общие характеристики: число полей 128-1024, число символов в записи от нескольких тысяч до десятков тысяч, число записей в таблице – почти не ограничено (только емкость памяти), число открытых файлов – не ограничено (dBase, MS Access).

По среде функционирования – одноплатформенные (ориентированы на работу на некотором классе компьютеров и операционных систем), многоплатформенные (могут работать на разных классах компьютеров с разными операционными системами).

Существуют и другие признаки классификации, например, **возможностям встроенного языка СУБД, наличие средств конструирования объектов БД и работы с ней, возможности работы с данными в сетях** (страницы HTML, электронная почта, изображения, звуковые файлы, видео). Но следует отметить, что практически все производители СУБД стремятся оснастить свои комплексы подобными возможностями.

Рынок программного обеспечения ПК располагает большим числом разнообразных по своим функциональным возможностям систем управления базами данных общего назначения, а также средствами их окружения практически для всех массовых моделей вычислительных машин и для различных операционных систем.

Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и надежной безопасности, что дает возможность разработчикам гарантировать большую безопасность данных при меньших затратах сил на низкоуровневое программирование.

2. Функциональные возможности субд

Возможности по созданию базы данных и ее актуализации, извлечению данных, созданию приложений базы данных, взаимодействию с другими системами, управлению базой данных.

В настоящее время создано множество СУБД имеющих приблизительно одинаковые возможности. Все СУБД позволяют создавать структуру БД и заполнять таблицы (создавать и просматривать файлы базы данных), редактировать их (обновлять данные, удалять ненужные данные, добавлять новые данные). Данные в базе можно упорядочить по одному или нескольким полям. Можно осуществить поиск данных в базе по различным критериям поиска. Важна возможность изменения существующей структуры базы данных. Также имеются возможности конструировать формы, формировать отчеты, реализовать нестандартные процедуры обработки данных (макросы, VB), создавать приложения пользователя (генераторы приложений, средства создания меню и панелей управления приложениями). Для реализации перечисленных возможностей в состав СУБД входят языковые средства.

Язык современной СУБД включает подмножество команд, относящихся к следующим специализированным языкам:

- *язык описания данных* — высокоуровневый непроцедурный язык декларативного типа, предназначенный для описания логической структуры данных;
- *язык манипулирования данными* — командный язык СУБД обеспечивающий выполнение основных операций по работе с данными — ввод, модификацию и вывод данных по запросам;
- *структурированный язык запросов* (Structured Query Language, SQL) — обеспечивает манипулирование данными и определение схемы реляционной БДП, является стандартным средством доступа к серверу БД.

Для обработки команд пользователя в СУБД используются *интерпретаторы* команд (операторов) и *компиляторы*. С помощью компьютеров в ряде СУБД можно получать исполняемые автономно приложения программы.

Обеспечение безопасности достигается в СУБД шифрованием прикладных программ, защитой паролем, поддержкой уровней доступа к базе данных, к отдельной таблице.

Расширение возможностей пользователя СУБД достигается за счет подключения систем построения графиков и диаграмм, а также подключения модулей, написанных на языках *Си* или *Ассемблера*.

Поддержка функционирования в сети обеспечивается:

- средствами управления доступом пользователей к совместно используемым базам, т.е. средствами блокировки файлов (таблиц), записей, полей, которые в равной степени реализованы в разных СУБДП;
- средствами механизма транзакций, обеспечивающими целостность БД функционировании в сети.

Поддержка взаимодействия с Windows-приложениями позволяет СУБД внедрять в отчет сведения, хранящиеся в файлах, созданных с помощью других приложений (например, в документе Word или в рабочей книге Excel), включая графику и звук. В СУБД поддерживаются механизмы, разработанные для среды Window, такие как DDE (Dynamic Data Exchange — динамический обмен данными) и OLE (Object and Embedding — связывание и внедрение объектов).

2. Режимы работы пользователя с субд. Направления развития субд

СУБД предполагает работу пользователя с базой в различных режимах :

- режим «ассистента» - используется разветвленное меню, не требуется специальной подготовки пользователя;
- командный режим – взаимодействие с системой посредством ввода команд, требуется подготовка пользователя (должен знать язык команд СУБД);
- программный режим – для взаимодействия используется язык СУБД, позволяющий создавать программы, имеющие «дружественный» пользовательский интерфейс.

На сегодняшний день известно более двух десятков серверных СУБД, однако наиболее популярными, исходя из числа продаж и установленных копий, следует признать Oracle, Microsoft SQL Server, Informix, Sybase, DB2.

Семейство СУБД фирмы Oracle. Очень популярные и престижные СУБД. Выпущенная в 1979 году СУБД фирмы Oracle была первой коммерческой реляционной СУБД, поддерживающей ставший ныне индустриальным стандартом язык SQL. Фактически все эти годы Oracle является бесменным лидером на рынке производителей коммерческих СУБД и второй (после Microsoft) по величине компаний, производящей программное обеспечение.

Ранние версии этой СУБД были предназначены для мэйнфреймов, а в качестве рабочих мест использовались «неинтеллектуальные» терминалы. Однако со временем появились версии Oracle, предназначенные для использования в архитектуре «клиент-сервер» (первой такой версией была Oracle 5, выпущенная в 1985 году).

Первоначально эти версии были предназначены для различных серверных платформ — разнообразных версий UNIX, VMS и др. Позже были выпущены версии сервера Oracle для Novell NetWare. Первые версии этого сервера для персональных компьютеров появились в середине 90-х (Personal Oracle 7 for Windows 3.1, Personal Oracle 7 for Windows 95, Personal Oracle Lite, Oracle Workgroup Server 7 for Windows NT). До появления этих версий персональные компьютеры могли использоваться исключительно в качестве клиентских рабочих станций — в состав Oracle для серверных платформ обычно входила клиентская часть для DOS.

Отметим, что Oracle была первой компанией, создавшей СУБД использовавшую предоставляемые некоторыми серверными платформами средства параллельных вычислений — Oracle Parallel Server (до его появления параллельные вычисления использовались только для решения научных задач). При использовании параллельных вычислений Oracle Parallel Server дает возможность нескольким процессорам обращаться к одной базе данных, что позволяет обеспечить высокую скорость обработки транзакций, а более поздние его версии дают возможность осуществить декомпозицию операций с большими объемами данных с целью параллельного выполнения их на нескольких процессорах.

Отличительными свойствами последних версий Oracle являются:

- наличие объектных расширений и соответствующих типов данных, таких как вложенные таблицы, массивы, объекты и др. Иными словами, Oracle являются объектно-ориентированными СУБД
- наличие функций аналитической обработки данных (например, вычисления процентных соотношений, ранжирования, сравнения временных периодов);
- возможность создания таблиц, содержащих агрегатные данные, и частичного их обновления при изменении данных, на основании которых они вычислены;
- поддержка Java;
- поддержка XML;
- поддержка HTML- и XML-страниц с включенным в них кодом PL/SQL;
- поддержка хранения мультимедиа-данных с возможностью индексации, построения контекстных запросов, поддержки разных языков для хранимых документов;
- набор процедур и функций для обработки пространственной информации;
- дополнительные возможности обеспечения безопасности, например шифрование данных, поддержка SSL, использование ролей уровня базы данных и уровня предприятия;
- возможность создания систем, устойчивых к сбоям, с использованием нескольких параллельных процессов;
- поддержка кластеров на основе Microsoft Cluster Server,
- наличие OLE DB-провайдера для доступа к данным сервера.

Объявлено, что следующая версия этой СУБД Oracle 9i, будет содержать в своем составе OLAP-средства.

Помимо различных версий сервера баз данных среди продуктов Oracle имеется также Designer/2000 — ориентированное на эту СУБД CASE-средство для анализа бизнес-процессов и проектирования данных, а также средства разработки клиентских приложений. Одно из них — Developer/2000 (называвшееся ранее Oracle Forms) — весьма популярно среди пользователей Oracle. Следует отметить, что приложения, созданные с помощью Developer/2000, могут выполняться на различных платформах. Язык PL/SQL, используемый в этом средстве разработки, является интерпретируемым и представляет собой тот же самый язык, что используется в Oracle для написания серверного кода. Это позволяет отлаживать с помощью Developer/2000 серверный код.

Производя собственные средства разработки, Oracle предоставляет своим пользователям возможность создавать клиентские приложения с помощью других средств. В частности, помимо стандартного в таких случаях клиентского API (Oracle Call Interface) клиентская часть Oracle содержит также объектную модель (Oracle Objects for OLE), позволяющую использовать клиентскую часть Oracle как набор COM-объектов для доступа к данным. Кроме того, обычно клиентская часть Oracle содержит также ODBC-драйвер для доступа к данным этой СУБД.

Многие другие компании производят ODBC-драйверы и OLE DB-провайдеры для доступа к Oracle. Компании, производящие средства разработки, использующие собственные библиотеки доступа к данным (Borland), также включают библиотеки доступа к Oracle в состав наиболее полных версий своих продуктов.

Из готовых информационных систем на базе Oracle следует особо отметить несколько крупных систем управления предприятием, в частности SAP/R3. На Западе также нередко используются готовые решения, производимые самой Oracle Corporation, объединенные под общим названием Oracle Applications. К ним, в частности, относятся Oracle Financials, Oracle Human Resources, Oracle Market Management, Oracle Project Systems и ряд других.

Microsoft SQL Server. Первая версия Microsoft SQL Server, совместно разработанная в 1988 году компаниями Microsoft и Sybase, предназначалась для платформы OS/2. Последующие версии этого сервера баз данных предназначались для платформы Windows NT и со временем стали тесно интегрированы с этой операционной системой. Для других платформ версии этого сервера не выпускались и не выпускаются.

Удобный пользовательский интерфейс утилит администрирования в сочетании с достаточно высокой производительностью и относительно невысокой стоимостью эксплуатации сделал эту серверную СУБД второй по популярности — после Oracle. Наибольший рост популярности этой СУБД пришелся на конец 90-х годов, когда были выпущены Microsoft SQL Server 6.0 (1995 г.), обладавший централизованными функциями администрирования и встроенными возможностями репликации данных, Microsoft SQL Server 6.5 (1996 г.) и Microsoft SQL Server 6.5 Enterprise Edition, поддерживающий параллельные вычисления в многопроцессорных системах.

На сегодняшний день наиболее широко используемыми являются выпущенная в 1998 году версия Microsoft SQL Server 7.0 и вышедшая ранее версия 6.5. Версия 7.0 Microsoft SQL Server отличается от предыдущих тем, что она была полностью переписана фирмой Microsoft исключительно под платформу Windows NT. В состав Microsoft SQL Server 7.0 входят еще более простые утилиты администрирования (Enterprise Manager), преобразования данных (Data Transformation Services), облегчающие перенос данных в SQL Server из других типов СУБД (поддержка распределенных запросов и транзакций, OLAP-сервер и утилиты для создания хранилищ данных (в том числе данных из других серверных СУБД), расширенная поддержка функций для создания Web-приложений).

Помимо собственно Microsoft SQL Server 7.0 в качестве встроенной СУБД для настольных приложений и приложений для небольших рабочих групп можно использовать Microsoft Data Engine (MSDE) — настольный сервер баз данных, совместимый с Microsoft SQL Server и предназначенный для использования в настольных системах или в сетевых приложениях с небольшим (до 2 Гбайт) объемом данных и небольшим количеством пользователей.

СУБД фирмы Sybase. Серверные продукты компании Sybase происходят от двух «предков». Первым из них является одна из ранних версий Microsoft SQL Server, созданная совместно Microsoft и Sybase. Начиная с 1994 года Microsoft и Sybase разрабатывают свои серверные продукты независимо друг от друга. Результатом деятельности компании Sybase в этом направлении является продукт под названием Adaptive Server Enterprise (в настоящее время наиболее часто используются его версии 11 и 12). Этот продукт существует для платформы Windows NT и некоторых версий UNIX (включая Linux) и предназначен для обслуживания крупных предприятий. В настоящее время этот сервер баз данных поддерживает:

- упреждающее асинхронное чтение, повышающее скорость выполнения сложных запросов;
- использование кластерных систем;
- распределенную обработку запросов, в том числе к базам данных на СУБД других производителей;
- расширенные хранимые процедуры, позволяющие осуществить легкий доступ к не-SQL функциям (Java, 3GL-системы, функции операционной системы и т.д.);
- параллельную обработку запросов в многопроцессорных системах;
- параллельную работу утилит администрирования;
- интеграцию с популярными системами безопасности (Kerberos).

Еще одна линия серверных продуктов Sybase ведет свое начало от сервера баз данных Watcom SQL Anywhere, отличавшегося компактностью и простотой администрирования. Последняя версия этого продукта называется Adaptive Server Anywhere 6.03. Этот сервер предназначен для обслуживания небольших рабочих групп, для применения в портативных компьютерах в качестве персонального сервера.

с периодической репликацией, а также в мобильных устройствах — существуют версии этого сервера для Windows CE 2.1 и версия UltraLite для разнообразных мобильных устройств.

Для управления распределенными транзакциями компания Sybase выпускает монитор транзакций — Jaguar CTS. Для создания многомерных хранилищ данных у Sybase существует еще один серверный продукт — Adaptive Server IQ, позволяющий создавать хранилища на основе данных не только из СУБД производства Sybase, но и из СУБД других производителей. Особенностью этого продукта является необычный способ хранения данных — в таблицах этой СУБД рядом находятся данные не из одной и той же записи, как это принято в большинстве СУБД, а из одного и того же поля разных записей. Такой способ хранения данных позволяет выполнять запросы на выборку данных с более высокой скоростью, чем в СУБД с традиционными способами хранения данных.

Отметим также, что существует ряд продуктов под общим названием Sybase Industry Warehouse Studio, ориентированных на обслуживание конкретных предметных областей: торговли (Retail Warehouse Studio), здравоохранения (Healthcare Warehouse Studio), страхования (Life Insurance Warehouse Studio) и ряда других.

Помимо серверных продуктов фирма Sybase производит средства разработки, ориентированные на создание клиентских приложений для них (PowerBuilder, PowerJ, PowerSite (для создания Web-приложений), а также средства проектирования данных и генерации кода приложений. Последние можно отнести к универсальным средствам - CASE-средство DataArchitect поддерживает широкий спектр СУБД различных производителей, а генератор приложений AppModeler способен генерировать код не только для PowerBuilder и Optima-H-, но и для Delphi, Visual Basic, Web-приложений с использованием технологии Active Server Pages.

СУБД фирмы Informix. Ведущий продукт фирмы Informix - Informix Dynamic Server (1999 г.). Данный продукт поддерживает платформы UNIX и Microsoft Windows NT и обеспечивает эффективную работу как на одно-, так и на многопроцессорных системах, а также в кластерах. Сервер построен по архитектуре Dynamic Scalable Architecture (DSA), обеспечивающей мощные средства для параллельной обработки данных. В числе основных характеристик Informix Dynamic Server следует отметить:

- использование для управления дисковым пространством как средств операционной системы (UNIX или Microsoft Windows NT), так и собственных функций, позволяющих обойти ограничения операционной системы и добиться более высокой производительности, - такое управление дисковым пространством называется Raw Disk Management;
- управление разделением памяти — поддержку одновременного доступа к данным, находящимся в памяти, несколькими приложениями;
- динамическое управление потоками;
 - поддержку фрагментации таблиц и индексов на нескольких дисках;
- распараллеливание запросов (parallel database query, PDQ);
- зеркалирование данных.

Сервер поддерживает двухфазное завершение транзакции, гетерогенные транзакции. Расширения функциональности сервера реализуются на базе DataBlade - коллекций объектов баз данных и подпрограмм на языке C, подключаемых к базе данных.

Входящие в состав Informix Dynamic Server клиентские утилиты предназначены для подключения к серверу и обработки информации (DB-Access), а также для выполнения функций администрирования (DB/Cockpit).

Клиентские приложения могут создаваться с использованием языков Informix ESQL (средство для разработки на языке C, позволяющее включать в приложения запросы к данным на языке SQL), а также C, C++, Java, Visual Basic и Delphi. Помимо этого существует собственное средство разработки — Informix-4GL и Informix Client Software Developer's Kit

Для поддержки универсальных механизмов доступа к данным фирма Informix выпускает Informix ODBC Driver, OLE DB Provider для Informix Dynamic Server и Informix ODBC Driver.

Для генерации отчетов предлагается Informix-ViewPoint—визуальное средство, рассчитанное на пользователей. Версия Pro также содержит средства администрирования.

Поддержка OLAP: продукт Informix MetaCube поставляется как часть Informix Decision Frontie - комплексного решения для создания хранилищ данных.

Среди других продуктов фирмы Informix следует отметить:

- Informix Internet Foundation.2000 — специально разработанный для Internet вариант Informix Dynamic Server;
- i.Reach — корпоративный репозиторий для хранения данных различного типа, интеллектуального управления информацией и извлечения данных. Основное назначение данного продукта - поддержка включения в содержимое корпоративных сайтов электронных документов и их последующее обслуживание;
- i.Sell — комплексное решение для электронной коммерции на базе Informix Dynamic Server.

IBMDB2. Семейство серверных СУБД фирмы IBM, известное под названием DB2 Universal Database, представляет собой стратегию IBM по объединению в единую линию продуктов DB2 для различных платформ. Впервые появившееся в 1996 году семейство DB2 Universal Database объединяло в себе функциональные возможности таких продуктов фирмы, как DB2 Common Server, DB2 Parallel Edition (DB2 PE), NetData, Data Propagator и технологии DataHub, и предназначалось для платформ UNIX, OS/2 и Microsoft Windows NT.

Отметим, что при переносе DB2 на не-IBM-платформы фирма старается максимально использовать уникальные функциональные возможности конкретной платформы. Помимо этого DB2 for Windows 2000 поддерживает Microsoft Transaction Services (MTS) в качестве координатора при создании приложений, использующих распределенные транзакции. Для разработчиков, использующих Microsoft Visual Studio, становятся доступными дополнительные модули, например Stored ProcedureBuilder, включаемый непосредственно в среду Visual Studio. IBM также предлагает собственные средства разработки, например IBM VisualAge for Java, позволяющие создавать приложения, работающие с данными в DB2. Продукт также поддерживает создание хранимых процедур на языке Java (Java Stored Procedure Builder).

Помимо этого IBM предлагает бесплатное средство для миграции данных из Microsoft Access в DB2, а также средства для миграции данных из СУБД фирм Oracle, Microsoft, Sybase и Informix.

К основным характеристикам СУБД можно отнести поддержку реляционных и комплексных данных через объектные расширения, возможность работы на мультипроцессорных платформах, поддержку кластеров, 64-битную архитектуру памяти и распараллеливание запросов, возможность создания Web-приложений (поддерживаются такие технологии, как Java, ODBC, SQL, XML) и наличие средства для гетерогенного администрирования и обработки данных.

Семейство DB2 функционирует на системах AS/400 и RISC System/6000, мэйнфреймах IBM, машинах от Hewlett-Packard и Sun Microsystems и на таких операционных системах, как Windows NT/2000 и т.д., OS/2, AIX, HP-UX, SCO UnixWare, Linux, Sun Solaris, и сейчас поддерживает портативные устройства под управлением Windows CE и Palm OS. DB2 Universal Database выпускается в редакциях DB2 Universal Database Enterprise — Extended Edition (платформы AIX, Solaris и Windows NT), DB2 Universal Database Enterprise Edition (платформы AIX, HP-UX, Linux, OS/2, Solaris и Windows NT), DB2 Universal Database Workgroup Edition (платформы Linux, OS/2 и Windows NT) и DB2 Universal Database Personal Edition (платформы OS/2, Linux, Windows 9x и Windows NT). К дополнительным продуктам можно отнести:

- DB2 OLAP Server — средство для онлайн-аналитической обработки данных и реализации хранилищ данных, интегрирующее ядро OLAP-сервера Hyperion Essbase с семейством DB2 Universal Database. (Business Objects), Lotus 1-2-3 (Lotus), Excel, Internet Explorer, Visual Basic (Microsoft);
- DB2 Connect - средство для управления соединениями различных клиентов с DB2 на AS/400; DB2 Universal Developer's Edition — рассчитанное на разработчиков средство для создания и тестирования приложений в архитектуре «клиент-сервер», работающих с данными DB2; DB2 DataJoiner — средство для доступа к реляционным и нереляционным данным, расположенным на различных платформах как к единому «образу» данных;
- DB2 Data Links Manager — средство для управления дополнительными файлами, подключаемыми к СУБД; DB2 Query Patrolier — набор средств для создания запросов и управления ресурсами для систем принятия решений. DB2 Query Patrolier получает ODBC-запросы от клиента, анализирует их и динамически распределяет запросы по различным узлам DB2 UDB Enterprise — Extended Edition;
- DB2 Net.Data — приложение, позволяющее Web-разработчикам создавать динамические Internet-приложения» используя Web Macros; DB2 Universal Database Satellite Edition — средство для внедрения масштабируемых мобильных решений, для управления удаленными пользователями. Поддерживает функции репликации, централизованное администрирование и средства управления через Web DB2 Web Control Center,
- DB2 Everywhere — СУБД для Palm OS и Windows CE, обеспечивающее SQL-функции и синхронизацию данных с другими реляционными базами данных, с данными Lotus Notes/Domino.

Таким образом, системы, основанные на серверных СУБД, обладают более высокой производительностью, генерируют меньший сетевой трафик по сравнению с системами, основанными на настольных СУБД. Такие СУБД имеют общие характеристики:

- Реализованы для нескольких платформ;
- Включают утилиты администрирования;
- Осуществляют резервное копирование данных;
- Поддерживают репликацию данных;
- Поддерживают параллельную обработку данных в многопроцессорных системах;
- Поддерживают OLAP и создание хранилищ данных;
- Поддерживают различные средства проектирования данных;
- Поддерживают публикацию данных в Интернете.
- Следовательно, в настоящее время существуют тенденции развития СУБД
- Расширение множества типов обрабатываемых данных — простые типы данных (текст, числа) дополняются мультимедийными данными, графическими образами, хронологическими рядами.
- Использование технологий Интернет (Интеграция технологий баз данных и Web-технологий) — современные информационные системы базируются на взаимодействии информационных ресурсов разных форматов и моделей представления. В среде Интернет данные распределены и не упорядочены как в базах данных.

Большинство существующих информационных систем используют простые СУБД, функционирующие на персональных компьютерах. Базой систем нового поколения являются профессиональные СУБД, расширяющие сервисные возможности (Oracle, SyBase, Informix, Ingres). Эти СУБД имеют средства обработки распределенной информации, они поддерживают технологию клиент-сервер. Современные СУБД поддерживают средства, значительно ускоряющие разработку приложений. Это языки, интегрирующие средства высокого уровня для создания интерфейса с элементами CASE-технологии, средства для создания сложных запросов, возможности для подключения

фрагментов на языках низкого уровня, поддерживают SQL-интерфейс. Все это ускоряет разработку приложений. Таким образом, выделяется еще несколько тенденций развития СУБД:

- распределенная обработка данных;
- применение средств быстрой разработки приложений.

Поскольку одной из современных тенденций развития баз данных является превращение их в базы знаний, необходимо Превращение СУБД в системы управления базами знаний.

Контрольные вопросы

1. Отличия СУБД общего назначения от специализированных.
2. Отличия однофайловых СУБД от многофайловых.
3. Особенности многопользовательских СУБД.
4. Классификация СУБД по технологии обработки данных.
5. Структура языка современной СУБД.
6. Какие механизмы СУБД обеспечивают целостность базы данных.
7. Какие механизмы СУБД обеспечивают безопасность базы данных.
8. Какие механизмы СУБД обеспечивают возможность использования базы данных в сети.
9. Отличия режима ассистента от командного режима работы пользователя.
10. Какие пользователи работают в программном режиме.
11. Характеристика семейства СУБД фирмы Oracle.
12. Характеристика СУБД Microsoft SQL Server.
13. Характеристика СУБД фирмы Sybase.
14. Характеристика СУБД фирмы Informix.
15. Характеристика СУБД фирмы **IBM (DB2)**.
16. Основные тенденции развития СУБД.

Датчики и микроконтроллеры

Введение

В системе управления технологической установкой снятие текущих показаний некоторой величины — температуры, влажности, давления, уровня жидкости, напряжения, тока и проч. осуществляется с помощью датчиков — устройств и механизмов, предназначенных для преобразования сигнала внешнего воздействия в форму, понятную системе управления. Например, датчик влажности генерирует электрический сигнал, пропорциональный текущему значению влажности воздуха.

Как правило, датчики используются не сами по себе, а входят в состав системы управления, обеспечивая сигнал обратной связи.

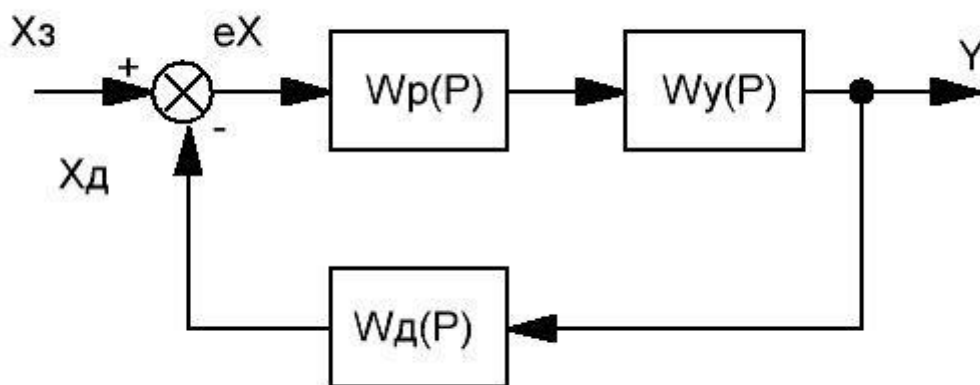


Рисунок 1. Типовая схема замкнутой системы регулирования

На рисунке 1 представлена типовая схема системы регулирования. Имеется сигнал задания X_z , который сравнивается с сигналом на выходе, получаемым с помощью датчика, имеющего передаточную функцию $W_d(p)$. Ошибка управления подается на регулятор, который, в свою очередь, формирует сигнал управления исполнительным узлом, формирующим выходной сигнал Y . [1]

Простой пример — центробежный регулятор частоты вращения двигателя, где датчиком является платформа с шарами, которая, вращаясь, устанавливает то или иное положение топливной рейки. Заслонка, управляемая этой рейкой, регулирует количество топлива, подаваемое на двигатель. Сигналом задания будет являться требуемое значение скорости.

1.1 Классификация датчиков

Классификация датчиков очень разнообразна. Приведу лишь ее часть исходя из [4].

Все датчики делятся на два основных класса:

- Пассивные, которые не нуждаются во внешнем источнике электроэнергии, и в ответ на входное воздействие генерируют электрический сигнал.

Примерами таких датчиков являются термопары, фотодиоды и пьезоэлектрические чувствительные элементы.

- Активные, которые требуют для своей работы внешний сигнал, называемой сигналом возбуждения. Поскольку, такие датчики меняют свои характеристики в ответ на изменение внешних сигналов, их называют параметрическими. Примерами активных датчиков являются терморезисторы, сопротивление которых можно вычислить путем пропускания через них электрического тока.

Следует отметить, что в литературе встречается и альтернативный вариант классификации, когда Генераторные датчики определяются как Активные, а Параметрические — как Пассивные. Здесь и далее я руководствуюсь вариантом согласно справочнику Фарйдена.

Другим важным критерием для нас является выбор точки отсчета данных. Таким образом датчики бывают

- Абсолютные, измеряемое значение физической величины которых не зависит от условий измерения и внешней среды.
- Относительные, когда выходной сигнал такого датчика в каждом конкретном случае трактуется по разному.

Ярким примером опять является терморезистор, сопротивление которого напрямую зависит только от температуры измеряемого объекта, и термопара, выходное напряжение которой зависит от разности температур между горячим и холодным концами.

При разработке радиоэлектронного оборудования важным фактором характеристик датчика также является характер выходного сигнала.

- Аналоговые датчики на выходе имеют непрерывный выходной сигнал, для снятия которого необходимо использовать аналого-цифровой преобразователь, после чего необходимо произвести преобразования значения АЦП в формат измеряемой величины.
- Цифровые датчики, информация с которых снимается с помощью различных цифровых интерфейсов. Как правило, информация доступна непосредственно в формате измеряемой величины и не требует проведения дополнительных преобразований.
- Дискретные датчики, имеющие только два варианта сигнала на выходе канала датчика — лог 0. и лог. 1. Примером такого датчика является конечный выключатель, имеющий состояния замкнут и разомкнут. Дискретный датчик может иметь несколько выходных каналов, каждый из

которых находится в одном из двух состояний. Например, 12-разрядный абсолютный датчик положения.

- Импульсные датчики, формирующие импульсы выходного сигнала, амплитуда или длительность которых зависит от измеряемой величины. Например, инкрементальный датчик положения формирует на выходе код Грея. При этом, чем выше частота вращения вала датчика, тем большая частота сигнала будет на выходе, что позволит с высокой точностью определить частоту вращения вала.

2 Характеристики датчиков

Большинство датчиков имеют сложную процедуру преобразования измеряемой величины в электрический сигнал. Например, в тензорезисторном датчике давления измеряемая величина воздействует на чувствительный элемент, изменяя его сопротивление. После подачи сигнала возбуждения, падение напряжения на резисторе позволит косвенно определить его сопротивление и, на основании зависимости сопротивления от давления, вычислить измеряемую величину.

Для разработчика датчик представляет собой черный ящик с известными соотношениями сигналов между входами и выходами.

2.1 Диапазон измеряемых и выходных значений

Диапазон измеряемых значений показывает, какое максимальное значение входного сигнала датчик может преобразовать в выходной электрический сигнал, не выходя за пределы установленных погрешностей. Данные цифры всегда приводятся в спецификации на датчик, одновременно отображая возможную точность измерений в том или иной диапазоне.

Следует понимать, что одни датчики при подаче входного сигнала больше максимальных значений просто войдут в насыщение и будут возвращать неверные данные. Другие же датчики (например датчики температуры) могут выйти из строя. В дальнейшем, для каждого типа датчика будут даны свои рекомендации.

Диапазон выходных значений датчика — это минимальное и максимальное напряжение, которое датчик способен выдать при минимальном и максимальном внешнем воздействии. Так как мы рассматриваем датчики, преобразующие входной сигнал в электрический, то диапазон выходных значений датчика будет определяться в вырабатываемом им напряжении, или пропускаемом через него токе. Одной из наших задач при подключении датчика будет согласование выходного диапазона датчика со входным диапазоном измерительного тракта.

2.2 Передаточная функция — статические и динамические характеристики

При работе с датчиком требуется знать соотношение уровней сигналов на входе и выходе. Отношение $W_d(p) = Y(p)/X(p)$ в операторном виде является передаточной функцией датчика и однозначно определяет характеристики датчика в статике и динамике.

Уравнение $Y(p) = W_d(p) * X(p)$ в реальной плоскости, т.е. функция $Y = f(x)$ будет являться статической характеристикой

Статическая характеристика может быть линейно и будет определяться как:

$$Y = aX + b \quad (1)$$

Где a – наклон прямой, определяемый чувствительностью датчика и b – постоянная составляющая (т.е. уровень выходного сигнала при отсутствии сигнала на входе)

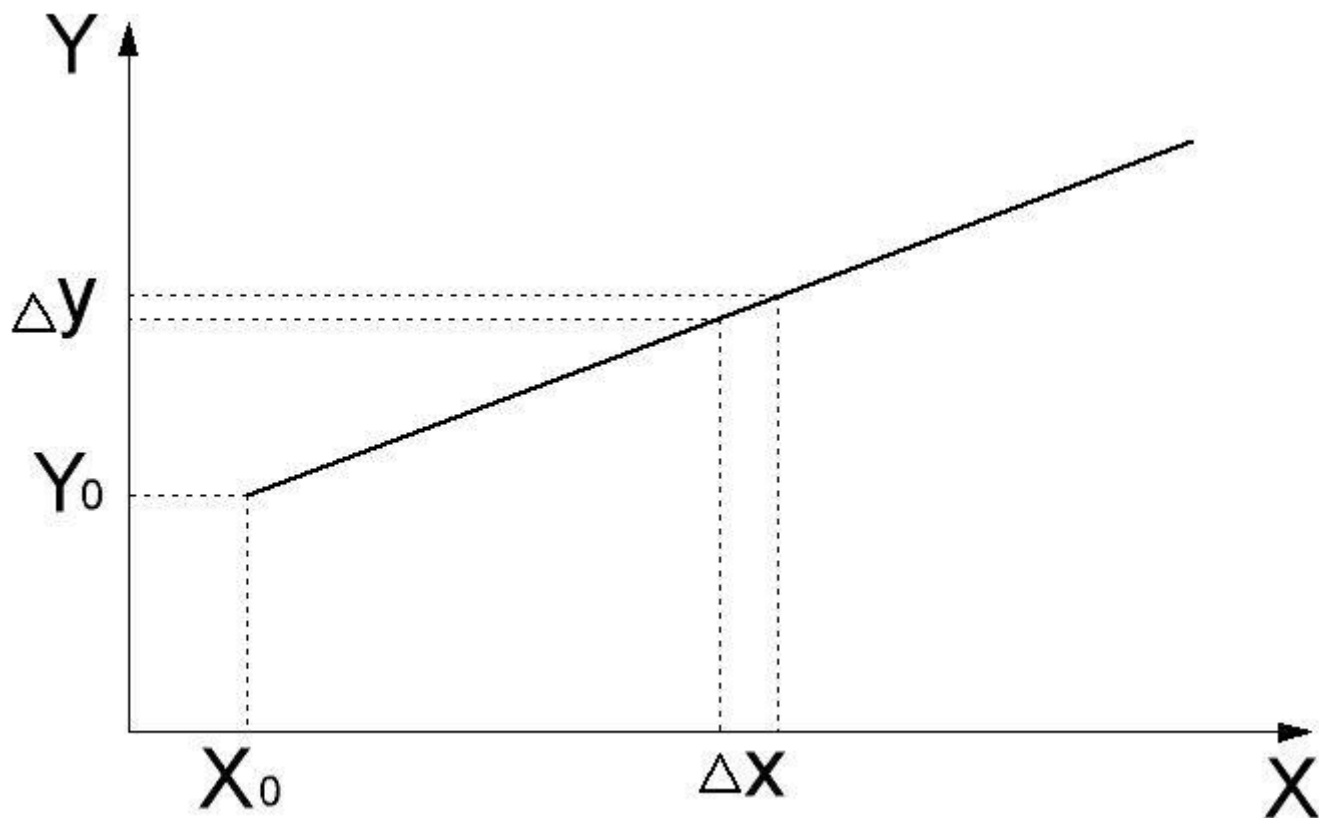


Рисунок 2. Линейная зависимость

Помимо датчиков с линейной зависимостью, могут быть датчики с логарифмической зависимостью, с уравнением вида

$$Y = a \ln X + b \quad (2)$$

Экспоненциальной:

$$Y = a e^{kX} + b \quad (3)$$

Или степенной:

$$Y = a_1 X^k + a_0 \quad (4)$$

Где k – постоянное число.

Существуют датчики с более сложной характеристикой. Но на то есть документация.

Однако, передаточная функция раскрывает и то, какими свойствами обладает датчик в динамике, т. е. насколько быстро и точно обрабатывает датчик выходной сигнал при быстром изменении входной величины. Практически каждый реальный датчик имеет в себе накопитель энергии — конденсатор, массу и т. п. Рассмотрим поведение датчика, динамические характеристики которого описываются уравнением первого порядка:

$$Y(t) = a_2 \frac{dX(t)}{dt} + a_1 X(t) + a_0 \quad (5)$$

В теории автоматического управления существует два тестовых входных сигнала. Это единичная функция — подача в нулевой момент времени единицы, и дельта-функция — подача сигнала бесконечной амплитуды и бесконечно малой длительности.

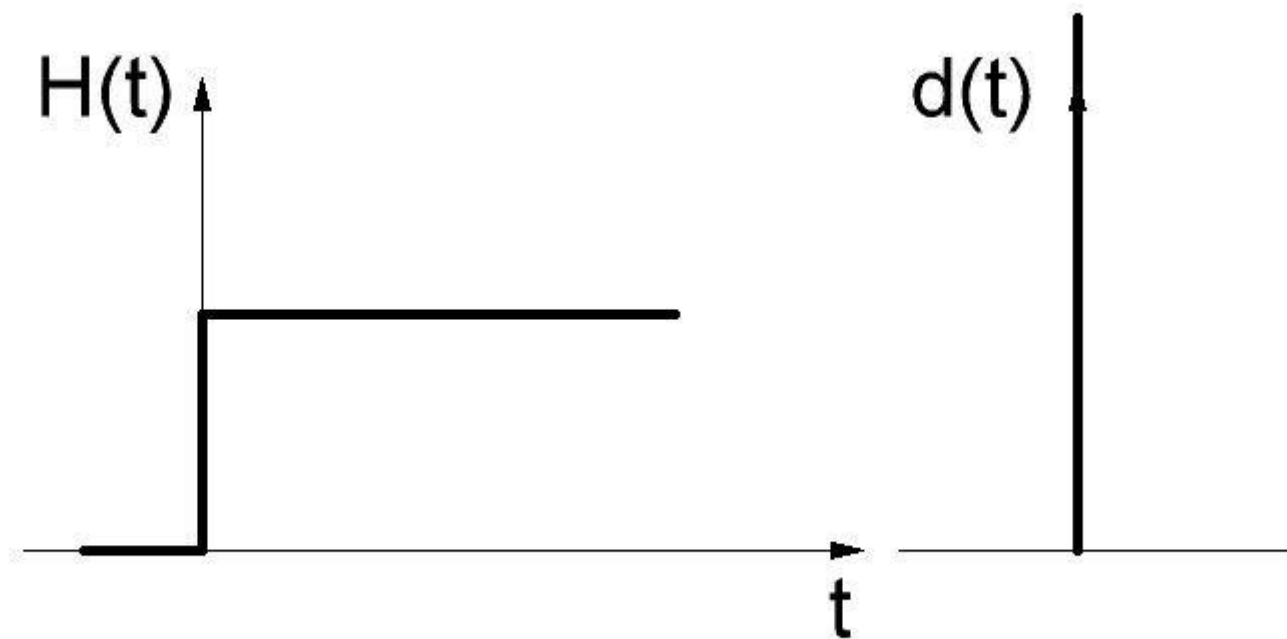


Рисунок 3. Единичная и дельта функции

Безынерционный, то бишь идеальный датчик в точности повторит форму входного сигнала. Реальный датчик, описанный формулой (5) выдаст следующую реакцию:

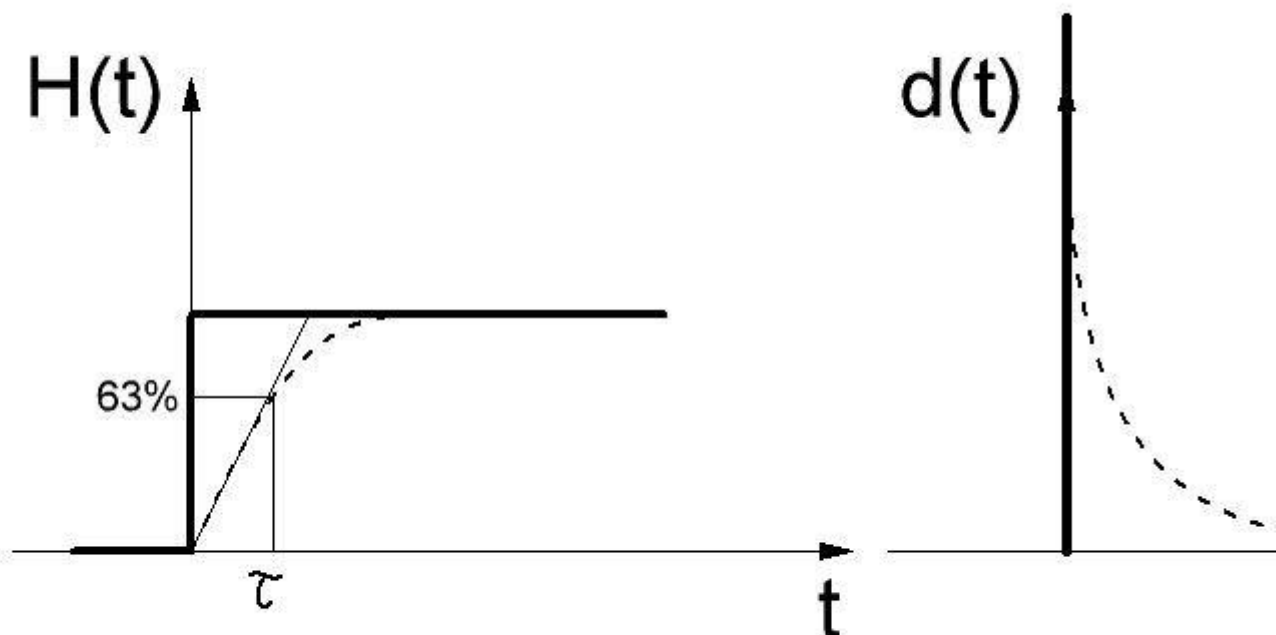


Рисунок 4. Реакция аperiodического звена первого порядка на тестовые сигналы

Следует отметить, что значение на выходе датчика будет соответствовать поданному на входе только после завершения переходного процесса, которое будет длиться $3-4\tau$, где τ — постоянная времени нашего звена. При $t = \tau$, выходное значение достигнет

$$1 - e^{-1} = 63\%$$

Нетрудно посчитать, что при $t = 2\tau$ выходное значение составит 86%, а при $t = 3\tau$ — 95% и переходный процесс будет считаться завершённым.

Таким образом нужно понимать, что, например, тот же датчик температуры будет реагировать на изменение температуры окружающей среды с некоторым запаздыванием из-за того, что между датчиком и окружающей средой имеется корпус, который должен поглотить тепло и нагреться. На это требуется время.

Разумеется, инерционные датчики могут описываться более сложными уравнениями, например представляться аperiodическими звеньями второго порядка, иметь задержку реакции и т. д. Особенности поведения таких звеньев подробно описаны в [1].

2.3 Точность, нелинейность

Одной из важных характеристик датчика является его **точность** в диапазоне измеряемых величин. Выходной сигнал датчика соответствует значению измеряемой величины с некоторой достоверностью, называемой погрешностью.

Например, датчик температуры имеет точность ± 2 градуса. Это означает, что при реальной температуре измеряемого объекта в 100 градусов, допустимые показания данного датчика температуры находятся в пределах 98 – 102 градусов. Погрешность датчика бывает разной.

Различают аддитивную и мультипликативную погрешность.

Аддитивная погрешность постоянна во всем диапазоне измерений.

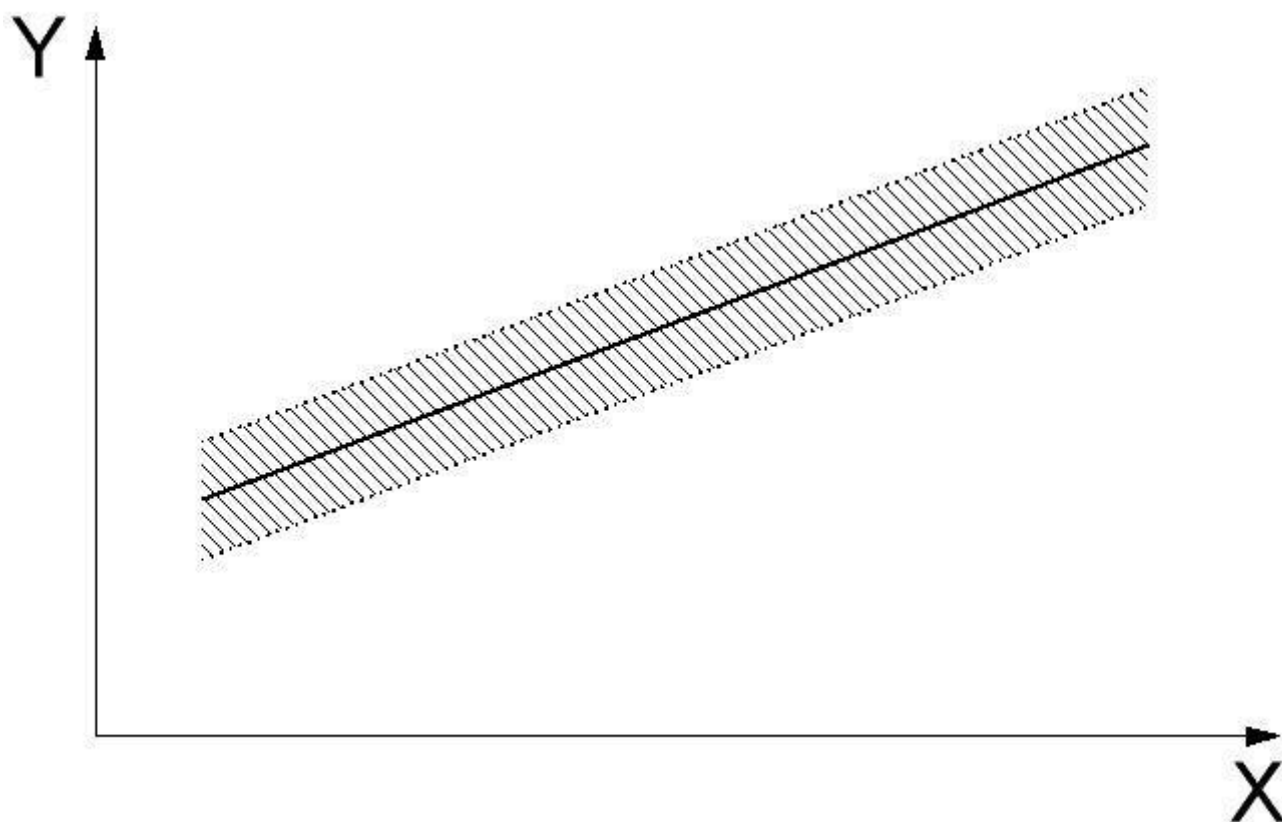


Рисунок 5. Аддитивная погрешность

Мультипликативная линейно зависит от уровня измеряемой величины:

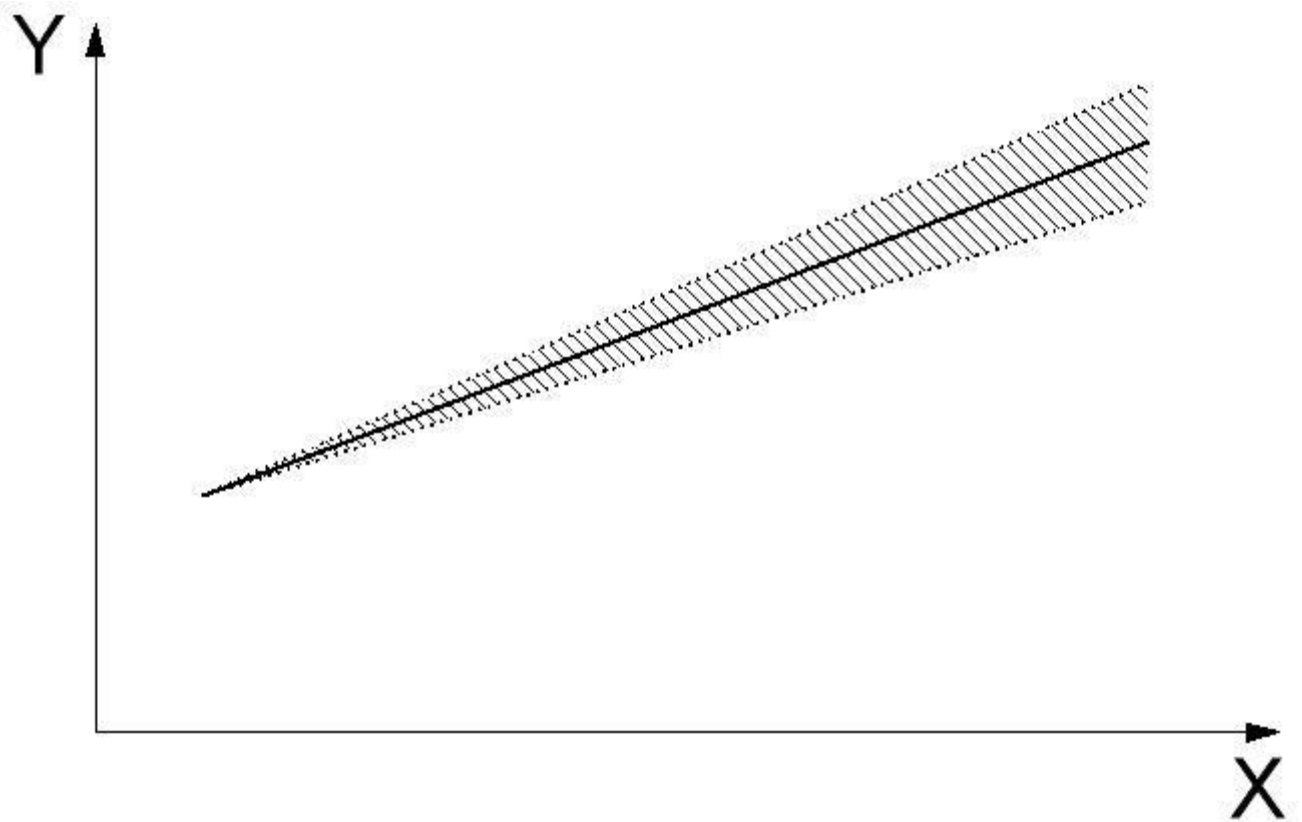


Рисунок 6. Мультипликативная погрешность

Кроме того, существует нелинейность датчика в измеряемом диапазоне. В зависимости от текущего диапазона измерения, коэффициент наклона передаточной функции изменяется в некоторых пределах. При этом, в спецификации указываются либо кривые изменения точности по диапазону, либо худшие показатели нелинейности в том или ином диапазоне.

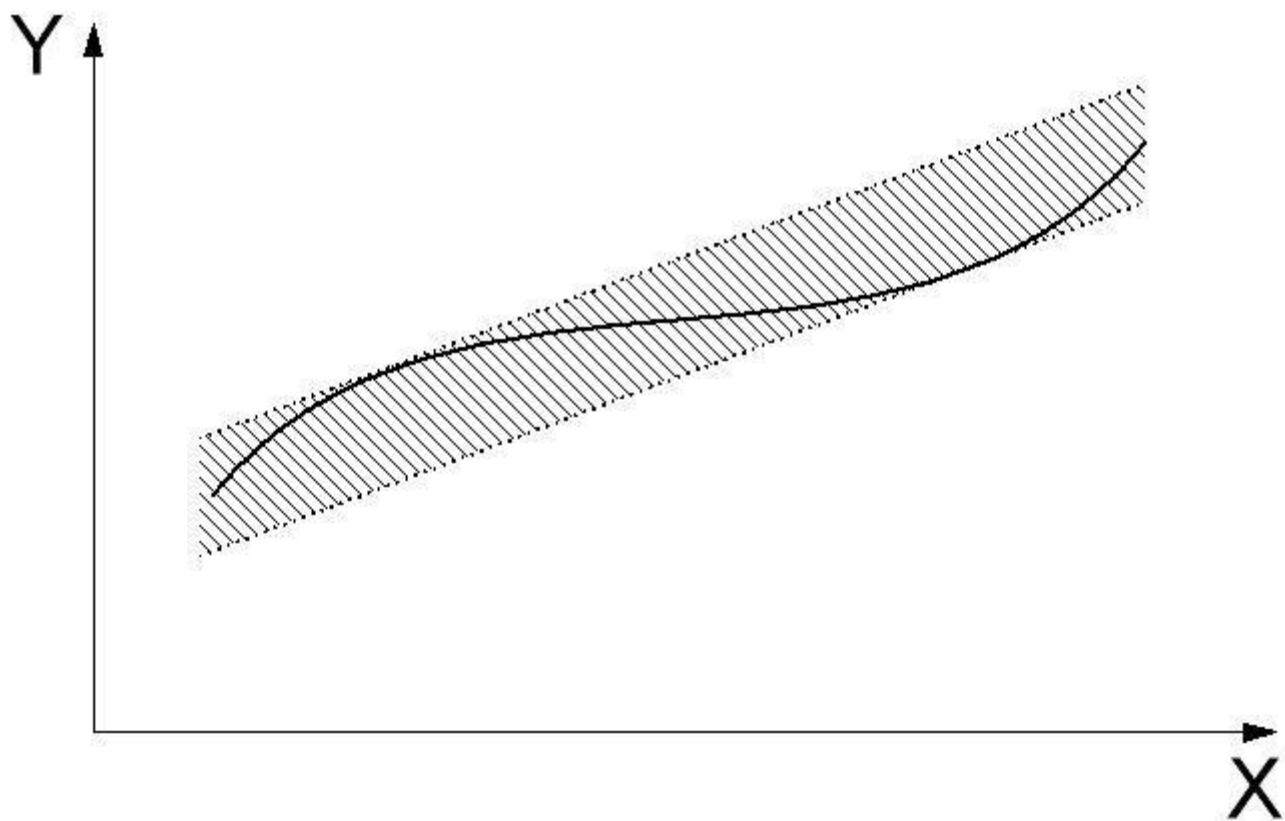


Рисунок 7. нелинейность датчика

Кроме того, некоторые датчики имеют эффект гистерезиса, когда для одного и того же входного сигнала после возрастания и убывания значения выходного сигнала получаются разными. Типичной причиной гистерезиса является трение и структурные изменения материалов. Наибольшему эффекту гистерезиса подвержены датчики на основе ферромагнитных материалов.

Для повышения точности и компенсации аддитивной и мультипликативной погрешности может производиться процесс калибровки датчика. Например, для линейного датчика необходимо с заведомо известной точностью определить показания в двух точках, находящиеся на разных концах рабочего диапазона. Для некоторых датчиков данные калибровки могут приводиться в паспорте на каждый конкретный экземпляр. Для проведения процедуры калибровки можно воспользоваться более точной аппаратурой, можно воспользоваться эталоном (например черное тело, эталонный килограмм и т. п.). Точность после калибровки естественно не сможет превышать точность эталона.

2.4 Чувствительность датчика, разрешающая способность и мертвая зона

Мертвая зона датчика — это нечувствительность датчика в определенном диапазоне входных сигналов. В пределах этой зоны выходные показания некорректны.

Для примера на рисунке 2 показания выходной величины для всех значений от 0 до x_0 не определены. Такой особенностью грешат, например, некоторые датчики тока, имеющие нулевое напряжение на выходе при токах меньших, к примеру, 10мА.

Во всем остальном диапазоне имеет место определенная **чувствительность** датчика, т. е. насколько силен прирост выходного сигнала на изменение входного сигнала. т. е. чувствительность определяется следующей формулой:

$$a = \frac{\Delta Y}{\Delta X}$$

Для линейного датчика, чувствительность будет постоянной на всем измеряемом диапазоне.

Разрешающая способность показывает, насколько малое изменение измеряемой величины способно вызвать изменение выходного сигнала. Например, какой-нибудь инкрементальный датчик положения имеет разрешающую способность в 1 градус. Аналоговые датчики обладают бесконечно большим разрешением, так как в их выходном сигнале нельзя определить отдельных ступеней его изменения.

3 Способ подключения датчиков

В зависимости от типа датчика, подключается он к измерительному тракту по-разному.

Подключение пассивного датчика

Так как пассивный датчик без посторонней помощи в ответ на внешнее воздействие самостоятельно вырабатывает для нас электрический сигнал, нам этот сигнал нужно считать.

В зависимости от того, будет ли наш датчик источником тока или источником напряжения, способ подключения будет отличаться.

К примеру, термопара является источником напряжения — напряжение на выходе не зависит от величины выходного тока (в разумных пределах конечно). Наша задача — измерить вырабатываемую ЭДС. Так как измерительный тракт будет иметь некоторое конечное сопротивление, схема подключения будет следующей:

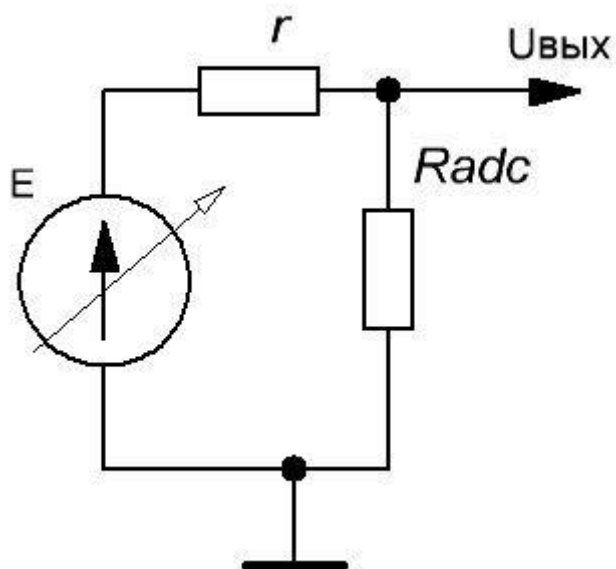


Рисунок 8. Подключение источника напряжения к АЦП

Если R_{adc} будет много больше внутреннего сопротивления r , тогда падение напряжения на нем будет стремиться к нулю и напряжение на входе АЦП будет стремиться к значению ЭДС.

Во второй части я подробно рассмотрю термопару, как один из самых точных и быстродействующих датчиков.

Другой случай, если наш датчик является источником тока, т.е генерируемое им напряжение зависит от пропускаемого через нагрузку тока.

Подключение датчика аналогично:

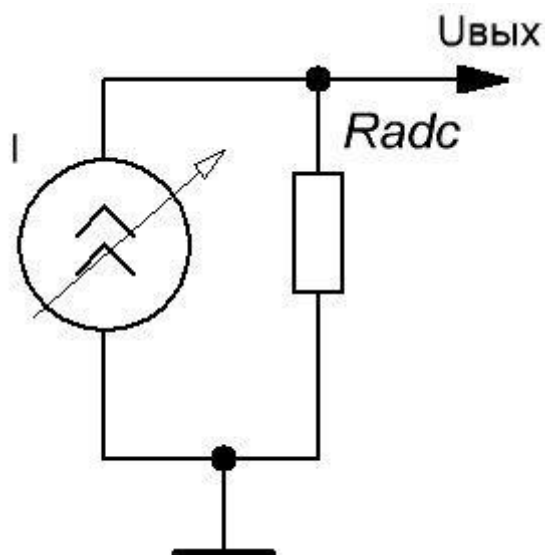


Рисунок 9. Подключение источника тока к АЦП

Однако, сопротивление нагрузки источника тока теперь должно стремиться к нулю. Для этого, датчик шунтируется резистором необходимого сопротивления, превращая тем самым, источник тока в источник напряжения:

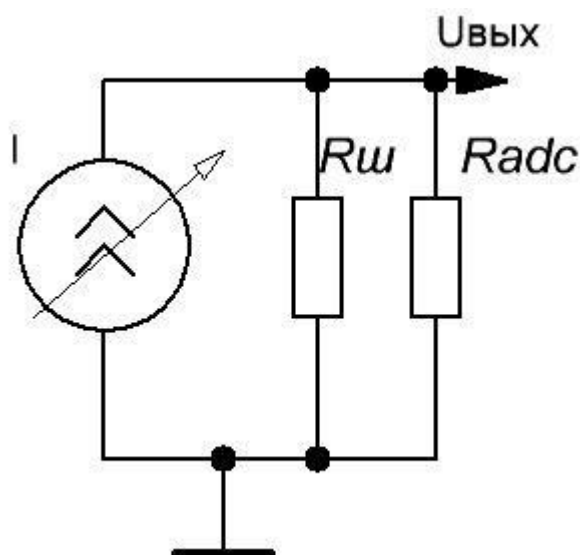


Рисунок 10. Правильное подключение источника тока к АЦП

Сопротивление резистора $R_{ш}$ рассчитывается как частное от деления максимального напряжения, подаваемого на вход АЦП на максимальный ток, который способен выдать датчик

$$R = \frac{U_{max}^{adc}}{I_{max}}$$

Наиболее яркий представитель такого датчика — датчик тока.

ВНИМАНИЕ: датчики, имеющие схему замещения в виде источника тока, следует обязательно шунтировать сопротивлением и не допускать обрыва цепи шунтирования при наличии сколь угодно малого входного воздействия. В противном случае, тот же датчик тока генерирует на свободных клеммах вторичной обмотки напряжение в киловольты до пробоя схемы измерения или самого датчика. Современные датчики тока тестируют на напряжении 1кВ и более, так что получить на выходе 2-3кВ, а еще попасть в них пальцем — не самая сложная задача.

Подключение активного датчика

Рассмотрим активные датчики, представляющие собой переменное сопротивление. В частности это терморезисторы, тензорезисторы и прочие подобные датчики. Чтобы сопротивление датчика измерить, его необходимо подключить к источнику тока и определить падение напряжения на нем:

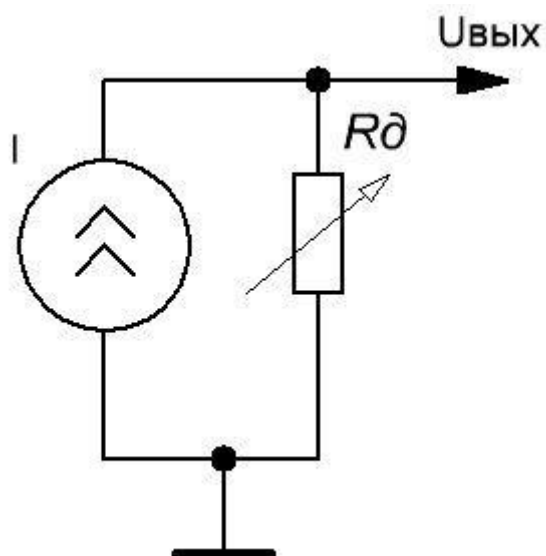


Рисунок 11. Подключение датчика к нерегулируемому источнику тока

Источник тока вырабатывает ток постоянного значения известной величины. Тогда, выходное напряжение будет определяться по формуле:

$$U_{\text{вых}} = I \cdot R_{\text{д}} \quad (7)$$

Например, рассчитаем выходное значение напряжения при токе источника 10мА если наш датчик изменяет сопротивление от 0,1кОм до 1 кОм. Тогда максимальное выходное напряжение будет равно

$$U_{\text{вых}}^{\text{max}} = 0,01 \cdot 1000 = 10\text{В} \quad (8)$$

Что вполне соответствует требуемому значению напряжения для аналоговой системы управления на базе операционных усилителей.

Где взять источник тока? Бывает так что он встроен в сам микроконтроллер. Например в микроконтроллерах ADuCM360/361 есть два встроенных источника тока 0,01-1мА. Правда там у них диагностическая задача — подавая малый ток через цепи датчика можно убедиться в его наличии и исправности.

Конечно, нам привычнее использовать источник напряжения с делителем:

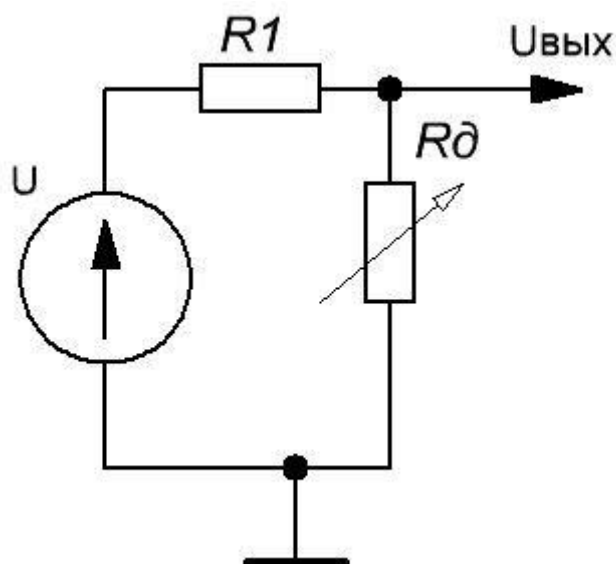


Рисунок 12. Подключение датчика к источнику напряжения с делителем

Если говорить на чистоту, то цепочка U-R1 образует тот же самый источник тока, только его параметры зависят от нагрузки — R_d . Напряжение на выходе будет определяться по следующей формуле:

$$U_{\text{вых}} = U \frac{R_1}{R_1 + R_d} \quad (9)$$

И тут всплывает главная проблема такого метода — от сопротивления нашего датчика в знаменателе не избавишься никак и показания становятся нелинейными, в отличие, кстати, от первого варианта.

Встает вопрос — каким должно быть сопротивление $R1$? Оно должно обеспечивать максимальный диапазон выходного напряжения. т. е. при известных значениях минимального и максимального сопротивления датчика R_{d1} и R_{d2} , $\text{abs}(U_{\text{вых}1} - U_{\text{вых}2}) \rightarrow \max$

С другой стороны, максимальное выходное напряжение у нас ограничено входными цепями измерительного устройства. Например, на вход микроконтроллера с питанием 5В необходимо подать напряжение, к примеру, не более 2,5В. Отмечу, что если максимально возможное напряжение, подаваемое на вход АЦП меньше напряжения питания, то мы сможем его туда подать.

Если наш датчик изменяет сопротивление от 0,1кОм до 1 кОм, то примем сопротивление резистора $R1$ равное верхней границе сопротивления датчика. Тогда $U_{\text{вых}}$ сможет изменяться в пределах от $1/11 U_{\text{вх}}$ до $1/2 U_{\text{вх}}$. В абсолютных цифрах данного примера — от 0,45 до 2,5В. И такими значениями мы используем $(2,5 - 0,45) / 2,5 = 82\%$ всего диапазона АЦП, что довольно неплохо.

Еще датчик можно воткнуть в состав измерительного моста и измерять разницу

напряжений в его плечах:

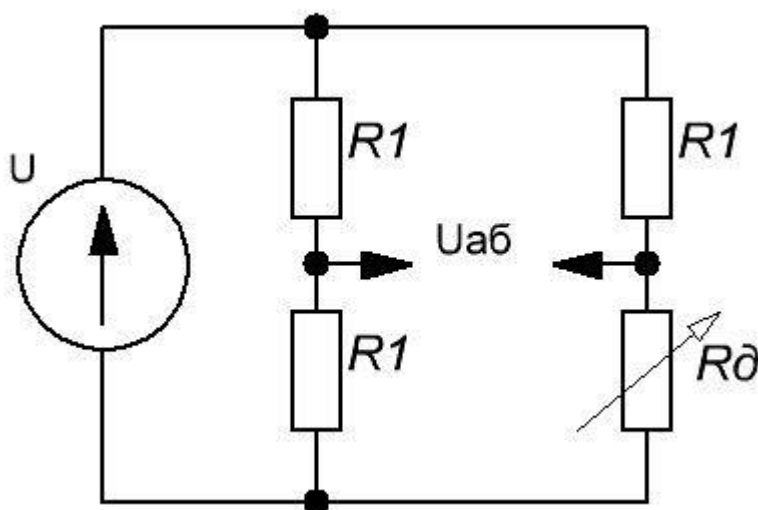


Рисунок 13. Датчик в составе измерительного моста

В этом случае мы работаем с дифференциальным АЦП, измеряя разность потенциалов U_{ab} . Она будет равна:

$$(10)$$

Причем сопротивление резистора $R1$ может быть таким, чтобы U_{ab} могло быть и отрицательным. Существуют датчики, внутренняя схема которых уже представляет собой балансный мост с необходимыми характеристиками. Позднее я рассмотрю примеры таких датчиков.

Существуют более удобные в использовании датчики. Они выдают необходимый аналоговый сигнал и без танцев с резисторами. Например, аналоговый датчик влажности ННН-4010-004 — трехвыводной корпус, 5В питание, линейный выход. Подключается это чудо так:

Рисунок 14. Подключение датчика влажности ННН-4010-004

Два провода к источнику опорного напряжения, выход — к АЦП микроконтроллера.

Подключение цифровых датчиков по стандарту 1-Wire

1-Wire это двунаправленная низкоскоростная цифровая шина передачи данных, требующая всего два провода — информационный провод и землю. Шина достаточно проста в использовании, поддерживает паразитное питание устройств от линии и позволяет подключать параллельно множество однотипных устройств вроде датчиков температуры(всеми любимыми DS18B20), или микросхем идентификации (iButton).

Паразитное питание организовывается следующим образом:

Рисунок 15. Паразитное питание устройств шины 1-Wire

А это обычное активное питание устройства, когда до источника рукой подать.

Рисунок 16. Питание устройства 1-Wire от внешнего источника

Количество подключенных параллельно датчиков фактически ограничено лишь параметрами линии.

Возможно горячее подключение и идентификация на ходу. Причем вычислительная сложность алгоритма идентификации $O(\log n)$

Более подробно с этим протоколом мы поработаем во второй части.

А пока, про сам протокол можно почитать по классической

ссылке: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

Подключение цифровых датчиков по стандарту I2C(Twi)/SMBus

Если 1-Wire требовала один провод данных, то эта шина, исходя из названия Two-Wire Bus — два.

Один из проводов — SCL будет тактирующим, по второму — SDA, полудуплексом будут передаваться данные.

Шина с открытым коллектором, следовательно обе линии необходимо подтянуть к питанию. Датчик будет подключаться следующим образом:

Рисунок 17. Подключение датчиков по I2C

Общее количество устройств, которые можно подключить к шине I2C — 112 устройств при 7-разрядной адресации. Каждому устройству на деле выделяется два последовательных адреса, младшим битом выставляется режим — на чтение или запись. Есть строгое требование по емкости шины — не более 400пФ.

Общеупотребительные значения скоростей — 100 кбит/сек и 10 кбит/сек, хотя последние стандарты допускают и скоростные режимы в 400 кбит/сек и 3.4мбит/сек.

Шина может работать как с несменяемым мастером, там и с передачей флага.

Огромное количество информации по протоколу можно найти по этой

ссылке: <http://www.esacademy.com/en/library/technical-articles-and-documents/miscellaneous/i2c-bus.html>

Подключение цифровых датчиков по стандарту SPI

Требуется как минимум три провода, работает в режиме полного дуплекса — т.е. организует одновременную передачу данных в обе стороны.

Линии связи:

- CLK — линия тактового сигнала.
- MOSI — выход мастера, вход слейва
- MISO — вход мастера, выход слейва
- CS — выбор чипа (опционально).

Одно из устройств выбирается мастером. Оно будет отвечать за тактирование шины. Подключение осуществляется перекрестным образом:

Рисунок 18. подключение по SPI и суть передачи

Каждое устройство в цепи содержит свой сдвиговый регистр данных. С помощью сигналов тактирования, спустя 8 тактов содержимое регистров меняется местами, тем самым, осуществляя обмен данными.

SPI — Самый скоростной из представленных интерфейсов передачи данных. В зависимости от максимально-возможных частот тактирования скорость передачи данных может составлять 20, 40, 75 мбит/сек и выше.

Шина SPI позволяет подключать устройства параллельно, но здесь возникает проблема — каждому устройству требуется своя линия CS до процессора. Это ограничивает общее количество устройств на одном интерфейсе.

Главная сложность в настройке SPI — это установить полярность сигнала тактирования. Серьезно. Настроить SPI не просто, а очень просто.

Коротко и ясно об SPI с описанием периферийных модулей SPI для AVR и MSP430 можно прочитать

здесь <http://www.gaw.ru/html/cgi/txt/interface/spi/index.htm>

4 Снятие показаний с датчиков

Пора бы уже прочесть с наших датчиков хоть какую-то информацию.

В зависимости от способа подключения датчика и его типа возможны различные способы снятия показаний. Следует отметить, что некоторые датчики, например цифровые датчики, или датчики состава газа, требуют предварительного запуска режима измерения, который может длиться некоторое время.

Таким образом, процесс измерения состоит из двух тактов — такт измерения

данных и такт снятия данных. При организации программы можно пойти по одному из следующих вариантов:

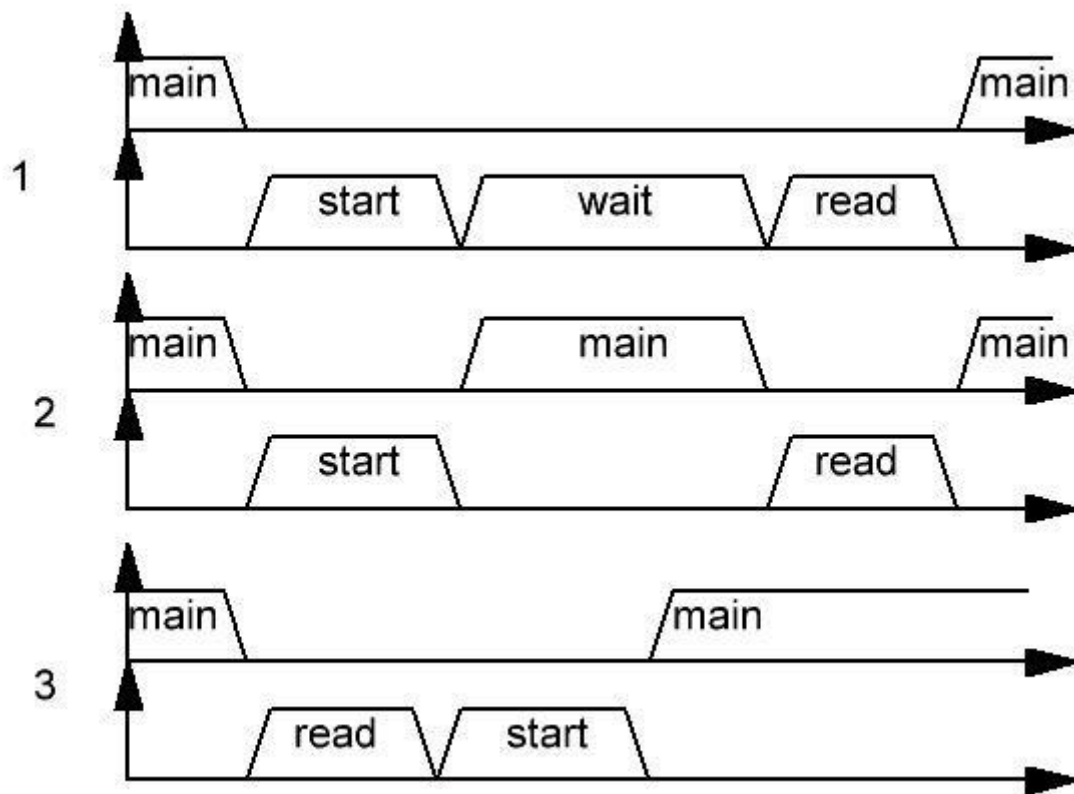


Рисунок 19. Процесс считывания показаний с датчика

Рассмотрим каждый вариант по отдельности и набросаем скелеты:

Вариант 1. запустили режим измерений, подождали, считали.

Вариант притягателен своей простотой, однако за ней кроется проблема — во время ожидания выполнения измерений микроконтроллер нагло простаивает, не выполняя задач. В большинстве систем автоматики такой режим — непозволительная роскошь.

В коде это будет выглядеть следующим образом:

```
Sensor.Start();//запустить процесс измерений
delay(MINIMAL_SENSOR_DELAY_TIME);//ждем завершения процесса
int var = Sensor.Read();//считываем данные
```

Вариант 2. запустили режим измерений, вернулись к другим задачам, по прошествии времени сработало прерывание, считали данные.

Один из лучших вариантов. Но наиболее сложный:

```
void Setup(){
  TimerIsr.Setup(MINIMAL_SENSOR_DELAY_TIME);//настраиваем прерывание по таймеру с необходимой периодичностью
  int mode = START;//переменная состояния
  Sensor.Start();//запускаем процесс измерений в первый раз
}
```

```

TimerIsr.Vector(){//обработчик прерывания по таймеру
if (mode == START{
mode = READ;
var = Sensor.Read();//если датчик был в режиме измерения, считываем данные
}
else
{
mode = START;
Sensor.Start();//если датчик был в режиме считывания данных, запускаем новый цикл измерений
}
}

```

Выглядит неплохо. позволяет варьировать время между циклами измерений и циклами считывания. например, датчик состава газов должен успеть остыть после предыдущих измерений, либо успеть нагреться во время измерений. Это разные периоды времени.

Вариант 3: Считали данные, запустили новый виток.

Если датчик позволяет после считывания данных запускать новый цикл измерений, то почему бы и нет — сделаем все наоборот.

```

void Setup(){
TimerIsr.Setup(MINIMAL_SENSOR_DELAY_TIME);//настраиваем прерывание по таймеру с необходимой периодичностью
Sensor.Start();//запускаем процесс измерений в первый раз
}
TimerIsr.Vector(){//обработчик прерывания по таймеру
var = Sensor.Read();//считываем данные
Sensor.Start();//запускаем новый цикл измерений
}

```

Отличный способ сэкономить время. и знаете что — такой метод отлично работает и без прерываний. Цифровые датчики хранят вычисленное значение вплоть до отключения питания. А с учетом того, что считывать сигналы с датчика влажности ввиду его инерционности в 15 секунд часто и не требуется, можно и вовсе сделать так:

```

void Setup(){
Sensor.Start();//запускаем процесс измерений в первый раз

while(1){
//много всякой остальной рутины
var = Sensor.Read();//считываем данные
Sensor.Start();//запускаем новый цикл измерений
}
}

```

Может быть и такой вариант, что наш датчик самостоятельно запускает новый цикл измерений и потом с помощью внешнего прерывания он сообщает о

завершении измерений. Например, АЦП можно настроить на автоматический режим считывания данных с частотой N Гц. С одной стороны, в обработчике прерывания будет достаточно реализовать только процесс считывания новых данных. С другой стороны, можно воспользоваться прерыванием АЦП с режимом Прямого Доступа к Памяти — ПДП(DMA). В этом случае по сигналу прерывания периферийный модуль АЦП на аппаратном уровне самостоятельно скопирует данные в определенную ячейку памяти в ОЗУ, тем самым обеспечивая максимальную скорость обработки данных и минимальное воздействие на рабочую программу (не надо уходить в прерывание, вызывать обработчик и проч.).

Но использование DMA сильно выходит за рамки данного цикла.

К сожалению, первый метод поголовно используется в библиотеках и примерах для Arduino, не позволяет этой платформе правильно использовать ресурсы микроконтроллера. Зато он проще в написании и отладке.

4.1 Работа с АЦП

Имея дело с аналоговыми датчиками имеем дело с АЦП. В данном случае рассматривается АЦП встроенный в микроконтроллер. Так как АЦП является по сути тем же датчиком — преобразует электрический сигнал в информационный — для него справедливо все что описано выше в разделе 2. Главными характеристиками АЦП для нас являются его эффективная разрядность, чувствительность, опорное напряжение и быстродействие. При этом, выходным значением АЦП преобразования будет некоторое число в выходном регистре, которое необходимо перевести в абсолютное значение в единицах измеряемой величины. В дальнейшем, для отдельных датчиков будут рассмотрены примеры таких расчетов.

4.1.1 Опорное напряжение

Опорное напряжение АЦП — это напряжение, которому будет соответствовать максимальное выходное значение АЦП. Опорное напряжение подается от источника напряжения, как встроенного в микроконтроллер, так и внешнего. От точности этого источника зависит точность показаний АЦП. Типовое опорное напряжение встроенного источника равняется напряжению питания или половине напряжения питания микроконтроллера. Могут быть и другие значения.

Например, таблица возможных значений опорного напряжения для микроконтроллера Atmega1280:

REFS1	REFS0	Voltage Reference Selection ⁽¹⁾
0	0	AREF, Internal V_{REF} turned off
0	1	AVCC with external capacitor at AREF
1	0	Internal 1.1V Voltage Reference with
1	1	Internal 2.56V Voltage Reference with

Рисунок 20. Выбор опорного напряжения для АЦП микроконтроллера Atmega1280

4.1.2 Разрядность АЦП и чувствительность

Разрядность АЦП определяет максимальные и минимальные значения в выходном регистре при минимальном и максимальном входном воздействии электрического сигнала.

Следует отметить, что максимальная разрядность АЦП может не соответствовать его эффективной разрядности.

Часть младших разрядов может быть отдана на шум. Обратимся к даташиту на микроконтроллер ADuCM360, имеющему 24-разрядный АЦП с эффективной разрядностью 14 бит:

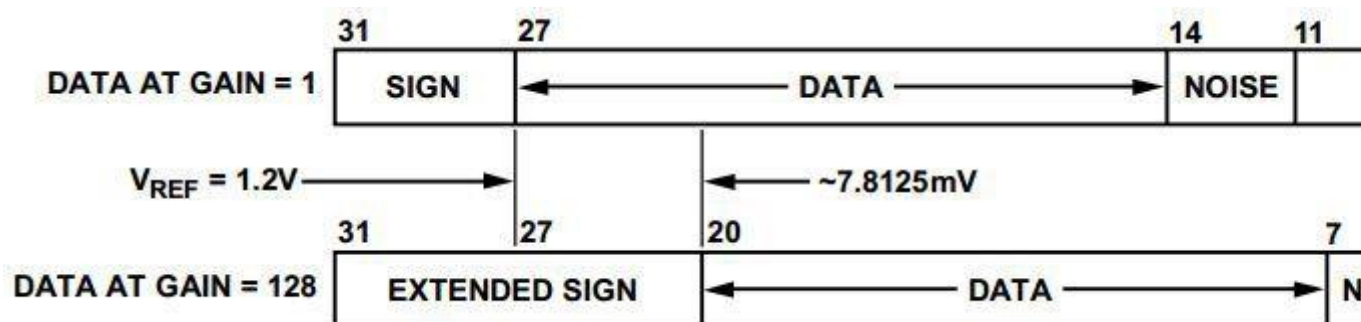


Рисунок 21. Назначение битов регистра данных АЦП

Как видно из рисунка, в 32-разрядном регистре, часть выделяется на знак, часть на нули и часть на шум. И лишь 14 разрядов содержат данные, имеющие указанную точность. В любом случае, эти данные всегда указываются в документации.

От эффективной разрядности АЦП зависит его чувствительность. Чем больше промежуточных ступеней выходного напряжения, тем выше будет чувствительность.

Допустим, опорное напряжение АЦП $U_{оп}$. Тогда, N-разрядный АЦП, имея 2^N возможных значений, имеет чувствительность

$$a = \frac{U_{оп}}{2^N}, B \quad (11)$$

Таким образом, для 12-разрядного АЦП и опорного напряжения в 3,3В его чувствительность составит $3,3/4096 = 0,8\text{мВ}$

Так как наш датчик также обладает определенной чувствительностью и точностью, будет неплохо, если АЦП будет обладать лучшими показателями

4.1.3 Быстродействие АЦП

Быстродействие АЦП определяет, насколько быстро считываются показания. Для АЦП последовательного приближения требуется определенное количество тактов, чтобы оцифровать уровень входного напряжения. Чем больше разрядность, тем требуется больше времени, соответственно, если к концу измерения уровень сигнала успевает измениться, это отразится на точности измерения.

Быстродействие АЦП измеряется в количестве семплов данных в секунду. Оно определяется как частота сигнала тактирования АЦП, деленная на требуемое для измерения число тактов. Например, имея частоту тактирования АЦП в 1МГц и 13 тактов для снятия показаний, быстродействие АЦП будет равно 77 килосемплов в секунду. Для каждого варианта разрядности возможно рассчитать свое быстродействие. В технической документации обычно указывается максимально-возможная частота тактирования АЦП и его максимальное быстродействие при той или иной разрядности.

4.2 Цифровые датчики

Главное преимущество цифровых датчиков перед аналоговыми — они предоставляют информацию об измеряемой величине в готовом виде. Цифровой датчик влажности вернет абсолютное значение влажности в процентах, цифровой датчик температуры — значение температуры в градусах.

Управление датчиком осуществляется с помощью имеющихся в нем регистром в форме вопрос-ответ. Вопросы следующие:

- Запиши в регистр А значение В
- Верни значение, хранящееся в регистре С

В ответ датчик, соответственно, либо записывает необходимые данные в регистр, производя настройку параметров или запуск какого-то режима, либо передает контроллеру измеренные данные в готовом виде.

На этом я закончу общий материал. В следующей части мы рассмотрим датчики HVAC с примерами.

После датчиков пойдет рассмотрение исполнительных устройств — там довольно много интересного с точки зрения значения теории автоматического управления, а потом доберемся и до синтеза и оптимизации регулятора всего этого безобразия.

UPD: Выражаю благодарность [amartology](#), [Arastas](#) и [Stross](#) за справедливые замечания по статье. Добавил материал по 2 и 4 разделам и пояснил некоторые спорные моменты.

Список полезной литературы:

1. Бессекерский В.А., Попов Е.П. Теория систем автоматического управления / В.А. Бессекерский, Е.П. Попов — Изд. 4-е, перераб. И доп. — Спб., Профессия, 2007. — 752с.
2. Датчики: Справочное пособие / В.М. Шарапов, Е.С. Полищук, Н.Д. Кошевой, Г.Г. Ишанин, И.Г. Минаев, А.С. Совлуков. — Москва: Техносфера, 2012. — 624 с.
3. Г. Виглеб. Датчики. Устройство и применение. Москва. Издательство «Мир», 1989
4. Современные датчики. Справочник. ДЖ. ФРАЙДЕН Перевод с английского Ю. А. Заболотной под редакцией Е. Л. Свинцова ТЕХНОСФЕРА Москва Техносфера-2005

Работа в сетях с TCP/IP.

Linux полностью поддерживает протокол TCP/IP (Transport Control Protocol/Internet Protocol). Протокол TCP/IP оказался наиболее успешным средством объединения компьютеров всего мира в единую сеть. Имея компьютер с системой Linux и адаптер Ethernet, можно подключить компьютер к локальной сети или (при наличии соответствующего подключения сети) к сети Интернет, являющейся всемирной сетью с протоколом TCP/IP.

Собрать небольшую локальную сеть из компьютеров с системой UNIX просто. Каждому компьютеру потребуется адаптер Ethernet, а кроме этого соответствующие кабели и другое аппаратное обеспечение. Если же некоторая фирма или университет предоставляют доступ в Интернет, то компьютер с системой Linux может быть легко подключен к этой сети.

Современная реализация протокола TCP/IP и связанных с ним протоколов для системы Linux называется ``NET-3'', ей предшествовала ``NET-2''. Это не имеет никакого отношения к так называемой версии NET-2 системы BSD UNIX; в действительности, ``NET-3'' в этом контексте означает вторую реализацию протокола TCP/IP для системы Linux.

Реализация NET-3 для системы Linux также поддерживает протоколы SLIP (Serial Line Internet Protocol: протокол последовательного подключения к Интернету) и PPP (Point-to-Point Protocol: протокол "точка-точка"). Протоколы SLIP и PPP позволяют подключаться к Интернету по телефонным линиям с помощью модема. Если соответствующая фирма или университет предоставляет доступ с помощью протокола SLIP или PPP, то нужно лишь дозвониться до SLIP или PPP-сервера. Если же компьютер имеет доступ в Интернет через адаптер Ethernet, то компьютер можно превратить в SLIP или PPP-сервер.

Полная информация о работе с протоколом TCP/IP в системе Linux может быть найдена в Практическом руководстве по использованию NET-3 в системе Linux (NET-3 HOWTO). Этот документ можно скопировать с анонимного доступа FTP к сайту sunsite.unc.edu. Документ NET-3 HOWTO является полным руководством к конфигурированию сетей с протоколом TCP/IP; в нем, в том числе, описываются подключения Ethernet, SLIP и PPP в системе Linux. Практическое руководство по использованию подключения Ethernet в системе Linux (Ethernet HOWTO) родственен ему документ; в нем описываются конфигурации различных драйверов для адаптеров Ethernet в системе Linux. Также имеется документ *Linux Network Administrator's Guide* (Руководство администратора сети в Linux), разработанный в рамках проекта Linux Documentation Project. Более подробно эти документы описаны в [приложении А](#).

Интересна будет также книга Крейга Ханта (Craig Hunt) под названием *TCP/IP Network Administration* (администрирование в сети TCP/IP). В этой книге имеется полная информация по использованию и конфигурированию протокола TCP/IP в системе Linux.

Требования к аппаратуре для работы в сети TCP/IP.

В системе Linux можно использовать протокол TCP/IP без каких-либо сетевых устройств, поскольку конфигурирование режима работы с сетевой "заглушкой" (loopback) позволяют вести диалог с самим собой. Это необходимо для некоторых программ и игр, в которых используется "заглушка".

Однако если предполагается использование системы Linux в сети с протоколом TCP/IP и подключением Ethernet, то адаптер Ethernet потребуется. В системе Linux имеется поддержка распространенных адаптеров Ethernet, таких, как 3com 3c503, HP PCLAN (серии 27245 и 27xxx), Western Digital WD80x3, Novell NE2000/NE1000 и многих других. Подробности можно найти в практических руководствах по адаптерам Ethernet и по аппаратному обеспечению в системе Linux (Linux Ethernet HOWTO, Linux Hardware HOWTO).

Нужно иметь в виду, что для некоторых карт поддержка пусть формально и имеется, но производительность оказывается невысокой, или присутствуют другие ограничения. Примерами могут служить адаптер 3Com 3C501, который работает, но с очень низкой производительностью, и адаптер Racal-Interlan NI6510 с микросхемой am7990, который не работает, если оперативная память превышает 16 МВ. Также многие адаптеры являются клонами, совместимыми с NE1000/NE2000, и могут быть причиной различных проблем. Более полное обсуждение вопроса о совместимости сетевого оборудования в системе Linux можно найти в документе Linux Ethernet HOWTO.

Linux также поддерживает протоколы SLIP и PPP, что позволяет заходить в Интернет через телефонные линии с помощью модема. В данном случае модем должен быть совместимым со SLIP или PPP-сервером: большинство серверов требуют как минимум модема со скоростью обмена 14.4bps и протоколом V.32bis. Производительность резко повышается при использовании модема со скоростью обмена 33.6bps и выше.

Конфигурирование TCP/IP.

В этом разделе обсуждается конфигурирование подключения к сети Ethernet с протоколом TCP/IP. Следует отметить, что этот метод будет работать для большинства компьютеров и вариантов системы Linux, но определенно не для всех. Обсуждение будет достаточным для того, чтобы указать верный путь к успешному конфигурированию подключения, но на этом пути имеется множество опасностей и тонких деталей, которые здесь не оговариваются. Более полная информация может быть найдена в документах *Linux Network Administrators' Guide* и NET-3-HOWTO.

Во-первых, предполагается, что на компьютере с Linux установлено программное обеспечение TCP/IP. Это обеспечение включает основные клиентские программы, такие как telnet и ftp, команды для системного администрирования, например, ifconfig и route (обычно эти файлы находятся в каталоге /etc), конфигурационные файлы (например, /etc/hosts). Если сетевое программное обеспечение еще не установлено, то сделать это помогут упомянутые выше документы по Linux.

Также предполагается, что ядро системы конфигурировано и откомпилировано так, что в нем имеется поддержка TCP/IP. Процедура компиляции ядра описывается в [разделе 4.9](#). Для того, чтобы задать опцию, соответствующую работе в сетях, надо ответить ``yes" на соответствующие вопросы на этапе make config и собрать ядро заново.

После того, как это будет сделано, нужно поменять несколько файлов, которые используются системой NET-3. По сути это простая процедура. К сожалению, разные дистрибутивы системы Linux помещают конфигурационные файлы и программы поддержки протокола TCP/IP в разные каталоги. Как правило, в каталог /etc, иногда в каталоги /etc, /usr/etc, /usr/etc/inet или другие странные каталоги. В наихудшем случае придется воспользоваться командой find, чтобы найти эти файлы. Следует помнить также, что не все дистрибутивы помещают конфигурационные файлы и программы NET-3 в одно и то же место; напротив, они могут быть рассеяны по разным каталогам.

Следующая информация касается прежде всего подключения Ethernet. Если планируется использовать протоколы SLIP или PPP, то этот раздел следует прочитать, чтобы узнать основные понятия, а при использовании руководствоваться последующими разделами.

Ваши параметры сети.

Прежде чем конфигурировать TCP/IP, нужно узнать определенную информацию о параметрах сети. В большинстве случаев необходимую информацию предоставит системный администратор. Потребуются следующие параметры:

- IP-адрес. Это уникальный адрес компьютера в формате четырех трехзначных десятичных чисел, разделенных точками, например, 128.253.153.54. Этот набор чисел должен предоставить системный администратор.

Если конфигурируется только режим работы с "заглушкой" (которая вообще-то называется закольцовывающим интерфейсом, loopback mode, т.е. когда нет подключения SLIP и нет адаптера Ethernet, а только моделируется подключение TCP/IP компьютера к самому себе, то IP-адресом будет 127.0.0.1.

- Маска подсети ("netmask"). Формат маски аналогичен формату IP-адреса. Маска определяет, какая часть IP-адреса соответствует номеру локальной подсети (subnetwork number), а какая номеру компьютера в сети. Маска представляет собой битовый шаблон. При наложении этого шаблона на адрес компьютера (хоста) можно узнать номер того участка сети, к которому относится этот адрес. Это очень важно для рассылки сообщений, поэтому если вдруг окажется, например, что можно установить связь с кем-либо вне локальной сети, а внутри сети такая связь не устанавливается, то может оказаться, что маска указана неверно.

Маска выбиралась администратором сети при ее проектировании, поэтому он должен ее точно знать. Большинство локальных сетей имеют класс C и используют маску 255.255.255.0. Сети класса B имеют маску 255.255.0.0. Программа NET-3 выбирает маску автоматически, предполагая по умолчанию, что подсети отсутствуют, если они не указаны.

То же относится и к адресу "заглушки". Поскольку адрес в этом случае всегда 127.0.0.1, то и маска всегда 255.0.0.0. Это можно указать явно или положиться на маску, даваемую по умолчанию.

- Адрес сети (network address). Этот адрес является результатом побитовой операции AND двух аргументов: IP-адреса компьютера и маски подсети. Например, если маска имеет вид 255.255.255.0, IP-адрес равняется 128.253.154.32, то адрес сети будет 128.253.154.0. Если маска равняется 255.255.0.0, то адрес сети будет равняться 128.253.0.0.

При использовании только loopback сетевой адрес отсутствует.

- Широковещательный адрес (broadcast address). Этот адрес используется для трансляции пакетов сообщений по всем компьютерам, объединенным в подсеть. Следовательно, если IP-адрес компьютера в сети определяется последним байтом (иными словами, если маска равна 255.255.255.0), то широковещательный адрес является результатом побитовой операции OR выражения 0.0.0.255 с IP-адресом Вашего компьютера.

Например, если IP-адрес компьютера 128.253.154.32, а маска 255.255.255.0, то широковещательный адрес будет равен 128.253.154.255.

Исторически сложилось, что в некоторых сетях в качестве широковещательного используется адрес сети. Этот факт следует знать, и если по этому вопросу возникают сомнения, надо обращаться к системному администратору. Разумеется, во многих случаях для конфигурирования сети достаточно будет взять соответствующие файлы с других компьютеров в той же сети и изменить в них только IP-адрес компьютера.

При использовании только loopback широковещательный адрес отсутствует.

- Адрес шлюза (gateway address). Это адрес компьютера, который действительно является "шлюзом" во внешний мир (т.е. к компьютерам вне локальной сети). Во многих случаях адрес шлюза получается из IP-адреса компьютера заменой последних групп цифр на .1. Например, если IP-адрес компьютера равняется 128.253.154.32, то адрес шлюза может иметь значение 128.253.154.1. Это значение может также сообщить системный администратор.

Реально шлюзов может быть несколько. В действительности *шлюз* это компьютер, который находится в двух различных сетях (т.е. имеет IP-адреса в различных подсетях). Шлюз занимается тем, что пересылает пакеты сообщений между этими двумя сетями. Многие сети имеют единственный шлюз во внешний мир (который представляет собой непосредственно примыкающую сеть). Однако в некоторых случаях к локальной сети примыкает несколько других сетей, и для каждой из них имеется свой шлюз.

При использовании только loopback адрес шлюза отсутствует. То же верно и в случае, когда локальная сеть изолирована от всех остальных.

- Адрес сервера имен в сети (name server address). У большинства компьютеров в сети имеется сервер, который преобразует имя компьютера в IP-адрес. Адрес этого сервера должен сообщить администратор сети. Такой сервер можно также запустить на собственном компьютере (программа называется *named*); в этом случае адрес сервера имен в сети равняется 127.0.0.1. Кроме тех случаев, когда *абсолютно необходимо* иметь собственный сервер имен, лучше воспользоваться тем, который имеется (если имеется). Совершенно отдельный вопрос конфигурирование программы *named*. Сейчас он рассматриваться не будет, поскольку цель данного раздела дать возможность приступить к работе в сети. Вопросы, касающиеся имен в сети, будут рассмотрены позднее.

При использовании только loopback адрес сервера имен отсутствует.

Пользователям SLIP/PPP информация, приведенная выше (кроме адреса сервера имен), может не требоваться. При подключении SLIP клиентский IP-адрес определяется либо статически, когда при каждом новом соединении клиенту дается один и тот же IP-адрес, либо динамически, когда адрес в момент соединения выбирается из некоторого множества свободных IP-адресов сервера. В последующих разделах конфигурирование подключения SLIP будет обсуждаться более подробно.

В реализации NET-3 заложена полная маршрутизация (full routing), поддерживаются множественные маршруты и подсети (последние на данный момент только по границам байта), и тому подобное. Приведенное выше иллюстрирует только принципиальные основы построения сетей TCP/IP. Конкретная сеть может обладать рядом особенностей; выяснить их можно у администратора сети. Также полезно будет почитать экранную документацию к программам *route* и *ifconfig*. Конфигурирование сетей TCP/IP лежит далеко за пределами темы данной книги, однако приведенное рассмотрение должно дать достаточно информации, чтобы пользователь смог выяснить остальное самостоятельно.


```

# место и продолжайте редактировать дальше.
GATEWAY="128.253.154.1" # ЗАМЕНИТЕ на адрес ВАШЕГО шлюза

/etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK} broadcast ${BROADCAST}

# Если у вас нет широковещательного адреса, то замените
# предыдущую строчку на следующую:
# /etc/ifconfig eth0 ${IPADDR} netmask ${NETMASK}

/etc/route add ${NETWORK}

# Следующая строчка нужна только в случае, если есть шлюз
# (т.е. сеть связана с внешним миром)
/etc/route add default gw ${GATEWAY} metric 1

```

Еще раз отметим, что для конкретной конфигурации компьютера этот файл, возможно, придется в чем-то изменить. Приведенных в данном примере инструкций достаточно для сети простой конфигурации, однако для всех возможных конфигураций этого примера будет определено недостаточно.

Файл `rc.inet2` запускает различные серверы, входящие в обеспечение протокола TCP/IP. Наиболее важным из них является сервер `inetd`. Этот сервер находится в фоновом режиме и наблюдает за работой сетевых портов. Когда к компьютеру пытаются обратиться через какой-либо порт (например, на этот порт приходит команда `telnet`), сервер `inetd` создает для этого порта копию соответствующего демона (в случае прихода команды `telnet` это будет демон `in.telnetd`). Это оказывается проще, чем иметь сразу несколько отдельных работающих демонов (т.е. отдельных копий `telnetd`, `ftpd` и т.д.). В принятой схеме они запускаются по мере надобности программой `inetd`.

`Syslogd` демон протоколирования системы. Он собирает сообщения о работе от разных программ и складывает их в файлы протоколов согласно конфигурации, содержащейся в файле `/etc/syslogd.conf`. Функции сервера `routed` поддерживать динамическую информацию о маршрутах. Если компьютер пытается послать пакеты сообщений на другую сеть, то для этого ему могут потребоваться дополнительные данные в таблице маршрутов. Заботы о действиях с таблицей маршрутов берет на себя сервер `routed`, и пользователь может не вмешиваться в этот процесс.

В приведенном ниже примере файл `rc.inet2` запускает лишь необходимый минимум серверов. Помимо указанных, существует еще много других серверов, и многие из них относятся к конфигурированию NFS. Однако при попытке конфигурировать программное обеспечение протокола TCP/IP лучше всего начинать с минимальной конфигурации, а более сложные элементы (такие, как NFS) добавлять, когда часть программ уже работает.

Также отметим, что в приведенном ниже файле предполагается, что демоны работы в сетях располагаются в каталоге `/etc`. Разумеется, для конкретной конфигурации этот файл может быть отредактирован.

```

#!/bin/sh
# Пример файла /etc/rc.d/rc.inet2

# Запуск демона syslogd
if [ -f /etc/syslogd ] then
    /etc/syslogd
fi

# Запуск сервера inetd
if [ -f /etc/inetd ] then
    /etc/inetd
fi

# Запуск сервера routed

```

```
if [ -f /etc/routed ] then
    /etc/routed -q
fi
```

Среди дополнительных серверов, которые могут быть запущены скриптом `rc.inet2`, имеется сервер `named`. Это сервер имен в сети, и его функция состоит в том, чтобы преобразовывать (локальные) IP-адреса в имена компьютеров и обратно. Если в сети нет другого сервера имен или если требуется дать (локальные) имена другим компьютерам того же домена, можно воспользоваться этим сервером (однако для большинства конфигураций это оказывается ненужным). Конфигурирование сервера `named` является довольно сложным и требует предварительного планирования; тем, кому это интересно, можно порекомендовать обратиться к книгам по администрированию сетей TCP/IP.

Файл `/etc/hosts`.

Файл `/etc/hosts` содержит список IP-адресов и соответствующих им имен компьютеров. Как правило, файл `/etc/hosts` содержит информацию только о данном компьютере и, возможно, о других важных компьютерах (таких, как сервер имен или шлюз). Для других компьютеров в локальной сети преобразования IP-адресов в имена и обратно будет обеспечиваться локальным сервером имен в сети.

Например, если имя компьютера `loomer.vpizza.com`, а IP-адрес `128.253.154.32`, то файл `/etc/hosts` на этом компьютере может выглядеть так:

```
127.0.0.1      localhost
128.253.154.32  loomer.vpizza.com loomer
```

Если используется только `loopback`, то единственная строка в файле `/etc/hosts` должна начинаться с чисел `127.0.0.1`, за которыми должны следовать `localhost` и затем имя компьютера (прим. переводчика: так должно быть вообще в любой конфигурации: дело в том, что если `loopback` будет первым в этом файле, системе не надо будет просматривать остальные строки).

Файл `/etc/networks`.

Файл `/etc/networks` содержит список имен и адресов как собственной сети, так и других сетей. Этот список используется командой `route` и дает возможность (при желании) для обозначения сетей использовать имена, а не адреса.

Каждая сеть, к которой предполагается создать отдельный маршрут командой `route` (обычно вызываемой скриптом `rc.inet1`), *должна* иметь соответствующую строку в файле `/etc/networks`.

Приведем пример:

```
default  0.0.0.0      # маршрут по умолчанию - обязателен
loopnet  127.0.0.0     # сеть для loopback - обязательна
mynet    128.253.154.0 # ИЗМЕНИТЕ на адрес ВАШЕЙ сети
```

Файл `/etc/host.conf`.

В этом файле определяется, как в системе будут разрешаться имена компьютеров. В нем должны содержаться две строки:

```
order hosts,bind
multi on
```

Эти строки сообщают соответствующим процедурам, что сначала все имена должны искажаться в файле `/etc/hosts`, а затем делается обращение к серверу имен (если он есть). Строка, начинающаяся с `multi`, позволяет в файле `/etc/hosts` для данного компьютера иметь несколько IP-адресов.

Файл `/etc/resolv.conf`.

В этом файле содержится конфигурация для программы разрешения имен. Здесь задается имя сервера имен (если таковой имеется) и имя домена. Имя домена получается, если из полного имени компьютера (например, для компьютера, зарегистрированного в сети Интернет) вычеркнуть имя компьютера (хоста). Например, если полное имя выглядит как `loomer.vpizza.com`, именем домена будет `vpizza.com`.

Пусть, например, имя компьютера `goober.norelco.com`, а сервер имен для него имеет IP-адрес, равный `128.253.154.5`. Тогда файл `/etc/resolv.conf` может выглядеть так:

```
domain    norelco.com
nameserver 127.253.154.5
```

Можно указать и несколько серверов имен, тогда для каждого из них в файле `/etc/resolv.conf` будет строка, начинающаяся с `nameserver`.

Установка имени хоста.

Имя компьютера устанавливается командой `hostname`. Она обычно вызывается из скрипта `/etc/rc` или `/etc/rc.local`; чтобы определить, где именно, надо просто просмотреть содержимое этих файлов и найти ее имя. Например, если (полное) имя компьютера должно быть `loomer.vpizza.com`, то надо отредактировать соответствующий `rc`-файл и вставить туда строку

```
/bin/hostname loomer.vpizza.com
```

Отметим, что исполняемый файл с именем `hostname` может и не присутствовать в каталоге `/bin`.

Проверка работоспособности.

После того, как все сконфигурировано, можно попробовать перезагрузить систему с новым ядром и попытаться использовать сеть. Неисправность может случиться во многих местах, так что разумно будет проверять правильность конфигурации сети по частям. К примеру, не следует начинать проверку работы сети с запуска программы `Mosaic` через сетевые функции системы `X`.

Для вывода на экран таблиц маршрутов можно использовать программу `netstat`. Именно эти таблицы являются наиболее частым источником неполадок. Точный синтаксис команды `netstat` подскажет `man`-страница. Для проверки подключения к сети мы предлагаем использовать клиентские программы, такие, как `telnet`, и попытаться подключиться к компьютерам как в своей локальной (под)сети, так и за ее пределами. Такой подход сузит область поиска возможных неполадок. Например, если не удастся соединиться с компьютерами в своей сети, но соединение с компьютерами вне этой сети устанавливается, то скорее всего ошибка связана с маской и с конфигурированием таблицы маршрутов. Для непосредственной работы с содержимым таблиц маршрутов можно вызвать (как пользователь `goot`) команду `route` для работы с записями в Вашей таблице маршрутизации.

Можно также проверить соединение через сеть путем указания точного IP-адреса вместо обращения по имени. Например, если не работает команда

```
telnet shoop.vpizza.com
```

то проблема может заключаться в неверной конфигурации сервера имен. Если при указании точного IP-адреса этого компьютера соединение устанавливается, то можно заключить, что почти наверняка в основном конфигурирование сети выполнено верно, а проблема связана с указанием адреса сервера имен.

Поиск ошибок в конфигурировании сети может оказаться трудной задачей, и она не может быть здесь освещена целиком. Если у вас под рукой нет специалиста, который помог бы вам, мы предлагаем обратиться к книге *Linux Network Administrators' Guide*, выполненной в рамках проекта Linux Documentation Project.

Настройка SLIP.

Протокол SLIP (Serial Line Internet Protocol) позволяет использовать протокол TCP/IP при подключении через последовательный порт. К этому порту может быть подключен либо модем, либо выделенная асинхронная телефонная линия (leased asynchronous line) какого-либо вида. Разумеется, при таком подключении требуется близко расположенный SLIP-сервер с соответствующими телефонными номерами. Многие университеты и коммерческие организации предоставляют доступ к SLIP-серверу за умеренную плату.

В настоящий момент имеется две основные программы, работающие с протоколом SLIP: `dip` и `slattach`. Обе программы предназначены для того, чтобы установить соединение SLIP через последовательное устройство. Для осуществления соединения SLIP *необходимо* воспользоваться одной из этих программ; просто позвонив на номер сервера одной из коммуникационных программ (например, `kermi`) и выполнив команды `ifconfig` и `route`, того же результата достичь не удастся. Дело в том, что программы `dip` и `slattach` выдают специальное обращение `ioctl()` к системе, которое перехватывает управление последовательным устройством и начинает использовать его как интерфейс SLIP.

Программа `dip` может позвонить на номер SLIP-сервера, произвести необходимый диалог для входа в систему (например, ввести по приглашению имя пользователя и пароль) и инициализировать подключение SLIP через открытый последовательный порт. Напротив, программа `slattach` не делает почти ничего помимо захвата последовательного устройства для использования его для подключения SLIP. Она полезна в случае, если имеется постоянное соединение со SLIP-сервером через выделенную телефонную линию, и для установления связи не надо звонить на сервер, вводить пароль и т.п. Большинству пользователей, работающих с протоколом SLIP, нужнее оказывается программа `dip`.

Программа `dip` может быть использована также для конфигурирования системы Linux в качестве SLIP-сервера. При этом другие компьютеры могут звонить на данный компьютер и подключаться к сети через вторичное подключение Ethernet (secondary Ethernet connection). Для более полной информации об этом следует обратиться к соответствующим документам и `man`-документации к `dip`.

Подключение SLIP кардинально отличается от подключения Ethernet тем, что в "сети" присутствует всего два компьютера: SLIP-клиент (ваш компьютер) и SLIP-сервер. По этой причине соединение SLIP часто называется соединением "точка-точка" (point-to-point connection). Обобщение этой идеи под названием "протокол PPP" (Point to Point Protocol) также реализовано в системе Linux.

При инициализации подключения к SLIP-серверу последний дает клиенту IP-адрес. При выборе адреса используется два основных метода. Некоторые SLIP-серверы дают статические IP-адреса, так что один и тот же клиент будет всякий раз при подключении получать один и тот же адрес. Однако многие SLIP-серверы дают динамические IP-адреса, которые могут быть новыми при каждом новом подключении. Как правило SLIP-сервер при соединении сообщает клиенту его IP-адрес и IP-адрес шлюза. Программа `dip` может прочесть эти данные, сообщаемые сервером при входе в систему, и использовать их затем при конфигурировании устройств SLIP.

По сути настройка SLIP напоминает настройку сетевой заглушки или адаптера Ethernet. Основные отличия обсуждаются ниже. Для ознакомления надо прочесть предыдущий раздел, посвященный конфигурированию основных файлов, относящихся к протоколу TCP/IP, и затем прочесть про отличия в данном разделе.

Соединение SLIP программой `dip` со статическими IP-адресами.

Если SLIP-сервер дает статические IP-адреса, их можно вписать в файл `/etc/hosts`. Можно также конфигурировать файлы, упомянутые в предыдущих разделах: `rc.inet2`, `host.conf` и `resolv.conf`.

Также можно сконфигурировать файл `rc.inet1` как описано выше. Для сетевой заглушки (вырожденной сети) надо выполнить только программы `ifconfig` и `route`. При использовании программы `dip` она сама выполнит команды `ifconfig` и `route` для устройства SLIP. Однако если используется программа `slattach`, то команды `ifconfig/route` *надо будет* включить в файл `rc.inet1` для устройства SLIP (см. ниже).

Во время соединения программа `dip` *должна* конфигурировать должным образом таблицы маршрутов (routing tables) для устройства SLIP. Однако для некоторых конфигураций поведение программы `dip` может оказаться неправильным, и после того, как программа `dip` произведет соединение с сервером, придется выполнять команды `ifconfig` и `route` вручную. Проще всего это сделать из скрипта командной оболочки, который запускается из программы `dip` и немедленно исполняет соответствующие команды конфигурирования. В большинстве случаев адрес шлюза совпадает с адресом SLIP-сервера. Этот адрес может быть известен заранее; кроме того, этот адрес будет выдан на экран SLIP-сервером при соединении. Диалоговый скрипт программы `dip` (см. ниже) может воспринимать и использовать эту информацию.

Если программа `dip` сконфигурировала интерфейс неправильно, то программа `ifconfig` может затребовать значение аргумента `pointopoint`. Например, если адрес SLIP-сервера равен 128.253.154.2, а IP-адрес клиента 128.253.154.32, то может потребоваться войти в систему как `root` и после соединения с помощью программы `dip` выполнить команду

```
ifconfig sl0 128.253.154.32 pointopoint 128.253.154.2
```

При выяснении деталей может оказаться полезной [man-документация к команде ifconfig](#).

Следует напомнить, что используемые командами `ifconfig` и `route` имена устройств имеют вид `sl0`, `sl1` и т.д., в отличие от устройств Ethernet, называемых `eth0`, `eth1` и т.д.

.

Соединение SLIP программой `slattach` со статическими IP-адресами.

При наличии выделенной телефонной линии (или кабеля), ведущей непосредственно к SLIP-серверу, нет необходимости использовать программу `dip` для инициализации соединения. Вместо нее для конфигурирования устройств SLIP может использоваться программа `slattach`.

В этом случае файл `/etc/rc.inet1` должен выглядеть примерно так:

```
#!/bin/sh
IPADDR="128.253.154.32" # ЗАМЕНИТЕ на ВАШ IP-адрес
REMADDR="128.253.154.2" # ЗАМЕНИТЕ на адрес ВАШЕГО SLIP-сервера

# Замените данные в следующих строках на параметры ваших
# последовательных устройств, используемых для соединения SLIP
slattach -p cslip -s 19200 /dev/ttyS0
/etc/ifconfig sl0 $IPADDR pointopoint $REMADDR up
/etc/route add default gw $REMADDR
```

`slattach` размещает первое свободное устройство SLIP (`sl0`, `sl1` и т.д.) на указанную последовательную линию.

Первым параметром программы `slattach` является тип используемого протокола SLIP. На настоящее время такими могут быть `slip` и `cslip`. Первый из них обычный протокол SLIP, а второй протокол SLIP со сжатием заголовка датаграммы (datagram header compression). В большинстве случаев нужно использовать `cslip`; однако если возникают проблемы, можно попробовать `slip`.

Если имеется более одного интерфейса SLIP, то надо создать некоторый принцип маршрутизации (routing considerations). Надо будет решить, какие нужно создать маршруты; такие решения принимаются только на основе знаний о конкретной структуре сети и подключений. При этом большую помощь окажет книга по конфигурированию сетей TCP/IP, а также [man-страница по программе route](#).

Соединение SLIP программой `dip` с динамическими IP-адресами.

Если SLIP-сервер назначает IP-адреса динамически, то никогда нельзя знать IP-адрес компьютера заранее. Следовательно, включать соответствующую запись в файл /etc/hosts нельзя (при этом IP-адрес для loopback 127.0.0.1 должен присутствовать).

Многие SLIP-серверы выдают IP-адрес, выданный клиенту (а также собственный), во время соединения. Например, некоторые типы SLIP-серверов выводят такие сообщения:

```
Your IP address is 128.253.154.44.  
Server address is 128.253.154.2.
```

Программа `dip` может перехватить эти цифры из выдачи сервера и использовать их для конфигурирования устройств SLIP.

Информация по составлению файлов конфигурации протокола TCP/IP для использования протоколом SLIP дана выше. Далее описывается конфигурирование программы `dip` для соединения со SLIP-сервером.

Использование `dip`.

`dip` заметно упрощает процесс подключения к SLIP-серверу, входа в систему и конфигурирования устройств SLIP. Программа `dip` будет весьма полезна кроме, может быть, случаев, когда для подключения к SLIP-серверу используется выделенная телефонная линия (leased line).

Для использования программы `dip` требуется написать диалоговый скрипт (chat script). В этом скрипте будет содержаться список команд, с помощью которых будет осуществляться диалог со SLIP-сервером во время входа в систему. Эти команды могут автоматически посылать на сервер имя пользователя и пароль, а также получить от сервера информацию о вашем IP-адресе.

Ниже дается пример диалогового скрипта программы `dip` для работы с сервером с динамическими IP-адресами. Для серверов со статическими адресами надо будет в начале скрипта установить значения переменных `$local` и `$remote`, равных соответственно вашему (local) IP-адресу и IP-адресу сервера. Подробно об этом написано в man-документации к `dip`.

main:

```
# Установка максимального размера передаваемого блока  
# (Maximum Transfer Unit). Это максимальный размер пакета,  
# передаваемого через устройство SLIP device. Многие  
# SLIP-серверы используют либо 1500 либо 1006; при сомнениях  
# следует спросить администратора сети.  
get $mtu 1500  
  
# Установить маршрут SLIP как маршрут по умолчанию  
default  
  
# Установить нужный последовательный порт и скорость передачи  
port cua03  
speed 38400  
  
# Выполнить начальную установку модема и терминальной линии.  
# Если эта команда создает проблемы, прокомментируйте ее.  
reset  
  
# Подготовка к набору номера. Подставьте в следующей строке  
# инициализационную команду (initialization string) для
```



```
# вашего модема.
send ATT&C1&D2\\N3&Q5%M3%C1N1W1L1S48=7\r
wait OK 2
if $errlvl != 0 goto error

# Набираем номер SLIP-сервера
dial 2546000
if $errlvl != 0 goto error
wait CONNECT 60
if $errlvl != 0 goto error

# Связь установлена. Входим в систему.
login:
sleep 3
send \r\n\r\n

# Ждем приглашения в систему
wait login: 10
if $errlvl != 0 goto error

# Посылаем имя пользователя
send USERNAME\n

# Ждем приглашения ввести пароль
wait ord: 5
if $errlvl != 0 goto error

# Посылаем пароль
send PASSWORD\n

# Ждем сигнала готовности SLIP-сервера
wait annex: 30
if $errlvl != 0 goto error

# Посылаем команды SLIP-серверу для инициализации соединения
send slip\n
wait Annex 30

# Получаем значение IP-адреса SLIP-сервера. Команда
# "get...remote" читает текст в формате xxx.xxx.xxx.xxx,
# и присваивает его переменной, указанной вторым аргументом
# (в данном случае переменной $remote).
get $remote remote
if $errlvl != 0 goto error
wait Your 30

# Получаем значение своего IP-адреса от SLIP-сервера и
# присваиваем его переменной $local.
get $local remote
if $errlvl != 0 goto error

# Запускаем связь через соединение SLIP
done:
print CONNECTED to $remote at $rmtip
print GATEWAY address $rmtip
print LOCAL address $local
mode SLIP
```

```
goto exit
error:
print SLIP to $remote failed.

exit:
```

dip автоматически исполняет команды `ifconfig` и `route` с учетом значений переменных `$local` и `$remote`. В данном примере этим переменным присваивает значение команда `get...remote`, которая получает текстовую строку от SLIP-сервера и присваивает ее указанной переменной.

Если запускаемые программой `dip` команды `ifconfig` и `route` не работают, то можно либо после выполнения программы `dip` запустить правильные команды из скрипта программной оболочки, либо непосредственно изменить текст скрипта для программы `dip`. Запуская программу `dip` с опцией `-v` можно получать протокол процесса подключения. Эта информация поможет выявить источник неудачи, если связь не будет устанавливаться.

Для того, чтобы запустить программу `dip` и установить подключение SLIP, можно использовать команду следующего типа:

```
/etc/dip/dip -v /etc/dip/mychat 2>&1
```

При этом различные файлы программы `dip` и диалоговый скрипт `mychat.dip` хранятся в каталоге `/etc/dip`.

Вышеприведенное обсуждение должно быть достаточным для вашего хорошего самочувствия на славном пути в сетевое сообщество через Ethernet или SLIP. И вновь мы настоятельно рекомендуем заглянуть в книгу по TCP/IP, особенно, если ваша сеть имеет специфику в маршрутизации, отличающую ее от рассмотренных здесь.

Что такое TCP/IP и как работает этот протокол

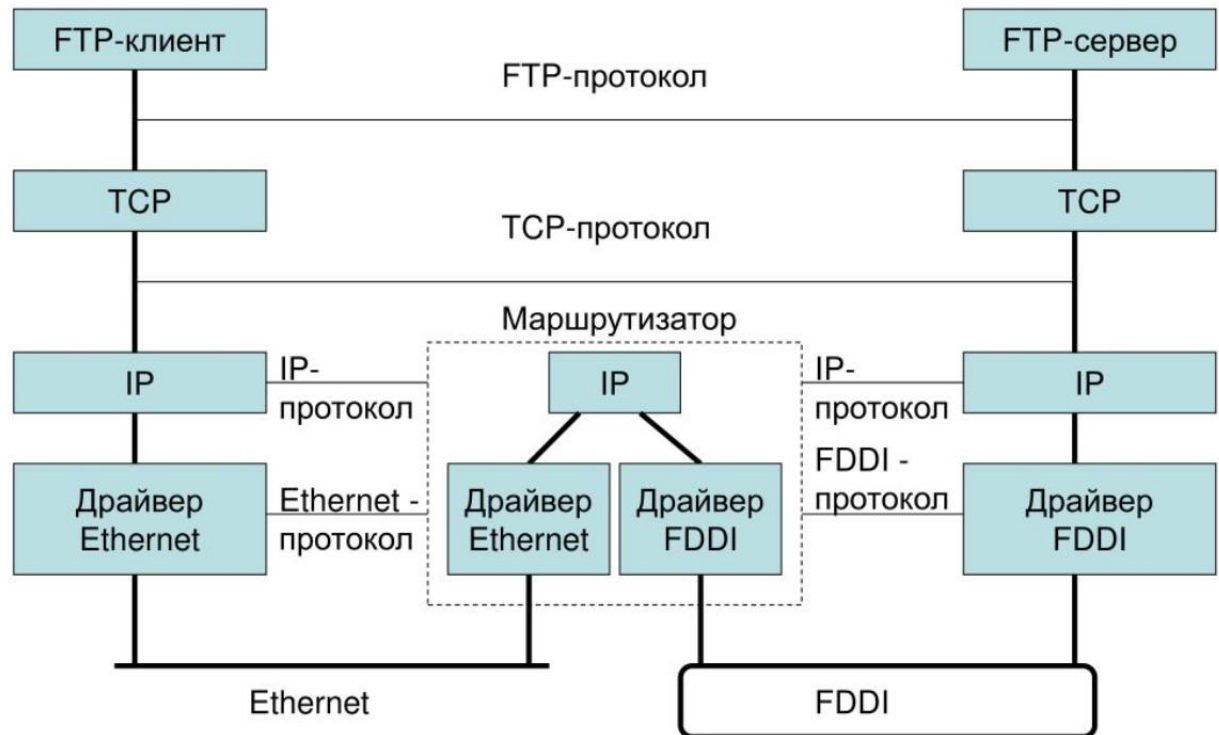
Протокол TCP/IP – это целая сетевая модель, описывающая способ передачи данных в цифровом виде. На правилах, включенных в нее, базируется работа интернета и локальных сетей независимо от их назначения и структуры.

Что такое TCP/IP

Произошло наименование протокола от сокращения двух английских понятий – Transmission Control Protocol и Internet Protocol. Набор правил, входящий в него, позволяет обрабатывать как сквозную передачу данных, так и другие детали этого механизма. Сюда входит формирование

пакетов, способ их отправки, получения, маршрутизации, распаковки для передачи программному обеспечению.

TCP/IP

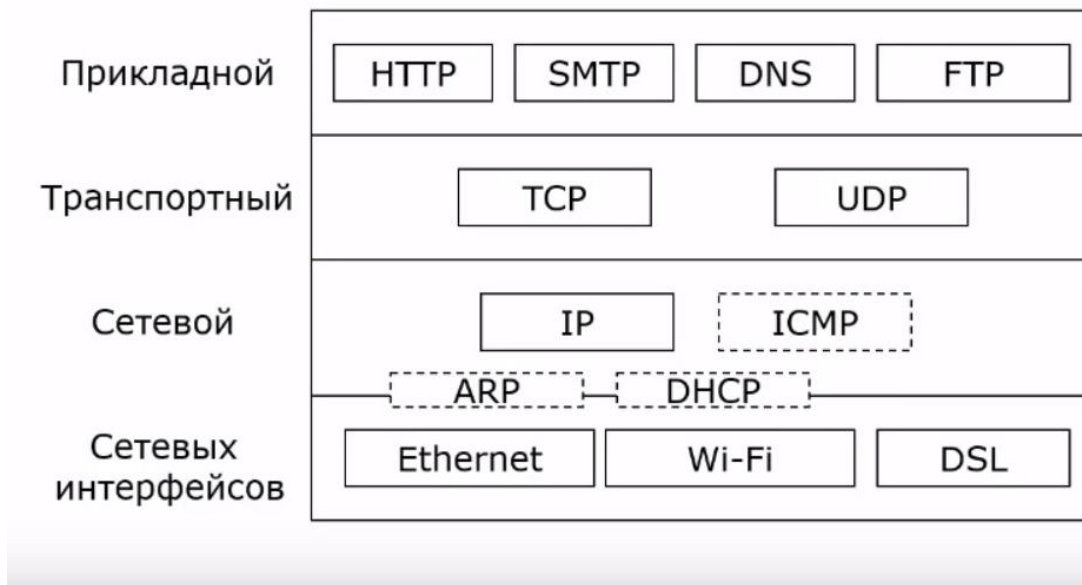


Стек протоколов TCP/IP был создан в 1972 году на базе NCP (Network Control Protocol), в январе 1983 года он стал официальным стандартом для всего интернета. Техническая спецификация уровней взаимодействия описана в документе RFC 1122.

В составе стека есть и другие известные протоколы передачи данных – UDP, FTP, ICMP, IGMP, SMTP. Они представляют собой частные случаи применения технологии: например, у SMTP единственное предназначение заключается в отправке электронных писем.

Уровни модели TCP/IP

Протокол TCP/IP основан на OSI и так же, как предшественник, имеет несколько уровней, которые и составляют его архитектуру. Всего выделяют 4 уровня – канальный (интерфейсный), межсетевой, транспортный и прикладной.



Канальный (сетевой интерфейс)

Аппаратный уровень обеспечивает взаимодействие сетевого оборудования Ethernet и Wi-Fi. Он соответствует физическому из предыдущего стандарта OSI. Здесь задача состоит в кодировании информации, ее делению на пакеты и отправке по нужному каналу. Также измеряются параметры сигнала вроде задержки ответа и расстояния между хостами.

Межсетевой (Internet Layer)

Интернет состоит из множества локальных сетей, объединенных между собой как раз за счет протокола связи TCP/IP. Межсетевой уровень регламентирует взаимодействие между отдельными подсетями. Маршрутизация осуществляется путем обращения к определенному IP-адресу с использованием маски.

Если хосты находятся в одной подсети, маркируемой одной маской, данные передаются напрямую. В противном случае информация «путешествует» по целой цепочке промежуточных звеньев, пока не достигнет нужной точки. Назначение IP-адреса проводится по стандарту IPv4 или IPv6 (они не совместимы между собой).

Транспортный уровень (Transport Layer)

Следующий уровень отвечает за контроль доставки, чтобы не возникало дублей пакетов данных. В случае обнаружения потерь или ошибок информация запрашивается повторно. Такой подход дает возможность полностью автоматизировать процессы независимо от скорости и качества связи между отдельными участками интернета или внутри конкретной подсети.

Протокол TCP отличается большей достоверностью передачи данных по сравнению с тем же UDP, который подходит только для передачи потокового видео и игровой графики. Там не критичны

потери части пакетов, чего нельзя сказать о копировании программных файлов и документов. На этом уровне данные не интерпретируются.

Прикладной уровень (Application Layer)

Здесь объединены 3 уровня модели OSI – сеансовый, представления и прикладной. На него ложатся задачи по поддержанию сеанса связи, преобразованию данных, взаимодействию с пользователем и сетью. На этом уровне применяются стандарты интерфейса API, позволяющего передавать команды на выполнение определенных задач.

Возможно и использование «производных» протоколов. Например, для открытия сайтов используется HTTPS, при отправке электронной почты – SMTP, для назначения IP-адресов – DHCP. Такой подход упрощает программирование, снижает нагрузку на сеть, увеличивает скорость обработки команд и передачи данных.

Порты и сокеты – что это и зачем они нужны

Процессы, работающие на прикладном уровне, «общаются» с транспортным, но они видны ему как «черные ящики» с зашифрованной информацией. Зато он понимает, на какой IP-адрес адресованы данные и через какой порт надо их принимать. Этого достаточно для точного распределения пакетов по сети независимо от месторасположения хостов. Порты с 0 до 1023 зарезервированы операционными системами, остальные, в диапазоне от 1024 до 49151, условно свободны и могут использоваться сторонними приложениями.

Комбинация IP-адреса и порта называется сокетом и используется при идентификации компьютера. Если первый критерий уникален для каждого хоста, второй обычно фиксирован для определенного типа приложений. Так, получение электронной почты проходит через 110 порт, передача данных по [протоколу FTP](#) – по 21, открытие сайтов – по 80.

Преобразование IP-адресов в символьные адреса

Технология активно используется для назначения буквенно-цифровых названий веб-ресурсов. При вводе домена в адресной строке браузера сначала происходит обращение к специальному серверу [DNS](#). Он всегда прослушивает порт 53 у всех компьютеров, которые подключены к интернету, и по запросу преобразует введенное название в стандартный IP-адрес.

После определения точного местонахождения файлов сайта включается обычная схема работы – от прикладного уровня с кодированием данных до обращения к физическому оборудованию на уровне сетевых интерфейсов. Процесс называется инкапсуляцией информации. На принимающей стороне происходит обратная процедура – декапсуляция

Стандартные стеки коммуникационных протоколов:
Стек tcp/ip

Стек TCP/IP был разработан по инициативе Министерства обороны США (DoD) более 20 лет назад для связи экспериментальной сети ARPANET с другими сетями как набор общих протоколов для разнородной вычислительной среды. Большой вклад в развитие стека TCP/IP, который получил свое название по популярным протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС Unix. Популярность этой операционной системы привела к широкому распространению протоколов TCP, IP и других протоколов стека. Сегодня этот стек используется для связи компьютеров в Интернете, а также в огромном числе корпоративных сетей.

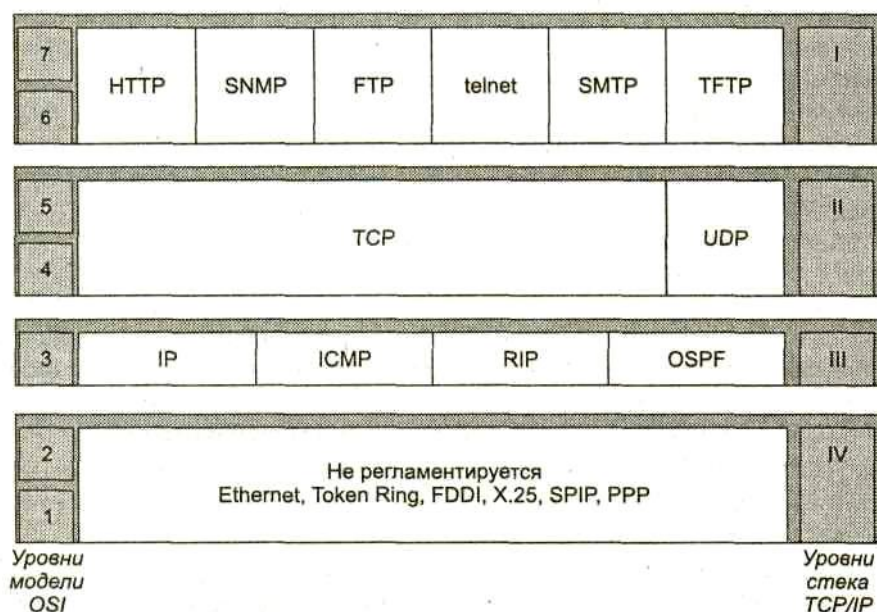
Поскольку стек TCP/IP изначально создавался для Интернета, он имеет много особенностей, дающих ему преимущество перед другими протоколами, когда речь заходит о построении сетей, включающих глобальные связи. В частности, очень полезным свойством, делающим возможным применение этого протокола в больших сетях, является его способность фрагментировать пакеты. Действительно, большая составная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть собственная величина максимальной длины единицы передаваемых данных (кадра). В таком случае переходе из одной сети, имеющей большую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого кадра на несколько частей. Протокол IP стека TCP/IP эффективно решает эту задачу.

Другой особенностью технологии TCP/IP является гибкая система адресации, позволяющая проще, чем другие протоколы аналогичного назначения включать в составную сеть сети разных технологий. Это свойство также способствует применению стека TCP/IP для построения больших гетерогенных сетей. В стеке TCP/IP очень экономно используются широковещательные рассылки. Это свойство совершенно необходимо при работе на медленных каналах связи, характерных для территориальных сетей.

Однако, как и всегда, за получаемые преимущества надо платить, и платой здесь оказываются высокие требования к ресурсам и сложность администрирования IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации больших вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети разнообразных централизованных служб типа DNS, DHCP и т. п. Каждая из этих служб направлена на облегчение администрирования сети, но в то же время сама требует пристального внимания со стороны администраторов.

В стеке TCP/IP определены 4 уровня.

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. Он объединяет службы, предоставляемые системой пользовательским приложениям. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие распространенные протоколы, как протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала (telnet), простой протокол передачи электронной почты (Simple Mail Transfer Protocol, SMTP), протокол передачи Ипертекста (HyperText Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах.



Архитектура стека TCP/IP

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- ☐ гарантированную доставку обеспечивает **протокол управления передачей** (Transmission Control Protocol, TCP);
- ☐ доставку по возможности, или с максимальными усилиями, обеспечивает **протокол пользовательских дейтаграмм** (User Datagram Protocol, UDP).

Для того чтобы обеспечить надежную доставку данных протокол TCP предусматривает установление логического соединения, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Этот протокол позволяет объектам на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме. TCP дает возможность без ошибок доставить сформированный на одном из компьютеров поток байтов в любой другой компьютер, входящий в составную сеть. TCP делит поток байтов на фрагменты и передает их нижележащему уровню межсетевого взаимодействия. После того как эти фрагменты будут доставлены средствами уровня межсетевого взаимодействия в пункт назначения, протокол TCP снова соберет их в непрерывный поток байтов.

Второй протокол этого уровня — UDP — является простейшим дейтаграммным протоколом, который используется в том случае, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня — прикладным уровнем или пользовательскими приложениями.

В функции протоколов транспортного уровня TCP и UDP входит также исполнение роли связующего звена между прилегающими к ним прикладным уровнем и уровнем межсетевого взаимодействия. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством, а после выполнения рапортует ему об этом. Нижележащий уровень межсетевого взаимодействия протоколы TCP и UDP рассматривают как своего рода инструмент, не очень надежный, но способный перемещать пакет в свободном и рискованном путешествии по составной сети.

Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на хостах.

Сетевой уровень, называемый также **уровнем интернета**, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах составной сети, образованной объединением множества сетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на

передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов, о функциях которого мы расскажем далее.

Основным протоколом сетевого уровня является межсетевой протокол (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями — от одного маршрутизатора до другого до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов прикладного и транспортного уровней протокол IP разворачивается не только на хостах, но и на всех шлюзах. Протокол IP — это дейтаграммный протокол, работающий без установления соединений по принципу доставки с максимальными усилиями.

К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, протоколы маршрутизации RIP и OSPF, занимающиеся изучением топологии сети, определением маршрутов и составлением таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены еще два протокола: протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP), предназначенный для передачи маршрутизатором источнику информации об ошибках, возникших при передаче пакета, и протокол групповой адресации (Internet Group Management Protocol, IGMP), использующийся для направления пакета сразу по нескольким адресам.

*Идеологическим отличием архитектуры стека TCP/IP от многоуровневой организации других стеков является интерпретация функций самого нижнего уровня — **уровня сетевых интерфейсов**.*

Нижние уровни модели OSI (канальный и физический) реализуют большое количество функций доступа к среде передачи, формированию кадров и согласованию уровней электрических сигналов, кодированию и синхронизации и некоторые другие. Все эти весьма конкретные функции составляют суть таких протоколов обмена данными, как Ethernet, Token Ring, PPP, HDLC и многих других.

У нижнего уровня стека TCP/IP задача существенно проще — он отвечает только за организацию взаимодействия с технологиями сетей, входящих в составную сеть. TCP/IP рассматривает любую сеть, входящую в составную сеть, как средство транспортировки пакетов до следующего на пути маршрутизатора.

Задачу обеспечения интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести:

- к определению способа упаковки (инкапсуляции) IP-пакета в единицу передаваемых данных промежуточной сети;
- к определению способа преобразования сетевых адресов в адреса технологии данной промежуточной сети.

Такой подход делает составную сеть TCP/IP открытой для включения любой сети, какую бы внутреннюю технологию передачи данных эта сеть не использовала. Для каждой новой технологии должны быть разработаны собственные интерфейсные средства. Следовательно, функции этого уровня нельзя определить раз и навсегда.

Уровень сетевых интерфейсов в стеке TCP/IP не регламентируется. Он поддерживает все популярные технологии; для локальных сетей — это Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, для глобальных сетей — протоколы двухточечных соединений SLIP и PPP, технологии X.25, Frame Relay, ATM.

Обычно при появлении новой технологии локальных или глобальных сетей она быстро включается в стек TCP/IP путем разработки соответствующего документа RFC, определяющего метод инкапсуляции IP-пакетов в ее кадры (например, спецификация RFC 1577, определяющая работу протокола IP через сети ATM, появилась в 1994 году вскоре после принятия основных стандартов ATM).

Каждый коммуникационный протокол оперирует некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 4.15).

Потоком данных, или просто потоком, называют данные, поступающие от приложений на вход протоколов транспортного уровня — TCP и UDP.

Протокол TCP «нарезает» из потока данных сегменты.



Рис. 4.15. Названия PDU в TCP/IP

Единицу данных протокола UDP часто называют **дейтаграммой**, или **датаграммой**. **Дейтаграмма** — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных также называют дейтаграммой. Однако очень часто используется и другой термин — пакет.

В стеке TCP/IP принято называть кадрами, или фреймами, единицы данных любых технологий, в которые упаковываются IP-пакеты для последующей переноски их через сети составной сети. При этом не имеет значения, какое название используется для этой единицы данных в технологии составляющей сети. Для TCP/IP фреймом является и кадр Ethernet, и ячейка ATM, и пакет X.25, так как все они выступают в качестве контейнера, в котором IP-пакет переносится через составную сеть.

Протокол SNA и операционные системы IBM

В устаревших мэйнфреймах IBM обычно используются протоколы стека *Systems Network Architecture*, SNA, который был изначально разработан в 1974 году. Фактически SNA — это набор частных протоколов, в которых в качестве метода доступа используется маркерное кольцо. Многие детали маркерных сетей, созданных компанией IBM, впоследствии были включены в стандарт IEEE 802.5. Однако в сети SNA кабельный участок обязательно строится на базе экранированной витой пары (STP), причем кабели имеют строго ориентированную маркировку (и разводку) (например, определенный конец кабеля должен идти к мэйнфрейму, а другой — к устройствам, подключенным к мэйнфрейму, таким как контроллеры дисковых накопителей или коммуникационных каналов). Это означает, что в сети SNA также используются частные (фирменные) кабельные разъемы и сетевые интерфейсы,

Стек протоколов SNA и эталонная модель OSI

Стек протоколов SNA базируется на семиуровневой модели (табл. 5.5), напоминающей эталонную модель OSI.

Таблица 5.5. Семиуровневая модель SNA

Уровень SNA	Эквивалентный уровень OSI	Назначение
Службы транзакций (Transaction Services)	Прикладной	Самый высокий уровень, управляет службами, от которых зависит работа прикладных программ (например, распределенных баз данных и приложений, выполняющихся одновременно на нескольких мэйнфреймах)
Представитель-ские службы (Presentation Services)	Представитель-ский	Управляет форматированием и преобразованием данных (например, перекодировкой из ASCII в EBCDIC и наоборот), также выполняет сжатие данных (хотя, в отличие от Представительского уровня OSI, этот уровень не обеспечивает шифрование данных)
Управление потоком данных (Data Flow Control)	Сеансовый	Устанавливает и поддерживает коммуникационные каналы между узлами, управляет потоками данных и обеспечивает восстановление после коммуникационных ошибок
Управление (Transmission Control)	Транспортный	Обеспечивает надежность передачи данных передачей от исходного узла к принимающему, а так же управляет шифрованием данных
Управление маршрутом (Path Control)	Сетевой	Управляет маршрутизацией и созданием виртуальных каналов, фрагментирует сообщения на блоки меньших размеров при передаче данных через разнородные сети (эту задачу выполняет Транспортный уровень OSI)
Управление(Data Link Control)	Канальный каналом	Форматирует данные на фреймы, обеспечивает маркерный доступ к сети при одноуровневых обменах данными между компьютерами
Управление Физическим Устройством (Physical Control)	Физический	Обеспечивает генерирование и кодирование электрических сигналов, работу физических интерфейсов, топологию сети и коммуникационную среду (например, кабель)

Достоинства и недостатки SNA

Аналогично любому стеку протоколов, SNA имеет как достоинства, так и недостатки. Отмечая достоинства, следует сказать, что архитектура SNA существует уже более четверти века и обеспечивает надежные и проверенные средства обмена данными с системами IBM. Существенным недостатком является то, что SNA – это частный (фирменный) стек протоколов, для которого нужны специальные устройства и дополнительное обучение процедурам конфигурирования, управления и отладки. По этим причинам сети SNA с мэйнфреймами IBM обычно работают очень хорошо, но это требует больших затрат на обучение персонала и поддержку сети.

Физические элементы сети SNA

В традиционной сети SNA с компьютерами IBM терминалы рассматривав как физические модули типа 2 (type 2). Физический модуль – это некоторое устройство, которое может подключаться к мэйнфрейму или управлять доступом к нему.

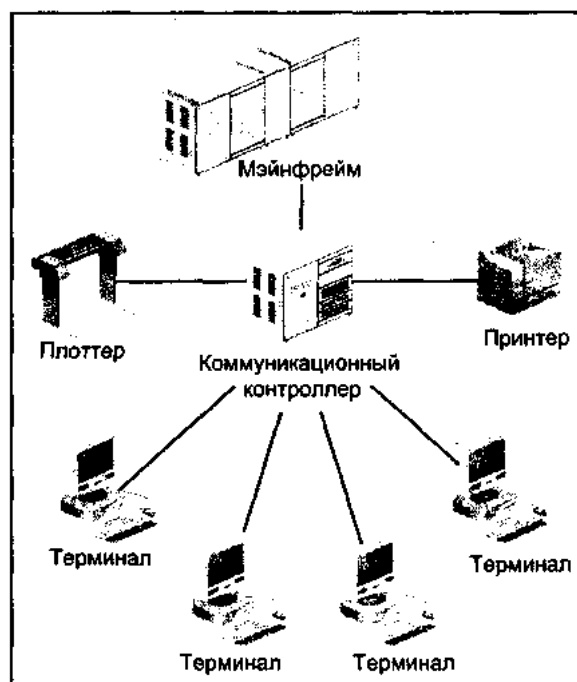


Рис. 5.6. Физические модули, подключенные к мэйнфрейму в сети SNA

К физическим модулям типа 2 относятся как терминалы, так и принтеры. Более сложные физические модули, обменивающиеся данными с мэйнфреймом, относятся к типу 2.1. В их число входят шлюзы (например, устройства, подключающие компьютеры с протоколом TCP/IP к мэйнфрейму IBM, работающему с SNA), мини-компьютеры и устройства, называемые контроллерами кластеров и позволяющие нескольким мэйнфреймам работать в тандеме. Терминалы и принтеры подключаются к физическим модулям типа 4, представляющим собой коммуникационные контроллеры, которые непосредственно соединены с мэйнфреймом и управляют трафиком обращений к нему (рис. 5.6).

Протоколы и приложения, работающие в стеке SNA

В табл. 5.6 перечислено множество протоколов и приложений, входящих в стек SNA.

Таблица 5.6. Протоколы и приложения, входящие в стек протоколов SNA

Аббревиатура или название	Полное название	Описание	Уровень модели SNA
APPN	Advanced Peer-to-Peer Networking (улучшенный протокол одноранговых сетей)	Обеспечивает одноранговые взаимодействия между устройствами, такими как мэйнфреймы, миникомпьютеры, шлюзы и контроллеры кластеров	Управление передачей
CICS	Customer Information Control System (абонентская информационно управляющая система)	Программная среда, предоставляющая программистам базовые средства взаимодействия с архитектурой SNA (в том числе безопасный доступ, управление файлами и накопителями). Альтернативой IMS является CICS	Управление потоком данных и Представительские службы
DDM	Distributed Data Management (управление распределенными данными)	Программы, обеспечивающие удаленный доступ к информации, хранящейся на мэйнфреймах IBM (например, по удаленному подключению со стороны другого мэйнфрейма,	Службы транзакций

Аббревиатура или название	Полное название	Описание	Уровень модели SNA
		находящегося в удалении)	
IMS	Information Management System (информационно-управляющая система)	Программная среда, предоставляющая программистам базовые средства взаимодействия с архитектурой SNA (в том числе безопасный доступ, управление файлами и накопителями). Альтернативой IMS является CICS	Управление потоком данных Представительские службы
NCP	Network Control Program (программа управления сетью)	Обеспечивает адресацию физических устройств и дополнительную логическую адресацию, а также маршрутизацию. Используется для шлюзовых коммуникаций SNA и управления ими (должна устанавливаться на любом шлюзе SNA для того, чтобы рабочие станции могли обращаться через шлюз к мейнфрейму; см. главы 1 и 4, где шлюзы рассматриваются подробнее)	Управление каналом и Управление маршрутом
SDLC	Synchronous Data Link Control (синхронное управление передачей данных)	Создает логические соединения (виртуальные каналы) в сетевом кабеле и координирует передачу данных по этим соединениям обеспечивает в каналах полудуплексную и полнодуплексную связь	Управление физическим устройством и Управление каналом
SNADS	SNA Distributed Services (распределенные службы SNA)	Программные средства, управляющие передачей документов. Используются системами электронной почты для передачи сообщений по указанным адресам	Службы транзакций
SSCP	System Services Control Point (точка управления системными службами)	Программное обеспечение, управляющее VTAM	Управлений передачей
Token Ring	Token Ring	Метод доступа, используемый сетях SNA	Управление физическим устройством Управление каналом
VTAM	Virtual Telecommunications Access Method (виртуальный телекоммуникационный метод доступа)	Управляет передачей данных в сети SNA (например, с помощью методов управления потоками). Обеспечивает обмен цифровыми данными	Управление передачей

Протокол DLC для доступа к операционным системам IBM

Если для доступа к мейнфрейму, работающему с SNA, используются компьютеры под управлением Windows 9x, Windows NT и Windows 2000, то альтернативой SNA-шлюзу является установка протокола *Data Link Control, DLC*. Этот протокол эмулирует SNA, и он может также применяться для подключения к некоторым устаревшим моделям сетевых принтеров, которые могут работать только с ним (например, старые принтеры Hewlett-Packard).

Совет

Протокол DLC не поддерживается в Windows XP. Если вы рассматриваете возможность перехода на эту систему, то учтите, что с ней вы не сможете использовать DLC для доступа к мейнфреймам IBM и, возможно, вам потребуется SNA-шлюз.

В основном протокол DLC является альтернативой TCP/IP в тех случаях, когда некоторый хост использует SNA-коммуникации. Недостатком этого протокола является то, что он не маршрутизируется. Кроме того, он на самом деле не предназначен для одноранговых взаимодействий между рабочими станциями, а служит только для подключения к старым мэйнфреймам IBM (например, ES9000) или мини-компьютерам IBM (например, AS/400). В практическом задании 5-7 рассказывается о том, как установить DLC в системе Windows 2000.

Протокол DNA для операционных систем компьютеров Digital (Compaq)

Созданная в 1974 году архитектура *Digital Network Architecture (DNA)* имеет такой же возраст, что и SNA. DNA использовалась в первых сетях компании Digital Equipment Corporation (DEC) и по-другому называлась DECnet. Затем этот стек протоколов применялся значительно реже.

Архитектура DNA предусматривает использование фреймов Ethernet II (или DIX – аббревиатура от названий компаний-разработчиков Digital, Intel и Xerox) в шинной топологии. Одним из достоинств DNA является то, что с самого начала эта архитектура близко следовала эталонной модели OSI. Не-Достаток DNA – то, что эта архитектура частная. Кроме того, после приобретения фирмы DEC компанией Compaq оригинальные компьютеры DEC и сети DNA стали менее популярными. Даже некогда известные компьютеры на базе процессора DEC Alpha все чаще заменяются продаваемыми под маркой Compaq рабочими станциями и серверами, реализованными с использованием процессоров Intel Itanium.

Поскольку DNA все реже встречается в сетях, уменьшается вероятность того, что вы столкнетесь с этой архитектурой на практике. Однако для общего представления в табл. 5.7 перечислены некоторые из протоколов и приложений, образующих стек DNA.

Таблица 5.7. Протоколы и приложения, входящие в стек протоколов

Аббревиатура	Полное название	Описание	Уровень модели OSI
CLNS	Connectionless-Mode Network Service (сетевая служба без установления соединения)	Обеспечивает работу служб без установления соединения (см. главу 2), а также маршрутизации	Сетевой
CONS	Connection Oriented Network Service (сетевая служба с установлением соединения)	Обеспечивает работу служб с установлением соединения для маршрутизации и контроля за ошибками маршрутизации	Сетевой
DDCMP	Digital Data Communications Message Protocol (протокол сообщений передачи цифровых данных)	Обеспечивает работу служб с установлением соединения и контролем ошибок. На уровне электрических сигналов позволяет осуществлять полудуплексную и полнодуплексную связь	Физический Канальный(подуровень LLC)
FTAM	File Transfer, Access, and Management (передача файлов, доступ и управление)	Позволяет передавать файлы с текстовым и двоичным содержанием	Прикладной
HDLC	High-Level Data Link Control (высокоуровневое управление каналом)	Создает логические соединения (виртуальные каналы) в сетевом кабеле и координирует передачу данных между ними. Управляет форматированием фреймов	Физический и Канальный
MAILbus	MAILbus	Соответствует стандарту X.400 на почтовые службы	Прикладной
Naming Service	Naming Service (служба имен)	Предоставляет сетевым устройствам службы именования, преобразующие адрес устройства в его имя и наоборот (что упрощает пользователям работу с устройствами)	Прикладной
NVTS	Network Virtual Terminal	Транслирует символы между Service	Представительский и

Аббревиатура	Полное название	Описание	Уровень модели OSI
	(служба сетевых виртуальных терминалов)	терминалами, сетями DNA и хост-компьютерами	Прикладной

Повышение производительности локальных сетей

Проще всего повысить производительность сети, если уменьшить количество протоколов, передаваемых через каждый маршрутизатор. При этом уменьшается рабочая нагрузка на маршрутизаторы, что позволяет им быстрее обрабатывать сетевой трафик. При меньшем количестве протоколов уменьшается и ненужный трафик, создаваемый в сети.

Вопросы для обсуждения

При выборе протоколов, используемых в сети, рассмотрите следующие вопросы.

- Должны ли пакеты маршрутизироваться?
- Какого размера сеть – маленькая (менее 100 узлов), средняя (100 – 500 узлов) или крупная (свыше 500 узлов)?
- Какие серверы используются и какие протоколы для них нужны?
- Имеются ли мэйнфреймы и какие протоколы для них требуются?
- Имеется ли непосредственный выход в Интернет или подключение к интранет-приложениям, использующим веб-технологии (виртуальная частная сеть)?
- Какая скорость требуется для подключений к глобальной сети?
- Имеются ли ответственные приложения?

Если фреймы нужно маршрутизировать (например, в корпоративной сети), то лучше всего применять протокол TCP/IP, поскольку он ориентирован на маршрутизацию и распространен во многих сетях. Для маленьких и средних немаршрутизируемых сетей (менее 200 узлов) на базе серверов Windows NT и при условии отсутствия подключения к Интернету наилучшим выбором остается протокол NetBEUI, обеспечивающий быстрые и надежные коммуникации. В сетях NetWare (с серверами версий ниже 5.0) можно использовать IPX/SPX, хотя в смешанной сети, где имеются старые серверы NetWare и новые серверы Windows 2000, могут понадобиться протоколы IPX/SPX и TCP/IP. Протокол NWLink является хорошим средством для подключения систем Windows 9x/NT/2000 к старым серверам NetWare.

Проблема каналов связи

Наличие подключения к Интернету или веб-службам требует развертывания Протокола TCP/IP, при этом службы FTP могут использоваться для передачи файлов. Также протокол TCP/IP лучше всего применять для связи с со временными мэйнфреймами и компьютерами UNIX, поскольку для подключения к мэйнфрейму или к приложению, работающему на компьютере UNIX, может потребоваться эмуляция терминала по протоколу Telnet. Для подключения к мэйнфреймам IBM и мини-компьютерам (если они работают в среде SNA) можно также использовать протокол DLC. И, наконец, протокол DNA по-прежнему может понадобиться в сети, где имеются старые компьютеры DEC (например, DEC VAX).

Примечание

TCP/IP – наилучший протокол для средних и крупных сетей. Он может маршрутизироваться, обладает надежностью для ответственных приложений, имеет надежный механизм контроля ошибок. В таких сетях важно иметь средства мониторинга сети и анализа неисправностей. Как изложено в главе 6, стек TCP/IP имеет протоколы, необходимые для решения подобных задач.

Во многих случаях для разных сетевых приложений нужно использовать различные протоколы локальных сетей. Иногда в современных сетях в любых сочетаниях применяются протоколы TCP/IP, NetBEUI, IPX/SPX, SM и даже DNA. Как вы уже знаете, развернутые протоколы связаны с типом используемых операционных систем. Также на их выбор влияет наличие связи с глобальными сетями (например, для выхода в Интернет нужен протокол TCP/IP, который может также потребоваться для связи локальных сетей между собой через глобальную сеть). Если, скажем, TCP/IP используется серверами в одной локальной сети, а рабочие станции из другой сети должны обращаться к этим серверам, то обе локальные сети и связывающая их глобальная сеть должны обеспечивать передачу протокола TCP/IP.

Удаление ненужных протоколов

Иногда рабочие станции в сети остаются настроенными на использование нескольких протоколов даже после того как все хосты и серверы переведены на протокол TCP/IP. В этом случае легко можно повысить производительность сети, удалив с рабочих станций ненужные протоколы. В практическом задании 5-8 рассказывается, как удалить протокол DLC из системы Windows 2000, а в задании 5-9 вы узнаете, как удалить службу Client Service for NetWare (и протокол NWLink IPX/SPX) из систем Windows 2000 и Windows XP Professional.

Резюме

- В значительной степени архитектуру сетей определяют протоколы. Во многих сетях используется несколько протоколов, с помощью которых осуществляется доступ к различным операционным системам сетевых серверов и хост-компьютеров.
- Обычно применяемые протоколы локальных сетей определяются типом сетевой серверной операционной системы, используемой в конкретной сети. Одной из старейших сетевых систем является NetWare, работающая со стеком протоколов IPX/SPX и обеспечивающая передачу данных между старыми версиями серверов NetWare и рабочими станциями (а также и другими серверами), подключенными к серверам. Протокол IPX/SPX реализован в тысячах локальных сетей, поскольку NetWare является одной из распространенных сетевых операционных систем. Однако в настоящее время благодаря тому, что многие сети связаны с Интернетом, новые версии NetWare (5.0 и выше) ориентированы на работу с более универсальным стеком протоколов TCP/IP.
- Родным протоколом для систем Windows NT Server является NetBEUI, появление которого связано с разработкой сетевой операционной системы LAN Manager, которую компания Microsoft начинала совместно с фирмой IBM. В средних и крупных сетях с серверами Windows NT чаще используется стек TCP/IP. С появлением систем Windows 2000 и Windows Server 2003 протокол TCP/IP пришел на замену NetBEUI, что определяется требованиями службы Active Directory и необходимостью доступа к Интернету.
- AppleTalk – это протокол, используемый компьютерами Macintosh с операционными системами Mac OS и Mac OS Server. Системы Windows NT, Windows 2000, Windows Server 2003 и Novell NetWare также поддерживают AppleTalk.
- Некоторые сетевые серверные операционные системы (в частности, UNIX) изначально были ориентированы на работу со стеком TCP/IP (а также и с Интернетом). В других сетевых операционных системах (например, NetWare, Windows NT и Mac OS Server) стек TCP/IP был реализован уже после создания этих систем.
- В первых системах IBM использовался стек протоколов SNA, который обеспечивал обмен данными между мэйнфреймами (мини-компьютерами) и терминалами, контроллерами и принтерами, а также между различными компьютерами. В операционных системах Windows имеется возможность установки протокола DLC для эмуляции коммуникаций SNA.
- Стек протоколов DNA был разработан для использования в сетях на базе компьютеров DEC, однако в настоящее время он применяется редко, поскольку количество таких компьютеров в сетях значительно уменьшилось.
- Простым и эффективным способом повышения производительности локальной сети является периодически проводимый анализ применяемых протоколов и удаление тех протоколов, которые больше не используются. Для доступа к компьютерам и принтерам.
- Вплоть до начала 1990-х годов сетевые технологии в первую очередь разбивались в области протоколов локальных сетей. В настоящее время архитектура этих протоколов нашла логическое завершение в стеке TCP/IP, а частные протоколы (такие как IPX/SPX и NetBEUI) используются реже.

Роль датчиков в сети интернета вещей

Интернет вещей (IoT) и периферийные вычисления (edge computing) спровоцировали внедрение во множество областей жизни smart-технологий: умный город, умные фабрики, умное сельское хозяйство, умная медицина и многое другое. Основа данных технологий – сбор данных в режиме реального времени с помощью различных датчиков и последующий анализ полученных данных.

TechJury, компания, специализирующаяся на программном обеспечении, проанализировав прогнозы различных фирм, предсказывает, что к 2025 году количество устройств, подключенных к интернету вещей, достигнет 64 миллиардов. В то же время маркетинговая компания [Grand View Research](#) прогнозирует, что рынок интернета вещей в том же году достигнет 949,42 миллиардов долларов. Постоянно ускоряющийся рост производства различных типов датчиков также говорит о росте индустрии интернета вещей. Обновление аппаратной и программной части датчиков и систем на их основе происходит стремительными темпами, и ожидается, что в ближайшие 25 лет размеры датчиков значительно сократятся, а сами датчики станут еще «умнее» и дешевле, что в свою очередь увеличит масштаб их использования.

В статье описана работа разных типов датчиков, а также принцип их взаимодействия с Интернетом вещей.

Как работают датчики?

Работа облачных серверов IoT и пограничных (шлюзовых) устройств зависят от датчиков, отвечающих за сбор данных в режиме реального времени. Так как окружающая нас действительность, как правило, оперирует сигналами, представленными в аналоговой форме, такими как температура в градусах Цельсия, расстояние в метрах, скорость в километрах в час, давление в ньютонах на квадратный метр и т. д., задача датчиков состоит в том, чтобы улавливать изменения данных параметров в окружающей среде и затем преобразовывать полученные данные в цифровой вид.

На рисунке 1 отображена конфигурация сети IoT с подключенными датчиками. Датчики являются конечной точкой в сети Интернета вещей, так как находятся дальше от облачных серверов, чем другие устройства сети. Несмотря на то, что датчики имеют небольшие размеры и не являются такой важной частью сети как облачные серверы, они могут сыграть решающую роль в проектировании и работе системы. Одним из примеров важности датчика в системе являются недавние крушения самолетов Boeing 737 Max, где одним из факторов, повлекших за собой обе трагедии, стала именно неисправность датчика.

Датчики должны подключаться и взаимодействовать с пограничными устройствами и облачными серверами для работы в сети IoT. В настоящее время для организации взаимодействия с серверами и пограничными устройствами используются преимущественно беспроводные технологии, такие как Bluetooth, NFC, RF, Wi-Fi, LoRaWAN и NB-IoT (сотовая связь). Организация сети и количество датчиков в ней определяют необходимое количество пограничных устройств, отвечающих за выполнение граничных вычислений (анализа данных с датчиков) перед отправкой информации на облачные серверы.

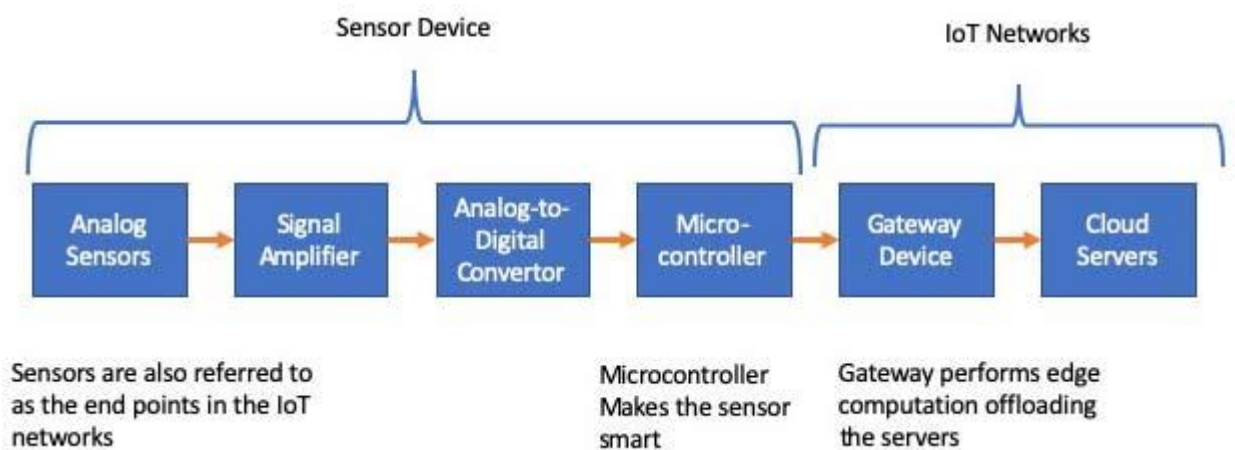


Рис. 1. Блок-схема сети интернета вещей, показывающая взаимосвязь между датчиком и сетью IoT

Сколько типов датчиков применяется в настоящее время?

Существует множество типов датчиков разных форм и размеров, способных измерить практически любую физическую величину. Некоторые из них представляют собой один единственный компонент, например, светочувствительный диод; другие - модуль с различной периферией и встроенным микроконтроллером.

Датчики могут быть использованы для измерения света, звука, температуры, давления, положения, изменения высоты и расстояния, газового состава воздуха, скорости и направления движения, плотности и состава жидкости и так далее. Существует также множество датчиков на основе технологий, используемых для обнаружения и распознавания объектов: радар, LiDAR, световой детектор, магнитный детектор, детектор инфракрасного излучения (IR), датчики на основе индуктивностей, устройства обработки изображений, устройства для работы с ультразвуковым излучением, сонары, устройства для работы с фотонным излучением, сенсорное распознавание, энкодеры и многое другое.

Ниже приведен примерный список доступных на сегодняшний день типов датчиков (рисунок 2):

- [Датчик атмосферного давления](#)
- [Датчик расстояния/ приближения](#)
- [Датчик влажности](#)
- [ИК-сенсор](#)
- [Датчик уровня](#)
- [Световой датчик](#)
- [Датчик движения](#)
- [Датчик дыма и газа](#)
- [Датчик температуры](#)
- [Датчик прикосновения](#)
- [Ультразвуковой датчик](#)

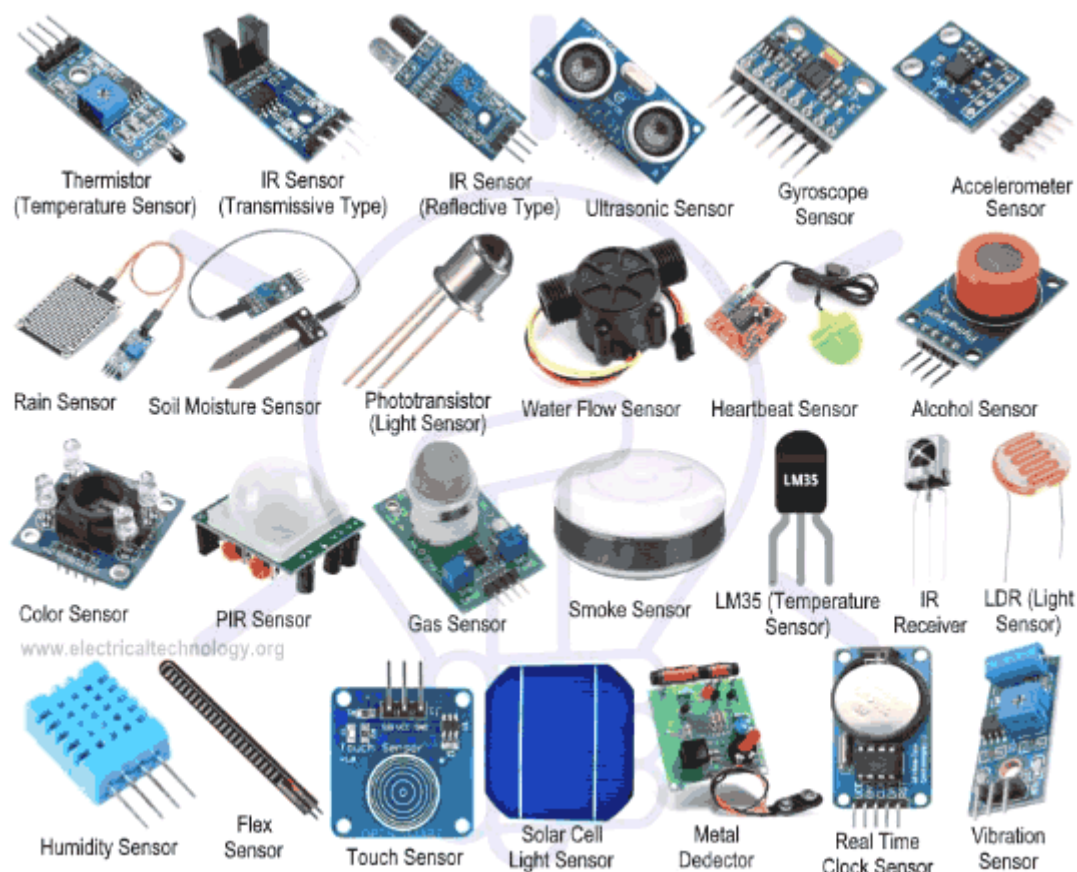


Рис. 2. Типы датчиков

Помимо прочего, датчики также могут быть классифицированы как активные и пассивные, аналоговые и цифровые.

Активные датчики обычно требуют внешней поддержки для работы. Это может быть внешний источник питания или беспроводная передача энергии на основе индуктивностей. В качестве примера можно привести дифференциальный трансформатор для измерения линейных перемещений (LVDT), который может использоваться для преобразования линейного перемещения в эквивалентные электрические сигналы: при работе с LVDT энергия приходит от линейного перемещения через индуктивности без внешнего источника энергии. Пассивные датчики не нуждаются в стимуляции для работы, например, термопара сможет преобразовывать тепло непосредственно в электрические сигналы и без дополнительных источников энергии.

В сети интернета вещей используются как аналоговые, так и цифровые датчики. Аналоговые датчики измеряют аналоговые сигналы, такие как температура, давление и так далее. Данные сигналы должны быть оцифрованы, прежде чем отправиться на микроконтроллер для последующей обработки. Цифровые датчики, в свою очередь, имеют только два состояния (0 и 1) и могут измерять входные сигналы, например, наличие или отсутствие света согласно настроенной чувствительности, выдавая при этом на выход соответствующие значения.

Что такое МЭМС?

МЭМС расшифровывается как «микроэлектромеханические системы». МЭМС-устройства обычно изготавливают на кремниевой подложке с помощью технологии микрообработки, аналогично технологии изготовления однокристалльных интегральных микросхем. Благодаря данной технологии, функционал датчика можно уместить в значительно меньшем пространстве, чем это было бы, если бы при изготовлении использовались дискретные компоненты. Например, в устройство МЭМС, построенное на базе интегральной микросхемы, можно добавить, в дополнение к осям x и y , третью ось для измерения перемещений без изменения размеров конечного устройства. Подобный датчик МЭМС будет реагировать на перемещения, создавая внутри себя разницу электрического потенциала, регистрируемую как изменение емкости датчика.



Рис. 3. Использование МЭМС-датчиков для моделирования функций человеческого организма

МЭМС-датчик объединяет электрические и механические функции в одном корпусе. Электрические элементы выполняют обработку данных, а механические элементы отвечают за реакцию на внешнее воздействие окружающей среды. Это похоже на миниатюрную машину, заключенную внутрь микросхемы с размерами от 1 до 100 мкм и толщиной меньше человеческого волоса. МЭМС-датчики имеют широкое применение (рисунок 3). В качестве примера можно привести датчики срабатывания подушек безопасности автомобиля, способные определять возникновение аварии и инициировать срабатывание подушек.

С целью продвижения и развития МЭМС-технологий был создан международный консорциум [SEMI-MEMS & Sensors \(MSIG\)](#), в который на данный момент входят 120 компаний. Ожидается, что технология МЭМС будет продолжать свое развитие и ее роль в мире датчиков и Интернета вещей будет только возрастать.

Что такое умные датчики?

Сам по себе датчик способен лишь реагировать на изменения определенной физической величины во внешней среде и преобразовывать полученные данные в цифровой формат для отправки на внешний микроконтроллер. Вся «умная» часть датчика находится именно в микроконтроллере, отвечающим за обработку данных с датчика. Объединение микроконтроллера и датчика в единый модуль позволило создать так называемый умный (smart) датчик, который, благодаря современным технологиям производства, может иметь достаточно небольшие размеры (рисунок 4).

Например, микросхема [BNA250](#) производства Bosch Sensortec (рисунок 5) объединяет в себе 32-разрядный микроконтроллер с 14-разрядным датчиком ускорения в корпусе 2,2 x 2,2 x 0,95 мм. Другой пример – решения компании [TE](#), представляющие собой датчики с разъемами для подключения, заключенные в компактный корпус, но при этом обеспечивающие широкий функционал.



Рис. 4. BNA250 от Bosch Sensortec, объединяющий 32-разрядный микроконтроллер с 14-разрядным датчиком ускорения в компактном корпусе 2,2 x 2,2 x 0,95 мм

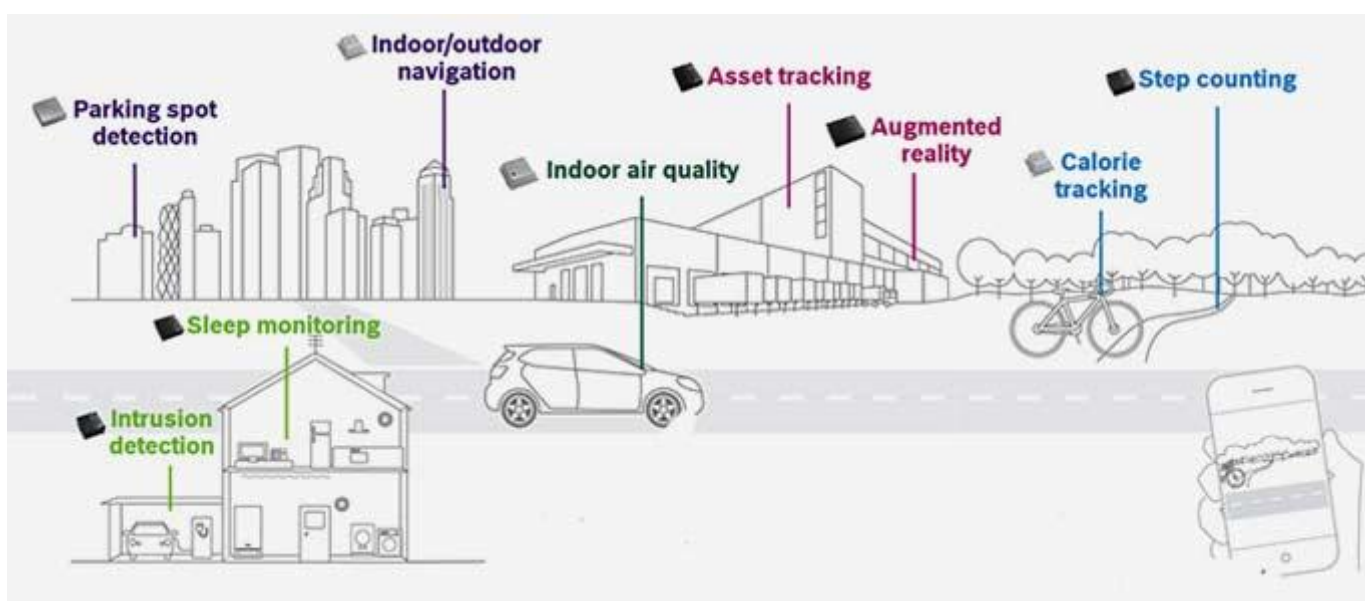


Рис. 5. Пример использования МЭМС-датчиков в приложениях IoT для обнаружения парковочных мест, отслеживания активности, подсчета шагов, мониторинга сна и т.д.

Насколько сложно проектировать системы с датчиками?

Многие производители модулей и микросхем датчиков упрощают процесс разработки системы, предоставляя своим клиентам готовые наборы для создания систем датчиков. Разработчики могут использовать подобные наборы для отладки и разработки, прежде чем создать окончательный вариант системы.

STMicroelectronics недавно представила гибкий модуль подобного типа под названием [SensorTile.box](https://www.st.com/en/development-tools/sensortile-box.html). Модуль включает в себя маломощный микроконтроллер ARM Cortex-M4 с DSP и блоком, производящим операции с плавающей точкой (FPU), плюс несколько датчиков. На борту SensorTile.box расположены следующие датчики:

1. Цифровой датчик температуры
2. IMU-датчик с шестью степенями свободы
3. Трехосевой акселерометр
4. Трехосевой магнитометр
5. Альтиметр/ датчик давления

6. Микрофон/ аудио датчик
7. Датчик влажности

SensorTile.box имеет небольшие размеры (57 x 38 x 20 мм) (рисунки 6 и 7) и Bluetooth на борту. Модуль способен работать «из коробки» без необходимости задания программного кода, что полезно для разработчиков, желающих сразу начать работу с устройством. Однако SensorTile.box также поддерживает программирование и отладку в открытой среде разработки STMicro STM32 (ODM STM32), которая позволяет пользователю самому задать алгоритм работы модуля и поддерживает различные типы библиотек, в том числе библиотеки по работе с искусственным интеллектом и нейронными сетями.

С каждым годом все больше производителей поставляют на рынок готовые решения, подобные SensorTile.box от STMicroelectronics, что позволяет упростить процесс разработки и создания новых систем на основе датчиков.



Рис. 6. Внешний вид модуля SensorTile.box с интегрированным Arm Cortex-M4 и датчиками

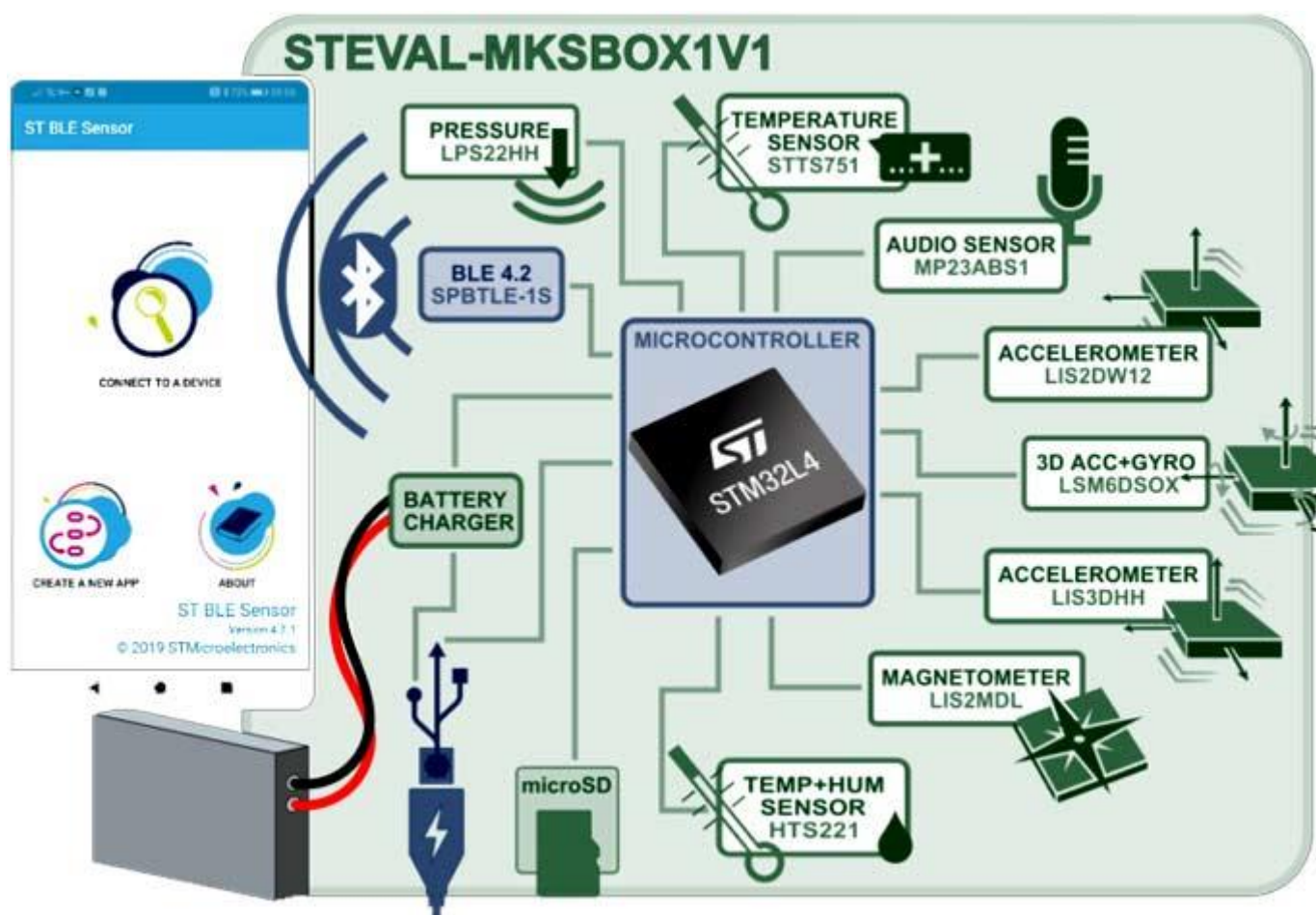


Рис. 7. Блок-схема SensorTile.box, отображающая функционал модуля

Что необходимо учесть при выборе датчика?

Перед выбором датчика следует обратить внимание на следующие критерии:

- Точность измерений с точки зрения погрешности и шага измерения. Заранее определите, какое разрешение (шаг измерений) вам нужно. К примеру, в случае измерения температуры может быть достаточно точности в пределах 1 градуса, с другой стороны, при измерении длины погрешность в 0,01 сантиметра может стать критической. Однако стоит учесть, что чем выше разрешение (чем меньше шаг измерений), тем выше стоимость датчика.
- Диапазон измерений. Необходимо определиться с верхним и нижним пределом измерений. От диапазона измерений также зависит стоимость датчика: чем больше диапазон, тем выше стоимость.
- Условия окружающей среды. Существуют типы датчиков, способные работать при высокой/низкой температуре с различными степенями влагозащиты и так далее.
- Надежность. Надежность датчика условно делится на два аспекта. Первый аспект заключается в том, как долго датчики будут работать без поломок. Параметр MTBF (среднее время безотказной работы) показывает, в течение какого времени датчики могут функционировать без сбоев или поломок. Данный показатель особенно важен для датчиков, работающих в удаленных и труднодоступных местах. Второй аспект заключается в том, как долго датчик может работать с необходимой точностью без дополнительной калибровки. Случаи, когда датчик начинает выдавать неточные данные, нередки, однако в большинстве случаев датчик просто требует калибровки.
- Экстремальные условия. В некоторых случаях датчики должны выдерживать экстремальные температуры, удары и вибрацию.
- Требования к размеру и весу. В зависимости от применения, датчики могут устанавливаться в устройствах с весьма ограниченными размерами, или для одного устройства может понадобиться установить несколько датчиков.
- Выбор между интегрированным решением и решением на дискретных компонентах. Интегрированное решение обычно имеет выигрыш в размерах и может значительно упростить разработку и сократить время проектирования. Например, датчик вибрации со встроенной функцией связи LoRaWAN или RF будет занимать значительно меньше места и времени на установку чем его аналог, выполненный на дискретных компонентах. Однако интегрированное решение также может иметь более высокую стоимость по сравнению с аналогом на дискретных компонентах.
- Если датчики имеют критическую важность для системы, стоит рассмотреть модули с дублированием датчиков.
- Общая стоимость системы. Помимо стоимости датчика, стоит также помнить о затратах на установку, проектирование и поддержку разрабатываемой системы.

Протокол SNA и операционные системы IBM

В устаревших мэйнфреймах IBM обычно используются протоколы стека *Systems Network Architecture, SNA*, который был изначально разработан в 1974 году. Фактически SNA – это набор частных протоколов, в которых в качестве метода доступа используется маркерное кольцо. Многие детали маркерных сетей, созданных компанией IBM, впоследствии были включены в стандарт IEEE 802.5. Однако в сети SNA кабельный участок обязательно строится на базе экранированной витой пары (STP), причем кабели имеют строго ориентированную маркировку (и разводку) (например, определенный конец кабеля должен идти к мэйнфрейму, а другой – к устройствам, подключенным к мэйнфрейму, таким как контроллеры дисковых накопителей или коммуникационных каналов). Это означает, что в сети SNA также используются частные (фирменные) кабельные разъемы и сетевые интерфейсы,

Стек протоколов SNA и эталонная модель OSI

Стек протоколов SNA базируется на семиуровневой модели (табл. 5.5), напоминающей эталонную модель OSI.

Таблица 5.5. Семиуровневая модель SNA

Уровень SNA	Эквивалентный уровень OSI	Назначение
Службы транзакций (Transaction Services)	Прикладной	Самый высокий уровень, управляет службами, от которых зависит работа прикладных программ (например, распределенных баз данных и приложений, выполняющихся одновременно на нескольких мэйнфреймах)
Представитель-ские службы (Presentation Services)	Представитель-ский	Управляет форматированием и преобразованием данных (например, перекодировкой из ASCII в EBCDIC и наоборот), также выполняет сжатие данных (хотя, в отличие от Представительского уровня OSI, этот уровень не обеспечивает шифрование данных)
Управление потоком данных (Data Flow Control)	Сеансовый	Устанавливает и поддерживает коммуникационные каналы между узлами, управляет потоками данных и обеспечивает восстановление после коммуникационных ошибок

Уровень SNA	Эквивалентный уровень OSI	Назначение
Управление (Transmission Control)	Транспортный	Обеспечивает надежность передачи данных передачей от исходного узла к принимающему, а так же управляет шифрованием данных
Управление маршрутом (Path Control)	Сетевой	Управляет маршрутизацией и созданием виртуальных каналов, фрагментирует сообщения на блоки меньших размеров при передаче данных через разнородные сети (эту задачу выполняет Транспортный уровень OSI)
Управление(Data Link Control)	Канальный каналом	Форматирует данные на фреймы, обеспечивает маркерный доступ к сети при одноуровневых обменах данными между компьютерами
Управление Физическим Устройством (Physical Control)	Физический	Обеспечивает генерирование и кодирование электрических сигналов, работу физических интерфейсов, топологию сети и коммуникационную среду (например, кабель)

Достоинства и недостатки SNA

Аналогично любому стеку протоколов, SNA имеет как достоинства, так и недостатки. Отмечая достоинства, следует сказать, что архитектура SNA существует уже более четверти века и обеспечивает надежные и проверенные средства обмена данными с системами IBM. Существенным недостатком является то, что SNA – это частный (фирменный) стек протоколов, для которого нужны специальные устройства и дополнительное обучение процедурам конфигурирования, управления и отладки. По этим причинам сети SNA с мэйнфреймами IBM обычно работают очень хорошо, но это требует больших затрат на обучение персонала и поддержку сети.

Физические элементы сети SNA

В традиционной сети SNA с компьютерами IBM терминалы рассматривав как физические модули типа 2 (type 2). Физический модуль – это некоторое устройство, которое может подключаться к мэйнфрейму или управлять доступом к нему.

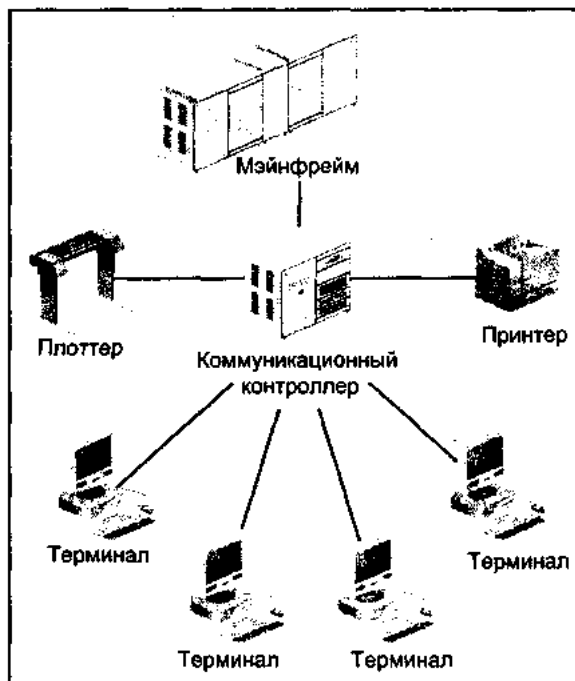


Рис. 5.6. Физические модули, подключенные к мэйнфрейму в сети SNA

К физическим модулям типа 2 относятся как терминалы, так и принтеры. Более сложные физические модули, обменивающиеся данными с мэйнфреймом, относятся к типу 2.1. В их число входят шлюзы (например, устройства, подключающие компьютеры с протоколом TCP/IP к мэйнфрейму IBM, работающему с SNA), мини-компьютеры и устройства, называемые контроллерами кластеров и позволяющие нескольким мэйнфреймам работать в тандеме. Терминалы и принтеры подключаются к физическим модулям типа 4, представляющим собой коммуникационные контроллеры, которые непосредственно соединены с мэйнфреймом и управляют трафиком обращений к нему (рис. 5.6).

Протоколы и приложения, работающие в стеке SNA

В табл. 5.6 перечислено множество протоколов и приложений, входящих в

стек SNA.

Таблица 5.6. Протоколы и приложения, входящие в стек протоколов SNA

Аббревиатура или название	Полное название	Описание	Уровень модели SNA
APPN	Advanced Peer-to-Peer Networking (улучшенный протокол одноранговых сетей)	Обеспечивает одноранговые взаимодействия между устройствами, такими как мэйнфреймы, миникомпьютеры, шлюзы и контроллеры кластеров	Управление передачей
CICS	Customer Information Control System (абонентская информационно управляющая система)	Программная среда, предоставляющая программистам базовые средства взаимодействия с архитектурой SNA (в том числе безопасный доступ, управление файлами и накопителями). Альтернативой IMS является CICS	Управление потоком данных и Представительские службы
DDM	Distributed Data Management	Программы, обеспечивающие удаленный доступ к	Службы транзакций

Аббревиатура или название	Полное название	Описание	Уровень модели SNA
	(управление распределенными данными)	информации, хранящейся на мейнфреймах IBM (например, по удаленному подключению со стороны другого мейнфрейма, находящегося в удалении)	
IMS	Information Management System (информационно - управляющая система)	Программная среда, предоставляющая программистам базовые средства взаимодействия с архитектурой SNA (в том числе безопасный доступ, управление файлами и накопителями). Альтернативой IMS является CICS	Управление потоком данных Представительские службы
NCP	Network Control Program (программа управления сетью)	Обеспечивает адресацию физических устройств и дополнительную логическую адресацию, а также маршрутизацию. Используется для шлюзовых коммуникаций SNA и управления ими (должна устанавливаться на любом шлюзе SNA для того, чтобы рабочие станции могли обращаться через шлюз к мейнфрейму; см. главы 1 и 4, где шлюзы рассматриваются подробнее)	Управление каналом и Управление маршрутом
SDLC	Synchronous Data Link Control (синхронное управление передачей данных)	Создает логические соединения (виртуальные каналы) в сетевом кабеле и координирует передачу данных по этим соединениям обеспечивает в каналах полудуплексную и полнодуплексную связь	Управление физическим устройством и Управление каналом
SNADS	SNA Distributed Services (распределенные службы SNA)	Программные средства, управляющие передачей документов. Используются системами электронной почты для передачи сообщений по указанным адресам	Службы транзакций
SSCP	System Services Control Point (точка управления системными службами)	Программное обеспечение, управляющее VTAM	Управлений передачей
Token Ring	Token Ring	Метод доступа, используемый	Управление

Аббревиатура или название	Полное название	Описание	Уровень модели SNA
		сетях SNA	физическим устройством Управление каналом
VTAM	Virtual Telecommunications Access Method (виртуальный телекоммуникационный метод доступа)	Управляет передачей данных в сети SNA (например, с помощью методов управления потоками). Обеспечивает обмен цифровыми данными	Управление передачей

Протокол DLC для доступа к операционным системам IBM

Если для доступа к мэйнфрейму, работающему с SNA, используются компьютеры под управлением Windows 9x, Windows NT и Windows 2000, то альтернативой SNA-шлюзу является установка протокола *Data Link Control, DLC*. Этот протокол эмулирует SNA, и он может также применяться для подключения к некоторым устаревшим моделям сетевых принтеров, которые могут работать только с ним (например, старые принтеры Hewlett-Packard).

Совет

Протокол DLC не поддерживается в Windows XP. Если вы рассматриваете возможность перехода на эту систему, то учтите, что с ней вы не сможете использовать DLC для доступа к мэйнфреймам IBM и, возможно, вам потребуется SNA-шлюз.

В основном протокол DLC является альтернативой TCP/IP в тех случаях, когда некоторый хост использует SNA-коммуникации. Недостатком этого протокола является то, что он не маршрутизируется. Кроме того, он на самом деле не предназначен для одноранговых взаимодействий между рабочими станциями, а служит только для подключения к старым мэйнфреймам IBM (например, ES9000) или мини-компьютерам IBM (например, AS/400). В практическом задании 5-7 рассказывается о том, как установить DLC в системе Windows 2000.

Протокол DNA для операционных систем компьютеров Digital (Compaq)

Созданная в 1974 году архитектура *Digital Network Architecture (DNA)* имеет такой же возраст, что и SNA. DNA использовалась в первых сетях компании Digital Equipment Corporation (DEC) и по-другому называлась DECnet. Затем этот стек протоколов применялся значительно реже.

Архитектура DNA предусматривает использование фреймов Ethernet II (или DIX – аббревиатура от названий компаний-разработчиков Digital, Intel и Xerox) в шинной топологии. Одним из достоинств DNA является то, что с самого начала эта архитектура близко следовала эталонной модели OSI. Не-Достаток DNA – то, что эта архитектура частная. Кроме того, после приобретения фирмы DEC компанией Compaq оригинальные компьютеры DEC и сети DNA стали менее популярными. Даже некогда известные компьютеры на базе процессора DEC Alpha все чаще заменяются продаваемыми под маркой Compaq рабочими станциями и серверами, реализованными с использованием процессоров Intel Itanium.

Поскольку DNA все реже встречается в сетях, уменьшается вероятность того, что вы столкнетесь с этой архитектурой на практике. Однако для общего представления в табл. 5.7 перечислены некоторые из протоколов и приложений, образующих стек DNA.

Таблица 5.7. Протоколы и приложения, входящие в стек протоколов

Аббревиатура	Полное название	Описание	Уровень модели OSI
CLNS	Connectionless-Mode Network Service (сетевая служба без установления соединения)	Обеспечивает работу служб без установления соединения (<i>см. главу 2</i>), а также маршрутизации	Сетевой
CONS	Connection Oriented Network Service (сетевая служба с установлением соединения)	Обеспечивает работу служб с установлением соединения для маршрутизации и контроля за ошибками маршрутизации	Сетевой
DDCMP	Digital Data Communications Message Protocol	Обеспечивает работу служб с установлением соединения и контролем ошибок. На	Физический Канальный(подуровень LLC)

Аббревиатура	Полное название	Описание	Уровень модели OSI
	(протокол сообщений передачи цифровых данных)	уровне электрических сигналов позволяет осуществлять полудуплексную и полнодуплексную связь	
FTAM	File Transfer, Access, and Management (передача файлов, доступ и управление)	Позволяет передавать файлы с текстовым и двоичным содержимым	Прикладной
HDLC	High-Level Data Link Control (высокоуровневое управление каналом)	Создает логические соединения (виртуальные каналы) в сетевом кабеле и координирует передачу данных между ними. Управляет форматированием фреймов	Физический и Канальный
MAILbus	MAILbus	Соответствует стандарту X.400 на почтовые службы	Прикладной
Naming Service	Naming Service (служба имен)	Предоставляет сетевым устройствам службы именования, преобразующие адрес устройства в его имя и наоборот (что упрощает пользователям работу с устройствами)	Прикладной
NVTS	Network Virtual Terminal (служба сетевых виртуальных терминалов)	Транслирует символы между Service терминалами, сетями DNA и хост-компьютерами	Представительский и Прикладной

Повышение производительности локальных сетей

Проще всего повысить производительность сети, если уменьшить количество протоколов, передаваемых через каждый маршрутизатор. При этом уменьшается рабочая нагрузка на маршрутизаторы, что позволяет им быстрее обрабатывать сетевой трафик. При меньшем количестве протоколов уменьшается и ненужный трафик, создаваемый в сети.

Вопросы для обсуждения

При выборе протоколов, используемых в сети, рассмотрите следующие вопросы.

- Должны ли пакеты маршрутизироваться?
- Какого размера сеть – маленькая (менее 100 узлов), средняя (100 – 500 узлов) или крупная (свыше 500 узлов)?
- Какие серверы используются и какие протоколы для них нужны?
- Имеются ли мэйнфреймы и какие протоколы для них требуются?
- Имеется ли непосредственный выход в Интернет или подключение к интранет-приложениям, использующим веб-технологии (виртуальная частная сеть)?
- Какая скорость требуется для подключений к глобальной сети?
- Имеются ли ответственные приложения?

Если фреймы нужно маршрутизировать (например, в корпоративной сети), то лучше всего применять протокол TCP/IP, поскольку он ориентирован на маршрутизацию и распространен во многих сетях. Для маленьких и средних немаршрутизируемых сетей (менее 200 узлов) на базе серверов Windows NT и при условии отсутствия подключения к Интернету наилучшим выбором остается протокол NetBEUI, обеспечивающий быстрые и надежные коммуникации. В сетях NetWare (с серверами версий ниже 5.0) можно использовать IPX/SPX, хотя в смешанной сети, где имеются старые серверы NetWare " новые серверы Windows 2000, могут понадобиться протоколы IPX/SPX и TCP/IP. Протокол NWLink является хорошим средством для подключения систем Windows 9x/NT/2000 к старым серверам NetWare.

Проблема каналов связи

Наличие подключения к Интернету или веб-службам требует развертывания Протокола TCP/IP, при этом службы FTP могут использоваться для передачи файлов. Также протокол TCP/IP лучше всего применять для связи с со временными мэйнфреймами и компьютерами UNIX, поскольку для подключения к мэйнфрейму или к приложению, работающему на компьютере UNIX, может потребоваться эмуляция терминала по протоколу Telnet. Для подключения к мэйнфреймам IBM и мини-компьютерам (если они работают в среде SNA) можно также использовать протокол DLC. И, наконец, протокол DNA по-прежнему может понадобиться в сети, где имеются старые компьютеры DEC (например, DEC VAX).

Примечание

TCP/IP – наилучший протокол для средних и крупных сетей. Он может маршрутизироваться, обладает надежностью для ответственных приложений, имеет надежный механизм контроля ошибок. В таких сетях важно иметь средства мониторинга сети и анализа неисправностей. Как изложено в главе 6, стек TCP/IP имеет протоколы, необходимые для решения подобных задач.

Во многих случаях для разных сетевых приложений нужно использовать различные протоколы локальных сетей. Иногда в современных сетях в любых сочетаниях применяются протоколы TCP/IP, NetBEUI, IPX/SPX, SM и даже DNA. Как вы уже знаете, развернутые протоколы связаны с типом используемых операционных систем. Также на их выбор влияет наличие связи с глобальными сетями (например, для выхода в Интернет нужен протокол TCP/IP, который может также потребоваться для связи локальных сетей между собой через глобальную сеть). Если, скажем, TCP/IP используется серверами в одной локальной сети, а рабочие станции из другой сети должны обращаться к этим серверам, то обе локальные сети и связывающая их глобальная сеть должны обеспечивать передачу протокола TCP/IP.

Удаление ненужных протоколов

Иногда рабочие станции в сети остаются настроенными на использование нескольких протоколов даже после того как все хосты и серверы переведены на протокол TCP/IP. В этом случае легко можно повысить производительность сети, удалив с рабочих станций ненужные протоколы. В практическом задании 5-8 рассказывается, как удалить протокол DLC из системы Windows 2000, а в задании 5-9 вы узнаете, как удалить службу Client Service for NetWare (и протокол NWLink IPX/SPX) из систем Windows 2000 и Windows XP Professional.

Резюме

- В значительной степени архитектуру сетей определяют протоколы. Во многих сетях используется несколько протоколов, с помощью которых осуществляется доступ к различным операционным системам сетевых серверов и хост-компьютеров.
- Обычно применяемые протоколы локальных сетей определяются типом сетевой серверной операционной системы, используемой в конкретной сети. Одной из старейших сетевых систем является NetWare, работающая со стеком протоколов IPX/SPX и обеспечивающая передачу данных между старыми версиями серверов NetWare и рабочими станциями (а также и другими серверами), подключенными к серверам. Протокол IPX/SPX реализован в тысячах локальных сетей, поскольку NetWare является одной из распространенных сетевых операционных систем. Однако в настоящее время благодаря тому, что многие сети связаны с Интернетом, новые версии NetWare (5.0 и выше) ориентированы на работу с более универсальным стеком протоколов TCP/IP.
- Родным протоколом для систем Windows NT Server является NetBEUI, появление которого связано с разработкой сетевой операционной системы LAN Manager, которую компания Microsoft начинала совместно с фирмой IBM. В средних и крупных сетях с серверами Windows NT чаще используется стек TCP/IP. С появлением систем Windows 2000 и Windows Server 2003 протокол TCP/IP пришел на замену NetBEUI, что определяется требованиями службы Active Directory и необходимостью доступа к Интернету.
- AppleTalk – это протокол, используемый компьютерами Macintosh с операционными системами Mac OS и Mac OS Server. Системы Windows NT, Windows 2000, Windows Server 2003 и Novell NetWare также поддерживают AppleTalk.
- Некоторые сетевые серверные операционные системы (в частности, UNIX) изначально были ориентированы на работу со стеком TCP/IP (а также и с Интернетом). В других сетевых операционных системах (например, NetWare, Windows NT и Mac OS Server) стек TCP/IP был реализован уже после создания этих систем.
- В первых системах IBM использовался стек протоколов SNA, который обеспечивал обмен данными между мэйнфреймами (мини-компьютерами) и терминалами, контроллерами и принтерами, а также между различными компьютерами. В операционных системах Windows имеется возможность установки протокола DLC для эмуляции коммуникаций SNA.
- Стек протоколов DNA был разработан для использования в сетях на базе компьютеров DEC, однако в настоящее время он применяется редко, поскольку количество таких компьютеров в сетях значительно уменьшилось.
- Простым и эффективным способом повышения производительности локальной сети является периодический проводимый анализ применяемых протоколов и удаление тех протоколов, которые больше не используются. Для доступа к компьютерам и принтерам.
- Вплоть до начала 1990-х годов сетевые технологии в первую очередь развивались в области протоколов локальных сетей. В настоящее время архитектура этих протоколов нашла логическое завершение в стеке TCP/IP, а частные протоколы (такие как IPX/SPX и NetBEUI) используются реже.

Наименование работы: Создание идеальной сети в аудитории

1. Цель работы: научиться создавать сеть в аудитории

2. Литература:

2.1. <http://www.overclockers.ru/articles/lan/index.shtml?page1#11>

3. Подготовка к работе:

Изучение материала по указанной литературе.

4. Перечень средств обучения:

4.1. IBM PC.

5. Задание:

5.1. Ознакомиться с видами сетевых кабелей, сетевых карт и коммутаторами;

5.2. Описать выбранные варианты;

5.3. Соединить 2 компьютера в сеть.

6. Содержание отчета:

1. Название и цель работы.
2. Порядок выполнения работы.
3. Ответы на контрольные вопросы.
4. Сделать выводы о проделанной работе.

7. Контрольные вопросы:

7.1. Зачем нужна Локальная Вычислительная Сеть (ЛВС)?

7.2. Что нужно, чтобы проложить локальную сеть?

7.3. Настройка Windows XP для работы в локальной сети.

7.4. Настройка общего доступа к сетевым ресурсам.

Приложение

<http://www.overclockers.ru/articles/lan/index.shtml?page1#11>

Локальная вычислительная сеть

Лока́льная вычисли́тельная сеть (ЛВС, локальная сеть; (англ. Local Area Network, LAN) — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт). Также существуют локальные сети, узлы которых разнесены географически на расстояние более 12 500 км (космические станции и орбитальные центры). Несмотря на такое расстояние, подобные сети относят к локальным.

Существует множество способов классификации сетей. Основным критерием классификации принято считать способ администрирования. То есть в зависимости от того, как организована сеть и как она управляется, её можно отнести к локальной, распределённой, городской или глобальной сети. Управляет сетью или её сегментом сетевой администратор. В случае сложных сетей их права и обязанности строго распределены, ведётся документация и журналирование действий команды администраторов.

Компьютеры могут соединяться между собой, используя различные среды доступа: медные проводники (витая пара), оптические проводники (оптоволоконные кабели) и через радиоканал (беспроводные технологии). Проводные связи устанавливаются через Ethernet, беспроводные — через Wi-Fi, Bluetooth, GPRS и прочих средств. Отдельная локальная вычислительная сеть может иметь шлюзы с другими локальными сетями, а также быть частью глобальной вычислительной сети (например, Интернет) или иметь подключение к ней.

Чаще всего локальные сети построены на технологиях Ethernet или Wi-Fi. Следует отметить, что ранее использовались протоколы Frame Relay, Token ring, которые на сегодняшний день встречаются всё реже, их можно увидеть лишь в специализированных лабораториях, учебных заведениях и службах. Для построения простой локальной сети используются маршрутизаторы, коммутаторы, точки беспроводного доступа, беспроводные маршрутизаторы, модемы и сетевые адаптеры. Реже используются преобразователи (конвертеры) среды, усилители сигнала (повторители разного рода) и специальные антенны.

Маршрутизация в локальных сетях используется примитивная, если она вообще необходима. Чаще всего это статическая либо динамическая маршрутизация (основанная на протоколе RIP).

Иногда, в локальной сети организуются рабочие группы — формальное объединение нескольких компьютеров в группу с единым названием.

Сетевой администратор — человек, ответственный за работу локальной сети или её части. В его обязанности входит обеспечение и контроль физической связи, настройка активного оборудования, настройка общего доступа и предопределённого круга программ, обеспечивающих стабильную работу сети.

Адресация

В локальных сетях основанных на протоколе IP могут использоваться специальные адреса, назначенные IANA (стандарты RFC 1918 и RFC 1597):

10.0.0.0—10.255.255.255;

172.16.0.0—172.31.255.255;

192.168.0.0—192.168.255.255.

Такие адреса называют локальными или серыми, эти адреса не маршрутизируются в Интернет. Необходимость использовать такие адреса возникла из-за того, что когда разрабатывался протокол IP не предусматривалось столь широкого его распространения, и постепенно адресов стало не хватать. Как вариант был придуман протокол IPv6. Однако, он пока не стал популярным и стали использовать локальные адреса. В различных пересекающихся LAN адреса могут повторяться, и это не является проблемой, так как доступ в другие сети происходит с применением технологий, подменяющих или скрывающих адрес внутреннего узла сети за её пределами — NAT или проху дают возможность подключить ЛВС к глобальной сети (WAN). Для обеспечения связи локальных сетей с глобальными применяются маршрутизаторы (в роли шлюзов и файрволов).

Конфликт адресов — это распространённая ситуация в локальной сети, при которой в одной IP подсети оказываются два или более компьютеров с одинаковыми IP адресами. Для предотвращения таких ситуаций, а также для облегчения работы сетевых администраторов применяется протокол DHCP с помощью которого можно автоматически назначать адреса компьютерам.

LAN и VPN

Связь с удалённой локальной сетью, подключенной к глобальной сети, из дома/командировки/удалённого офиса часто реализуется через VPN. При этом устанавливается VPN-подключение к пограничному маршрутизатору.

Особенно популярен следующий способ организации удалённого доступа к локальной сети:

Обеспечивается подключение снаружи к маршрутизатору по протоколам PPPoE, PPTP или L2TP (PPTP+IPSec).

Так как в этих протоколах используется PPP, то существует возможность назначить абоненту IP-адрес. Назначается свободный (не занятый) IP-адрес из локальной сети.

Маршрутизатор (VPN, Dial-in сервер) добавляет прохуаг запись на локальной сетевой карте для IP-адреса который он выдал VPN-клиенту. После этого, если локальные компьютеры попытаются обратиться напрямую к выданному адресу, то они после ARP-запроса получают MAC-адрес локальной сетевой карты сервера и трафик пойдет на сервер а потом и в VPN-туннель.

Зачем нужна Локальная Вычислительная Сеть (ЛВС)?

Это типичный вопрос тех, кому вы предложите вступить в свою сеть, да и для себя стоит определиться, зачем тратить своё время, силы и деньги на подобный проект.

Какие же возможности открывает ЛВС?

1. Обмен информацией между членами сети: документами, фильмами, играми, музыкой, программами и т.д. Скорость современных сетевых адаптеров позволяет совершенно свободно смотреть фильмы и слушать музыку с удалённого компьютера, даже не переписывая их себе на жесткий диск, а если это вдруг потребуется, то гигабайт данных можно переписать всего за 5-15 минут, в зависимости от скорости сети. Фактически, все компьютеры вашей сети объединяются в единую систему, что позволяет на порядок повысить интенсивность обмена информацией, к тому же очень удобно иметь постоянное соединение при совместной работе над каким-либо проектом (разработка web-сайтов, видеороликов, программного обеспечения и т.д.). 2. Возможность совместно использовать такое оборудование как принтеры, CD-RW/DVD/DVD-RW. Единственное ограничение - согласие владельца устройства на его использование, кроме того, вам все равно придется зайти за отпечатанными листами/дисками. Если все компьютеры находятся в пределах одного помещения, то все намного проще.

3. Совместное использование и оплата канала доступа в интернет.

Когда канал доступа в интернет достаточно широк - выделенная линия или ADSL, то даже при одновременном доступе большого количества пользователей ощутимого падения скорости не произойдет. При этом ваш телефон будет совершенно свободен, а для тех, у кого нет телефона, это вообще практически единственная возможность получить доступ в Сеть. Даже если у вас нет ресурсов на покупку выделенного канала, можно совместно использовать модемы.

4. Совершенно новый уровень общения (Голосовая связь, видео и чат)

Это скорее не достоинство, а само собой вытекающее следствие из прокладки сети, ведь членам сети нужно как-то общаться между собой, а для этого есть множество программ. Использование, микрофонов и дешевых Web-камер позволяет легко осуществлять видеосвязь друг с другом.

5. Мультиплатформность С помощью ЛВС можно объединять компьютеры разных видов (Например: PC и Macintosh) и с любыми операционными системами поддерживающими протокол TCP/IP. (Разные версии Windows, Linux, и т.д.).

6. Распределенные вычисления

Если вы работаете в 3DMax, Maya или любой другой подобной программе, вы сможете использовать сетевой рендеринг (когда над сценой работают все машины, объединенные в сеть) и во много раз ускорить скорость обработки кадров, особенно это важно при работе с рендеренными видеороликами высокого качества. Работа, ранее занимавшая несколько дней, будет выполняться за часы. Пожалуйста, обратитесь к документации вашего 3D программного пакета за дополнительными деталями относительно настройки программного обеспечения для сетевого рендеринга.

7. Сетевые службы

Создание системы терминалов, установка Windows по локальной сети, удаленное администрирование систем и многое другое.

8. Многопользовательские игры по локальной сети.

Практически все современные игры поддерживают тот или иной режим мультиплеер. Играть с другими людьми здорово, это даёт совершенно иные впечатления, появляется азарт, совершенно не присущий одиночной игре. Локальная сеть идеально подходит для разворачивания мультиплеер-баталлий. Скорость связи несоизмеримо выше модемной и даже DSL, телефон всегда свободен, участвовать могут все члены сети, игра бесплатна. К тому же, если ваша ЛВС подключена к высокоскоростному каналу доступа в Интернет, никто не мешает вам поискать противников и во всемирной паутине, или даже сразиться целой командой, например с пользователями другой подобной сети.

Что нужно, чтобы проложить локальную сеть?

В первую очередь желание, причем не только ваше, но и всех тех, кто будет состоять в будущей сети. На самом деле, самое сложное в создании домашней сети вовсе не прокладка проводов и настройка оборудования. Самое сложное - убедить людей в том, что это здорово, и нужно всем. Чаще всего проекты прокладки домашних сетей упираются в банальную лень и непонимание. Многие просто до конца не осознают, что может дать им сеть,

другие не хотят ничего сверлить, третьим не нравится сетевой кабель, который, как им кажется, портит весь интерьер. Если будущие члены сети - дети, то им могут запрещать родители. Кроме того, к вам совершенно не обязательно будут относиться с симпатией, равно как и к вашему проекту. Хорошо, если всех соседей вы давно знаете. А если нет? Вряд ли, когда вы начнете звонить подряд во все квартиры и спрашивать "Есть ли у вас компьютер?", "Не хотите ли вступить в локальную сеть?", вас будет ждать особо теплый прием. Вообще, наилучший способ найти тех, с кем стоит тянуть сеть - это сначала поговорить с теми, кого вы знаете, а потом достать телефонный справочник и выяснить телефоны жильцов вашего и соседнего дома, а затем методично обзвонить их всех, предлагая вступить в вашу будущую сеть. Лучше всего говорить уверенным официальным тоном. Ещё может помочь печать объявлений о создании локальной сети и расклейка их по подъездам домов кандидатов. Все гораздо проще, если вы строите офисную сеть или сеть для личного пользования, например, объединяете компьютеры в пределах одной квартиры.

Созданию сети, особенно крупной, помогает согласование проведения работ с эксплуатирующими организациями (такими как ЖЭК, ДЭЗ и т.п.) или владельцами здания, впрочем, это не всегда возможно, данные организации предпочитают работать с компаниями, а не с частными лицами. Так же это не всегда финансово оправдано, если вы строите небольшую сеть на несколько квартир.

Но, предположим, что все организационные моменты вы уже уладили. Вне зависимости от того, нужно ли вам объединить десятки компьютерных систем в нескольких зданиях или же просто соединить ноутбук и настольный компьютер, Вам понадобится:

- Сетевой кабель – то, что собственно и связывает компьютеры между собой.
- Сетевые карты - по одной для каждого участника сети.
- Switch (Коммутатор) – если в сети участвует больше 2 компьютеров, то это устройство объединяет всю систему, к нему сходятся все сетевые кабели.

Перспективы использования беспроводных ZigBee-интерфейсов в электроприводе

Внедрение беспроводных технологий является весьма перспективным направлением развития промышленных сетей. В статье сформулированы требования к беспроводным сетям в промышленности, отражены преимущества и особенности технологии ZigBee. Рассмотрены методы построения аппаратно-программных средств и решения по применению технологии ZigBee в конечных устройствах промышленного назначения.

Введение

В современном мире уже никого не удивит устройство беспроводной передачи данных. В нашу жизнь давно проникли технологии мобильной связи и интернета, системы глобальной навигации и радиоиентификации. Однако лишь относительно недавно беспроводные технологии связи начали применяться в сферах коммунального хозяйства и промышленности. Причем, если в сфере коммунального хозяйства уже множество задач коммуникации решается с помощью беспроводных технологий (сбор показаний счетчиков воды/электроэнергии, управление освещением, сбор информации с разнообразных датчиков, построение системы «умный дом»), то в сфере промышленности аналогичные работы только начинаются. При этом эффективность от внедрения таких систем в промышленности очень высока, так как именно здесь насчитывается множество объектов автоматизации различной сложности, связь между которыми удобно осуществлять через беспроводные каналы. А в случае расположения части системы на движущихся объектах, как это может быть в автоматизированных складах, применение беспроводных сетей оказывается единственно возможным решением. До последнего времени внедрение этих технологий в промышленность сдерживалось из-за проблем, связанных с надежностью каналов связи в жестких условиях эксплуатации при большом уровне промышленных помех, а также с

защитой беспроводных промышленных сетей от несанкционированного доступа. Сейчас ситуация кардинально меняется, и из области «экзотики» беспроводные промышленные сети переходят в область целесообразных технических решений. Это обстоятельство не могут игнорировать и производители исполнительных устройств, в частности, комплектных электроприводов, которые должны проектироваться с учетом наличия встроенного беспроводного интерфейса.

Стандарты беспроводной связи

Среди наиболее известных беспроводных технологий можно выделить: Wi-Fi, Wi-Max, Bluetooth, Wireless USB и относительно новую технологию — ZigBee, которая изначально разрабатывалась с ориентацией на промышленные применения (см. рис. 1).

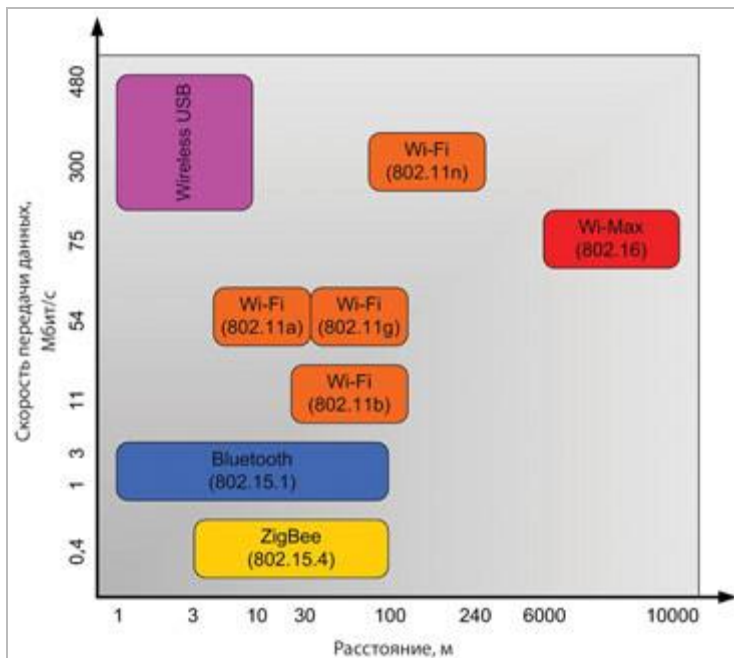


Рис. 1. Стандарты беспроводной связи

Каждая из этих технологий имеет свои уникальные характеристики, которые определяют соответствующие области применения.

Попытаемся сформулировать требования, которым должна удовлетворять технология связи для ее успешного применения в промышленности. Допустим, имеется некий промышленный объект (см. рис. 2), состоящий из нескольких насосных электроприводов, устройства сбора информации с различных технологических датчиков, например, датчиков давления, температуры, расхода, в том числе установленных удаленно, операторского пульта и диспетчерского пункта. Управление насосами производится с операторского пульта, а в диспетчерском пункте производится непрерывный мониторинг системы.



Рис. 2. Пример структуры промышленного объекта

Очевидно, что оптимальным вариантом с точки зрения простоты и удобства было бы объединение всех устройств, участвующих в обмене информацией, в единую информационную сеть, работающую в одном стандарте. Поскольку на промышленном объекте могут быть установлены устройства различной сложности и, соответственно, стоимости, то программно-аппаратный комплекс, обеспечивающий доступ каждого устройства в информационную сеть, должен быть достаточно дешевым. Также технология связи должна обеспечивать необходимую дальность и скорость соединений. А если принять во внимание то, что промышленная установка может быть дополнена новыми узлами (например, еще одним

насосом или устройством сбора информации), то от технологии связи требуется возможность масштабирования. Ну и, конечно же, коммуникационная технология должна обеспечивать надежность и безопасность передачи информации. Рассмотренный случай является типичным примером распределенной системы управления, где каждый из узлов, являясь интеллектуальным, выполняет свою локальную задачу автоматизации, а связи между узлами являются «слабыми» — в основном по сети передаются команды оперативного управления и смены уставок регулируемых переменных, сообщения о состоянии оборудования и технологического процесса. Каждый узел, например, на базе преобразователя частоты имеет собственные каналы связи с технологическими датчиками, и необходимость в передаче больших потоков данных отсутствует.

Анализ беспроводных технологий [1], [2], [3] показывает, что высокоскоростные технологии Wi-Fi, Wi-Max, Bluetooth, Wireless USB предназначены в первую очередь для обслуживания компьютерной периферии и устройств мультимедиа. Они оптимизированы для передачи больших объемов информации на высоких скоростях, работают в основном по топологии «точка-точка» или «звезда» и мало пригодны для реализации сложных разветвленных промышленных сетей с большим количеством узлов. Напротив, технология ZigBee имеет достаточно скромные показатели скорости передачи данных и расстояния между узлами, но обладает следующими важными, с точки зрения применения в промышленности, преимуществами.

1. Она ориентирована на преимущественное использование в системах распределенного мульти-микропроцессорного управления со сбором информации с интеллектуальных датчиков, где вопросы минимизации энергопотребления и процессорных ресурсов являются определяющими.
2. Предоставляет возможность организации самоконфигурируемых сетей со сложной топологией, в которых маршрут сообщения автоматически определяется не только числом исправных или включенных/выключенных на текущий момент устройств (узлов), но и качеством связи между ними, которое автоматически определяется на аппаратном уровне.
3. Обеспечивает масштабируемость — автоматический ввод в работу узла или группы узлов сразу после подачи питания на узел.
4. Гарантирует высокую надежность сети за счет выбора альтернативного маршрута передачи сообщений при отключениях/сбоях в отдельных узлах.
5. Поддерживает встроенные аппаратные механизмы шифрации сообщений AES-128, исключая возможность несанкционированного доступа в сеть.

Стандарт беспроводной связи ZigBee

ZigBee — относительно новый стандарт беспроводной связи, который изначально разрабатывался как средство для передачи небольших объемов информации на малые расстояния с минимальным энергопотреблением. Фактически этот стандарт описывает правила работы программно-аппаратного комплекса, реализующего беспроводное взаимодействие устройств друг с другом. Модель реализации стандарта связи ZigBee представлена на рис. 3.

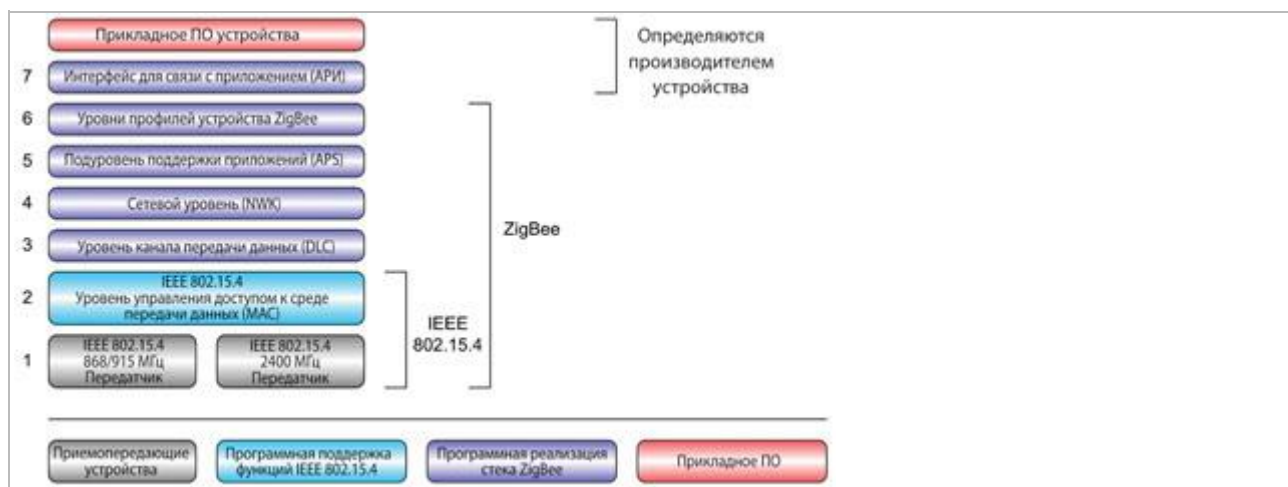


Рис. 3. Многоуровневая модель стандарта связи ZigBee

Стандарт ZigBee описывает все уровни, через которые проходит поток передаваемой информации, начиная с физического уровня и заканчивая уровнем поддержки профилей устройств. Нижние два уровня описываются стандартом IEEE 802.15.4 [4] и определяют физические параметры приемопередатчика, структуру радиочастотной посылки, число адресуемых устройств, механизмы проверки и подтверждения целостности принятых данных, процедуры оценки качества канала связи и алгоритмы предотвращения коллизий. Приемопередающие устройства, предназначенные для работы в стандарте IEEE 802.15.4, обладают следующими основными характеристиками.

– Трансивер работает на частоте 2,4 ГГц по 16-ти каналам с шагом 5 МГц между каналами (разрешено для использования во всех странах мира). Дополнительно для Европы выпускаются трансиверы на 868 МГц, работающие по 1-му каналу, для Америки — трансиверы на 915 МГц, работающие по 10-ти каналам с шагом 2 МГц. Далее рассматриваются только устройства, работающие на частоте 2,4 ГГц.

– Максимальная скорость передачи данных по радиоканалу составляет 250 Кбит/с.

– Максимальное расстояние связи на открытой территории должно быть не меньше 100 м.

Уровни с третьего по шестой описываются непосредственно спецификацией стека ZigBee [5]. На этих уровнях определяется, какими свойствами должны обладать устройства, входящие в сеть, каким образом пакет информации передается от одного узла сети к другому, как обеспечивается безопасность передачи информации, как новое устройство подключается к сети и ее топология, какой узел в сети является главным, какой — подчиненным.

В спецификации стека предусмотрены три типа устройств: координатор, маршрутизатор и оконечное устройство. Координатор инициализирует сеть, управляет ее узлами, хранит информацию о настройках каждого узла, задает номер частотного канала и идентификатор сети PAN ID, а в процессе работы может являться источником, приемником и ретранслятором сообщений. Маршрутизатор отвечает за выбор пути доставки сообщения, передаваемого по сети от одного узла к другому, и в процессе работы также может являться источником, приемником или ретранслятором сообщений. Оконечное устройство не участвует в управлении сетью и ретрансляции сообщений, являясь только источником/приемником сообщений.

Среди свойств ZigBee следует особо выделить поддержку сложных топологий сетей (см. рис. 4). Именно за счет этого, при относительно малой максимальной дальности связи двух близлежащих устройств, возможно расширить зону покрытия сети в целом. Также этому способствует 16-битная адресация, позволяющая объединять в одну сеть более 65 тыс. устройств.

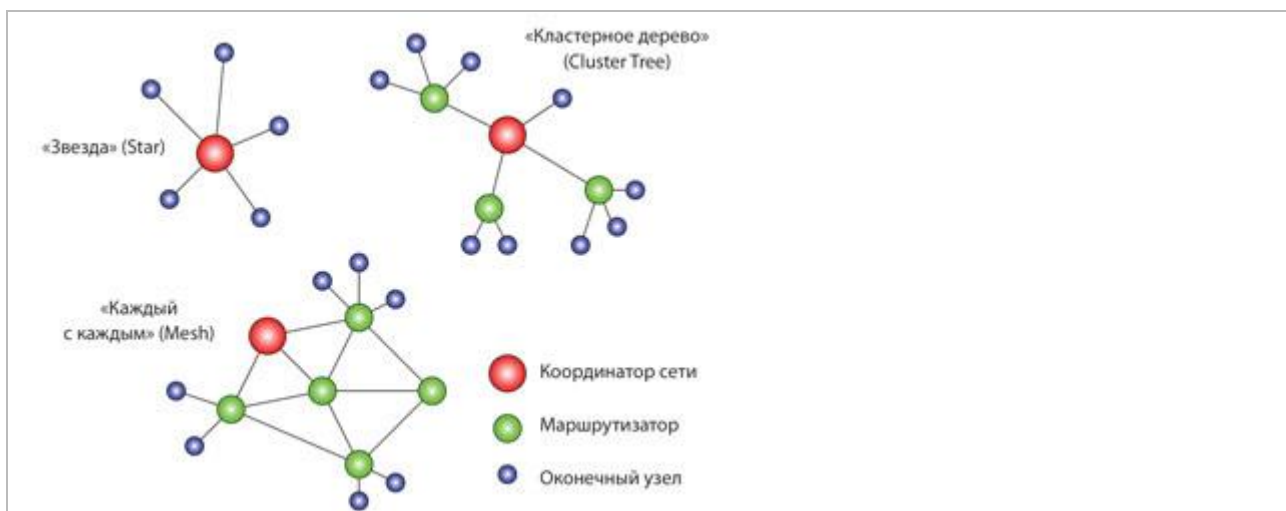


Рис. 4. Топологии сетей ZigBee

Другой отличительной чертой стандарта является возможность самовосстановления дееспособности сети в случае выхода из строя отдельных ее узлов. Это свойство основано на том, что каждый узел следит за своими соседями, постоянно обновляя маршрутные таблицы на основе оценки мощности принятых от них сигналов. В результате, при изменении пространственного расположения соседей или выходе из сети одного из устройств, вычисляется новый маршрут следования сообщения. Это преимущество является исключительно важным в сетях, функционирующих на промышленных объектах в жестких условиях эксплуатации при наличии промышленных помех, а также в тех случаях, когда часть узлов находится на движущихся устройствах, например электрокарах.

Микросхемы и модули для обеспечения связи по стандарту ZigBee

Сегодня на рынке предлагаются два основных вида решений по применению технологии ZigBee в конечных устройствах.

В первом случае разработчику конечного устройства предлагается самостоятельно сконструировать аппаратную часть, используя различные микросхемы с поддержкой радиоинтерфейса, работающего по стандарту IEEE 802.15.4. При этом разработчику, как правило, доступна документация, описывающая правила конструирования и примеры проектирования аппаратной части. Разработчик может полностью и самостоятельно создать программную часть (используя спецификацию протокола) либо сделать это с применением стандартных сетевых протоколов, разработанных и отлаженных производителями микросхем. Эти протоколы могут иметь разный уровень совместимости со стандартом и поставляться в виде библиотек или наборов разработчика программного обеспечения (SDK).

В настоящее время выпускаются 5 основных типов микросхем, выполняющих функции приема/передачи данных по радиоканалу в стандарте IEEE 802.15.4.

1. «Классический» радиотрансивер в стандарте IEEE 802.15.4. Выполняет только функции приема и передачи информации по радиоканалу и не содержит программного обеспечения, реализующего стек протокола ZigBee. Программная реализация стека ZigBee возлагается на внешний микроконтроллер, к которому микросхема трансивера подключается по одному из стандартных интерфейсов, например по синхронному периферийному интерфейсу (SPI) или параллельному интерфейсу. Дополнительно для работы трансивера требуется обвязка, включающая в себя антенну, кварцевый резонатор, а также небольшое число пассивных компонентов.

2. Интегрально гибридные микросхемы (SiP), объединяющие в одном корпусе кристаллы «классического» радиотрансивера и микроконтроллера общего назначения. Данное решение позволяет разместить программный стек ZigBee непосредственно во встроенной памяти микроконтроллера и тем самым существенно снизить нагрузку на внешний микроконтроллер, обеспечивающий решение задач конкретного приложения. Эти микросхемы требуют минимальной обвязки, включая антенну.

3. Системы на кристалле (SoC). Данный вид микросхем фактически повторяет возможности SiP, но в отличие от них и трансивер, и встроенный микроконтроллер выполнены на одном кристалле. Некоторые фирмы, например, Texas Instruments называют такие кристаллы ZigBee-процессорами. Они взаимодействуют с внешним контроллером приложения по одному из стандартных интерфейсов (SPI, UART) и допускают загрузку в память сетевых протоколов разной сложности в зависимости от предполагаемой сложности и структуры реализуемой сети.

4. ZigBee-акселераторы. Микросхемы аналогичные SoC, но с уже «зашитым» во внутренний микроконтроллер стеком ZigBee протокола. Данный вид микросхем является идеальным выбором для тех разработчиков, которые предпочитают работать с привычным для него микроконтроллером и хотят добавить ZigBee-функционал без изучения сложного набора

протоколов ZigBee-стека. Небольшая внешняя обвязка требуется и в этом случае.

5. Платформы в едином корпусе (PiP). Наиболее функциональное решение, подразумевающее объединение высокопроизводительного микроконтроллера с широким набором встроенных периферийных модулей (таймеров, АЦП и т.д.) и трансивера IEEE 802.15.4 на одном кристалле. В этом случае вычислительных возможностей встроенного микроконтроллера и набора встроенной периферии достаточно не только для реализации стека ZigBee, но и для поддержки конкретного приложения. Дополнительный процессор не требуется.

Следует обратить внимание на то, что ZigBee-микросхемы от разных производителей обладают различными вспомогательными периферийными модулями, которые позволяют увеличить их быстродействие и функциональность.

Таковыми модулями, например, могут быть: модуль аппаратной поддержки функций уровня управления доступом к среде передачи данных (MAC) или модуль аппаратной поддержки функций шифрования передаваемых данных (AES-128).

Во втором случае разработчику конечного устройства предлагается использовать готовые ZigBee-модули, не требующие ни дополнительной доработки, ни обвязки. Такие модули представляют собой размещенные на одной плате IEEE 802.15.4-трансивер, микроконтроллер, содержащий в себе стек ZigBee-протокола, необходимые внешние элементы обвязки (антенну, схемы питания, тактирования и др). Они выпускаются небольшими компаниями, специализирующимися на этой продукции. Использование готовых модулей наиболее целесообразно на начальной стадии проектирования, когда необходимо быстро оценить принципиальную возможность решения задачи.

Поясним описанные выше возможности на примере комплектного частотно-регулируемого электропривода.

Преобразователь частоты (ПЧ) состоит из силовой и управляющей части. Управляющая часть преобразователя построена на специализированном DSP-микроконтроллере, выполняющем функции управления силовой частью ПЧ и двигателем (см. рис. 5). Для интерактивной настройки, мониторинга состояния привода и управления в реальном времени преобразователь необходимо снабдить беспроводным сетевым интерфейсом ZigBee.

Варианты построения управляющей части преобразователя частоты на базе ZigBee-микросхем представлены на рис. 6.



Рис. 5. Структура современного преобразователя частоты

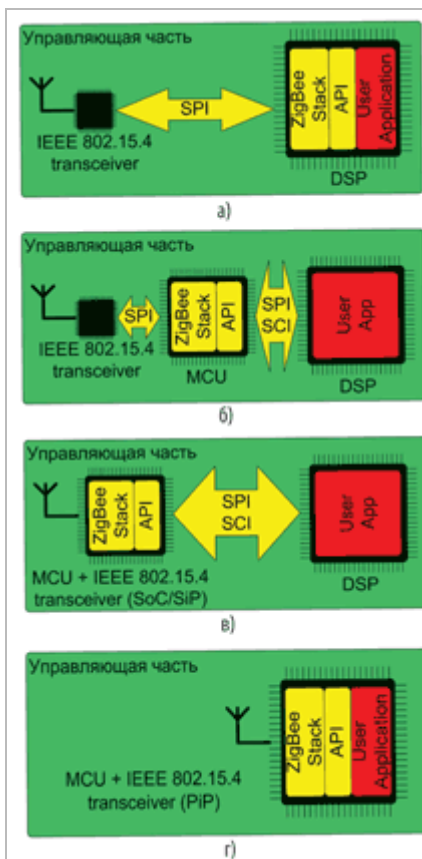


Рис. 6. Структура управляющей части ПЧ с различными вариантами реализации ZigBee-интерфейса

Схема на рис. 6а предполагает, что ресурсов мощного DSP-микроконтроллера достаточно не только для реализации основных функций приложения, но и стека ZigBee. В схему контроллера ПЧ добавляется микросхема радио-трансивера IEEE 802.15.4, имеющая интерфейс SPI для сопряжения с базовым процессором. Вся нагрузка по реализации ZigBee-стека лежит на разработчиках комплектного привода.

Как правило, в современных системах векторного управления приводами центральный процессор значительно загружен прикладными задачами. В этом случае для реализации беспроводного интерфейса больше подходят решения, представленные на рис. 6б или 6в. Стек ZigBee реализуется на дополнительном микроконтроллере, что позволяет существенно снизить нагрузку на основной DSP-контроллер. Решение на рис. 6в основано на использовании системы на кристалле или ZigBee-процессора. Применительно к силовой электронике и электроприводу это наиболее правильное решение, если речь идет о создании серийной продукции, а не опытно-промышленных образцов.

Наконец, в случае если к основному DSP-контроллеру не предъявляются особых требований по быстродействию, то управляющую часть ПЧ можно реализовать по схеме рис. 6г. В этом случае система управления реализована на одной микросхеме. Пока такое решение может быть рекомендовано только для относительно простых систем привода, например со скалярным управлением двигателями, а также для источников питания и регуляторов напряжения.

При использовании готовых ZigBee-модулей серьезной модификации управляющей части преобразователя вообще не требуется (см. рис. 7). При этом ZigBee-стек реализуется внутри модуля. Важным является лишь наличие у модуля нужного интерфейса для сопряжения с целевым микроконтроллером. Это решение позволяет существенно уменьшить время разработки конечного устройства с поддержкой беспроводной сети.

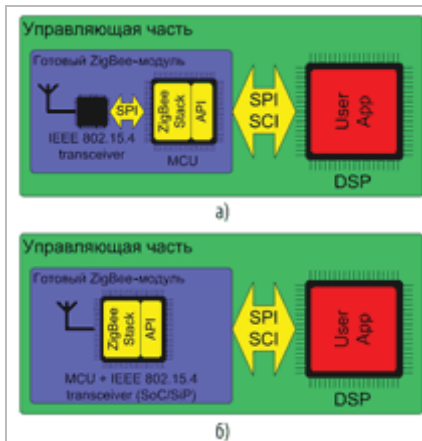


Рис. 7. Структура управляющей части ПЧ с готовыми ZigBee-модулями

Окончательный выбор элементной базы для конкретной разработки остается за прикладными специалистами. Для облегчения этой задачи мы приводим краткий обзор решений в этой области от наиболее крупных производителей, среди которых по степени интеграции и комплектности решений выделяются Freescale Semiconductor и Texas Instruments (см. табл. 1).

Табл. 1. Решения в области ZigBee-технологии от различных производителей

Производитель	Аппаратные решения	Программные решения
Freescale Semiconductor [6]	Радиотрансиверы IEEE 802.15.4: — MC13192 — MC13193 — MC13202 — MC13203 Гибридные интегральные микросхемы (SiP): — MC13212 — MC13213 — MC13214 Платформы в едином корпусе (PiP): — MC1322x	BeeKit — программный инструмент для сборки и конфигурирования программных модулей реализующих коммуникационную технологию ZigBee
Texas Instruments [7]	Радиотрансиверы IEEE 802.15.4: — CC2420 — CC2520 Системы на кристалле (SoC): — CC2430 — CC2431 ZigBee акселераторы: — CC2480A1	Простой сетевой протокол SimplicityTI для сетей с небольшим числом узлов до 256. Более сложный набор ПО TIMAC, поддерживающий топологии «точка-точка» и «звезда». Z-stack набор программных модулей для реализации полной технологии ZigBee. Сетевые протоколы поддерживают все аппаратные платформы вплоть до систем на кристалле
Ember	Системы на кристалле (SoC): — EM250 ZigBee-акселераторы: — EM260	ZigBee Development Tools — набор программных инструментов для создания и конфигурирования программных модулей для реализации технологии ZigBee
Digi International	Готовые ZigBee-модули: — XB24-X — XBP24-X	Программное обеспечение для модулей, позволяющее создавать ZigBee-сети на их базе (ZB и ZNet firmware).
STMicroelectronics	Готовые ZigBee-модули: — SPZB250 — SPZB260	Предлагается использовать программные решения от Ember, поскольку модули основаны на микросхемах этой компании
Panasonic	Готовые ZigBee-модули: — PAN802154HAR00 — PAN4555 — PAN4570	Предлагается использовать программные решения от Freescale Semiconductor и Ember, поскольку модули основаны на микросхемах этих компаний

Компания Freescale Semiconductor одной из первых начала предлагать свои решения в области ZigBee-коммуникаций и на сегодняшний день выпускает широкий ряд различных устройств и необходимые программные пакеты, предназначенные для организации связи согласно этому стандарту. Из ее ассортимента можно выделить «классические» IEEE 802.15.4-трансиверы, гибридные интегральные микросхемы (SiP) и платформы PiP. Основные характеристики этих микросхем приведены в табл. 2, 3 и 4.

Табл. 2. Основные характеристики трансиверов компании Freescale Semiconductor

Параметр	MC13192/MC13193/MC13202/MC13203
Напряжение питания	2...3,4 В
Рабочий частотный диапазон	2,405... 2,485 ГГц
Число каналов/шаг	16/5 МГц
Максимальная скорость радиопередачи информации	250 Кбит/с
Выходная мощность радиопередатчика	3,6 <дБм (настраивается программно)

Чувствительность радиоприемника	-92 дБм
Наличие встроенного Rx/Tx переключателя	MC13192/MC13193 — Нет, MC13202/MC13203 — Есть
Возможность подключения внешнего усилителя	Есть
Поддерживаемые сетевые топологии	«Точка-точка», «звезда», «кластерное дерево», «каждый с каждым»
Интерфейс связи с ведущим микроконтроллером	4-проводной SPI
Рабочий температурный диапазон	-40...85°C
Особенности	— 4 канала событий внутреннего таймера. — Возможность тактирования внешнего микроконтроллера. — Адаптирован под применение ZigBee стека Figure 8 Wireless Z-Stack*. * Только для трансиверов MC13193 и MC13203

Табл. 3. Основные характеристики SiP от компании Freescale Semiconductor

Параметр	MC13212/MC13213/MC13214
Интегрированный трансивер	MC1320x
Интегрированный микроконтроллер	MC9S08GTx
Частота ядра	40 МГц
Flash память	32/60/60 Кбайт
РАМ память	2/4/4 Кбайт
Напряжение питания	1,8...3,6 В
АЦП	10-бит, 8 каналов
Последовательные интерфейсы	SPI, 2SCI (UART), I ² C
Рабочий температурный диапазон	-40...85°C
Особенности	MC13214 полностью адаптирован под применение ZigBee стека Figure 8 Wireless Z-Stack

Табл. 4. Основные характеристики PiP от компании Freescale Semiconductor

Параметр	MC1322X
Рабочий частотный диапазон	2,405... 2,485 ГГц
Число каналов/шаг	16/5 МГц
Максимальная скорость радиопередачи информации	250 Кбит/с
Выходная мощность радиопередатчика	-30.. 4 дБм (настраивается программно)
Чувствительность радиоприемника	-95 дБм
Наличие встроенного переключателя Rx/Tx	Есть
Возможность подключения внешнего усилителя	Есть

Поддерживаемые сетевые топологии	«Точка-точка», «звезда», «кластерное дерево», «каждый с каждым»
Ядро микроконтроллера	ARM7
Частота ядра	26 МГц
Flash-память	До 128 Кбайт
RAM-память	96 Кбайт
ROM-память	80 Кбайт
Напряжение питания	2,0...3,6 В
Рабочий температурный диапазон	-40...105°C
Особенности	— 6 активных режимов работы, 3 энергосберегающих. — Наличие аппаратного MAC-акселератора. — Аппаратный кодировщик AES-128. Расширенный набор периферии микроконтроллера. — Требуется малое количество внешних компонентов

Помимо аппаратных решений компания Freescale Semiconductor предлагает необходимые программные инструменты для создания и конфигурирования программных модулей, реализующих коммуникационную технологию ZigBee на основе описанной выше аппаратной части.

Компания Texas Instruments также предлагает широкий ряд продукции, предназначенной для решения задач коммуникации устройств по стандарту ZigBee. В ее ассортименте присутствуют «классические» IEEE 802.15.4-трансиверы и «системы на кристалле» (SoC), а также так называемые «ZigBee-акселераторы», которые фактически являются системой на кристалле, но с уже интегрированным обработчиком ZigBee-стека. Основные характеристики решений от Texas Instruments представлены в табл. 5, 6 и 7.

Табл. 5. Основные характеристики трансиверов компании Texas Instruments

Параметр	CC2420	CC2520
Напряжение питания	2,1...3,6 В	1,8...3,8 В
Рабочий частотный диапазон	2,405...2,485 ГГц	2,405... 2,485 ГГц
Число каналов/шаг	16/5 МГц	16/5 МГц
Максимальная скорость радиопередачи информации	250 Кбит/с	250 Кбит/с
Выходная мощность радиопередатчика	-25...0 дБм (настраивается программно)	-20...5дБм
Чувствительность радиоприемника	-95 дБм	-98 дБм
Наличие встроенного переключателя Rx/Tx	Есть	Есть
Возможность подключения внешнего усилителя	Есть	Нет
Поддерживаемые сетевые топологии	«Точка-точка», «звезда», «кластерное дерево», «каждый с каждым»	«Точка-точка», «звезда», «кластерное де «каждый с каждым»
Интерфейс связи с ведущим микроконтроллером	4-проводной SPI	4-проводной SPI
Рабочий температурный диапазон	-40...85°C	-40...125°C
Особенности	— Наличие аппаратного MAC-акселератора. — Аппаратный кодировщик	— Возможность тактирования внешнего микроконтроллера. — Наличие аппаратн MAC-акселератора. — Аппаратный

	AES-128	кодирущик AES-128

Табл. 6. Основные характеристики SoC от компании Texas Instruments

Параметр	CC2430/CC2431
Интегрированный трансивер	CC2420
Ядро интегрированного микроконтроллера	8051
Частота ядра	32 МГц
Flash-память	До 128 Кбайт
РАМ-память	До 8 Кбайт
Напряжение питания	2,0...3,6 В
АЦП	12-бит, 8 каналов
Последовательные интерфейсы	2SCI (UART)
Рабочий температурный диапазон	-40...85°C
Особенности	CC2431 содержит модуль определения положения узла

Табл. 7. Основные характеристики ZigBee-акселераторов от компании Texas Instruments

Параметр	CC2480A1
Напряжение питания	2,0...3,6 В
Рабочий частотный диапазон	2,405...2,485 ГГц
Число каналов/шаг	16/5 МГц
Максимальная скорость радиопередачи информации	250 Кбит/с
Выходная мощность радиопередатчика	До 0 дБм (настраивается программно)
Чувствительность радиоприемника	-92 дБм
Наличие встроенного переключателя Rx/Tx	Нет
Возможность подключения внешнего усилителя	Есть
Поддерживаемые сетевые топологии	«Точка-точка», «звезда», «кластерное дерево», «каждому каждому»
Интерфейс связи с ведущим микроконтроллером	4-проводной SPI/UART
Рабочий температурный диапазон	-40...85°C

Важно обратить внимание, что ZigBee-акселераторы предназначены лишь для выполнения стека протокола, а пользовательское приложение должно выполняться на внешнем микроконтроллере. Для удобства внедрения ZigBee-технологии в конечные устройства большинство фирм производителей соответствующих решений, предлагает различные вспомогательные и отладочные средства и комплекты.

Первый опыт разработки преобразователей частоты и регуляторов напряжения с ZigBee-интерфейсом Отечественными фирмами ООО НПФ «Вектор», ООО «Нанко» и ООО «Конвир» ведутся совместные исследования по созданию преобразовательной техники и систем комплексной автоматизации и диспетчеризации с использованием беспроводного ZigBee-интерфейса с ориентацией на энергосберегающие технологии в коммунальном и сельском хозяйстве, а также в «умных домах».

Преобразователи частоты (ПЧ) (рис. 8а) и многоканальные регуляторы напряжения (МРН) (рис. 8б) имеют высокопроизводительную систему управления на базе DSP-микроконтроллера TMS320F2810 с интегрированным готовым ZigBee-модулем XBP24-X от компании Digi International (рис. 9). Для обеспечения качественного приема использована всенаправленная внешняя антенна с возможностью наклона оси до 90° и поворота до 180°. Интерфейс ZigBee предназначен для беспроводного конфигурирования, настройки параметров и оперативного управления сетью исполнительных устройств на базе ПЧ и МРН от персонального компьютера с помощью демонстрационной интерфейсной платы XBIB-U-DEV.



а) б)

Рис. 8. Преобразователь частоты ПЧ (а) и многоканальный регулятор напряжения МРН (б)



Рис. 9. Управляющий контроллер с интегрированным ZigBee-модулем

На данный момент достигнуты относительно невысокие скорости передачи информации (10...15 Кбит/с) при полномасштабной реализации протокола ZigBee, количестве узлов в сети не более 10 и расстояниях между узлами до 300 м (в прямой видимости). С учетом того, что критичные по скорости выполнения процессы управления оборудованием выполняются в реальном времени на встроенных высокопроизводительных DSP-микроконтроллерах, а по сети передается относительно небольшое число команд оперативного управления и смены уставок, достигнутые показатели по скорости являются приемлемыми.

Несомненным достоинством беспроводной технологии является, в частности, возможность изготовления удобных пультов для обслуживания сложной преобразовательной техники, которые не требуют подключения к сети и объекту и автоматически обеспечивают гальваническую развязку оборудования. Исследования в этом направлении проводятся в настоящее время.

Литература

1. Wi-Fi alliance//www.wi-fi.org
2. Bluetooth special interests group (SIG)//www.bluetooth.org
3. ZigBee alliance//www.zigbee.org
4. IEEE Computer Society. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Std 802.15.4™-2006//standards.ieee.org/getieee802/802.15.html
5. ZigBee Alliance. ZigBee Specification. Q4/2007//www.zigbee.org/en/spec_download/zigbee_downloads.asp
6. Официальный сайт компании Freescale Semiconductor//www.freescale.com
7. Официальный сайт компании Texas Instruments//www.ti.com
8. Официальный сайт компании «Конвир»//www.konvir.ru

ПРАКТИЧЕСКАЯ работа. Протокол NETBIOS

Цель работы: Получить основные теоретические сведения о протоколе NETBIOS.

Теоретическая справка.

В NetWare протокол NETBIOS является надстройкой над протоколом IPX и используется для организации обмена данными между рабочими станциями.

Протокол NETBIOS реализован в виде резидентной программы NETBIOS.EXE, входящей в комплект поставки NetWare. Для обмена данными между этими резидентными программами используются пакеты IPX с номером гнезда 0x0455 и типом пакета 20.

Для идентификации рабочей станции протоколы IPX и SPX используют номер сети, адрес станции в сети и номер гнезда. Адрес станции определяется на аппаратном уровне и представляет собой число длиной 6 байтов.

Номер сети занимает 4 байта. Номер гнезда выделяется динамически протоколом IPX или может быть получен в фирме Novell. Номер гнезда занимает 2 байта /1/.

Протокол NETBIOS использует другой механизм адресации станций и программ. Для адресации станций используются имена размером 16 байтов. Каждая станция имеет одно постоянное имя (permanent name), которое образуется из аппаратного адреса добавлением к нему с лева десяти нулевых байтов. Кроме постоянного имени имеются обычные имена и групповые имена, которые протокол NETBIOS позволяет добавлять (и удалять). Обычные имена служат для идентификации рабочей станции, групповые имена могут служить для посылки пакетов одновременно нескольким станциям в сети. Постоянное имя удалять нельзя, так как оно полностью определяется аппаратным обеспечением станции.

При добавлении обычного имени протокол NETBIOS опрашивает всю сеть для проверки уникальности имени. Групповое имя может быть одинаковым для нескольких станций, поэтому при добавлении группового имени опрос сети не выполняется /1/.

После добавления новому имени присваивается так называемый номер имени (name number), который используется для передачи данных по сети.

Сравнивая методы адресации, используемые протоколами IPX/SPX и NETBIOS, можно заметить, что метод адресации протокола NETBIOS более удобен. Вы можете адресовать данные не только одной станции (как в IPX и SPX) или всем станциям сразу (как в IPX), но и группе станций, имеющим одинаковое групповое имя /1/.

Чтобы выполнить функцию NETBIOS, в прикладной программе необходимо:

- заполнить поля блока NCB (Network Control Block);
- загрузить в регистр ES:BX дальний адрес блока NCB;
- вызвать программное прерывание 5C (INT 5C).

Назначение полей блока NCB.

Поле *Cmd* содержит код команды, которую необходимо выполнить.

Поле *CCode* содержит код ошибки, возвращаемый после проверки параметров до выполнения команды.

Поле *LocalSessionNumber* содержит номер канала, установленного с другой программой. Оно используется только при выдаче команд передачи данных через каналы.

Поле *NetworkNameNumber* содержит номер имени, который присваивается при добавлении обычного или группового имени. Это поле должно быть заполнено при приёме датаграмм.

Поле *Buffer* представляет собой дальний указатель на буфер, который должен содержать данные перед выполнением передачи, или на буфер, который будет использован для приёма данных.

Поле *Size* определяет размер буфера, используемого для приёма или передачи данных.

В поле *CallName* указывается имя станции-получателя.

Поле *OurName* содержит имя станции-отправителя. Обычно используется в командах создания имени станции или создания канала.

Поля *ReceiveTimeout* и *SendTimeout* содержат интервал времени (измеряемый в 1/2с), в течение которого ожидается завершение соответственно команд приёма и передачи.

Поле *PostRoutine* - ноль или указатель на программу (POST-программу), которая получает управление после завершения команды.

Поле *AdapterNumber* используется, если на рабочей станции установлено несколько сетевых адаптеров (в сетях Ethernet этого обычно не бывает). В этом поле указывается номер адаптера, для которого предназначена команда. Первый адаптер имеет номер 0, второй -1 и т. д.

Поле *FinalCCode* содержит во время выполнения команды значение 0xFF. После завершения выполнения команды в это поле записывается ноль или код ошибки, который относится к выполнению команды в целом (в отличие от кода в поле *CCode*).

Поле *Reserved* зарезервировано для использования протоколом NETBIOS.

Перед выполнением команды её код должен быть записан в поле *Cmd* блока NCB. Каждая команда NETBIOS реализована в двух вариантах: с ожиданием и без ожидания окончания выполнения команды /4/.

Все команды NETBIOS можно разделить на несколько групп:

1. Для работы с именами:

- 0x30, 0xB0 - добавить новое имя в таблицу имён станции (с ожиданием и без ожидания);
- 0x36, 0xB6 - добавить новое групповое имя в таблицу станции;

- 0x31, 0xB1 - удалить имя из таблицы имён станции.

2. Для приёма и передачи датаграмм:

- 0x20, 0xA0 - передать одной или группе станций блок данных в виде датаграммы;
- 0x22, 0xA2 - передать всем станциям блок данных в виде датаграммы;
- 0x21, 0xA1 - принять блок данных, переданный с помощью команды 0x20 или 0xA0;
- 0x23, 0xA3 - принять блок данных, переданный с помощью команды 0x22 или 0xA2.

3. Для работы с каналами (для приёма и передачи данных):

- 0x10, 0x90 - установить канал между двумя именами, заданными в блоке NCB;
- 0x11, 0x91 - организовать канал с вызываемой стороны (работают в паре с командами 0x10 или 0x90);
- 0x12, 0x92 - закрыть канал;
- 0x34, 0xB4 - опросить состояние канала;
- 0x14, 0x94 - передать блок данных (до 64 Кб) по каналу;
- 0x71, 0xF1 - передать блок данных (до 64 Кб) по каналу без проверки доставки блока;
- 0x17, 0x97 - передать два буфера (каждый по 64 Кб) по каналу как один блок;
- 0x72, 0xF2 - передать два буфера (каждый по 64 Кб) по каналу как один блок без проверки доставки блока;
- 0x15, 0x95 - принять блок данных, переданный по каналу;
- 0x16, 0x96 - принять блок данных, переданный по любому каналу, который организовала принимающая станция.

5. Другие команды:

- 0x32 - удалить все имеющиеся каналы и имена;
- 0x35 - отменить ранее запущенную команду.

Выполнение работы:

- изучить назначение протокола NETBIOS;
- изучить основную структуру пакета NETBIOS;
- изучить команды NETBIOS;
- изучить описание полей блока NCB (Network Control Block);
- оформить отчет.

Отчет по лабораторной работе должен содержать следующее:

- название и цель работы;
- основные теоретические пункты общих сведений;
- выводы по выполненной работе;
- список использованных источников.

Контрольные вопросы:

- назначение протокола NETBIOS?;
- как устроен механизм адресации в протоколе NETBIOS?;
- что делает прикладная программа для того, чтобы выполнить функцию NETBIOS?;
- какие поля содержит блок NCB (Network Control Block)?;
- как реализована каждая команда NETBIOS?;
- на какие группы делятся команды NETBIOS?

Стек NetBIOS/SMB

Этот стек широко используется в продуктах компаний IBM и Microsoft. На физическом и канальном уровнях этого стека используются все наиболее распространенные протоколы Ethernet, Token Ring, FDDI и другие. На верхних уровнях работают протоколы NetBEUI и SMB.

Протокол NetBIOS (Network Basic Input/Output System) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. В дальнейшем этот протокол был заменен так называемым протоколом расширенного пользовательского интерфейса NetBEUI — NetBIOS Extended User Interface. Для обеспечения совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. Протокол NetBEUI разрабатывался как эффективный протокол, потребляющий немного ресурсов и предназначенный для сетей, насчитывающих не более 200 рабочих станций. Этот протокол содержит много полезных сетевых функций, которые можно отнести к сетевому, транспортному и сеансовому уровням модели OSI, однако с его помощью невозможна маршрутизация пакетов. Это ограничивает применение протокола NetBEUI локальными сетями, не разделенными на подсети, и делает невозможным

его использование в составных сетях. Некоторые ограничения NetBEUI снимаются реализацией этого протокола NBF (NetBEUI Frame), которая включена в операционную систему Microsoft Windows NT.

Протокол 8MB (Server Message Block) выполняет функции сеансового, представительного и прикладного уровней. На основе 8MB реализуется файловая служба, а также службы печати и передачи сообщений между приложениями.

Модель OSI	IBM/Microsoft	TCP/IP	Novell	Стек OSI
Прикладной	SMB	Telnet, FTP, SNMP, SMTP, WWW	NCP, SAP	X.400 X.500 FTAM
Представительный				Представительный протокол OSI
Сеансовый	NetBIOS	TCP	SPX	Сеансовый протокол OSI
Транспортный				Транспортный протокол OSI
Сетевой	IP, RIP, OSPF		IPX, RIP, NLSP	ES-ES IS-IS
Канальный	802.3 (Ethernet), 802.5 (Token Ring), FDDI, Fast Ethernet, SLIP, 100VG-AnyLAN, X.25, ATM, LAP-B, LAP-D, PPP			
Физический	Коаксиал, экранированная и неэкранированная витая пара, оптоволокно, радиоволны			

Рисунок 4.1 - Соответствие популярных стеков протоколов модели OSI

Стеки протоколов SNA фирмы IBM, DECnet корпорации Digital Equipment и AppleTalk/AFP фирмы Apple применяются в основном в операционных системах и сетевом оборудовании этих фирм.

На (рис.4.1) показано соответствие некоторых, наиболее популярных протоколов уровням модели OSI. Часто это соответствие весьма условно, так как модель OSI — только руководство к действию, причем достаточно общее, а конкретные протоколы разрабатывались для решения специфических задач, причем многие из них появились до разработки модели OSI. В большинстве случаев разработчики стеков отдавали предпочтение скорости работы сети в ущерб модульности — ни один стек, кроме стека OSI, не разбит на семь уровней. Чаще всего в стеке явно, выделяются 3-4 уровня: уровень сетевых адаптеров, в котором реализуются протоколы физического и канального уровней, сетевой уровень, транспортный

уровень и уровень служб, вбирающий в себя функции сеансового, представительного и прикладного уровней.

Контрольные вопросы:

1. Что является в компьютерных сетях идеологической основой стандартизации?
2. Дайте развернутое определение протоколу?
3. Для чего используют стек коммуникационных протоколов?
4. Для чего используют открытые системы?
5. Перечислите наиболее популярными являются стеки?

Практическая работа. Протокол IPX/SPX

Цель работы: Получить основные теоретические сведения о протоколе IPX/SPX

Теоретическая справка.

Протокол IPX NetWare является реализацией протокола обмена межсетевыми пакетами фирмы Xerox. Целью IPX является предоставление возможности прикладным системам, работающим в рабочих станциях NetWare, использовать драйверы NetWare для установления прямой связи с другими рабочими станциями, серверами или другими устройствами в объединенной сети (интерсети). IPX является сервисом, который обеспечивает прикладным системам возможность передавать и принимать пакеты информации по интерсети. Структура пакетов определена сетевым стандартом Xerox (XNS). В сетевом оборудовании NetWare (сеть включает одну или более сетей, использующих NetWare) каждый узел (устройство, подключенное к сети) имеет уникальный интерсетевой адрес. Используя IPX, рабочая станция NetWare может передать/принять пакет к/от любого узла интерсети. Маршрутизация пакетов, источник и приемник которых расположены в физически разных локальных сетях, является в значительной степени автоматической и прозрачной. Эта прозрачность достигается средствами IPX в совокупности с сервисом, предоставляемым мостами NetWare.

Сетевые драйверы реализуют физическую доставку пакетов, но не гарантируют их доставку. Реализация надежной доставки, протоколов передачи данных и других протоколов взаимодействия может быть построена на основе протоколов IPX, в соответствии с требованиями различных прикладных программ /1/.

Протокол IPX предназначен для использования как основа для построения сложных прикладных систем, таких как коммуникационные серверы, концентраторы. IPX полностью поддерживается на всех топологиях локальных сетей, которые поддерживаются NetWare V2.0.

Асинхронный планировщик событий (AES) является вспомогательным сервисом, который обеспечивает средства для измерения прошедшего времени и/или запуска события по окончании отмеряемых временных интервалов.

Событиями являются: завершение требования передачи IPX, прием пакета при завершении требования просмотра IPX, окончание временного интервала, определенного прикладной программой. Каждому событию может соответствовать сервисная программа события, которая может вызываться, когда событие произойдет. При необходимости, сервисная программа события может перепланировать себя для выполнения по истечении определенного интервала времени.

Структура пакетов IPX полностью соответствует структуре пакетов протокола XNS Xerox.

Основная структура пакетов IPX.

Пакет IPX содержит 30 байтов заголовка, за которыми следуют от 0 до 546 байтов данных. Таким образом, минимальная длина пакета будет 30 байтов и максимальная длина пакета будет равна 576 байтов.

Содержимое и структура части пакета, в которой содержатся данные, находится полностью в ведении прикладных программ, использующих IPX, и может быть любой требуемой структуры /5/.

Таблица 1 - Структура заголовков пакетов IPX

Началь-ный байт	Поле	Размер	Тип данных
	Контрольная сумма	2 байта	Старшее-младшее целое
	Длина	2 байта	Старшее-младшее целое
	Транспортный контроль	1 байт	Целое без знака
	Тип пакета	1 байт	Целое без знака
	Сеть назначения	4 байта	Старшее-младшее целое без знака
	Узел назначения	6 байт	Старшее-младшее целое без знака
	Сокет назначения	2 байта	Старшее-младшее целое без знака

	Сокет источника	4 байта	Старшее-младшее целое без знака
	Узел источника	6 байт	Старшее-младшее целое без знака
	Сокет источника	2 байта	Старшее-младшее целое без знака

Численные поля, состоящие больше чем из одного байта, могут размещаться двумя способами: старшее-младшее и младшее-старшее. При размещении старшее-младшее самый старший значащий байт размещается в первом байте поля, самый младший – в последнем байте поля. При размещении младший-старший байты хранятся в противоположном порядке.

Описание полей заголовка.

Контрольная сумма. Это поле содержит контрольную сумму 16-ти битовых слов заголовка пакета. Это поле будет содержать -0 (FFFFh), если контрольная сумма не требуется. Если контрольная сумма равна -0, тогда это поле должно содержать +0 (0000h).

Это поле содержит контрольную сумму только 30-ти байтового заголовка, если прикладной программе необходимо иметь контрольную сумму тогда она должна обеспечить контрольную сумму своих данных и разместить их в области данных.

В данной реализации оболочки NetWare контрольная сумма может как проверяться так и нет. Если проверка контрольной суммы необходима, она должна обеспечиваться прикладной программой, которой приходит пакет /1/.

Длина. Это поле содержит полную длину сетевого пакета, которую составляет длина заголовка плюс длина поля данных. Таким образом, минимальное значение этого поля равно 30, максимальное значение равно 576.

Транспортный контроль. Это поле используется межсетевыми мостами NetWare и должно быть установлено в 0 перед передачей пакета.

Тип пакета. Это поле определяет тип услуг, которые предлагаются или требуются пакету. Xerox определяет следующие типы пакетов:

0 - неопределенный тип пакета;

1 - пакет информации маршрутизации;

2 - эхо пакет;

3 - пакет ошибок;

4 - пакет обмена;

5 - пакет последовательного протокола;

16-31 - экспериментальные протоколы.

Все пакеты пользователей должны быть либо типа 0, либо типа 4.

Сеть назначения. Это поле содержит номер сети, в которой может быть расположен узел, которому предназначается данный пакет.

В NetWare сетям, объединяемым в интерсеть администратором сети назначаются 4-х байтовые адреса. Каждая сеть в интерсети имеет уникальный адрес.

Если это поле устанавливается в 0, предполагается, что узел-приемник расположен в той же физической сети, что и узел источник; пакет в этом случае будет передаваться без обращения к межсетевому мосту NetWare.

Узел назначения. Это поле содержит 6-ти байтовый номер, который идентифицирует физический адрес (в сети назначения) узла, которому предназначается пакет.

Не все сети используют одинаковое адресное пространство. Так в сети EtherNet используются все 6 байтов под адрес, в то время как в сети OmniNet используется для адресации только один байт.

Если в данной физической сети для адресации требуется менее 6 байтов, то необходимая часть адреса должна занимать последние байты поля, первые байты поля должны быть установлены в 0.

Установка всех 6 байтов этого поля в FFh показывает что данный пакет является широковещательным для сети назначения.

Поддерживается или нет широковещательный пакет зависит от физических характеристик той сети, в которую пакет доставляется.

Сокет назначения. Это поле содержит адрес сокета системной программы, которой предназначается пакет.

Номера сокетов используются для маршрутизации пакетов к различным программам в данном узле /1/.

Хегох резервирует следующие сокеты для специальных целей:

1 - пакет информации маршрутизации;

2 - пакет эхо-протокола;

3 - пакет обработки ошибок;

32-63 - экспериментальные.

Сокеты с номерами меньшими 3.000 (десятичное значение) считаются статически назначаемыми сокетами, сокеты с номерами большими 3.000 считаются динамически назначаемыми сокетами.

Сеть источник. Это поле содержит номер сети, в которой находится узел -источник пакета. В этом поле может быть установлено значение 0. Если пакет передается через межсетевой мост NetWare, нулевое значение в этом поле будет заменено на реальный номер сети-источника. Только пакеты, передаваемые в рамках одной физической сети могут содержать нулевое значение в этом поле.

Узел источник. Это поле содержит физический адрес узла, из которого пакеты передаются. Количество байт этого поля, используемых для адресации данного узла зависит от физической сети, в которую этот узел входит. Это поле формируется аналогично полю узел назначения, описанному выше.

Сокет источника. Это поле содержит адрес сокета, используемый прикладной программой, которая выдает пакет. Хотя это не требуется протоколами IPX, обычно для всех станций, взаимодействующих по отдельной задаче на равных правах, для передачи и приема используются одинаковые номера сокетов. В ситуации клиент/сервер, узел, работающий как сервер (возможно как коммуникационный шлюз), должен просматривать определенный сокет для обслуживания требований на прием. В этом случае номер сокета источника не обязательно будет таким же.

События и блоки управления событиями.

Блоки управления событиями (ЕСВ) - это структура данных, которая содержит информацию, требуемую для координации планирования и/или активизации определенных операций. Почти все функции IPX или AES работают, используя ЕСВ и/или информацию, которую они содержат. Существует два различных типа событий, которые могут произойти: события, относящиеся к операциям приема или передачи IPX и специальные события, определяемые прикладными процессами.

Например, когда прикладной процесс хочет передать пакет IPX, он должен подготовить ЕСВ, в котором указывает, где содержаться данные для пакета. Завершенный процесс передачи пакета составляет "событие передачи IPX".

Аналогично, когда прикладному процессу требуется принимать пакеты, он должен создать один или более ЕСВ, в которых будет указано, куда размещать принимаемые данные. Прием пакета вызовет "событие приема IPX".

Кроме того, прикладной процесс может определить и спланировать "специальное" событие для своих целей. Блок управления событием должен обеспечиваться планировщиком события в течении определенного интервала времени. AES будет уменьшать время в данном интервале, когда таймер достигнет нуля, произойдет планируемое событие.

Существует два способа, которыми прикладная система может определить и задействовать события: опрашивая флаг "в использовании" ("in use") или обеспечивая сервисную программу события (ESR), которая может быть вызвана в режиме прерывания /1/.

Каждый ЕСВ имеет флаг статуса "in use", который устанавливается в ненулевое состояние как только ЕСВ передается функциям IPX или AES для обеспечения требования события (блок находится "в использовании"). Прикладная программа не должна изменять содержимое блока управления событием, пока флаг его статуса имеет ненулевое значение.

Когда IPX или AES заканчивают выполнение требуемой функции, включающей в себя данный ЕСВ, флаг "in use" этого ЕСВ будет автоматически устанавливаться в 0. После этого прикладная программа может изменять содержимое ЕСВ и любую, связанную с ним информацию.

Второй способ заключается в том, что каждый блок управления событием определяет адрес сервисной программы события, которая будет вызываться в режиме прерывания, когда событие произойдет.

Структура блока управления событием для функций IPX несколько отличается от структуры блока управления событием для AES, однако, одинаковый формат первых нескольких полей обоих типов ЕСВ позволяет использовать оба типа ЕСВ асинхронным планировщиком событий. Таким образом, сервисная программа события, вызываемая событием IPX может использовать связанные с этим событием данные для перепланирования самой себя с помощью планировщика событий /1/.

ЕСВ для событий IPX.

Блок управления событием для событий IPX ("ЕСВ-IPX") имеет структуру, показанную в таблице 2.

Таблица 2 – Структура блока управления событием

Начальный байт	Поле	Размер	Тип данных
	Линия	4 байта	Длинный адрес 8086 (смещение, сегмент)
	Адрес ESR	4 байта	Длинный адрес 8086 (смещение, сегмент)
	"in use"	1 байт	Целое без знака
	Код завершения	1 байт	Целое без знака
	Номер сокета	2 байта	Старшее-младшее целое без знака
	Рабочая область	4 байта	Не определено IPX
	Рабочая область	12 байт	Не определено драйверов
	Непосредственный	6 байт	Старшее-младшее целое без знака
	Количество фрагментов	2 байта	Старшее-младшее целое без знака

Как видно из приведенной структуры, блок управления событием состоит из двух частей. Первая, фиксированная (36 байт длиной), содержит

поля статуса/информации, а также рабочую область для использования IPX и сетевыми драйверами. Вторая часть, переменной длины, является списком одного или более описателей фрагментов. Каждый описатель включает адрес и размер области памяти (фрагмента), из которой будет браться пакет для передачи или куда будет записываться пакет после приема /5/.

Описание полей IPX-ЕСВ.

Линия. Это поле используется IPX, когда ЕСВ имеет статус "in use". Если ЕСВ не находится в режиме "in use", прикладная система может использовать это поле для своих целей. В наиболее общем случае, это поле может использоваться как связное для хранения ЕСВ в списках и очередях.

Адрес ESR. Это поле содержит адрес определенной прикладной системой сервисной программы, которая будет вызываться IPX, когда ожидаемое событие (передача пакета или прием пакета) завершиться.

Поскольку IPX использует поля "in use" и "код завершения", прикладная программа может определять статус своих требований опросом в определенные моменты времени состояния этих полей и не использовать сервисную программу события. В этом случае поле "адрес ESR" должно быть установлено в ноль (четыре байта нулей).

"In use". Это поле имеет ненулевое значение в том случае, когда ЕСВ используется IPX или AES. В этом случае это поле может иметь следующие значения:

FFh - ЕСВ используется для передачи пакета;

FEh - ЕСВ используется для просмотра определенного сокета при ожидании пакета;

FDh - ЕСВ используется с AES и ожидает окончания временного интервала;

FBh - событие передачи или приема уже произошло, но ЕСВ временно хранится в очереди, ожидая обработки.

IPX будет устанавливать это поле в ноль, когда обработка будет закончена.

Код завершения. Это поле устанавливается программами IPX и отражает результат обработки требования IPX. Это поле не может считаться действительным до тех пор пока IPX не установит флаг статуса "in use" в ноль.

Номер сокета. Это поле содержит номер сокета, с которым связан ЕСВ. Если ЕСВ используется для передачи, это поле содержит номер сокета, из которого передается пакет. Если ЕСВ используется для приема, это поле содержит номер сокета, в который будет помещаться принятый пакет.

Рабочая область IPX. Это поле резервируется для использования стандартными программами IPX. Это поле может не устанавливаться в начальное состояние, но оно не должно изменяться пока блок управления событием используется программами IPX. Когда ЕСВ не используется IPX, эта поле может использоваться для каких-либо других целей.

Рабочая область драйвера. Это 12-ти байтовое поле резервируется для использования сетевыми драйверами. Оно может не устанавливаться в какое-то начальное значение, но оно не должно изменяться пока блок управления используется программами IPX. Когда ЕСВ не используется IPX, это поле может использоваться для других целей.

Непосредственный адрес. Это поле содержит адрес узла, которому был только что передан пакет или от которого только что прибыл пакет (это будет адрес межсетевого моста, если пакет передавался или был принят от узла, расположенного вне локальной сети).

Количество фрагментов. Это поле содержит количество буферных фрагментов, из которых пакет будет формироваться для передачи или количество буферов, в которых принятый пакет должен быть размещен.

Значение этого поля должно быть больше 0. Значение 1 показывает, что пакет передается/принимается из/в смежную область памяти, то есть один фрагмент содержит целый пакет.

Список описателей фрагмента. Описатель фрагмента точно определяет, где размещена часть пакета, который должен быть передан или где расположена часть принятого пакета. Описатель фрагмента имеет два поля:

адрес - адрес буфера, из которого данные должны быть взяты при формировании пакета для передачи или в который данные должны быть помещены при приеме пакета;

размер - размер буферного фрагмента, указанного предшествующим адресом.

Любой блок управления событием может иметь произвольное число описателей фрагмента но не менее одного. Эти описатели составляют список описателей фрагмента. Необходимо, чтобы буферный фрагмент, определенный первым в списке описателей фрагментов должен быть по крайней мере 30 байт длиной и должен включать полный заголовок пакета IPX. Полный размер пакета (сумма размеров отдельных фрагментов) не должен превышать 576 байт /1/. Формат передаваемых по сети пакетов представлен на рисунке 1.

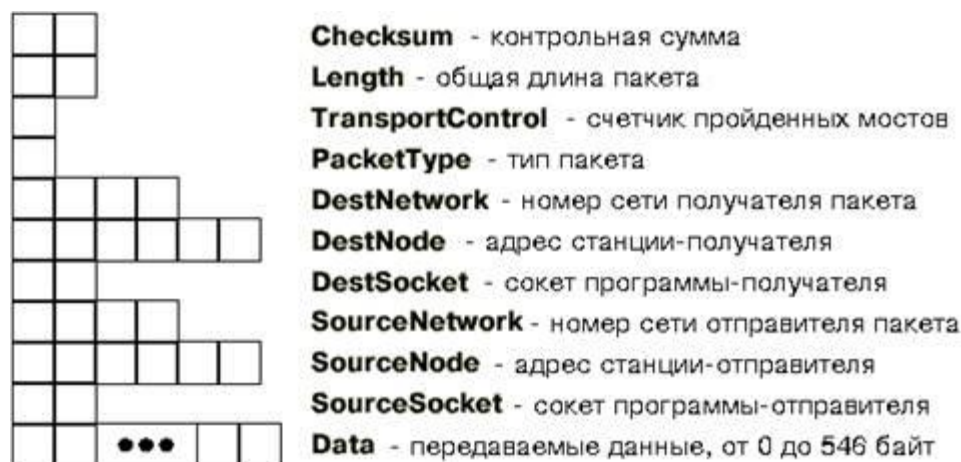


Рисунок 1 - Формат передаваемых по сети пакетов

Пакет можно разделить на две части - заголовок и передаваемые данные. Все поля, кроме последнего (Data), представляют собой заголовок пакета.

Поле Checksum предназначено для хранения контрольной суммы передаваемых пакетов. При формировании собственных пакетов вам не придется заботиться о содержимом этого поля, так как проверка данных по контрольной сумме выполняется драйвером сетевого адаптера.

Поле Length определяет общий размер пакета вместе с заголовком. Длина заголовка фиксирована и составляет 30 байт. Размер передаваемых в поле Data данных может составлять от 0 до 546 байт, следовательно, в поле Length в зависимости от размера поля Data могут находиться значения от 30 до 576 байт. Если длина поля Data равна нулю, пакет состоит из одного заголовка. Как это ни странно, такие пакеты тоже нужны. При формировании собственных пакетов не надо проставлять длину пакета в поле Length, протокол IPX сделает это сам (вернее, программный модуль, отвечающий за реализацию протокола IPX, вычислит длину пакета на основании длины поля Data).

Поле TransportControl служит как бы счетчиком мостов, которые проходит пакет на своем пути от передающей станции к принимающей. Каждый раз, когда пакет проходит через мост, значение этого счетчика увеличивается на единицу. Перед передачей пакета IPX сбрасывает содержимое этого поля в нуль.

Поле PacketType определяет тип передаваемого пакета. Программа, которая передает пакеты средствами IPX, должна установить в поле PacketType значение 4. Протокол SPX, реализованный на базе IPX, использует в этом поле значение 5.

Поле DestNetwork определяет номер сети, в которую передается пакет. При формировании собственного пакета вам необходимо заполнить это четырехбайтовое поле.

Поле DestNode определяет адрес рабочей станции, которой предназначен пакет. Вам необходимо определить все шесть байт этого поля.

Поле DestSocket предназначено для адресации программы, запущенной на рабочей станции, которая должна принять пакет. При формировании пакета вам необходимо заполнить это поле.

Поля SourceNetwork, SourceNode и SourceSocket содержат соответственно номер сети, из которой посылается пакет, адрес передающей станции и сокет программы, передающей пакет.

Поле Data в пакете IPX содержит передаваемые данные. Как мы уже говорили, длина этого поля может быть от 0 до 546 байт. Если длина поля Data равна нулю, пакет состоит из одного заголовка. Такой пакет может использоваться программой, например, для подтверждения приема пакета с данными.

Для приема или передачи пакета прикладная программа должна подготовить пакет данных, сформировав его заголовок, и построить так называемый блок управления событием ECB (Event Control Block). В блоке ECB задается адресная информация для передачи пакета, адрес самого передаваемого пакета в оперативной памяти и некоторая другая информация.

Подготовив блок ECB, прикладная программа передает его адрес соответствующей функции IPX для выполнения операции приема или передачи пакета.

Если программе требуется измерять временные интервалы, она может воспользоваться асинхронным планировщиком событий AES, реализованным в рамках драйвера IPX.

Для функций AES можно использовать тот же формат ECB, что и для функций IPX. Однако поля используются немного по-другому.

Для некоторых приложений (например, для программ, передающих файлы между рабочими станциями) удобнее использовать сетевой протокол более высокого уровня, обеспечивающий гарантированную доставку пакетов в правильной последовательности /1/.

Протокол SPX - протокол последовательного обмена пакетами (Sequenced Packet Exchange Protocol), разработанный Novell.

Пакет, передаваемый при помощи протокола SPX, имеет более длинный заголовок. Дополнительно к 30 байтам стандартного заголовка пакета IPX добавляется еще 12 байт /4/.

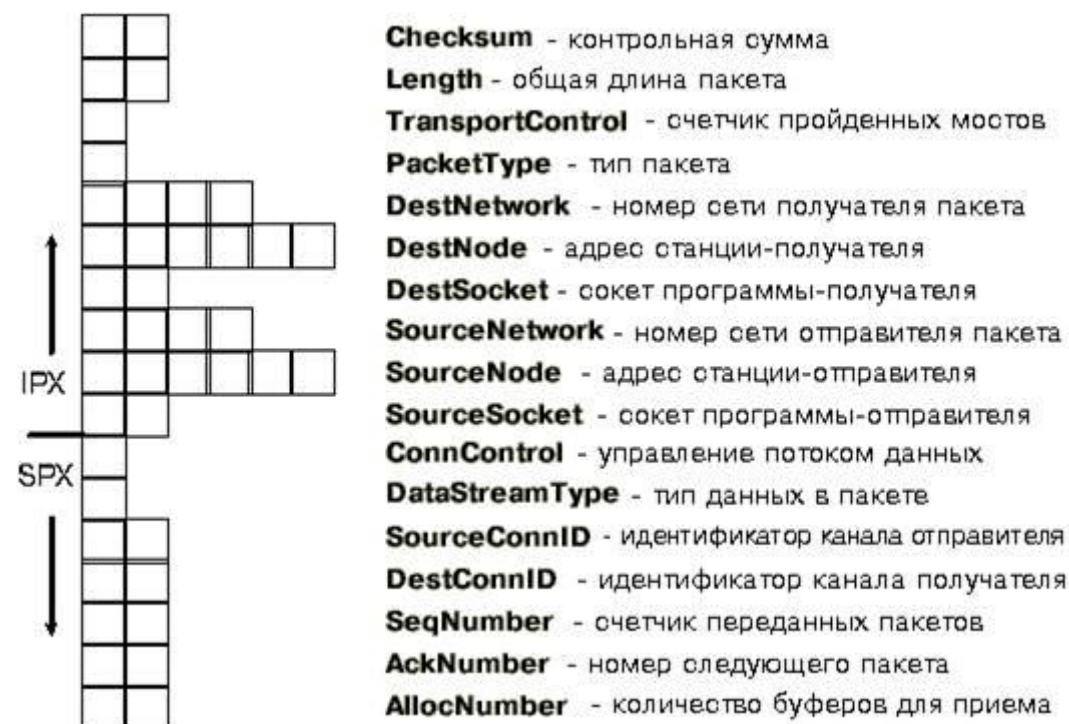


Рисунок 2 – Структура пакета протокола SPX

Выполнение работы:

- изучить назначение протокола IPX/SPX;
- изучить основную структура пакета IPX;
- изучить структуру заголовка пакета IPX;

- изучить описание полей заголовка IPX;
- изучить структуру блока управлением событием;
- изучить описание полей ESB;
- изучить формат передаваемых по сети пакетов;
- изучить основную структура пакета SPX;
- оформить отчет.

Отчет по лабораторной работе должен содержать следующее:

- название и цель работы;
- основные теоретические пункты общих сведений;
- выводы по выполненной работе;
- список использованных источников.

Контрольные вопросы:

- назначение протокола IPX/SPX?;
- назначение сетевых драйверов?;
- назначение асинхронного планировщика событий (AES)?;
- что является событиями?;
- что представляет собой основная структура пакетов IPX?;
- из каких полей состоит структура заголовка пакета IPX?;
- что такое сокет и его назначение?;
- какие сокеты используются в протоколе ?;
- назначение узла приемника и узла передатчика?;
- что представляют собой «События и блоки управления событиями»?;
- что представляет собой «Структура блока управлением событием»?;
- какие поля используются в «Структуре блока управлением событием»?;
- что такое «Рабочая область IPX» ?;
- что такое «Рабочая область драйвера» ?;
- что такое «Непосредственный адрес» ?;
- что такое «Количество фрагментов» ?;
- что такое «Список описателей фрагмента» ?;
- что представляет собой формат протокола IPX ?;
- что представляет собой формат протокола SPX ?.

Сетевые протоколы и службы

A
|

Обзор сетевых протоколов NetBEUI, IPX/SPX; служб DHCP, WINS, RRAS

Сетевые протоколы IPX/SPX, NetBEUI

Существует достаточно много стеков протоколов, широко применяемых в сетях. Это и стеки, являющиеся международными и национальными стандартами, и фирменные стеки, получившие распространение благодаря распространенности оборудования той или иной фирмы. Примерами популярных стеков протоколов могут служить: стек IPX/SPX фирмы Novell, стек TCP/IP, используемый в сети Internet и во многих сетях на основе операционной системы UNIX, стек OSI международной организации по стандартизации и некоторые другие.

Протокол NetBEUI

Протокол *NetBEUI* (*NetBIOS Extended User Interface*) ведет свою историю от сетевого программного интерфейса *NetBIOS* (*Network Basic Input/Output System*), появившегося в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM.

Протокол *NetBEUI* разрабатывался как эффективный протокол, потребляющий немного ресурсов, для использования в сетях, насчитывающих не более 200 рабочих станций. Этот протокол содержит много полезных сетевых функций, которые можно отнести к сетевому, транспортному и сеансовому уровням модели OSI, однако с его помощью невозможна маршрутизация пакетов. Это ограничивает применение протокола *NetBEUI* локальными сетями, не разделенными на подсети, и делает невозможным его использование в *составных сетях*. Некоторые ограничения *NetBEUI* снимаются реализацией этого протокола NBF (*NetBEUI Frame*), которая включена в операционную систему Microsoft Windows NT.

В настоящее время в операционных системах семейства Windows 2000 данный протокол уже практически не используется, а в системах Windows XP/2003 отсутствует даже возможность добавления данного протокола в *Свойствах* сетевого подключения системы (хотя, если необходима совместимость с какими-либо приложениями, унаследованными от старых систем и работающих только по протоколу *NetBEUI*, в дистрибутивах систем Windows XP/2003 имеются установочные файлы для добавления данного протокола).

Стек протоколов IPX/SPX

Этот стек является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х годов. Протоколы сетевого и сеансового уровня *Internetwork Packet Exchange* (IPX) и *Sequenced Packet Exchange* (SPX), которые дали название стеку, являются прямой адаптацией протоколов *XNS* фирмы Xerox, распространенных в гораздо меньшей степени, чем стек IPX/SPX. Популярность стека IPX/SPX непосредственно связана с операционной системой Novell NetWare.

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare (до версии 4.0) на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти и которые бы быстро работали на процессорах небольшой вычислительной мощности. В результате протоколы стека IPX/SPX до недавнего времени хорошо работали в локальных сетях и не очень — в больших корпоративных сетях, так как они слишком перегружали медленные *глобальные связи* широковещательными пакетами, которые интенсивно используются несколькими протоколами этого стека (например, для установления связи между клиентами и серверами). Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell, и на его реализацию нужно получать у нее лицензию, долгое время ограничивали распространенность его только сетями NetWare. С момента выпуска версии NetWare 4.0 Novell внесла и продолжает вносить в свои протоколы серьезные изменения, направленные на приспособление их для работы в корпоративных сетях. Сейчас стек IPX/SPX реализован не только в NetWare, но и в нескольких других популярных сетевых ОС, например, Microsoft Windows NT. Начиная с версии 5.0 фирма Novell в качестве основного протокола своей серверной операционной системы стала использовать протокол TCP/IP, и с тех пор практическое применение IPX/SPX стало неуклонно снижаться.

Как уже говорилось выше, стек протоколов IPX/SPX является фирменным запатентованным стеком компании Novell. Реализация данного протокола в операционных системах Microsoft называется NWLink (или IPX/SPX-совместимый протокол). Добавить данный протокол можно через *Свойства "Подключения по локальной сети"* (кнопка "Установить", выбрать "Протокол", кнопка "Добавить", выбрать "NWLink", кнопка "OK"; [рис. 10.1](#)).

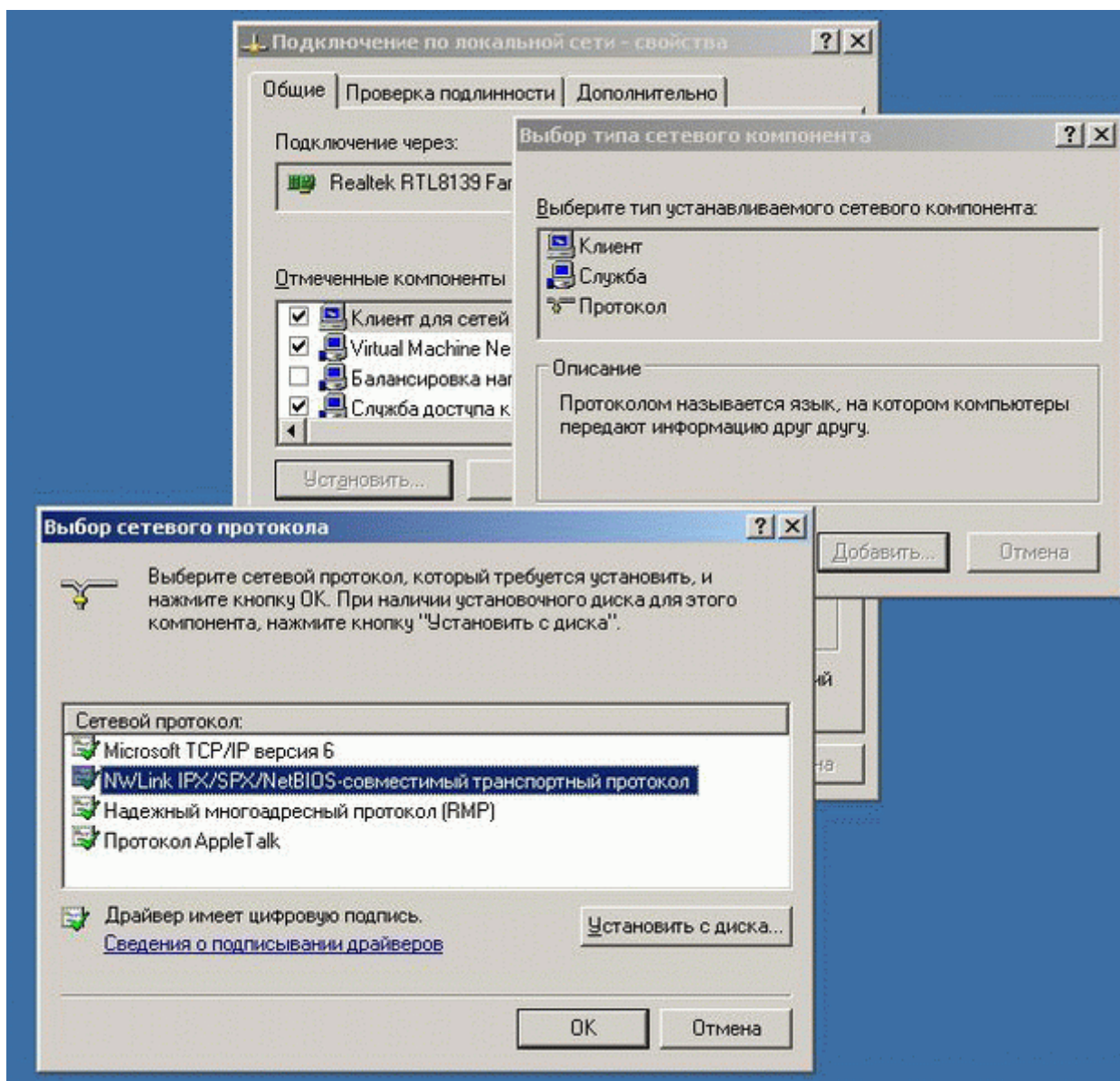


Рис. 10.1.

Для того, чтобы из сети под управлением систем семейства Windows получить доступ в сеть под управлением служб каталогов Novell, кроме NWLink, необходимо установить также клиента сетей Novell (*Свойства "Подключения по локальной сети"*, кнопка "Установить", выбрать "Клиент", кнопка "Добавить", выбрать "Клиент для сетей NetWare", кнопка "OK"; [рис. 10.2](#)).

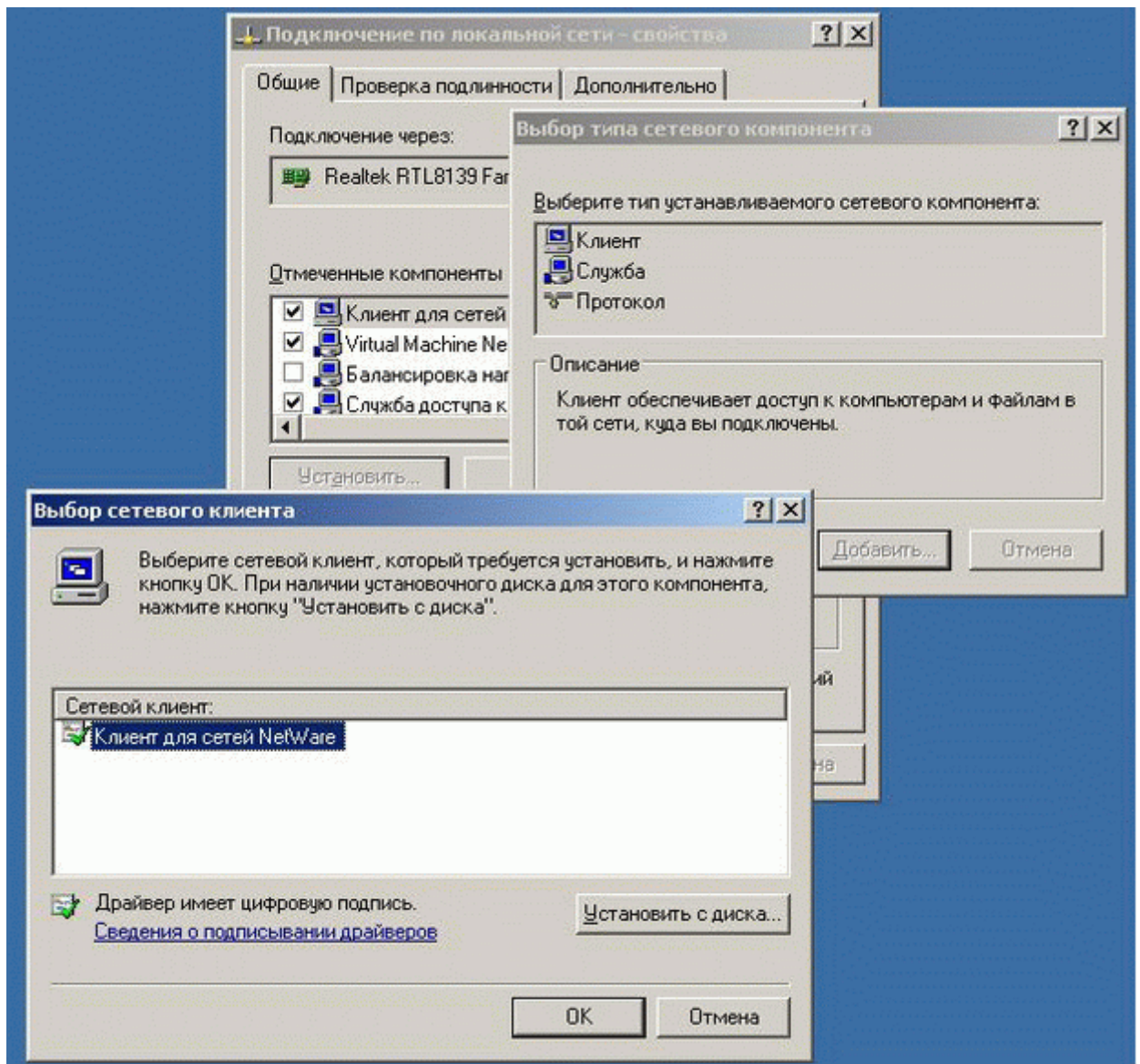


Рис. 10.2.

В серверных системах Windows (до Windows 2000 включительно) имелась также служба под названием " *Шлюз для сетей NetWare* ", позволявшая клиентам сетей Microsoft без установки клиента сетей NetWare получать доступ к ресурсам серверов под управлением Novell NetWare (через шлюз, установленный на сервере Windows NT/2000). В системе Windows 2003 служба шлюза отсутствует.

Служба DHCP

Служба *DHCP* (*Dynamic Host Configuration Protocol*) — это одна из служб поддержки протокола TCP/IP, разработанная для упрощения администрирования IP-сети за счет использования специально настроенного сервера для централизованного управления IP-адресами и другими параметрами протокола TCP/IP, необходимыми сетевым узлам. Сервер DHCP избавляет сетевого администратора от необходимости ручного выполнения таких операций, как:

- автоматическое назначение сетевым узлам IP-адресов и прочих параметров протокола TCP/IP (например, маска подсети, адрес основного шлюза подсети, адреса серверов DNS и WINS);
- недопущение дублирования IP-адресов, назначаемых различным узлам сети;
- освобождение IP-адресов узлов, удаленных из сети;
- ведение централизованной БД выданных IP-адресов.

Особенности службы DHCP в системах семейства Windows Server:

- **Интеграция с DNS** — DHCP-серверы могут осуществлять динамическую регистрацию выдаваемых IP-адресов и FQDN-имен сетевых узлов в базе данных DNS-сервера (это особенно актуально для сетевых клиентов, которые не поддерживают динамическую регистрацию на сервере DNS, например, Windows 95/98/NT4);
- **Авторизация сервера DHCP в Active Directory** — если сетевой администратор установит службу DHCP на сервере Windows 2000/2003, то сервер не будет функционировать, пока не будет авторизован в AD (это обеспечивает защиту от установки несанкционированных DHCP-серверов);
- **Резервное копирование базы данных DHCP** — Созданная резервная копия может использоваться впоследствии для восстановления работоспособности DHCP-сервера.

Определим основные термины, относящиеся к службе DHCP

- **Клиент DHCP** — сетевой узел с динамическим IP-адресом, полученным от сервера DHCP;
- **Период аренды** — срок, на который клиенту предоставляется IP-адрес;
- **Область** — это полный последовательный диапазон допустимых IP-адресов в сети (чаще всего области определяют отдельную физическую подсеть, для которой предоставляются услуги DHCP);
- **Исключаемый диапазон** — это ограниченная последовательность IP-адресов в области, которая исключается из числа адресов, предлагаемых службой DHCP (исключаемые диапазоны гарантируют, что сервер не предложит ни один адрес из этих диапазонов DHCP-клиентам в сети);
- **Доступный пул адресов** в области — адреса, оставшиеся после определения области DHCP и исключаемых диапазонов (адреса из пула могут быть динамически назначены сервером DHCP-клиентам в сети);
- **Резервирование** — назначение DHCP-сервером определенному сетевому узлу постоянного IP-адреса (резервирования гарантируют, что указанный сетевой узел будет всегда использовать один и тот же IP-адрес).

Рассмотрим технологию предоставления IP-адресов DHCP-сервером DHCP-клиентам

При загрузке компьютера, настроенного на *автоматическое получение IP-адреса*, или при смене статической настройки IP-конфигурации на динамическую, а также при обновлении IP-конфигурации сетевого узла происходят следующие действия:

1. компьютер посылает широковещательный *запрос на аренду IP-адреса* (точнее, на обнаружение доступного DHCP-сервера, *DHCP Discover*);
2. DHCP-серверы, получившие данный запрос, посылают данному сетевому узлу свои *предложения IP-адреса* (*DHCP Offer*);
3. клиент отвечает на предложение, полученное первым, соответствующему серверу запросом на *выбор арендуемого IP-адреса* (*DHCP Request*);
4. DHCP-сервер регистрирует в своей БД выданную IP-конфигурацию (вместе с именем компьютера и физическим адресом его сетевого адаптера) и посылает клиенту подтверждение на *аренду IP-адреса* (*DHCP Acknowledgement*).

Данный процесс изображен на [рис. 10.3](#):

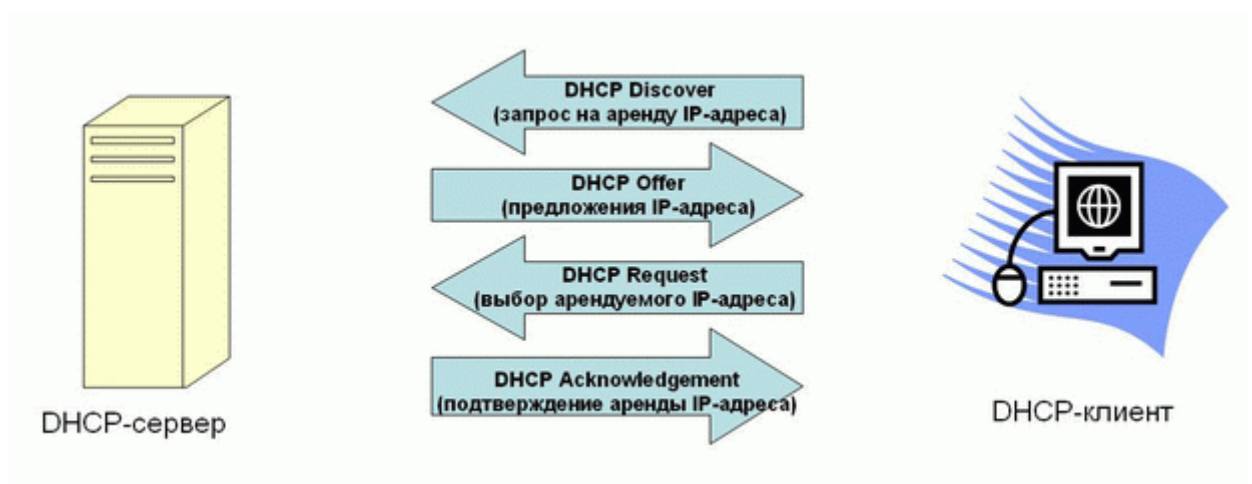


Рис. 10.3.

Интеллектуальные системы автоматического управления электрическим освещением

В статье анализируются основные требования к системам электрического освещения, рассматриваются разновидности структурного построения систем освещения. Автор описывает специализированные интерфейсы, сетевые протоколы и международные стандарты систем интеллектуального управления освещением, приводит примеры технических решений и интегральных электронных компонентов для систем управления светом, предлагаемые производителями микросхем. Также дается прогноз развития технологий управления электрическим освещением.

ОСНОВНЫЕ ТРЕБОВАНИЯ К СИСТЕМАМ ЭЛЕКТРИЧЕСКОГО ОСВЕЩЕНИЯ

Главное требование, предъявляемое к системе освещения — обеспечение заданного уровня освещенности. Существенная роль отводится также санитарным требованиям, таким как спектральный состав света и частота мерцания. Все эти требования призваны создать максимальный комфорт для зрения и, как следствие, обеспечить максимальную эффективность и безопасность производственных или учебных процессов.

Следующим по значимости является требование экономической эффективности, т.е. минимум затрат на создание и эксплуатацию системы освещения. Следует подчеркнуть особую важность второй составляющей. Она напрямую связана с энергетической эффективностью системы освещения. Последняя зависит не только от коэффициента полезного действия осветительных приборов, но и от интенсивности их использования как по уровню производимого света, так и по времени использования. Экономия электроэнергии на освещение — не только экономическая, но и экологическая задача: чем меньше требуется электроэнергии, тем меньше выделяется тепла от самих электроосветительных приборов, меньше сжигается топлива на электростанциях. В мировом масштабе это приводит к снижению угрозы глобального потепления и уменьшению загрязнения атмосферы.

Совершенно очевидно, что существуют два направления технического прогресса в области электрического освещения: совершенствование электрических осветительных приборов и совершенствование систем управления электрическим освещением. Первое направление связано с появлением новых типов галогенных и флуоресцентных ламп, а также т.н. белых осветительных светодиодов. Второе направление подробно рассматривается в настоящей статье.

СТРУКТУРНОЕ ПОСТРОЕНИЕ СИСТЕМ ЭЛЕКТРИЧЕСКОГО ОСВЕЩЕНИЯ

Сфера применения систем освещения очень разнообразна. Для простоты можно рассмотреть построение системы освещения в отдельной комнате с окном [1], обеспечивающей заданный уровень освещенности с 6 утра до 22 часов. Колоколообразная кривая, показанная на рисунке 1, соответствует естественному освещению. Согласно графику, с 9 часов утра до 17 часов (зона В) нет необходимости в искусственном освещении. Более того, есть избыток естественного освещения (показано пунктирной штриховкой), который необходимо устранить путем ограничения доступа естественного света в помещение. С 6 до 9 утра и с 17 до 22 часов естественного освещения недостаточно, необходимо включать искусственное освещение (зоны А). При этом существует потенциальная возможность экономии электроэнергии, если включать освещение не на полную мощность, а ровно на столько, чтобы восполнить недостаток естественной освещенности (см. области с вертикальной штриховкой). Естественно, что картина, показанная на рисунке 1, не является постоянной, а подвержена календарным и погодным изменениям.

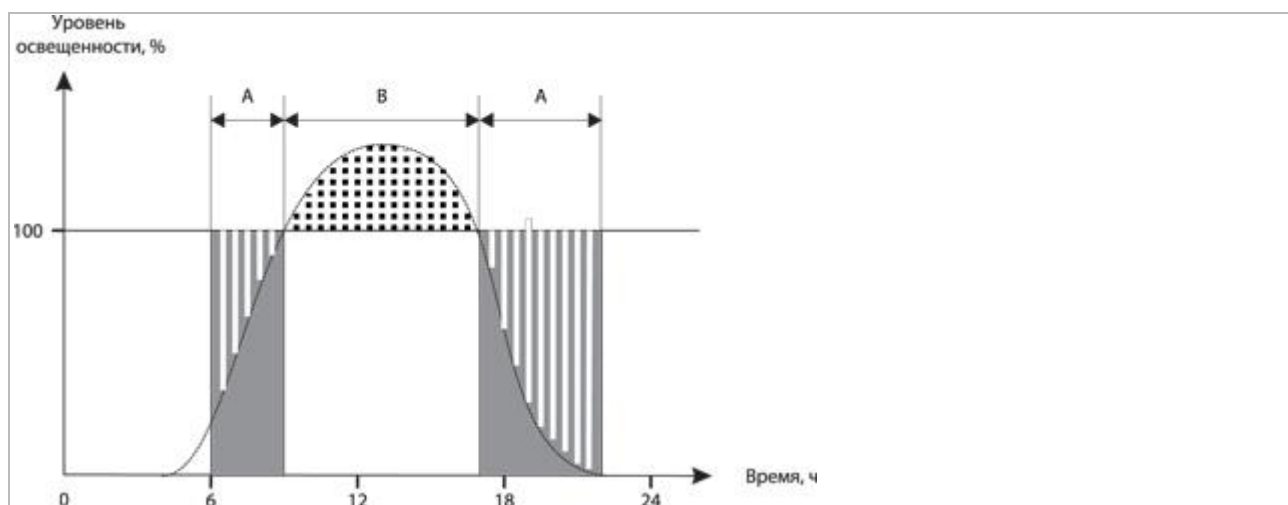


Рис. 1. Суточный цикл освещенности помещения

Таким образом, простейшая схема системы освещения с ручным управлением будет иметь вид, показанный на рисунке 2. Она содержит исполнительные устройства (электрические светильники и жалюзи с электроприводом) и управляющие устройства (электрические выключатели), соединенные силовыми проводами с электрической сетью. Управление — открытие/закрытие жалюзи, включение/выключение светильников — осуществляется человеком вручную на основе

визуальной оценки уровня освещенности. Подобные системы используются в настоящее время на производстве и в быту. Главными недостатками такой системы являются непостоянство освещенности, перерасход электроэнергии и цветных металлов на электропроводку.

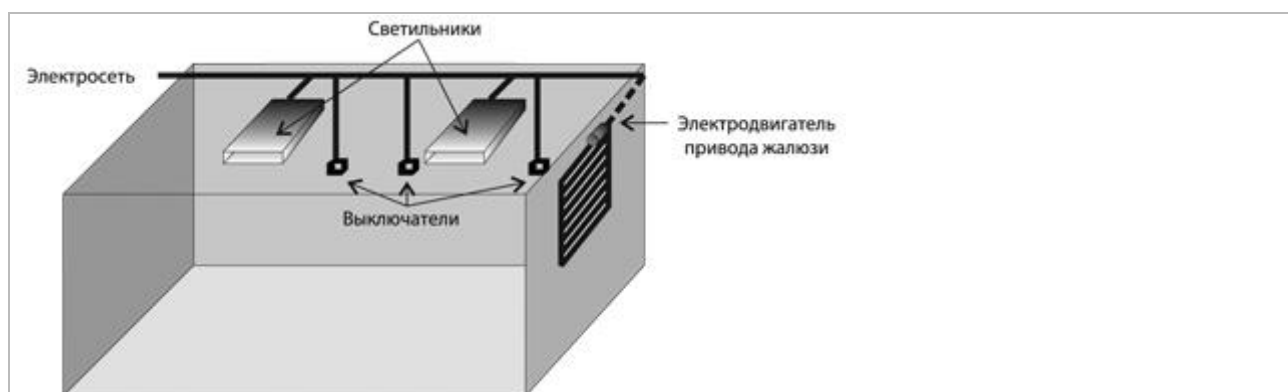


Рис. 2. Система освещения с ручным управлением

Первые два недостатка можно компенсировать, если осуществлять управление системой освещения автоматически. Возможный вариант системы с централизованным автоматическим управлением показан на рисунке 3. В такой системе постоянный уровень освещения поддерживается путем регулирования степени открытия жалюзи (в часы естественного освещения) и регулирования силы света в светильниках (в часы искусственного освещения) с помощью автоматического контроллера. Такая система требует применения датчика освещенности и датчика времени. Она гораздо более эффективна, однако не способствует сокращению длины силовых проводов.

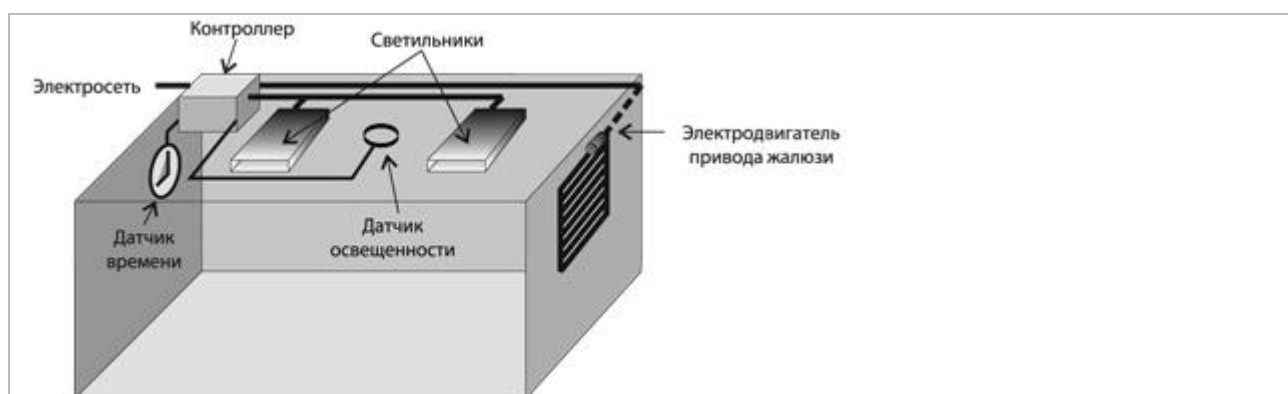


Рис. 3. Автоматическая система управления светом

Построение системы освещения с распределенным сетевым автоматическим управлением [2], которое иллюстрирует рисунок 4, позволяет избавиться от этого недостатка. Система освещения строится на основе интеллектуальных актуаторов (actuator), управляющих исполнительными устройствами, и интеллектуальных сенсоров (sensor). Слово «интеллектуальный» подчеркивает тот факт, что в составе каждого устройства имеется искусственный интеллект — микроконтроллер. Сенсоры и актуаторы запитываются от общей электрической сети и могут обмениваться сообщениями через локальную сеть на основе информационной шины, в соответствии с определенным стандартным протоколом. Такая шина может быть реализована в виде витой пары проводников, виртуального канала с частотным уплотнением непосредственно в силовой сети или в виде радиоканала. Кроме экономии цветных металлов, такая система имеет дополнительно два очень существенных преимущества. Первое из них состоит в том, что очень просто реализуется подключение дополнительных устройств и, следовательно, расширение функций.

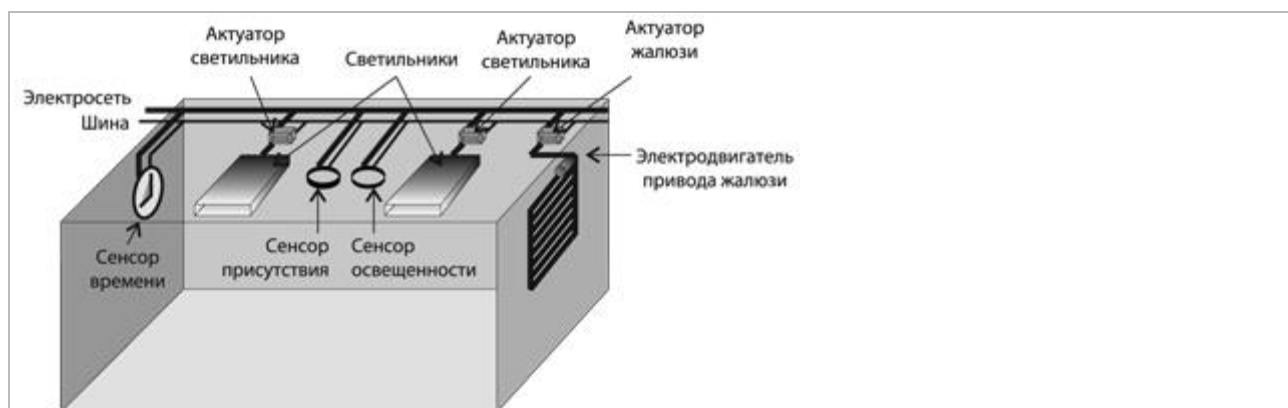


Рис. 4. Система освещения с распределенным сетевым автоматическим управлением

Например, можно подключить датчик присутствия людей в помещении, как это изображено на рисунке 4, и отключать освещение полностью в случае, если людей нет в помещении. Нетрудно предположить, что за счет такой дополнительной функции можно получить существенную экономию электроэнергии, особенно во вспомогательных помещениях: складах, коридорах, туалетах, лифтах и т.п. Второе преимущество состоит в возможности построения иерархической сети, позволяющей объединить локальные сети отдельных помещений в систему освещения этажа, а системы управления этажей — в систему управления освещением всего здания и т.п. Подключение к такой иерархической сети компьютера, связанного с интернетом, делает возможным дистанционное управление системой из любой точки мира.

Использование распределенного сетевого управления позволяет интегрировать систему управления освещением в интеллектуальную систему управления типа «Умный дом» [3]. Системы последнего типа, кроме функции управления освещением, включают также следующие функции:

- кондиционирование воздуха;
- управление мультимедийной аппаратурой;
- охрана от несанкционированного вторжения;
- управление безопасностью технических систем электро-, водо- и газоснабжения;
- дистанционное управление всеми подсистемами.

Нетрудно видеть, что для реализации всех указанных функций достаточно добавить в систему на рисунке 4 определенный набор интеллектуальных сенсоров и актуаторов, приведенный в таблице 1.

Таблица 1. Сенсоры и актуаторы системы «Умный дом»

Функция	Сенсоры	Актуаторы
Управление освещением	Освещенности Присутствия людей Реального времени	Управление светильниками различных типов Управление жалюзи
Кондиционирование воздуха	Температуры воздуха Влажности воздуха Метеорологических параметров	Управление системой отопления Управление кондиционером Управление системой вентиляции
Управление мультимедийной аппаратурой	Пульт программирования	Управление аудиоаппаратурой Управление видеоаппаратурой Управление светомузыкальной аппаратурой
Охрана от несанкционированного вторжения	Сенсоры охранной сигнализации Видеокамеры наблюдения Сенсоры идентификации личности	Управление замками Управление тревожной сигнализацией Связь с правоохранительными службами
Управление безопасностью технических систем	Сенсоры состояния электроснабжения Сенсоры затопления Сенсоры утечки газа	Управление аварийным отключением Связь с ремонтными службами
Дистанционное управление	Сенсоры пультов дистанционного управления (инфракрасными, радио и т.п.)	Управление информационными дисплеями

Таким образом, весьма перспективно построить систему управления электрическим освещением по принципу распределенного сетевого управления. С этой целью разработан целый ряд промышленных стандартов на интерфейсы и протоколы для построения таких систем.

СТАНДАРТНЫЕ ИНТЕРФЕЙСЫ И ПРОТОКОЛЫ СИСТЕМ УПРАВЛЕНИЯ СВЕТОМ

Одним из первых стандартных интерфейсов [4], разработанных после появления первых биполярных транзисторов, является аналоговый стандарт 0-10V DC — управление уровнем света пропорционально величине постоянного напряжения в диапазоне 0...10 В. В соответствии с этим стандартом до сих пор производятся актуаторы светильников: диммеры (dimmer — затемнитель, регулятор света) для ламп накаливания и балласты (ballast) для регулирования свечения флуоресцентных ламп. Физический интерфейс этого типа, рассчитанный на 6 каналов управления, использовал 8-контактный разъем типа DIN или XLR и многожильный кабель. В связи с распространенностью этого интерфейса в системах управления освещением, в 2001 г. Национальный институт стандартов США утвердил стандарт E1.3-2001.

Дальнейшим развитием аналоговых интерфейсов явился стандарт AMX192 (Analog Multiplex Data Transmission). Предложенный впервые в 1975 г. фирмой Strand Lighting и получивший статус стандарта в 1985 г., он широко

распространен в США и Канаде. Интерфейс использует 4-контактный разъем XLR и 2 пары витых проводников: по одной передается мультиплексированный аналоговый сигнал уровня света (0...5 В), по другой — синхриимпульсы разделения каналов. На одной шине может быть установлено 96—192 приемников-диммеров, преобразующих передаваемый сигнал к стандартному 0...10 В.

Более сложный протокол аналогового мультиплексирования используется в промышленном стандарте Microplex (MPX), или LMX-128. Физический интерфейс использует 3-контактный разъем XLR и микрофонный кабель. По линии данных передается аналоговый сигнал в диапазоне от 6—8 В. Отрицательная часть диапазона используется для передачи импульсов синхронизации (т.е. адресной информации), а положительная — для передачи аналогового сигнала уровня света (0 В — выключено, 8 В — максимум, т.е. 100%). К одному кабелю можно подключать до 128 диммеров.

Следующим по популярности является интерфейс DMX-512 (Digital MultipleX), обеспечивающий управление 512-ю устройствами. Система на основе DMX-512 предполагает наличие одного передатчика и до 512-ти приемников. Физический интерфейс использует 5- или 3-контактный разъем типа XLR и 2 витых пары проводников (вторая пара является не обязательной, и в случае 3-контактного разъема не используется). Электрические параметры сигналов в шине DMX-512 полностью соответствуют стандарту последовательных каналов RS-485. Данные передаются со скоростью 250 Кбит/с. Способ передачи — асинхронный (СТАРТ — 1 бит, СТОП — 2 бита). Фрейм содержит стартовый байт и 512 байтов данных. Каждый байт данных несет информацию об уровне света соответствующего канала (0 — выключено, 255 — максимум, т.е. 100%), в порядке следования данных. Протокол не предусматривает каких-либо подтверждений приема и средств контроля ошибок. Впервые интерфейс был разработан Институтом театральных технологий США в 1986 г. для управления осветительными приборами, затем улучшен в 1990 г. Современная версия DMX-512-A принята Институтом стандартов США под номером E1.11.

Более продвинутым и современным интерфейсом для сетевых систем управления электрическим освещением является DALI (Digital Addressable Lighting Interface). Этот интерфейс считается расширением международного стандарта IEC60929 и очень популярен у европейских производителей. Он пригоден для применения как в отдельных комнатах, так и для иерархических систем больших зданий. Система управления светом DALI основана на принципе master-slave: пользователь системы через контроллер, работающий в режиме master, отдает команды контроллерам, работающим в режиме slave и управляющим диммерами и балластами светильников. Каждый slave имеет свой индивидуальный 6-битный адрес, т.е. к одной линии связи DALI можно подсоединить до 64-х диммеров или балластов. Есть возможность разделения управляемых устройств на группы по 16 устройств в каждой. Связь осуществляется по двухпроводной линии дифференциально: разность потенциалов более 9,5 В принимается за логическую 1, менее 6,5 В — за логический 0. Скорость передачи — 1200 бит/с. Полудуплексная передача осуществляется двухфазным (манчестерским) кодом. Формат передачи master содержит бит СТАРТ, 2 байта данных и 2 бита СТОП. Первый байт это адрес (индивидуальный или групповой), второй байт — команда. Стандарт определяет специальную систему команд, задающую режимы работы диммера или балласта (уровень свечения, режим уменьшения или увеличения уровня света и т.п.), а также команды для проверки статуса управляемого контроллера и лампы. Последние требуют ответа. Формат сообщения slave содержит бит СТАРТ, 1 байт и 2 бита СТОП.

И наконец, разработаны всеобъемлющие системы стандартов для комплексной автоматизации зданий, поддерживающие и управление электрическим освещением. Наиболее применяемые из них — это X10 и EIB.

Стандарт домашней автоматики X10 [5] базируется на идее передачи управляющей информации непосредственно по проводам питающей сети. Эта технология PLC (Power Line Carrier) была запатентована в начале 70-х гг. прошлого столетия фирмой PICO Electronics в Великобритании. Позднее команда инженеров этой фирмы переехала в США и продолжала работу. 9 экспериментов оказались неудачными, а 10-й дал положительный результат. Отсюда и появилось название X10, а процветающая и поныне фирма X10 USA является ведущей фирмой США в области домашней автоматизации.

Стандартный фрейм передачи содержит уникальный 4-битный старт-код (1110), 4-битный код дома (первая часть адреса приемника) и 5-битный код устройства. Последний может быть второй частью кода адреса приемника (5-й бит равен 0) или кодом функции команды (5-й бит равен 1). Одна из функций (Extended Data — расширенные данные) предусматривает присоединение дополнительных байтов, передаваемых без перерыва, причем первый из них содержит число передаваемых байтов. Биты кода дома и кода адреса устройства передаются двухфазным способом (бит и дополнение до 1). Из-за этого фрейм, содержащий 13 информационных бит, в действительности содержит 22 бита ($4+9\times 2$) и передается в течение 11-ти периодов частоты сети 60 Гц (каждые полпериода — 1 бит).

Начало передачи каждого бита синхронизируется моментами пересечения несущего напряжения через 0 (фаза 0 и 180°). Единичное значение бита передается путем посылки пакета из трех импульсов длительностью 1 мс, заполненных синусоидальным сигналом частотой 120 кГц и сдвинутых по фазе на 30°. Нулевое значение бита передается путем отсутствия пакета в соответствующий момент времени. Каждый фрейм для надежности передается дважды. Для передачи команды необходимо послать 2 двойных фрейма: адресный, содержащий код устройства, и командный, содержащий код функции. С учетом необходимого промежутка в 3 периода между двойными фреймами, время передачи одной команды требует 47 периодов напряжения силовой сети (для частоты 60 Гц — 0,78 с, для адаптированного европейского варианта с частотой 50 Гц — 0,96 с).

Современный X10 допускает использование также радиоканала с частотой 310 МГц (США) или 433 МГц (Европа), по которому передаются описанные выше фреймы.

Европейский стандарт EIB (European Installation Bus) [6] рассчитан на построение управляющей сети с помощью витой пары, на основе передачи данных непосредственно по проводам силовой сети, на базе радиоканала или с использованием оптического инфракрасного канала. Протокол передачи по всем видам связи идентичен. К каждой линии EIB можно подключить до 64-х устройств; иерархическая сеть может содержать до 65536-ти устройств. Способ передачи по витой паре — балансный, без модуляции, асинхронный, скорость передачи — 9600 бит/с. Снабжение устройств питающим

напряжением и передача информации производится по тем же проводам, поэтому логический 0 передается двухполярным импульсом на уровне питания, а логическая 1 — отсутствием импульса. В системе может быть несколько устройств, инициирующих передачу, поэтому для разрешения конфликтов на шине используется стандартный алгоритм CSMA/CA.

Согласно спецификациям EIB, сенсоры и актуаторы обмениваются телеграммами. Сенсоры используют индивидуальные физические адреса, а актуаторы — групповые. Это дает возможность сенсору послать одну команду сразу многим актуаторам. Каждая телеграмма содержит пакет данных и требует обязательного подтверждения. Пакет данных содержит управляющее поле, поле адреса источника, поле адреса приемника, байт длины пакета, блок данных сообщения и байт контроля ошибок. EIB поддерживает все принятые в вычислительных сетях типы данных и протоколы, что позволяет строить сложные системы управления электрическим освещением с управлением от компьютеров с использованием интернета.

В мае 1999 г. ассоциация производителей изделий EIB объединилась с двумя другими ассоциациями в ассоциацию Konnex, которая приняла единый расширенный стандарт KNX для автоматизации жилых и промышленных зданий. В его основу был положен стандарт EIB. Поэтому ссылки на этот стандарт в технической литературе часто обозначаются как KNX/EIB. Основываясь на описанных стандартах, производители оборудования и приборов предлагают потребителям ряд интегрированных технологий для построения сетевых систем автоматического управления электрическим освещением.

ИНТЕГРИРОВАННЫЕ ТЕХНОЛОГИИ ДЛЯ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ СВЕТОМ
Современный уровень интеллектуальных технологий управления освещением можно проследить на примере одной из крупнейших европейских фирм, поставляющей такие технологии на мировой рынок, — германской фирмы INSTA [7] (она представляет группу компаний, куда входят также ELKA, Berker, Gira и Jung). Продукция этой фирмы — технологии и компоненты для реализации проектов освещения жилых домов и промышленных зданий. В таблице 2 приведено сравнение 6-ти интегрированных технологий, предлагаемых этой фирмой.

Таблица 2. Интегрированные технологии систем управления освещением фирмы INSTA

Характеристика	Наименование				
	Instalight PROFI	Instalight PLUS	Instalight DALI	Instalight IP	Instalight DMX-512
Назначение	Управление светом во множестве комнат с интеграцией в систему управления зданием	Управление светом во множестве комнат	Управление светом в отдельных комнатах	Мультисистемная мультимедиа технология для сложных систем	Управление светом во множестве комнат для динамического комплексного применения
Примеры использования	Гостиницы Административные здания Производственные здания Банки Торговые центры	Предприятия малого бизнеса Магазины Квартиры	Конференц-залы и офисы	Иллюминация фасадов зданий светодиодами панелями, аудиовизуальные и видеоэффекты	Иллюминация фасадов зданий сложными светодиодами приборами каскадная технология
Базовая система	Instabus KNX/EIB	Istafunk	DALI	IP-протокол	DMX-512
Передача данных	Стандартная KNX/EIB	Беспроводная, на основе техники радиосвязи	2-проводная линия	Кабель CAT 5 или CAT7 Оптический кабель Радиосвязь (WLAN)	3-проводная линия
Программирование	Программируемые компоненты	Не требует программного оборудования	Необходимо программное оборудование	Необходимы программирование и начальные установки	Ручное настраивание адреса, предварительное программирование по требованию заказчика
Преимущества	Удобство использования Прямая интеграция	Простота монтажа и модификации	Простой способ подключения	Высокая гибкость и производительность Интеграция общих	Интеграция с DMX-передатчи

	в техническую систему здания		и выбора групп светильников	IP-компонентов Использование высоконадежной серверной технологии	актуаторов Интеграция светодиодного применения DMX-технологии

Среди этих технологий особенно интересна Instalight PLUS — управление освещением с помощью техники радиосвязи. Система радиосвязи Istafunk работает на частоте 433 МГц с использованием узкой полосы частот. Разделение каналов осуществляется путем присвоения каждому передатчику индивидуального номера (до 16-ти млн возможных комбинаций). Каждый передатчик может посылать сообщения группе приемников, каждый приемник может быть настроен на прием сигналов от 30-ти передатчиков. Повышению надежности передачи телеграмм в системе способствует многократное дублирование передачи. Дальность действия передатчика (без ретрансляции) составляет 100 м. При этом мощность его радиоизлучения в 100 тыс. раз меньше, чем у сотового телефона, что допускает безопасное использование системы в жилых помещениях.

Номенклатура продукции систем освещения включает сенсоры (в т.ч. выключатели и потенциометры регулирования уровня света), диммеры и балласты для разных типов ламп и светодиодов, сетевые устройства, источники питания, контроллеры, программное обеспечение, установочные и тестирующие инструменты и приборы.

Для построения сенсоров и актуаторов интеллектуальных систем управления электрическим освещением требуются специализированные электронные компоненты. Они занимают существенный сегмент рынка, поддерживаемый ведущими производителями интегральных микросхем.

ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ ЭЛЕКТРООСВЕЩЕНИЕМ

Примеры предлагаемых технических решений и компонентов для управления электроосвещением трех ведущих производителей интегральных схем Freescale Semiconductor, Microchip Technology и STMicroelectronics представлены в таблице 3.

Таблица 3. Технические решения, интегральные микросхемы и программное обеспечение для систем управления освещением

Компания	Продукты				
	Техническое решение	Базовый микроконтроллер	Программное обеспечение	Специальные компоненты	Источники питания
Freescale Semiconductor	Управляемый балласт для флуоресцентных ламп с коррекцией коэффициента мощности	MC68HC908LB8	Исходный код на языке С и языке ассемблера		
	Цифровой интерфейс DALI для осветительных сетей	MC68HC908KX8	Исходный код на языке С	MLD1N06CL MC74AC14	
	Недорогие беспроводные сенсоры	MC9RS08KA2		MC33493	
Microchip Technology	Управляемый балласт DALI	PIC16F628P		L6561D IR2159	
	Использование микроконтроллера PIC для связи по линии DMX-512	PIC18F24J10			
	Интеллектуальные решения для управления освещением	PIC10F PIC12F PIC16F		MCP1630 MCP9700	
STMicroelectronics	Использование микроконтроллера ST7ULTRALITE для управления семистором	ST7FLITEUS5B6	Исходный код на языке С		
	Коррекция коэффициента мощности и управление балластами			L6585D	

В частности, Freescale Semiconductor предлагает технологию построения управляемого балласта для флуоресцентных ламп с коррекцией коэффициента мощности [8] на базе 8-битного 20-выводного микроконтроллера (МК) MC68HC908LB8 (128 байт оперативной памяти и 8 Кбайт программной). Особенностью МК является наличие двухканального ШИМ высокого разрешения для управления силовыми ключами. Другое предлагаемое техническое решение — цифровой интерфейс DALI для осветительных сетей [9] — базируется на использовании того же МК, а также силового транзистора MLD1N06CL и инвертора MC74AC14.

Определенный интерес для проектировщиков систем электрического освещения представляет базовый проект беспроводных сенсоров [10]. Фирма предлагает использовать миниатюрный и дешевый 8-битный МК MC9RS08KA2, имеющий всего 6 внешних выводов. На кристалле МК, кроме процессора RS08, размещены 63 байта оперативной памяти, 2 Кбайта памяти программ, таймер, аналоговый компаратор и другие узлы. МК имеет всего 4 линии ввода/вывода. Большим преимуществом этого МК является наличие встроенной защиты в случае сбоя управления или обнаружения некорректного кода операции, адреса и т.п. Для беспроводной передачи данных в проекте используется передатчик той же фирмы — Tango MC33493, работающий на частоте 315...434 МГц или 868 МГц.

Microchip Technology предлагает базовый проект управляемого балласта, принимающего команды по интерфейсу DALI [11], на базе 8-битного 20-выводного МК PIC16F628P, микросхемы коррекции коэффициента мощности L6561D фирмы STMicroelectronics и микросхемы управляемого балласта IR2159 компании International Rectifier. Самое прямое отношение к системам освещения имеет техническое решение Microchip Technology, связанное с программной реализацией интерфейса DMX-512 на 8-битном МК PIC18F24J10 [12]. Заслуживает внимания также выпущенный фирмой сборник технических решений для интеллектуальной светодиодной светотехники [13], где анонсированы новейшие компоненты, такие как 6-выводной МК PIC10F для управления диммерами, MCP1630, преобразователь DC/DC для управления светодиодными панелями, датчик температуры для мощных осветительных светодиодов MCP9700 и др. новейшие компоненты.

Фирма STMicroelectronics также предлагает технические решения и компоненты для систем электроосвещения. Техническое решение для диммера [14] базируется на использовании малогабаритного 8-выводного 8-битного МК ST7FLITEUS5B6. Фирма производит большое количество современных интегральных микросхем для управления различными типами ламп и светодиодов, в т.ч. многофункциональных. Например, 20-выводная микросхема L6585D [15] обеспечивает коррекцию коэффициента мощности и прямое управление высоковольтными флуоресцентными лампами. Анализируя данные таблицы 3, можно заключить, что главным компонентом интеллектуальных систем управления электроосвещением являются маловыводные (6—20) малогабаритные 8-битные МК, которые выполняют все необходимые функции начиная от протоколов связи и заканчивая генерированием ШИМ-сигналов для силовых драйверов, тем самым обеспечивая энергетическую эффективность системы освещения.

ОСНОВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ ТЕХНОЛОГИЙ УПРАВЛЕНИЯ ЭЛЕКТРИЧЕСКИМ ОСВЕЩЕНИЕМ

Охрана окружающей среды является главным критерием совершенствования систем управления электрическим освещением. Загрязнение окружающей среды и угроза глобального потепления в немалой степени связаны с недостаточной эффективностью систем электрического освещения. Повысить ее можно, снизив энергетические потери на всем пути от источника электроэнергии до конечного преобразования электрической энергии в световое излучение. Исходя из этого, можно сформулировать следующие направления развития технологий управления освещением:

- совершенствование полупроводниковых осветительных приборов;
- разработка новых, более эффективных элементов систем автоматического управления электроосвещением (сенсоров, микроконтроллеров, элементов силовой электроники, источников питания и т.п.);
- создание систем аккумуляторов света, способных накапливать энергию во время светового дня, и отдавать ее для освещения в ночные часы.

Несмотря на кажущуюся фантастичность последнего направления, зачатки подобных систем просматриваются в проектах «Зеленый свет» [16]. Такие системы пока еще достаточно громоздки и дороги. Однако другие примеры (сотовый телефон, компьютер и т.п.) подсказывают нам, что уже недалек тот день, когда световой аккумулятор будет в каждом доме и автомобиле.

ЛИТЕРАТУРА

1. DiLouie C., Daylighting is the use of daylight as a primary source of general illumination in a space: Part 1//Automated Buildings, August 2007. (www.automatedbuildings.com).
2. Как сделать ваш дом по настоящему «умным». Merten. (www.merten.de).
3. «Умный дом»: идеология или технология?//www.intelkey.ru/system/for/articles/intel-house.htm.
4. Control Protocols for Lighting Systems// <http://rocketsciencecanada.com/>.
5. <http://www.smarthomeusa.com/info/>.
6. EIBA Handbook Series. Release 3.0. Volume 1: Primer. Part 2: Introduction to the System, 19.04.1999. (www.knx.org).
7. <http://www.insta.de/>.
8. Dimmable Light Ballast with Power Factor Correction. Designer Reference Manual. Freescale Semiconductor, Inc., 2005. (www.freescale.com).
9. Digitally Addressable Lighting Interface (DALI) Unit Using the MC68HC908KX8. Designer Reference Manual. Freescale Semiconductor, Inc., 2002. (www.freescale.com).
10. Low Cost Wireless Sensors. Designer Reference Manual. Freescale Semiconductor, Inc., 2007. (www.freescale.com).
11. Digitally Addressable DALI Dimming Ballast. Application Note AN 809. Microchip Technology, Inc., 2002. (www.microchip.com).
12. Using a PIC® Microcontroller for DMX512 Communication. Application Note AN 1076. Microchip Technology, Inc., 2007. (www.microchip.com).

13. Adding Intelligence to Lighting Applications. LED Lighting Design Guide. Microchip Technology, Inc., 2008. (www.microchip.com).
14. Using an ST7ULTRALITE Microcontroller to Drive a TRIAC or an AC Switch for a Mains Supply. Application Note AN2425. STMicroelectronics, 2006. (www.st.com).
15. L6585D — Combo IC for PFC and ballast control. Data Sheet. STMicroelectronics, 2006. (www.st.com).
16. Sampath R. Green Lighting: Solar-based HBLED Lighting Solutions//Beyond Bits, Issue 3, 2008, pp. 10—12. (www.freescale.com).

