
В. М. ЛОПАТИН

ИНФОРМАТИКА ДЛЯ ИНЖЕНЕРОВ



Учебное пособие

Издание второе, стереотипное



САНКТ-ПЕТЕРБУРГ
МОСКВА
КРАСНОДАР
2021

УДК 004
ББК 32.973-018.2я73

Л 77 **Лопатин В. М.** Информатика для инженеров: учебное пособие для вузов / В. М. Лопатин. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 172 с. : ил. — Текст : непосредственный.

ISBN 978-5-8114-8614-4

В компактной и доступной форме представлен общий курс дисциплины «Информатика», предназначенный для студентов высших учебных заведений технического профиля. Основная цель издания — внесение вклада в решение актуальной задачи по базовой подготовке технических специалистов.

В издание включены основные разделы общего курса: системы кодирования данных, элементы и операции в алгебре логики, архитектура вычислительных систем, базовая конфигурация и программное обеспечение компьютера, методы моделирования и построения алгоритмов, системы и технологии программирования, компьютерные сети. Приведены также описания и примеры использования прикладных программ, которые дают читателю полезную информацию из разных областей знания и позволяют оценить значение информатики в практической деятельности.

УДК 004
ББК 32.973-018.2я73



Обложка
Е. А. ВЛАСОВА

© Издательство «Лань», 2021
© В. М. Лопатин, 2021
© Издательство «Лань»,
художественное оформление, 2021

Введение

Информатика относится к общетехническим дисциплинам и входит в систему базовых технических знаний, которые активно используются при обучении в высшем учебном заведении и в дальнейшем в процессе всей трудовой деятельности. Знания по информатике образуют научный базис, который является основой для разработки аппаратных и программных средств, а также для всестороннего развития цифровых технологий и их использования в различных секторах экономики.

Информатика как учебная дисциплина характеризуется довольно быстрой сменой устаревающих положений новыми научными и практическими достижениями, поэтому отличается ускоренным пополнением учебного материала, а также периодическим внесением изменений в учебные программы и учебную литературу. Учебный материал настоящего издания также составлен с учетом факторов современного развития информационных технологий. В процессе изложения материала уровень современных информационных технологий рассматривается в аспекте исторического развития основных понятий науки об информации и компьютерной технике. В целом предлагаемое издание можно рассматривать как очередной шаг к формированию современного быстроменяющегося учебного курса, направленного на повышение уровня подготовки студентов по информатике, включая приобретение навыков обращения к компьютеру при решении производственных и социальных задач.

Материал издания сформирован как последовательность основных разделов дисциплины. Тематика разделов посвящена вопросам представления информации в информационно-вычислительной технике, изложению основ аппаратного и программного обеспечения компьютера, описанию широко распространенных операционных систем и программных продуктов, правилам и принципам устройства компьютерных сетей и методам их защиты. Каждый из разделов структурирован по тематическим вопросам, основные разделы дополнены примерами использования компьютера при решении практических задач. Для облегчения восприятия и запоминания в тексте выделены основные определения и термины. Предложенная форма изложения упрощает процесс подготовки студентов к зачетам и экзаменам, а при необходимости может использоваться для оперативного восстановления знаний в области информатики и компьютерной техники.



Информатика – наука об автоматической обработке информации

Термин «информатика» связан с двумя французскими словами: *information* (информация) и *automatique* (автоматика). Объединение и совместное толкование этих слов позволяет определить информатику как науку об автоматической обработке информации. Это определение полностью соответствовало уровню развития науки в середине прошлого века. Со временем по мере совершенствования вычислительной техники и увеличения ее вклада в промышленную, административную и социальную сферу, первоначальное определение уточнялось и дополнялось.

Одно из современных определений может быть представлено через отношение информатики к техническим средствам [1].

Информатика – техническая наука, систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.

Другое определение отражает тесную связь информатики с понятием «информация» [2].

Информатика – это наука, изучающая структуру и наиболее общие свойства информации, ее поиск, хранение, передачу и обработку с применением ЭВМ.

Каждое из приведенных определений не дает полного представления о содержании информатики, а их совокупное рассмотрение позволяет связать информатику с двумя основными понятиями – «информация» и «вычислительная техника». Поскольку современные аппаратные средства наряду с вычислениями выполняют большой объем работ по информационному обеспечению, то понятие вычислительной техники можно заменить информационно-вычислительной техникой. С учетом этого дополнения определение информатики можно представить в простой форме.

Информатика – наука об информации и способах ее обработки с помощью информационно-вычислительных средств.

На практике термин «информатика» зачастую подменяют термином «информационные технологии», поскольку информатика тесно связана со способами обработки информации, с совокупностью операций, с технологиями.

Информационная технология (ИТ) – процесс, использующий совокупность средств и методов сбора, обработки и передачи данных для получения информации нового качества о состоянии объекта, процесса или явления [3].

К объектам, изучаемым информатикой, относятся все средства, которые задействованы в процессе автоматической или автоматизированной обработки данных (рис. 1):

- аппаратное обеспечение – совокупность электронных и механических составляющих информационно-вычислительной техники;
- программное обеспечение – совокупность программ, обеспечивающих функционирование аппаратной части и решение задач пользователя;

- средства взаимодействия программного и аппаратного обеспечения (программно-аппаратный интерфейс);
- средства взаимодействия пользователя с аппаратным и программным обеспечением (интерфейс пользователя).

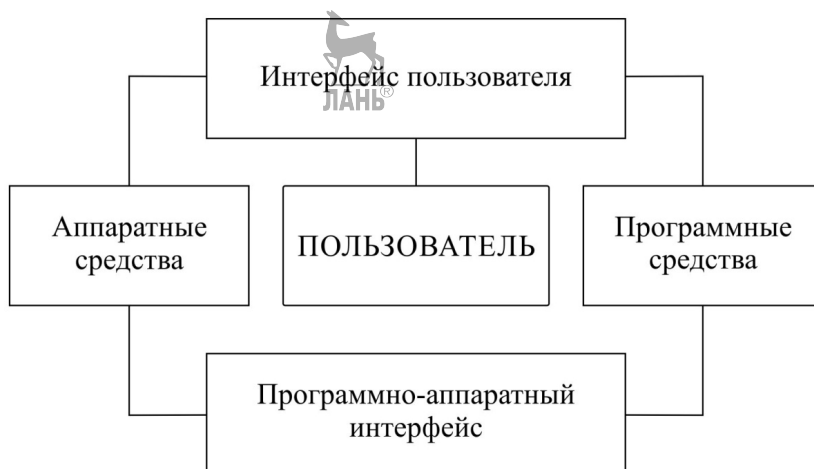


Рис. 1. Схема взаимодействия пользователя с программно-аппаратными средствами

Основная цель науки информатики заключается в поиске, развитии и внедрении автоматических способов обработки информации в технической, экономической, образовательной, общественной и других сферах деятельности. Поставленная цель достигается путем решения ряда взаимосвязанных задач по различным направлениям деятельности, в том числе:

- создание математических моделей и алгоритмов;
- создание эффективных программных решений;
- совершенствование технических систем управления;
- разработка аппаратных средств на новых физических принципах;
- разработка социально значимых систем управления;
- создание систем искусственного интеллекта и т. д.

Результаты развития каждого направления проходят проверку на практике, которая позволяет выделить наиболее эффективные решения и определить основной вектор развития по каждому из направлений. Основным критерием выбора является повышение эффективности новых решений и предпочтения общества пользователей. Основным результатом развития является создание и внедрение современных *информационных систем*.

Информационная система – система, предназначенная для хранения, обработки, поиска, распространения, передачи и предоставления информации [4].

К наиболее известным результатам, полученным на основе достижений в области информатики и информационных технологий, которые признаны и активно поддержаны во всех странах, относятся:

- создание глобальной компьютерной сети Интернет;
- создание глобальной сети мобильной телефонной связи, совмещенной с доступом к информационным ресурсам сети Интернет;
- создание и развитие системы социально значимых услуг, в том числе электронных продаж, социальных сетей, образовательных систем и др.;
- внедрение локального и глобального контроля в системах жизнеобеспечения, транспортных системах, системах навигации и т. д.;
- постепенный перевод производственной и социальной сферы на компьютерные системы управления и многое другое.

С технической точки зрения все современные информационные системы имеют общие принципы функционирования.

1. В системе осуществляется постоянный сбор данных и контроль характера изменения наблюдаемых данных.

2. Все изменения входных данных автоматически или с участием человека отслеживаются, учитываются и обрабатываются на основе компьютерных программ, которые заложены в систему.

3. На основе обработки поступающих данных программы формируют управляющие сигналы или выдают рекомендации, которые используются для корректировки в автоматизированном или ручном режиме.

Результатом работы любой информационной системы является выдача информации, которая необходима для управления процессом или принятия решения.



Информация и ее представление в информационно-вычислительной технике

Жизнь современного человека, общества, государства наполнена информацией. Информационные потоки окружают нас и регулируют нашу жизнь, многие виды общественной деятельности невозможны без постоянно обновляемой информации.

В окружающих нас неупорядоченных, на первый взгляд, потоках информации можно отметить некоторые закономерности. В частности, разделить всю информацию на отдельные категории.

Категорию простой информации составляют сообщения, поступающие от отправителя к получателю в форме некоторого физического сигнала, который выделяется на фоне шумового информационного потока, например телефонное сообщение или световой сигнал светофора.

Другая категория информации представляется в форме некоторого отображения объекта, формируемого в сознании наблюдателя. Для этой категории наличие отправителя необязательно, важно наличие наблюдателя, который может сформировать отображение, например представить окружающий ландшафт на основе изучения карты местности.

Выделяется также категория информации, которая формируется внутри одного объекта, переносится на другой объект неосознанно и влияет на развитие этого объекта без осмысления происходящего. К категории такой информации относятся, например, наследственные признаки.

Наличие категорий свидетельствует о широком спектре информационного потока и существенных различиях между составляющими этого потока. По этой причине не всегда удастся понять и определить, в чем заключается суть информации.

Определения и основные свойства информации

Энциклопедический словарь [5] дает следующее определение термину «информация».

Информация (от лат. *informatio* – осведомление, разъяснение, изложение) – первоначально сведения, передаваемые людьми устным, письменным или другим способом (с помощью условных сигналов, технических средств и т. д.). С середины XX в. в общенаучное понятие информации входят обмен сведениями между людьми, человеком и автоматом, автоматом и автоматом; обмен сигналами в животном и растительном мире; передача признаков от клетки к клетке, от организма к организму. Информация – одно из основных понятий *кибернетики*.

Кибернетика – наука о системах и методах управления, т. е. об организации и реализации целенаправленных действий в машинах, живых организмах и обществе. Кибернетика занимается общими законами преобразования информации в сложных управляющих системах [6]. Основоположник кибернетики – американский математик и философ Норберт Винер.

Информация относится к абстрактному понятию, которое имеет множество значений в зависимости от контекста. В узком смысле этого слова – сведения (сообщения, данные) независимо от формы их представления. В широком и более общем смысле не выработано единого определения этого термина. С точки зрения различных областей знания данное понятие описывается своим специфическим набором признаков.

Рассмотрим некоторые определения информации, которые используются в технической литературе.

1. Информация – специфический атрибут реального мира, представляющий собой его реальное отражение в виде совокупности сигналов и проявляющийся при взаимодействии с «приемником» информации, позволяющим выделять, регистрировать эти сигналы из окружающего мира и по тому или иному критерию их идентифицировать [6].

2. Информация – это сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления, предоставляемые получателю или оцениваемые им с целью выполнения действий, осмысления значения на основе обработки и анализа данных [7].

3. Информация – это продукт взаимодействия данных и адекватных им методов. Информация возникает в процессе взаимодействия данных и соответствующих методов [1].

Перечень аналогичных определений можно продолжить и дополнить список примерно сотней формулировок, собранных из разных источников и приведенных в [8]. Многообразие формулировок свидетельствует о том, что единого подхода к определению понятия «информация» пока не выработано, но каждое из определений дополняет и уточняет наши знания.

Для получения более полного представления об информации можно рассмотреть различные свойства информации.

Прагматические свойства информации, которые проявляются в процессе использования и характеризуют отношение информации к потребителю, выражаются в следующем.

1. Объективность информации – независимость от методов фиксации, от субъективного мнения или суждения. Поскольку субъективный вклад всегда в той или иной мере присутствует, то объективность информации следует считать величиной относительной. Более объективной считается информация, которая использует минимум субъективных показателей. Видеоинформация будет более объективной в сравнении с рассказом очевидца.

2. Достоверность информации – отражение истинного положения дел, минимальное число ошибок при передаче данных. Степень достоверности зависит от соотношения полезного сигнала к сопровождающему его информационному шуму. Достоверность информации увеличивается при увеличении соотношения полезного сигнала к уровню шума и наоборот. Для повышения достоверности необходимо принимать меры по очистке от информационного шума.

3. Адекватность информации – уровень соответствия полученной информации реальному состоянию объекта. Степень адекватности зависит от досто-

верности и полноты данных. Недостоверные и неполные данные увеличивают неадекватную составляющую информации.

4. Полнота информации – достаточность для понимания и принятия решений или для создания новых данных. Существующую информацию зачастую можно дополнить новыми данными, которые изменят принятое решение, поэтому полнота информации всегда относительна. Для получения полной информации следует пользоваться различными методами и выбирать те из них, которые обеспечивают достаточность.

5. Доступность информации – возможность получения информации пользователем, которая зависит как от наличия данных, так и от присутствия адекватного метода для обработки этих данных. Присутствие данных, которые невозможно изучить по причине незнания адекватного метода обработки, закрывает доступ к информации. Неадекватные методы в свою очередь приводят к получению недостоверной информации.

6. Точность информации – степень близости данных к реальному состоянию объекта, процесса, явления. Точность связана с техническими данными и зависит от погрешности приборов, которые используются для определения реального состояния объекта.

7. Актуальность информации – важность для настоящего момента времени или насущность. Актуальность информации связана с продолжительностью ее доставки. В случае задержки с доставкой информация становится неактуальной или даже ненужной.

8. Релевантность информации – степень соответствия запросам потребителя. Термин является одним из основных показателей работы автоматических поисковых систем. Релевантность информации повышается, если по запросу потребителя поисковая система выдает ожидаемые варианты ответов.

9. Репрезентативность информации – способность к отражению адекватного состояния объекта посредством оценки ограниченного объема данных. Применяется при статистической обработке данных, отражает соответствие характеристик выборки и полного массива данных.

10. Эргономичность информации – наглядность и удобство в пользовании с точки зрения потребителя. Эргономичность определяется разными параметрами представляемых данных: формат, объем, скорость и т. д.

11. Полезность (ценность) информации определяет степень важности для применения в практической деятельности. Полезность может быть оценена применительно к нуждам потребителей или по отношению к результатам решения конкретных задач. Полезность информации оценивается также по ее отношению к совокупности сведений, которыми владеет пользователь. Совокупность сведений, которыми располагает пользователь, называется *тезаурусом пользователя*.

Атрибутивные свойства информации объединяют характеристики, без которых она не может существовать.

1. Неотрывность от источника информации. Информация возникает из первоисточника, без которого она существовать не может. Со временем связь с пер-

воисточником может быть утеряна, но это не значит, что его совсем не было. Восстановление первоисточника иногда бывает непростой задачей.

2. Неотрывность от физического носителя. Информация всегда где-то хранится и не может существовать без привязки к носителю. Одна и та же информация может перемещаться между носителями, но всегда хранится хотя бы на одном из них. Отсутствие конкретной информации на всех возможных носителях означает, что такая информация не существует.

3. Непрерывность информации характеризуется потоковым распространением и постепенным накоплением данных на носителях. Непрерывность информации лежит в основе постоянного развития.

4. Дискретность информации отражает ее способность к распространению отдельными блоками, каждый из которых связан с предыдущим и дополняет его по мере поступления. Каждый из блоков несет дозированную информацию об ограниченном наборе свойств. Совокупность блоков расширяет наши представления об объекте.

Динамические свойства отражают процессы изменения информации во времени.

1. Рост информации проявляется в многократном повторении различными источниками, при повторении информация полностью остается у передающей стороны, поэтому рост сопровождается быстрым увеличением числа источников информации.

2. Старение информации связано с потерей ее актуальности, с уменьшением ценности со временем; причиной старения является новая информация, которая частично или полностью опровергает предыдущую.

Большой вклад в развитие теории информации внес американский математик и инженер Клод Шеннон. В середине прошлого века он предложил связать информацию с понятием «информационная энтропия». В подходе, предложенном Шенноном, информация определяется как снятая или уменьшенная неопределенность, при этом процесс получения информации характеризуется уменьшением неопределенности. Снятие неопределенности можно рассматривать как уменьшение числа возможных вариантов в процессе поиска наиболее подходящего решения.

Информационная энтропия используется для количественной оценки неопределенности. Максимальное значение энтропии соответствует полной неопределенности, которая характеризуется множеством возможных вариантов решения, ни один из которых не может быть подходящим. Напротив, минимальное значение энтропии означает, что неопределенность полностью исчезла, поскольку вся информация получена и найдено верное решение. Процесс поиска решения в этой ситуации сопровождается уменьшением энтропии, с одной стороны, и, с другой стороны, получением информации, которая требуется для снятия неопределенности и уменьшения энтропии.

Наряду с термином «информация» в информатике часто используют понятие «данные». Данные в отличие от информации появляются в результате энергетического обмена между объектами, который сопровождается изменением свойств

объектов. Следствием изменения свойств объекта является регистрация сигналов, а некоторое множество зарегистрированных сигналов объединяется понятием «данные».

Данные – зарегистрированные в определенном виде сигналы, поступающие из окружающей среды в форме сведений об измерении, наблюдений о событиях, математических выражений, которые пригодны для хранения, переработки и передачи.

Как правило, данные фиксируются на носителе в удобной форме, которая позволяет хранить зарегистрированные сигналы и обрабатывать их при необходимости или передавать на другой носитель. В процессе хранения данные остаются без движения и не являются информацией, но по мере преобразования и передачи могут привести к получению информации и последующему действию, связанному с полученной информацией. Другими словами, преобразование и обработка данных позволяют получить информацию.

С понятием «информация» связан ряд терминов.

Сигнал – это закодированное представление данных, которое используется для их передачи с помощью изменяющегося во времени физического процесса.

Сообщение – это совокупность данных, которая в результате преобразования и передачи обретает смысловую окраску.

Канал связи – это среда для распространения сигналов в совокупности с техническими средствами, которые используются для передачи сообщения от источника к приемнику.

Носитель информации – это некоторая среда или объект, в структуре которого сохраняются данные в кодированном виде.

Накопитель информации – это устройство для долговременного хранения больших объемов данных.

В технике для формирования и передачи сигналов используются электронные устройства, которые допускают работу с сигналами в двух формах: *аналоговой* и *цифровой*.

Аналоговый сигнал описывается непрерывной функцией, зависящей от времени. Передача аналогового сигнала обеспечивается за счет плавного изменения одного или нескольких параметров сигнала: амплитуды, частоты, фазы. Аналоговый сигнал можно описывать с помощью математической зависимости. На рис. 2 представлен фрагмент гармоничного аналогового сигнала, который описывается математической зависимостью

$$A(t) = A_1 + A_0 \sin(\omega t + \varphi_0),$$

где A_1 – постоянная составляющая; A_0 – амплитуда сигнала; ω и φ_0 – частота и начальная фаза.

Наши органы чувств настроены на прием аналоговых сигналов. Звук мы воспринимаем как колебания воздушной среды с определенной частотой и амплитудой. Нахождение и прием аналогового сигнала сильное влияние могут оказывать *помехи*.



Рис. 2. Аналоговый гармонический сигнал

Цифровая форма сигнала основана на представлении аналогового сигнала в виде последовательности чисел заданной разрядности. Для перевода сигнала в цифровую форму выполняют дискретизацию – разбиение временного промежутка следования сигнала на короткие временные интервалы, а затем осуществляют квантование – деление максимальной амплитуды сигнала на определенное число уровней. В результате на каждом коротком интервале времени можно определить приближенное или квантованное значение сигнала и записать это значение в численной форме. Если полученную последовательность чисел перевести в двоичную систему счисления, то сигнал преобразуется в двоичный код – последовательность нулей и единиц. На графике перевод аналогового сигнала в цифровой выглядит как замена непрерывной кривой на ступенчатую линию (рис. 3). Ширина ступени при этом зависит от степени дискретизации, а высота определяется числом уровней квантования.

Процесс преобразования аналогового сигнала в цифровой называется оцифровкой. Устройство, которое преобразует аналоговый сигнал в цифровой, называется аналого-цифровым преобразователем (АЦП), для обратного преобразования используется цифро-аналоговый преобразователь (ЦАП). Точность оцифровки зависит от количества уровней квантования и интервала дискретизации. Минимальное изменение величины аналогового сигнала, которое может быть зарегистрировано АЦП, связано с его разрядностью. Разрядность характеризует количество дискретных значений, которые преобразователь обеспечивает на выходе.

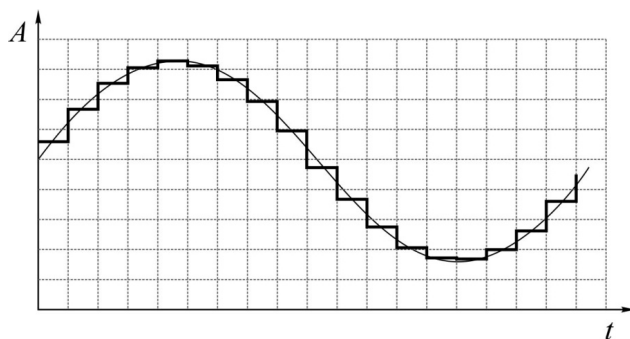


Рис. 3. Оцифрованный гармонический сигнал

Вся компьютерная техника работает с цифровыми сигналами, представленными в двоичном коде. Такое представление обеспечивает надежную помехозащищенность, высокую скорость передачи больших объемов данных, возможность компактного хранения и др. Цифровое формирование сигнала позволяет представить в единой форме (в двоичном коде) числовые, текстовые, графические, видео-, звуковые и другие данные. Это означает, что всю совокупность цифровых данных можно хранить на одном носителе, передавать по одному каналу связи, использовать одни и те же устройства для просмотра, копирования, обработки, удаления данных.

Единицы измерения и хранения данных

Для измерения количества информации используют в основном два подхода.

1. Символьный (алфавитный) подход основан на делении информационного блока на простые информационные элементы с последующим подсчетом числа этих элементов. В случае текстового блока простым информационным элементом выбирают символ и для каждого символа используют 1 *байт* памяти. При записи в памяти каждый байт регистрируется в виде двоичного восьмиразрядного кода. Количество информации измеряют при этом простым подсчетом числа символов и выражают его в количестве байтов. Использование числа символов для определения количества информации в тексте лежит в основе алфавитного подхода.

Алфавитный подход основан на том, что всякое сообщение можно закодировать с помощью конечной последовательности символов некоторого алфавита. Если допустить, что все символы алфавита встречаются в тексте с одинаковой частотой (равновероятно), то количество информации, которое несет каждый символ (информационный вес одного символа), вычисляется по формуле

$$I = \log_2 N, \quad (1)$$

где N – мощность алфавита (полное количество символов, составляющих алфавит выбранного кодирования). Из (1) следует, что мощность алфавита можно вычислить по формуле

$$N = 2^I. \quad (2)$$

В алфавите, который состоит из двух символов (двоичное кодирование), каждый символ несет 1 бит ($\log_2 2 = 1$) информации. Если мощность алфавита составляет четыре символа, то каждый символ несет 2 бита информации ($\log_2 4 = 2$); в алфавите из восьми символов – 3 бита ($\log_2 8 = 3$) и т. д. Один символ из алфавита мощностью 256 несет в тексте 8 бит ($\log_2 256 = 8$) информации.

Если весь текст состоит из k символов, то при алфавитном подходе размер содержащейся в нем информации H определяется по формуле

$$H = k \cdot I, \quad (3)$$

где I – информационный вес одного символа в используемом алфавите.

Максимальное количество слов L из m букв, которое можно составить из алфавита мощностью N , определяется по формуле

$$L = N^m. \quad (4)$$

При увеличении объемов информации используют единицу измерения байт с приставками кило-, мега-, гига- и т. д. Соотношения между единицами измерения количества информации представлены в табл. 1.

Таблица 1

Единицы измерения количества информации

Наименование единицы	Форма записи	Кол-во единиц	Кол-во байтов	Кол-во бит
Килобайт	Кбайт	1024 байт	2^{10}	2^{13}
Мегабайт	Мбайт	1024 Кбайт	2^{20}	2^{23}
Гигабайт	Гбайт	1024 Мбайт	2^{30}	2^{33}
Терабайт	Тбайт	1024 Гбайт	2^{40}	2^{43}
Петабайт	Пбайт	1024 Тбайт	2^{50}	2^{53}
Экзабайт	Эбайт	1024 Пбайт	2^{60}	2^{63}

При переходе от одной единицы измерения к другой используется два простых соотношения:

$$1 \text{ байт} = 8 \text{ бит}; \quad 1 \text{ Кбайт} = 1024 \text{ байт}.$$

Используя данные, приведенные в табл. 1, можно выполнять перевод из одних единиц измерения в другие. Например, для перевода величины 100 Мбайт в биты необходимо выполнить следующие преобразования

$$100 \text{ Мбайт} = 100 \cdot 1024 \cdot 1024 \text{ байт} = 100 \cdot 1024 \cdot 1024 \cdot 8 \text{ бит}.$$

Символьный подход чаще всего применяется для оценки емкости памяти запоминающих устройств, объема файлов или объема передаваемых сообщений. При этом содержание самой информации не учитывается и не принимается во внимание.

Пример. Рассчитаем мощность алфавита, с помощью которого записано сообщение, содержащее 8192 символа и занимающее объем 1/256 Мбайт.

Исходные данные:

$$k = 8192,$$

$$H = 1/256 \text{ Мбайт}.$$

Найти мощность алфавита N .

Решение:

Мощность алфавита находится по формуле $N = 2^I$, где I – информационный вес одного символа.

Информационный вес символа I связан с общим объемом информации формулой $I = H/k$.

Определяем значение I , соблюдая размерность исходных данных:

$$I = H/k = 1024 \cdot 1024 \cdot 8 / 256 \cdot 8192 = 2^{10} \cdot 2^{10} \cdot 2^3 / 2^8 \cdot 2^{13} = 2^2 = 4 \text{ бит.}$$

Определяем $N = 2^4 = 16$ символов.

Ответ: Мощность алфавита составляет 16 символов.

2. Содержательный (вероятностный) подход, в котором учитывается содержание информации, а точнее – вероятность реализации события. При этом более информативным считается то событие, которое менее вероятно, т. е. менее всего ожидалось. В содержательном подходе для подсчета количества информации I используют формулу Шеннона:

$$I = -\log_2 P, \quad (5)$$

где P – вероятность реализации события.

Так, при случайном доставании белого шара из коробки, в которой находятся 6 черных и 2 белых шара, количество информации I , соответствующей этому событию, равно

$$I = -\log_2 2 / 8 = \log_2 4 = 2 \text{ бит.}$$

Единицей измерения информации, рассчитанной по формуле (5), является *бит*. 1 бит соответствует ситуации, при которой возможен исход двух равновероятных событий, т. е. $P = 1/2$. Единица измерения «бит», или «bit», получила свое название от английского словосочетания «binary digit» – «двойная цифра». В памяти компьютера на физическом уровне бит представлен одной ячейкой, которая может находиться в одном из двух возможных состояний, первое из которых соответствует единице, второе – нулю.

Наряду с формулой (5), в которой используется логарифм по основанию 2, а единицей измерения служит *бит*, могут применяться другие основания алгоритма и другие единицы измерения. Десятичному алгоритму соответствует единица измерения *дит* (за единицу измерения принимается такое количество информации, которое необходимо для исхода десяти равновероятных событий), а натуральному алгоритму – единица измерения *нат*.

Формула (5) предназначена для оценки информативности одиночного события. Для оценки информативности нескольких событий, имеющих разную вероятность, формула Шеннона преобразуется к виду

$$I = -\sum_{i=1}^n P_i \log_2 P_i, \quad (6)$$

где n – количество ожидаемых событий.

Формула (6) отражает также величину изменения энтропии, которое соответствует получению информации I .

Пример. Определим количество информации, связанное с появлением каждого символа в сообщениях, записанных на русском языке при условии: а) появление каждого из 34 символов равновероятно; б) символы в тексте встречаются с разной вероятностью p_i (в соответствии с табл. 2).

Таблица 2

Вероятность появления символов в текстах на русском языке

i	Символ	p_i	i	Символ	p_i	i	Символ	p_i
1	Пробел	0,175	13	К	0,028	24	Г	0,012
2	О	0,090	14	М	0,026	25	Ч	0,012
3	Е	0,072	15	Д	0,025	26	И	0,010
4	Ё	0,072	16	П	0,023	27	Х	0,009
5	А	0,062	17	У	0,021	28	Ж	0,007
6	И	0,062	18	Я	0,018	29	Ю	0,006
7	Т	0,053	19	Ы	0,016	30	Ш	0,006
8	Н	0,053	20	З	0,016	31	Ц	0,004
9	С	0,045	21	Ь	0,014	32	Щ	0,003
10	Р	0,040	22	Ъ	0,014	33	Э	0,003
11	В	0,038	23	Б	0,014	34	Ф	0,002
12	Л	0,035						

а) При условии равновероятного появления каждый символ несет информацию:

$$I_a = \log_2 N = \log_2 34 = 5,09;$$

б) с учетом разной вероятности имеем

$$I_{\sigma} = -\sum_{i=1}^n P_i \log_2 P_i = -\sum_{i=1}^{34} P_i \cdot \log_2 P_i = -(-4,71) = 4,71.$$

Округляем в большую сторону для равновероятного случая и получаем $I_a = 6$ бит, а при условии разной вероятности $I_{\sigma} = 5$ бит. Это значит, что количество информации для каждого символа в тексте на русском языке равно 5 бит, а при упрощенном равновероятном условии – на 1 бит больше.

Если события равновероятны и возможное число этих событий равно N , то $P = 1/N$, и формула Шеннона при этом преобразуется в формулу Хартли,

$$I = -\log_2\left(\frac{1}{N}\right) = \log_2 N. \quad (7)$$

Например, при случайном доставании туза пик из колоды, в которой находится 32 карты, количество информации I , соответствующей этому событию, равно

$$I = \log_2 32 = 5 \log_2 2 = 5 \text{ бит.}$$

Если же в колоде находится не 32, а 36 карт, то расчет по формуле (7) дает результат

$$I = \log_2 36 = 5,17,$$

который означает, что 5 бит будет недостаточно и полученный результат нужно округлить в сторону большего целого значения, т. е. $I = 6$ бит.

В содержательном подходе учитывается исключительно вероятность реализации события и при этом не имеет значения объем информации, необходимый для регистрации события или сообщения.

Преобразование и обработка данных

На практике для измерения количества информации чаще используется символичный подход, в котором наименьшей единицей измерения является байт. При этом группа байтов выстраивается по тематическому признаку и образует *файл*.

Файл — это совокупность некоторого числа байтов, которая имеет уникальное собственное имя. Обычно в отдельном файле хранят данные одного типа. Файл выступает в виде единицы хранения данных, которая подвергается операциям обработки и преобразования.

Имя файла складывается из *имени*, которое присваивает пользователь, и *расширения* файла.

Имя, которое присваивает пользователь, может иметь до 256 символов (русских или английских), расширение – 3–4 символа (английских).

Расширение файла – последовательность символов, отделяемая от имени точкой и предназначенная для идентификации типа файла. Расширение файла отражает формат файла – правила кодирования данных, которые используются в соответствующих программах. Если говорят, что данные сохраняются в некотором формате, то это значит, что имеется программа, которая умеет обрабатывать этот формат. Файлы, имеющие одинаковую информацию, могут отличаться форматом представления данных. Например, графические файлы, содержащие одинаковое изображение, в разных графических редакторах имеют разное расширение. Перевод файла из одного представления в другое называется *периформатированием*, или *конвертацией*.

Конвертация файла – преобразование с видоизменением данных, предназначенное для перехода к другим условиям обработки. Конвертация выполняется с помощью программ, которые называются *конверторами*. Иногда конвертация файла связана с потерей части данных.

Сведения о файлах собираются в *файловой структуре*.

Файловая структура построена по табличному принципу. В таблицу заносятся имена файлов и данные о размещении файлов. При размещении файлов в качестве наименьшей единицы измерения используют один *сектор*, размер которого фиксирован (обычно 512 байт). Группа секторов объединяется в *кластер*, размер которого строго не фиксирован (рис. 4). Каждый кластер имеет адрес, который используется при обращении к данным. Кластер является наименьшей единицей адресации, а данные об адресах кластеров собираются в таблицах файловой структуры [1].

Кластер – минимальный объем дискового пространства, выделяемый для сохранения файла и состоящий из одного или нескольких смежных секторов.

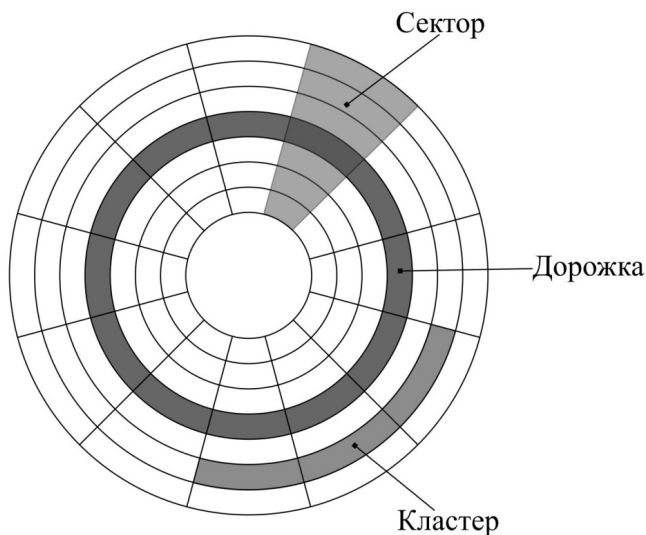


Рис. 4. Структурные элементы диска

При сохранении файла его данные записываются в один или несколько последовательно расположенных кластеров. После сохранения файла кластеры становятся занятыми, при этом кроме имени файла сохраняются сведения о его *размере, дате и времени* сохранения, а также *адрес первого кластера* файла. После каждого преобразования файла его данные перезаписываются на новом месте. Если на диске не хватает свободного неразрывного места для сохранения файла, то возможна *фрагментация* файла – разделение на части с распределением частей в несмежных кластерах.

В процессе хранения данные, которые представлены в файлах, могут быть обработаны и преобразованы. Обработка данных включает в себя набор различных операций, некоторые из них перечислены ниже.

1. Сбор данных – накопление данных с целью получения достоверной информации, достаточной для принятия решения.

2. Формализация – приведение данных из разных источников к одинаковой форме с целью сравнения и дальнейшей обработки.

3. Фильтрация – отсеивание избыточных данных или удаление информационного шума с целью повышения достоверности информации.

4. Сортировка – перегруппировка данных и их упорядочение по заданному параметру для удобства пользования и доступности.

5. Архивация – организация хранения данных в компактной и доступной форме для снижения затрат и защиты информации.

6. Защита – включает комплекс мер, направленных на предотвращение копирования, модификации или утраты данных.

7. Транспортировка – передача или прием данных, осуществляемых между отправителем и получателем, в соответствии с принятыми правилами обмена.

8. Преобразование – обработка и переформатирование данных или приведение к удобной для практического использования форме. Обработку и преобразование данных выполняют с помощью специальных программ, каждая из которых предназначена для выполнения специфического набора операций и представления обработанных данных в новом формате.



Представление числовых данных

Окружающая нас информация всегда существует в закодированной форме. Форма кодирования определяется системой, с помощью которой осуществляется представление информации. Известны следующие *системы кодирования*:

- естественные человеческие языки – кодирование понятий с помощью речи;
- азбука Морзе – кодирование алфавита последовательностью коротких и длинных сигналов;
- система Брайля – кодирование алфавита рельефно-точечным шрифтом (для незрячих и плохо видящих людей).

В общем виде можно сказать, что кодирование – это изменение типа данных или переход от одного формата к другому, более подходящему для хранения, передачи или обработки информации. В информатике и вычислительной технике существует своя система кодирования, которая называется *двоичным кодированием*. Система двоичного кодирования основана на представлении данных через двоичную систему счисления.

Системы счисления

Система счисления – способ записи чисел с помощью заданного набора специальных символов, которые называются цифрами.

Основание системы счисления – количество цифр, используемых для записи чисел.

Различают *позиционные* и *непозиционные* системы счисления.

Непозиционные системы счисления – система, в которой символы не меняют своего значения при смене их местоположения (позиции) в изображении числа.

К непозиционной системе относится римская система, используемые в ней обозначения показаны в табл. 3. Запись чисел в римской системе счисления осуществляется по простым правилам:

1) если цифра слева меньше, чем цифра справа, то левая цифра вычитается из правой (IX: $10 - 1 = 9$; XC: $100 - 10 = 90$);

Таблица 3

Непозиционная римская система счисления

Римские цифры	Значение (обозначаемое количество)	Римские цифры	Значение (обозначаемое количество)
I	1	C	100
V	5	D	500
X	10	M	1000
L	50		

- 2) если цифра справа меньше или равна цифре слева, то эти цифры складываются (VII: $5 + 1 + 1 = 7$; XXXV: $10 + 10 + 10 + 5 = 35$);
 3) нельзя записывать подряд 4 одинаковых цифры.

Пример записи числа 984 в римской системе счисления:

900 – CM, 80 – LXXX, 4 – IV, отсюда $984 = \text{CMLXXXIV}$.

В математическом аппарате информатики используются *десятичная, двоичная, восьмеричная и шестнадцатеричная* системы счисления (табл. 4). Все эти системы счисления относятся к системам позиционного типа, в которых величина, обозначаемая цифрой в записи числа, зависит от ее позиции.

Таблица 4

Позиционные системы счисления

Название системы	Основание системы	Обозначение	Цифры и символы для записи чисел
Двоичная	2	Bin	0 1
Восьмеричная	8	Oct	0 1 2 3 4 5 6 7
Десятичная	10	Dec	0 1 2 3 4 5 6 7 8 9
Шестнадцатеричная	16	Hex [®]	0 1 2 3 4 5 6 7 8 9 A B C D E F

Для перевода чисел из одной системы счисления в другую используют простые математические алгоритмы или делают перевод с помощью калькулятора. Использование калькулятора, например, позволяет представить и сравнить число R в разных системах счисления:

$$R = 1234_{10} = 10011010010_2 = 2322_8 = 4D2_{16}.$$

В этой записи подстрочный индекс используется для обозначения *основания* системы счисления. Общая формула для представления числа R в системе счисления с основанием q имеет вид:

$$R_q = \sum_{i=n}^0 a_i \cdot q^i + \sum_{j=-1}^{-k} a_j \cdot q^j, \quad (8)$$

где n – количество символов a_i с положительными показателями степеней; k – количество символов a_j с отрицательными показателями степеней.

Формула (8) позволяет представить любое число в развернутой форме записи:

$$386,15_{10} = 3 \cdot 10^2 + 8 \cdot 10^1 + 6 \cdot 10^0 + 1 \cdot 10^{-1} + 5 \cdot 10^{-2},$$

$$1111,01_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-2},$$

$$6BF,A_{16} = 6 \cdot 16^2 + B \cdot 16^1 + F \cdot 16^0 + A \cdot 16^{-1}.$$

Перевод чисел из одной системы счисления в другую

Развернутая форма записи числа используется для перевода чисел в десятичную систему счисления. Обратный перевод из десятичной системы в двоичную и другие системы осуществляется с помощью поэтапного деления числа на основание системы счисления с определением остатка от целочисленного деления. В табл. 5 в качестве примера показан процесс перевода десятичного числа 204 в двоичную, восьмеричную и шестнадцатеричную системы. Стрелкой показаны порядок считывания и записи остатков от целочисленного деления.

При переводе дробного десятичного числа в двоичную систему отдельно переводится целая и дробная часть. Целая часть переводится путем подсчета остатков поэтапного целочисленного деления на 2, а дробная – путем подсчета целых частей при поэтапном умножении на 2.

Пример перевода дробного числа в двоичную систему счисления – в табл. 6.

Таблица 5

Перевод целого числа из десятичной системы счисления

Двоичная	Восьмеричная	Шестнадцатеричная
$204 : 2 = 102 + 0$	$204 : 8 = 25 + 4$	$204 : 16 = 12 + 12$
$102 : 2 = 51 + 0$	$25 : 8 = 3 + 1$	$12 : 16 = 0 + 12$
$51 : 2 = 25 + 1$	$3 : 8 = 0 + 3$	
$25 : 2 = 12 + 1$		
$12 : 2 = 6 + 0$		
$6 : 2 = 3 + 0$		
$3 : 2 = 1 + 1$		
$1 : 2 = 0 + 1$		
$204_{10} = 11001100_2$	$204_{10} = 314_8$	$204_{10} = CC_{16}$

Для перевода чисел между системами с основанием 2, 8 или 16 используют замену двоичных цифр на их восьмеричные или шестнадцатеричные эквиваленты. Замену производят на основе данных, представленных в табл. 7.

Для перевода целого двоичного числа в восьмеричное сначала число разбивают справа налево на группы по три цифры (триады), а затем каждой группе присваивают ее восьмеричный эквивалент.

Аналогично поступают с шестнадцатеричным числом, которое разбивают на группы по четыре цифры (тетрады). Примеры перевода чисел из восьмеричной и шестнадцатеричной систем в двоичную и обратно показаны в табл. 8.

Таблица 6

Перевод дробного числа
из десятичной системы счисления

Целая часть	Дробная часть
$45 : 2 = 22 + 1$ $22 : 2 = 11 + 0$ $11 : 2 = 5 + 1$ $5 : 2 = 2 + 1$ $2 : 2 = 1 + 0$ $1 : 2 = 0 + 1$	$0,75 * 2 = 1, 5 = 1 + 0,5$ $0,5 * 2 = 1 + 0$
$45_{10} = 101101_2$	$0,75_{10} = 11_2$
$45,75_{10} = 101101,11_2$	

Таблица 7

Соответствие между целыми числами
в разных системах счисления

Dec $q = 10$	Bin $q = 2$	Oct $q = 8$	Hex $q = 16$
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Примеры перевода чисел с основанием $q = 2, 8, 16$

Перевод числа 1325_8 в двоичную систему				
Восьмеричные цифры	1	3	2	5
Двоичные триады	001	011	010	101
<i>Результат</i>		$1325_8 = 1011010101_2$		
Перевод числа 101010110_2 в восьмеричную систему				
Двоичные триады	000	101	010	110
Восьмеричные цифры	0	5	2	6
<i>Результат</i>		$101010110_2 = 526_8$		
Перевод числа $12BD_{16}$ в двоичную систему				
Шестнадцатеричные цифры	1	2	B	D
Двоичные тетрады	0001	0010	1011	1101
<i>Результат</i>		$12BD_{16} = 1001010111101_2$		
Перевод числа 10001110101100_2 в шестнадцатеричную систему				
Двоичные тетрады	0010	0011	1010	1100
Шестнадцатеричные цифры	2	3	A	C
<i>Результат</i>		$10001110101100_2 = 23AC_{16}$		

Приведенные примеры показывают, что восьмеричную и шестнадцатеричную системы удобно использовать в качестве вспомогательных систем для представления двоичных чисел, в частности, для компактной записи двоичных кодов.

Двоичная система занимает особое место среди позиционных систем, поскольку является *основной системой* в вычислительной технике. Преимущество двоичной системы обеспечивается простотой технической реализации, так как технические элементы зачастую существуют в двух возможных состояниях. В качестве примеров можно привести противоположные состояния некоего физического объекта: намагниченное или размагниченное, отражающее свет или поглощающее, пропускающее электрический ток или не пропускающее. Каждому из двух возможных состояний элемента можно присвоить значения 0 и 1 и назвать этот технический элемент *ячейкой*, в которой можно хранить 1 бит данных.

Если количество ячеек, в каждой из которых можно хранить 1 бит, увеличивать, то возможности по числу представляемых или кодируемых значений будут существенно вырастать. Одним битом могут быть выражены два значения: 0 или 1, двумя битами – 4 значения: 00, 01, 10, 11, тремя битами – 8 значений:

000, 001, 010, 100, 011, 101, 110, 111. Далее количество возможных значений вырастает в геометрической прогрессии (табл. 9).

Таблица 9

Возможности представления данных в системе двоичного кодирования

Количество бит	1	2	3	4	5	6	7	8
Количество кодируемых значений	2	4	8	16	32	64	128	256
Математическая закономерность роста	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8

В общем виде количество представляемых или кодируемых значений N может быть рассчитано по формуле

$$N = 2^m,$$

где m – число разрядов в системе двоичного кодирования.

Из табл. 9 видно, что с помощью 1 байта, состоящего из 8 бит ($m = 8$), можно закодировать 256 значений. Если таблицу продолжить, то при $m = 10$ количество кодируемых значений N будет равно 1024.

Двоичное кодирование целых чисел

Кодирование целого числа осуществляют простым переводом этого числа в двоичную систему счисления. При этом возможны 3 формы записи двоичного числа:

- прямой код – используют для положительных чисел и получают простым переводом числа в двоичный код;
- обратный код – используют при кодировании отрицательных чисел, получают инвертированием всех двоичных чисел прямого кода за исключением первой, указывающей знак числа;
- дополнительный код – прибавляют единицу к значению обратного кода целого отрицательного числа.

Обратный и дополнительный коды используются для представления отрицательных чисел, что позволяет заменить машинное *вычитание* на более быструю операцию *сложения*. Кроме формы записи большое значение имеет длина кода или количество разрядов в записи двоичного числа. При простом математическом переводе длина кода равна количеству знаков в двоичной записи числа. Для записи полученного кода в вычислительной системе необходимо учитывать ограничения, связанные с машинным представлением. Например, при записи в 2-байтовой машинной системе любое число должно быть представлено не более чем 16 разрядами, а в 4-байтовой – 32 разрядами.

Для примера рассмотрим представление числа 1607 в 2-байтовой (16-разрядной) машинной ячейке. Воспользуемся при этом равенством $1607_{10} = 11001000111_2$. Результаты перевода и представления числа в ячейке памяти приведены в табл. 10.

Таблица 10

Перевод целого числа в двоичный машинный код

Представление	Число
Десятичное	1607
Двоичное в прямом коде	11001000111
Машинное в 16-разрядном прямом коде	0000 0110 0100 0111
Обратный код	0111 1001 1011 1000
Дополнительный код	0111 1001 1011 1001

Для физического представления целого числа в ячейках памяти используется разрядная сетка. Число 1607, например, в 16-разрядной сетке представляется следующим образом (табл. 11).

Таблица 11

Разрядная сетка

Номер разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Значение разряда	0	0	0	0	0	1	1	0	0	1	0	0	0	1	1	1

Число заносится в разрядную сетку справа налево, при этом свободные разряды заполняются нулями. В старшем разряде сетки представляются данные о знаке числа: ноль соответствует положительному значению, единица – отрицательному.

Основной недостаток системы кодирования целых чисел – возможность переполнения разрядной сетки для больших значений чисел.

Двоичное кодирование действительных чисел

Действительными или вещественными числами (в отличие от целых) называются числа, имеющие дробную часть. Для кодирования действительного числа и сохранения его в памяти каждое число преобразуют следующим образом:

– представляют число R как произведение мантиссы m ($|m| < 1$) на основание системы счисления q , возведенное в целую степень p (p называют порядком или характеристикой); для представления числа используют формулу

$$R = m \cdot q^p;$$

– полученные значения мантиссы и порядка переводят в двоичный код;
– двоичный код разбивают на группы, число разрядов в группе определяют равным четырем, восьми или десяти байтам.

Такой способ записи чисел называется *нормализованным*, или представлением с *плавающей точкой*. Последовательность преобразования действительного числа в двоичную форму показана в табл. 12. Из таблицы видно, как указанное число сохраняется в машинных ячейках.

Перевод действительного числа в двоичный машинный код

Представление	Число
Десятичное	12,34
Десятичное нормализованное	$0,1234 \times 10^2$
Двоичное	10011010010 10
Двоичное в машинном коде	$\underbrace{0000\ 0010}_{\text{порядок}} \underbrace{0000\ 0100\ 1101\ 0010}_{\text{мантисса}}$

В разрядной сетке для действительного числа отводятся отдельные группы разрядов для мантиссы, порядка, знака числа и знака порядка. Структура разрядной сетки принимает при этом вид, показанный на рис. 5.

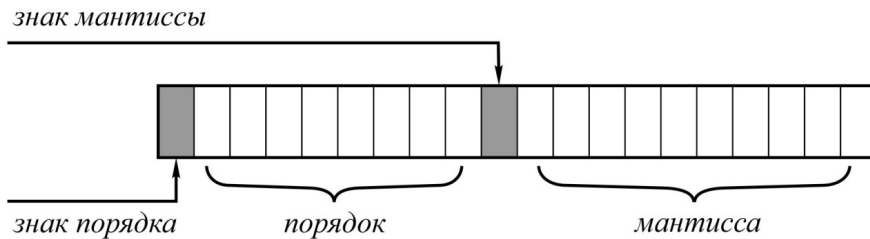


Рис. 5. Структура разрядной сетки для записи действительного числа

В представленной структуре можно менять соотношение между числом разрядов для записи порядка и мантиссы. Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа. Чем больше разрядов занимает порядок, тем шире диапазон представляемых чисел.

Представление и обработка текстовых символов

Первоначально вычислительные машины создавались исключительно для обработки численных данных. По мере совершенствования вычислительной техники числовые данные стали дополняться текстовыми комментариями, которые со временем преобразовались в электронные текстовые документы. С 80-х гг. прошлого века текстовые документы, созданные в электронной форме, стали активно вытеснять традиционные машинописные документы. С появлением компьютеров новая технология создания текстовых документов окончательно закрепилась и компьютеры стали широко использоваться для создания, обработки и хранения текстовых данных.

Способы кодирования текста

Компьютерное представление текста связано с системой его кодирования, поэтому для представления текста нужно знать систему кодирования и декодирования текстовых символов.

Кодировать языковую речь можно различными способами: письменно, жестами или сигналами любой другой природы (светофор, телефонные звонки). Чаще всего кодированию подвергаются тексты на естественных языках. Для естественных языков существуют различные способы кодирования, остановимся на некоторых характерных и используемых на практике способах.

1. Графический способ – основан на использовании специальных рисунков или знаков. Пример графического кодирования описан в литературном произведении Артура Конана Дойла «Пляшущие человечки», где для шифрования сообщений использовалась последовательность пляшущих человеческих фигурок (рис. 6).

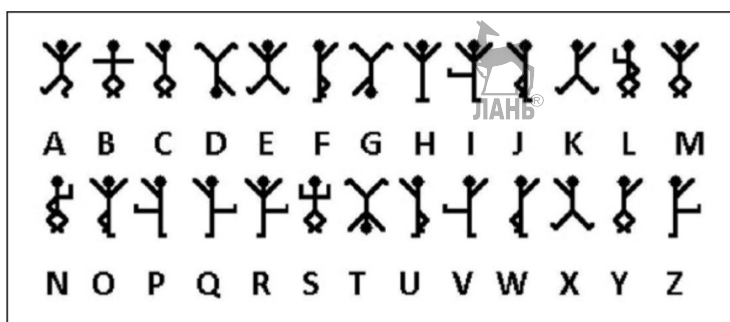


Рис. 6. Таблица кодов «Пляшущие человечки»

Пользуясь ключом к данному шифру, можно прочесть закодированную фразу:



Другим примером графического кодирования является азбука Морзе, созданная американским изобретателем Самюэлем Морзе в 1837 г. для телеграфического кодирования сообщений. В азбуке Морзе каждая буква представлена комбинацией точек и тире (табл. 13). До настоящего времени в мореходной практике может использоваться сигнал бедствия SOS (...---...), несмотря на существование автоматизированных систем оповещения о бедствии.

Графические коды азбуки Морзе

Аналоги букв латиницы и кириллицы, близкие по звучанию						Аналоги букв латиницы и кириллицы, различные по звучанию		
Лат.	Символ Морзе	Кир.	Лат.	Символ Морзе	Кир.	Лат.	Символ Морзе	Кир.
A	•–	А	L	•–••	Л	Q	–••–	Щ
B	–•••	Б	M	––	М	V	•••–	Ж
C	–•–•	Ц	N	–•	Н	X	–••–	Ь
D	–••	Д	O	–––	О	Y	–•––	Ы
E	•	Е	P	•–	П	Нет аналогов		
F	••–•	Ф	R	•–•–	Р	–•–•		Ч
G	–••	Г	S	•••	С	–•–•		Ш
H	••••	Х	T	–	Т	•–•–•		Ъ
I	••	И	U	••–	У	••–••		Э
J	•–•–	Й	W	•––	В	••––		Ю
K	–•–	К	Z	–•••	З	••–•		Я

2. Символьный способ – на основе символов (букв) того же алфавита, что и исходный текст. Способ используется в криптографии при создании шифрованных сообщений. Одним из первых применений способа является кодирование английского алфавита, предложенное в 1580 г. Фрэнсисом Бэконом [9]. Шифр Бэкона (табл. 14) выполнен на основе двоичного 5-разрядного кода, состоящего из букв А и В. Для создания сообщений на основе предложенной Бэконом системы требуется двухсимвольный алфавит, при этом длина самого сообщения возрастает в 5 раз, поскольку каждая буква заменяется набором из 5 символов.

Таблица 14

Кодирование английского алфавита

a	AAAAA	g	AABBA	n	ABVAA	t	BAABA
b	AAAAB	h	AABBB	j	ABVAB	v	BAABB
c	AAABA	i	AVAAA	p	ABVVA	w	BAVAA
d	AAABV	k	AVAAV	q	ABVVV	x	BAVAV
e	AABA	l	AVABA	r	BAAAA	y	BAVVA
f	AABAV	m	AVABV	s	BAAAV	z	BAVVV

3. Числовой способ – основан на кодировании символов с помощью чисел. Широкое распространение способ получил в связи с созданием и развитием электронно-вычислительных машин. В ЭВМ для кодирования букв используется два числа: 0 и 1. В отличие от шифра Бэкона, где достаточно 5-разрядного представления, в компьютерной технике принято 8-разрядное, или 8-битовое, представление символов. Последовательность из 8 бит образует 1 байт, байт используется для кодирования одного символа. С помощью одного байта путем смены последовательности записи нулей и единиц можно закодировать 256 различных символов.

Кодирование компьютерных символов

Система числового кодирования компьютерных символов принята на международном уровне и построена следующим образом.

1. Каждому символу, обозначенному на клавиатуре компьютера, присваивается двоичный восьмиразрядный код. Последовательность символов и их кодов формируется в кодировочную таблицу под названием ASCII (American Standard Code for Information Interchange) – американский стандартный код для обмена информацией.

2. В таблице ASCII все кодируемые символы делятся на две группы, в первую из которых (*базовую*) включается 128 символов (от 0 до 127) стандартного, или неизменяемого, кода, а во вторую (*расширенную*) – следующие 128 символов (от 128 до 255), которые относятся к расширяемой или изменяемой части кода (табл. 15).

3. В первой базовой группе кодируются буквы латинского алфавита, арабские цифры, математические знаки, знаки препинания и др. Кодировка первой группы символов является международной и обязательной для всех компьютеров, участвующих в процессе глобального обмена информацией.

4. Вторая группа из 128 символов предоставлена для кодирования национальных или специальных алфавитов, а также некоторых графических символов. Национальный алфавит русского языка (а также украинского, белорусского, болгарского) известен всему миру под названием кириллица.

Таблица 15

Представление символов в кодовой таблице ASCII

Базовая группа (0–127)	Расширенная группа (128–255)
Управляющие коды	Национальные и специальные алфавиты; алфавиты в России, Украине, Белоруссии, Сербии, Болгарии – кириллица
Латинский алфавит (26 букв)	
Арабские цифры	
Математические знаки	
Знаки препинания	

С появлением компьютеров кириллица была закодирована на основе расширенной группы символов, исторически сложилось 6 способов кодировки кириллицы, различные версии кодировки появлялись одна за другой в процессе развития компьютерной техники. Со временем некоторые кодировки исчезли из употребления, к действующим относятся 4 вида кодировок (табл. 16).

Первые из трех приведенных в таблице кодировок выполнены в однобайтовом формате, а кодировка Unicode, принятая в конце 1990-х, имеет 2-байтовый формат. Кодировка Unicode отличается от предыдущих версий широкими количественными возможностями, поскольку число возможных вариантов кодирования возрастает до $2^{16} = 65536$, но длина текстового сообщения увеличивается при этом в два раза. Расширенная кодировка может использоваться для представления большинства языков на планете Земля, в том числе для кириллицы.

Таблица 16

Виды кодировок кириллицы

Наименование	Назначение
CP 1251	для работы под операционной системой Windows
КОИ8-Р	для представления кириллических текстов в сети Интернет
ISO-8859	для представления кириллических текстов в соответствии с международным стандартом
Unicode	для расширения возможностей по числу кодируемых символов и алфавитов

Наличие многих кодировок кириллицы может создавать трудности по переходу из одной кодировки в другую. Для перехода между кодировками используются специальные программы, которые называются *текстовыми конверторами*. Современные конверторы способны самостоятельно определять исходную кодировку текста, расшифровывать текст и представлять его на естественном языке.

Компьютерные шрифты

Для представления текста в компьютере наряду с кодированием каждого символа используются различные варианты воспроизведения этого символа в тексте, другими словами, различные *шрифты* символа. Шрифт отражает форму печатного знака и включает в себя полный набор буквенных, цифровых и других символов, выполненных в едином стиле.

Шрифт относится к одному из основных изобразительных элементов текстового документа. Шрифты по своему назначению делятся на книжные, газетные, плакатные, картографические, декоративные, рекламные и другие.

Среди большого количества используемых на практике шрифтов выделяются *компьютерные шрифты*, которые отличаются разнообразием и лежат в основе форматирования электронного документа или web-страницы.

Компьютерный шрифт – это файл с описанием набора буквенных, цифровых, служебных и других символов, который используется специальной программой (текстовым редактором) для отображения этих символов в тексте. Каждый компьютерный шрифт имеет ряд отличительных признаков или характеристик. Характеристики шрифтов отображаются в меню текстового редактора, которое позволяет выбрать *наименование* шрифта, его *начертание*, особенности изображения его элементов и т. д.

1. Наименование шрифта – краткое обозначение совокупности символов, которые выполнены в едином стиле. В компьютерном меню наименование выражается термином «Шрифт».

2. Гарнитура шрифта – совокупность нескольких шрифтов, объединенных общими стилевыми признаками. Гарнитура имеет условное название, например обыкновенная, литературная, плакатная. В компьютерном меню совокупность близких по стилю шрифтов имеет похожие названия, например Arial, **ArialBlack**, ArialUnicode.



Рис. 7. Кегль шрифта – Н (пт)

3. Кегль шрифта – высота прямоугольника, в который вписан любой знак данного алфавита; выражается в типографических *пунктах* (пт), 1 пт = 1/72 дюйма (рис. 7). В компьютерном меню кегль обозначен словосочетанием «Размер шрифта».

4. Начертание шрифта – насыщенность и толщина штрихов, высота знаков и характер заполнения, отклонение от вертикального положения. В компьютерном меню выделяют следующие виды начертания: *обычный*, *курсив*, *полужирный*, *полужирный курсив*.

5. Видоизменение шрифта – изменение формы отдельных знаков шрифта. С помощью видоизменения выделяются ПРОПИСНЫЕ буквы, подстрочные и ^{надстрочные} индексы, *зачеркнутые* символы и т. п.

6. Интервал между отдельными символами в слове может быть выбран *обычным*, *уплотненным* или *разреженным*. Величина интервала плавно регулируется и выражается в пунктах. При необходимости к регулировке интервала можно подключить кернинг – процесс подбора расстояния между символами в зависимости от конкретного сочетания соседствующих символов. Кернинг предназначен для гармонизации визуального восприятия текста.

7. Смещение – сдвиг символов в вертикальном направлении; различают сдвиги *вверх* и *вниз* (рис. 8). Единицей измерения сдвига является пункт.

Компьютерные шрифты имеют и другие отличительные характеристики. К ним относятся *цвет текста*, *цвет заливки* текста, *масштаб* текста и др. Разнообразие характеристик предоставляет широкие возможности по выбору шрифтов и оформлению текстовых документов. Для соблюдения гигиенических требований детям младшего возраста следует выбирать четкие, простые по рисунку шрифты с кеглем 12–16 пт, для взрослого читателя – шрифты 8–10 пт, в справочных и газетных изданиях кегль шрифта может быть еще меньше.



Рис. 8. Смещение символов

Электронные документы

С помощью компьютера текст, представленный в бумажном документе, можно преобразовать в электронную форму. У электронных документов имеются очевидные преимущества перед бумажными копиями, например снижение затрат при создании или расширенные возможности преобразования и пересылки. Несмотря на бесспорные преимущества, электронный документ не вытесняет полностью бумажный, поэтому на практике используют и различают различные виды документооборота:

- бумажный с использованием автономных компьютеров – создание документов производится на компьютере, а остальные операции с документами выполняются в бумажной форме;
- смешанный – все операции с документами осуществляются на компьютерах, объединенных в сеть, но юридическую силу документ получает только в бумажном виде;
- электронный – все операции с документами, включая электронно-цифровую подпись, производятся в электронном виде.

На предприятиях и в организациях можно встретить все виды документооборота, но преобладающим является смешанный оборот. При этом после внедрения электронно-цифровой подписи постоянно возрастает доля полностью электронного документооборота.

Дальнейшее развитие и переход на более полный безбумажный документооборот сдерживается по нескольким причинам.

1. Бумажные документы с оттиском печати и подписью зачастую необходимы для соблюдения требований налогового, банковского, судебного и другого законодательства.
2. Электронно-цифровая подпись медленно внедряется в практическую деятельность.
3. Безбумажный документооборот связан с необходимостью введения дополнительных мер обеспечения компьютерной безопасности.

Несмотря на существующие ограничения, электронный документ все шире внедряется в практическую деятельность. При внедрении электронных доку-

ментов предприятия и организации сталкиваются с необходимостью рассылки и обмена большим количеством электронных файлов. На небольшом предприятии документооборот организуют обычно с помощью электронной почты. На большом предприятии для работы с электронными документами используются системы автоматизированного документооборота и делопроизводства (САДД).

Автоматизированные системы обеспечивают:

- автоматическое определение маршрута документа;
- рассылку и быстрое прохождение документа в сети;
- единую адресную книгу для всех пользователей;
- единую базу нормативов, словарей и классификаторов;
- потоковое сканирование и копирование документов.

Компьютерные программы, обеспечивающие автоматизированный документооборот и делопроизводство, относятся к САДД-продуктам, которые представлены на рынке программных продуктов.

Формирование графических данных

Классификация цифровых изображений

По типу представления информации и алгоритмам обработки цифровых данных различают *двумерную* и *трехмерную* графику. Двумерная компьютерная графика в свою очередь делится на графику *растровую* и *векторную*.

Растровая графика построена на принципах формирования изображения из отдельных точек. Основой растрового изображения является прямоугольная матрица, каждая ячейка которой представлена цветным единичным элементом квадратной формы.

Сетка матрицы называется растровой картой, а единичный элемент сетки – пикселем. Пиксели подобны зернам фотографии, при значительном увеличении они становятся заметными. Избыточное увеличение приводит к возникновению «лестничного эффекта» – последовательности наложенных друг на друга прямоугольных пикселей.

Под каждый пиксель отводится определенное количество битов, которое называется битовой глубиной. Если битовая глубина равна 1, то изображение может быть только черно-белым, при этом 0 соответствует белому цвету, а 1 – черному. Цветовая гамма черно-белого изображения равна $2^1 = 2$. Если битовая глубина равна 2, то цветовая гамма каждого пикселя равна 4 (2^2), для битовой глубины 3 – 8 (2^3) и т. д. Общее число цветов N , отображаемых в пикселе, вычисляется по формуле

$$N = 2^i,$$

где i – битовая глубина, или информационная емкость одного пикселя.

Если размер растрового изображения составляет по высоте X и по ширине Y (в пикселях), то произведение этих величин записывается в форме $X \cdot Y$ и называется разрешением экрана, например разрешение 1024 · 768 пикселей. Общий объем изображения с учетом цветности каждого пикселя вычисляется по формуле

$$V = X \cdot Y \cdot i.$$

Объем компьютерной памяти, который требуется для хранения цветного изображения, содержащего 256 цветов и имеющего размер $20 \cdot 30$ пикселей, вычисляется по формуле

$$V = X \cdot Y \cdot i = \log_2 256 = 20 \cdot 30 \cdot 8 = 600 \text{ байт.}$$

Растровые изображения получают чаще всего с помощью *сканеров, цифровых фото- и видеокамер*. Растровые изображения можно получить также в ручном режиме, используя программы *растровой графики*. С помощью растровой графики можно отразить и передать все тонкости реального изображения. Растровое изображение ближе к фотографии, поскольку позволяет более точно воспроизводить изображение объекта.

Основные недостатки растровой графики – большие массивы данных и рост зерна с увеличением изображения.

Векторная графика принципиально отличается от растровой графики, поскольку основана на других принципах.

Основными неделимыми элементами векторного изображения являются *линии* и *объекты*, которым назначают определенные атрибуты (свойства): размер, цвет и т. п.

Изображения векторной графики строятся из множества линий и объектов, положение и размеры которых задаются с помощью математических формул. Перед выводом на экран векторного изображения программа производит вычисления координат объектов, поэтому векторную графику иногда называют *вычисляемой графикой*.

Изображения чаще всего получают в ручном режиме и используют преимущественно в оформительских работах, а также в конструкторской и научной деятельности. Векторная графика отличается сравнительно *малыми массивами* данных. Увеличение размеров векторного изображения *не приводит к потере его качества* (в отличие от растрового).

Основные недостатки векторной графики выражаются в высокой трудоемкости создания реалистичных изображений и необходимости преобразования в растровую форму перед выводом изображения на печать.

Трехмерная компьютерная графика, или 3D-графика, оперирует с объектами в трехмерном пространстве. Все объекты 3D-графики формируются в объемном виде, а результаты представляются в виде проекции – плоской картины. Объемная форма обеспечивается представлением объекта из набора плоских частиц. Минимально возможную частицу называют *полигоном*. Обычно в качестве полигона используют треугольники, каждый из которых имеет три координаты вершин.

Для визуального преобразования в 3D-графике используют *матрицы*, которые бывают трех видов:

- матрица поворота;
- матрица сдвига;
- матрица масштабирования.

С математической точки зрения преобразование выражается в умножении координат треугольника на соответствующую матрицу. Матричное преобразование всех полигонов объекта приводит к повороту, сдвигу или изменению масштаба всего объекта.

Трехмерная компьютерная графика широко используется в компьютерных играх, кино и телевидении. Основным недостатком является высокая сложность программ 3D-графики и необходимость профессионального обучения для работы с этими программами.

Моделирование цветовых оттенков, законы Грассмана

При формировании цветных изображений удобно пользоваться понятием *цветовой модели*.

Цветовая модель – это упрощенный геометрический способ разделения цветового оттенка на составляющие компоненты.

Цветовые модели принято располагать в трехмерной системе координат, которая называется *цветовым пространством*. Цвет при этом представлен точкой в цветовом пространстве. Формирование цвета в точке подчиняется определенным законам, которые в середине XIX в. сформулировал немецкий математик, физик и филолог Герман Грассман. Учение Грассмана о цветоделении включает три закона.

Закон трехмерности. Цвет точки можно выразить тремя независимыми составляющими, каждую из которых невозможно получить сложением двух остальных составляющих.

Закон непрерывности. Непрерывное изменение уровня одной из составляющих приводит к непрерывному изменению цвета смеси, поэтому цвет смеси можно подобрать бесконечно близко к любому другому цвету.

Закон аддитивности. Любой цвет в цветовом пространстве, которое представлено в системе координат RGB, изображается вектором, который описывается уравнением

$$\vec{C}_n = R_n \vec{R} + G_n \vec{G} + B_n \vec{B}.$$

При этом направление вектора характеризует цветность излучения, а модуль выражает яркость.

Законы Грассмана положены в основу построения цветовых моделей. В компьютерной графике принято использовать два типа цветовых моделей. Первый тип предназначен для моделирования *самосветящихся* или *излучающих* объектов, связанных с аддитивным (суммарным) цветовоспроизведением, второй тип – для *несветящихся* или *отражающих* объектов, связанных с субтрактивным (вычитающим) цветовоспроизведением.

Цветовая модель RGB

Цветовая модель RGB предназначена для моделирования излучающих объектов на основе *аддитивного цветовоспроизведения* или *аддитивного синтеза*. Цветовой оттенок в этой модели складывается из трех основных цветов:

- красного (Red);
- зеленого (Green);
- синего (Blue).

Модель RGB относится к *аппаратно-ориентированным* моделям, которые используются в дисплеях для формирования оттенков светящихся пикселей экрана. Для геометрического представления модели используется трехмерная система координат, или так называемый RGB-куб (рис. 9).

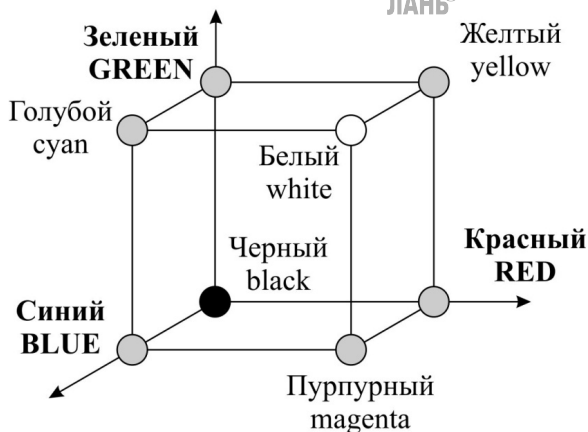


Рис. 9. Цветовой куб модели RGB

Начало отсчета в этом кубе $(0,0,0)$ соответствует *черному* цвету, а максимальное значение $(255,255,255)$ – *белому* цвету. Любая точка на поверхности цветового куба имеет *три цветовых координаты* или три значения цветности, которые определяют *суммарный* цвет одного пикселя на экране.

Для задания цвета пикселя требуется три значения цветности. Для каждого цвета можно использовать 8-битовое разрешение (256 оттенков). Это позволяет воспроизвести любой цветовой оттенок в точке, но одновременно создает трудности в регистрации и сохранении большого количества возможных цветовых оттенков.

Для упрощения цветовоспроизведения и уменьшения числа возможных цветовых оттенков используют так называемую *цветовую электронную палитру*, состоящую из конечного числа ячеек, каждая из которых содержит определенный цветовой оттенок.

Электронная палитра охватывает ограниченный набор цветов, общее число цветов или цветовых оттенков зависит от принятого цветового разрешения – числа возможных оттенков. Если для модели RGB выбрать 8-битовое разрешение, то электронная палитра будет содержать 256 возможных оттенков. Цветовая палитра, соответствующая 8-битовому разрешению, называется *индексной палитрой*.

В индексной палитре каждому цвету присваивается определенный номер или индекс, значение которого задает цвет пикселя. Это позволяет заменить

цветовые координаты индексом при хранении файла и уменьшить тем самым размер графического файла.

Цветовая модель СМУК

Цветовая модель СМУК предназначена для моделирования *отражающих* объектов на основе *субтрактивного* (вычитающего) *цветовоспроизведения* или *субтрактивного синтеза*. Основное назначение модели – цветовоспроизведение *полиграфических* печатных объектов, которые не излучают, а только отражают падающий на них свет. Цветовой оттенок в модели СМУК складывается не из основных цветов, а из дополнительных, которые получают в результате вычитания основного цвета из белого.

Дополнительные цвета или цветовые компоненты модели СМУК формируют следующим образом:

- голубой (Cyan) = белый – красный = зеленый + синий;
- пурпурный (Magenta) = белый – зеленый = красный + синий;
- желтый (Yellow) = белый – синий = красный + зеленый.

Для геометрического представления модели используется трехмерная система координат, или так называемый *цветовой куб* модели СМУК (рис. 10), по осям которого откладываются направления дополнительных цветов. Начало отсчета в этом кубе соответствует *белому* цвету, а максимальное значение – *черному* цвету.

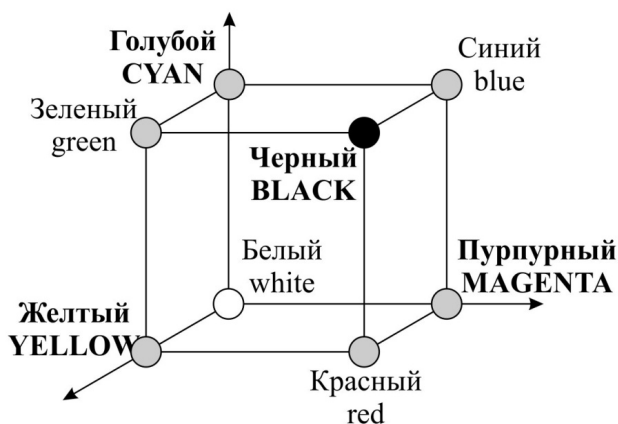


Рис. 10. Цветовой куб модели СМУК

Из названий цветов складывается аббревиатура СМУ. Основной недостаток модели СМУ выражается в том, что наложение трех дополнительных цветов не дает чистый черный цвет, поэтому модель дополняется четвертым цветом – *черным*. В название модели добавляется буква К (от слова black), откуда получается полное название СМУК. При этом цвет каждой точки на цветовом кубе имеет четыре составляющих и обозначается *четырьмя* координатами.

Основное назначение модели СМΥК – это разложение изображения на четыре составляющих, соответствующих цветовым компонентам этой модели. Процесс разложения изображения называют *цветоделением* и используют при подготовке цветных изображений к тиражированной печати. Все современные графические редакторы позволяют выполнять цветоделение изображений. В процессе печати каждую из цветовых составляющих наносят последовательно одну за другой, формируя результирующее изображение послойно в несколько приемов.

Формирование цветных изображений на экране и бумаге

Цветное изображение на экране монитора формируют способом аддитивного синтеза на основе модели RGB путем оптического смешивания красного, зеленого и синего цветов. Поверхность экрана при этом разбивают на множество точек – *пикселей*, которые располагают в строго определенной геометрической последовательности. Каждый пиксель в свою очередь образуют сложением трех *субпикселей* — красного (R), зеленого (G) и синего (B). Существует несколько различных вариантов поверхностного размещения субпикселей, на рис. 11 показан вариант размещения пикселей на поверхности жидкокристаллического монитора.

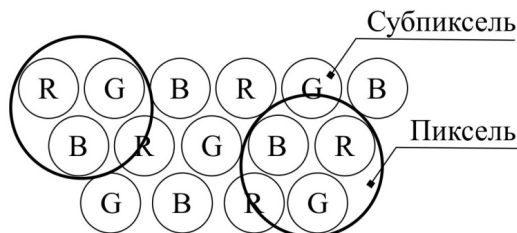


Рис. 11. Формирование цветного изображения на экране

Пропорция цветов в пикселе задается уровнем яркости субпикселей. Черный цвет получается при полностью выключенных субпикселях, белый – при полностью включенных. Другие цвета и оттенки задаются соотношением яркости субпикселей. Чистый красный цвет получается при полностью выключенных субпикселях синего и зеленого цвета, а желтый цвет получается смешиванием красного и зеленого в равных пропорциях.

Цветное изображение на бумаге формируют способом субтрактивного синтеза на основе модели СМΥК путем цветоделения исходного изображения и последовательного набора голубой, пурпурной, желтой и черной красок. Основная проблема формирования цветного изображения на бумаге выражается в том, что большинство печатающих устройств не позволяет в процессе печати смешивать краски и наносить на разные участки бумаги разные смесевые цвета. Для решения этой проблемы используют способ *точечного дозированного* нанесения краски каждого цвета, который реализуют с помощью механизма *амплитудно-модулированного растрирования*.

Растривание – это метод передачи изображения с помощью растровых точек, расположенных в определенной геометрической последовательности. Форма и частота расположения растровых точек зависит от вида *печатающего устройства*. Возможные формы растровых точек показаны на рис. 12.

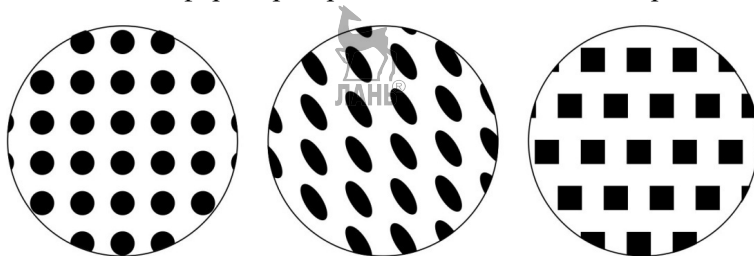


Рис. 12. Формы растровых точек на печатном материале

Суть *амплитудно-модулированного* растривания выражается в том, что размер точек в процессе печати может изменяться при неизменном количестве самих точек. Это приводит к тому, что на светлых участках изображения точки достаточно малы и малозаметны. В полутоновой части они занимают примерно половину запечатываемой площади изображения. В темных же участках точки настолько велики, что занимают почти всю площадь запечатываемого материала.

Качество печатной продукции, полученной способом амплитудно-модулированного растривания, оценивается по следующим параметрам.

1. Форма растровой точки определяется печатающим устройством и задает особенности визуального восприятия изображения.

2. Линиатура – число растровых точек на единице длины (обычно на одном дюйме). Чем выше число растровых точек, тем меньше заметна дискретность изображения, обусловленная его растровой структурой.

3. Угол наклона раstra – это угол поворота растровых точек разного цвета относительно друг друга и относительно общей оси изображения. Этот параметр имеет особое значение при цветной печати, так как печать каждой краски осуществляется с использованием разных углов наклона. Значения углов наклона раstra стандартизированы, и менять их не рекомендуется. При нарушении порядка расположения углов наклона может возникать *муар* – паразитный, раздражающий глаз визуальный эффект.

Выбор параметров растривания определяет качество печатной продукции и соответствующую стоимость печатного тиража.

Операции с логическими данными

Компьютер предназначен не только для обработки числовых, текстовых и графических данных. К числу сохраняемых и обрабатываемых данных относятся также логические данные. Для обработки логических данных используется *арифметико-логическое устройство* (АЛУ), которое входит в состав микропроцессора и отвечает за выполнение логических операций. Математической основой всех логических операций в АЛУ является *алгебра логики*.

Основные понятия алгебры логики

Основоположником алгебры логики является английский математик и логик Джорж Буль, в честь которого математические основы логики называют *булевой алгеброй*. В XIX в. он написал более 50 статей и монографий, в которых изложил основные понятия современной алгебры логики.

Алгебра логики – это раздел математики, в основе которого лежат операции с логическими высказываниями.

Логическое высказывание – это утверждение в языковой форме, которое может иметь только одно из двух возможных значений: ИСТИНА (*логическая 1*) или ЛОЖЬ (*логический 0*).

Примеры логических высказываний:

Луна – спутник планеты Земля – истинное высказывание;

Волга впадает в Черное море – ложное высказывание;

Завтра ожидается хорошая погода – фраза не относится к логическому высказыванию.

В алгебре логики не рассматривается конкретное содержание высказывания, важно только, *истинно* оно или *ложно*. Для обозначения истинности и ложности логических высказываний используют разные варианты:

ИСТИНА	И	True	T	1
ЛОЖЬ	Л	False	F	0

Простые высказывания могут объединяться в *составные* или *сложные* высказывания. Объединение высказываний выполняется с помощью *логических операций* и записывается в виде *логических формул* или *логических выражений*.

Логическое выражение – это запись высказывания в математической форме, которая состоит из логических величин, связанных между собой *логическими операциями*.

Логические операции выступают в роли связок между логическими величинами.

Рассмотрим пример преобразования логического высказывания в логическое выражение и перевод полученного выражения в математическую форму (табл. 17).

Таблица 17

Формирование логического выражения

Полное высказывание F	диаметр Земли больше диаметра Венеры и меньше диаметра Юпитера
Высказывание A	диаметр Земли больше диаметра Венеры
Высказывание B	диаметр Земли меньше диаметра Юпитера

Полное высказывание F	диаметр Земли больше диаметра Венеры и меньше диаметра Юпитера
Союз «и»	логическая операция между двумя высказываниями
Логическое выражение	$F = A \text{ и } B$
Значение выражения	$F = (1 \text{ и } 1) = 1$

Пример, приведенный в табл. 17, показывает, что полное высказывание F с помощью союза «и» связывается из двух исходных высказываний и является истинным, поскольку истинны исходные высказывания A и B .

Базовые логические операции

В записи выражения логические высказывания принято обозначать латинскими буквами A , B , C , а логические операции – словами естественного языка или специальными символами. Названия и способы обозначения основных или базовых логических операций показаны в табл. 18. Каждая логическая операция имеет свое наименование, определение и форму записи. В естественном языке, в алгебре логики и в языках программирования приняты разные формы записи одних и тех же операций.

Таблица 18

Обозначения базовых логических операций

Наименование	Формы записи		
	в естественном языке	в алгебре высказывания	в языках программирования
КОНЪЮНКЦИЯ	И 	\wedge или $\&$	<i>and</i>
ДИЗЪЮНКЦИЯ	ИЛИ	\vee или $+$	<i>or</i>
ИНВЕРСИЯ	НЕ, НЕВЕРНО, ЧТО...	\neg или $-$	<i>not</i>

Описание базовых логических операций и соответствующих определений этих операций представлено в табл. 19.



Логические операции

Наименование операции	Форма записи	Определение
КОНЬЮНКЦИЯ (логическое умножение)	$A \wedge B$	Новое выражение, которое будет истинным только тогда, когда истинны оба исходных простых выражения. Определяет соединение логических выражений с помощью союза И
ДИЗЬЮНКЦИЯ (логическое сложение)	$A \vee B$	Новое выражение, которое будет истинным только тогда, когда истинно хотя бы одно из исходных выражений. Определяет различие двух логических выражений с помощью союза ИЛИ
ИНВЕРСИЯ (логическое отрицание)	$\neg A$	К исходному выражению добавляется частица НЕ или слова НЕВЕРНО, ЧТО
ИМПЛИКАЦИЯ (логическое следование)	$A \rightarrow B$	Связывает два логических выражения, из которых первое является условием (А), а второе (В) – следствием. Результатом является ЛОЖЬ только тогда, когда условие А истинно, а следствие В ложно. Выражается словами ЕСЛИ, ТО
ЭКВИВАЛЕНТНОСТЬ (логическая равнозначность или тождество)	$A \leftrightarrow B$	Результатом является новое логическое выражение, которое будет истинным тогда и только тогда, когда оба исходных выражения одновременно истинны или ложны

В табл. 19 показано пять логических операций: конъюнкция, дизъюнкция, инверсия, импликация и эквиваленция. Эти функции связаны между собой определенными соотношениями. В частности, верны следующие выражения.

1. Импликацию можно выразить через дизъюнкцию и отрицание:

$$A \rightarrow B = \bar{A} \vee B.$$

2. Эквивалентность можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\bar{A} \vee B) \wedge (\bar{B} \vee A).$$

Таким образом, для описания и обработки всех логических выражений достаточно иметь три основные операции: инверсию, дизъюнкцию и конъюнкцию. Если в выражении использована не одна, а несколько логических операций, то действия выполняются в определенном порядке: 1 – инверсия, 2 – конъюнкция, 3 – дизъюнкция, 4 – импликация, 5 – эквивалентность. Для изменения указанного порядка действий используют скобки.

Все возможные значения любой логической операции представляются в *таблице истинности* (см. табл. 20). Таблица истинности показывает, каким образом связаны комбинации входных логических величин с соответствующими значениями на выходе логической операции.

Таблица 20

Таблицы истинности логических операций

Наименование операции	Таблица истинности		
	A	B	A∧B
КОНЪЮНКЦИЯ	1	1	1
	1	0	0
	0	1	0
	0	0	0
ДИЗЪЮНКЦИЯ	A	B	A∨B
	1	1	1
	1	0	1
	0	1	1
	0	0	0
ИНВЕРСИЯ	A	¬A	
	1	0	
	0	1	
ИМПЛИКАЦИЯ	A	B	A→B
	1	1	1
	1	0	0
	0	1	1
	0	0	1
ЭКВИВАЛЕНТНОСТЬ	A	B	A↔B
	1	1	1
	1	0	0
	0	1	0
	0	0	1

В качестве примера составим таблицу истинности и рассчитаем значение логического выражения $(X \wedge Y) \rightarrow Y$. В левых столбцах табл. 21 представим возможные комбинации значений X и Y , а в правых – результат применения логических операций.

Полученный в табл. 21 результат показывает, что выражение принимает значение «ИСТИНА» при любых значениях истинности входящих переменных. Такое выражение называется тождественно истинным, или *логической тавтологией*. Логические тавтологии называют также *логически истинными высказываниями*.

Если выражение принимает значение «ЛОЖЬ» при любых значениях истинности входящих в них переменных, то оно называется тождественно ложным выражением, или *противоречием*.

Противоречие называют также *логически ложным высказыванием*.

Таблица 21

Пример расчета таблицы истинности

X	Y	$X \wedge Y$	$(X \wedge Y) \rightarrow Y$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Логические элементы и схемы

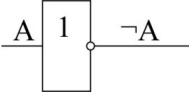
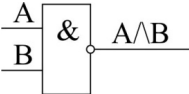
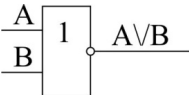
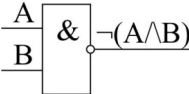
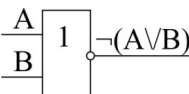


Математический аппарат логики используется для описания процесса функционирования вычислительной техники. Вычислительная техника основана на двоичной системе счисления, в которой используются цифры 0 и 1 и две логические переменные, которые также равны 0 и 1. Отсюда следует, что базовые логические операции можно представить в форме *технических элементов*, которые реализуют *логические функции*.

Логические элементы – простые конструкции, предназначенные для реализации базовых логических операций и обработки данных в цифровой форме. Наглядное описание логических элементов может быть представлено в графической форме (табл. 22).

Логические элементы имеют *один* или *несколько входов* и, как правило, *один выход*. Как входные, так и выходные сигналы имеют дискретную форму. Значение сигнала на выходе логического элемента определяется комбинацией входных сигналов и видом логической функции элемента.

Графическое представление логических элементов

Логические функции	Обозначения логических элементов
ИНВЕРСИЯ (НЕ)	
КОНЪЮНКЦИЯ (И)	
ДИЗЪЮНКЦИЯ (ИЛИ)	
ИНВЕРСИЯ КОНЪЮНКЦИИ (И – НЕ)	
ИНВЕРСИЯ ДИЗЪЮНКЦИИ (ИЛИ – НЕ)	

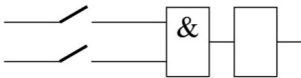


Рис. 13. Система безопасности механического пресса

Из нескольких логических элементов можно построить простую или сложную логическую схему. Пример простой схемы показан на рис. 13, где представлена система безопасности механического пресса, предназначенная для исключения травмирования конечностей работника. В этой схеме сигнал от оператора к силовому узлу проходит через элемент «И» (&), который пропускает его тогда, когда включены оба пускателя, расположенные под его правой и левой рукой.

Другой пример простой схемы – логический триггер (рис. 14). Логическая схема триггера строится из двух элементов «ИЛИ-НЕ», соединенных таким образом, что при подаче внешнего сигнала элементы переключаются из одного состояния в другое и устойчиво хранят это состояние до следующего переключения. Способность к сохранению одного из двух возможных состояний позволяет использовать триггер в качестве ячейки памяти, в которой хранится 1 бит данных.

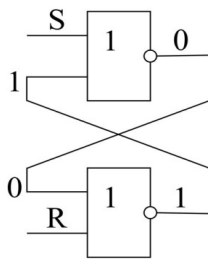


Рис. 14. Логическая схема триггера

Пример. Составим логическую схему для выражения $F = \neg(A \vee B) \& \neg C$ и заполним таблицу истинности.

В схеме должно быть 3 входа: А, В, С. Собираем логическую схему устройства из трех элементов: инверсия дизъюнкции, инверсия и конъюнкция (рис. 15).

Заполняем таблицу истинности для всех возможных комбинаций на входе схемы (табл. 23).

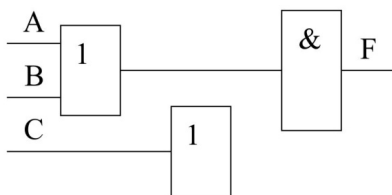


Рис. 15. Логическая схема

Таблица 23

Таблица истинности выражения $F = \neg(A \vee B) \& \neg C$

Исходные данные			Промежуточные данные			Результат
A	B	C	$A \vee B$	$\neg(A \vee B)$	$\neg C$	F
0	0	0	0	1	1	1
1	0	0	1	0	1	0
0	1	0	1	0	1	0
0	0	1	0	1	0	0
1	1	0	1	0	1	0
0	1	1	1	0	0	0
1	0	1	1	0	0	0
1	1	1	1	0	0	0

Получаем, что функция F на выходе будет равна 1 только тогда, когда все три входных сигнала равны нулю.

Из основных логических элементов складываются и более сложные логические узлы, в частности:

– регистры – устройства для хранения n -разрядных двоичных чисел и выполнения преобразований над ними;

– комбинационные преобразователи кодов (шифратор, дешифратор, мультиплексор – устройство, в котором имеется один выход, зависящий от комбинации нескольких входных сигналов);

– счетчики – устройства для подсчета количества поступающих импульсов и вывода результатов в двоичном коде;

– арифметико-логические узлы (сумматор, узел сравнения).

Из логических узлов строятся интегральные микросхемы более высокого уровня интеграции, например микропроцессоры, модули ОЗУ, контроллеры внешних устройств.

Логические операции с множествами

Множество можно представить как совокупность элементов, обладающих некоторым общим свойством. Объекты любой природы, составляющие множество, называют его элементами. Например, автомобиль с конкретным номером является элементом множества автомобилей и т. п.

Общее количество множеств не ограничено. Множества между собой могут *пересекаться*, *объединяться* или *содержать подмножества*. Для наглядного представления множеств и отношений между подмножествами используются диаграммы, которые называют *кругами Эйлера*, или кругами Эйлера – Венна.

При построении кругов Эйлера предполагается, что они выделяют подмножества из одного и того же большого множества, которое называют *универсальным* множеством. Предполагается, что универсальное множество содержит все используемые нами множества (обозначают символом U).

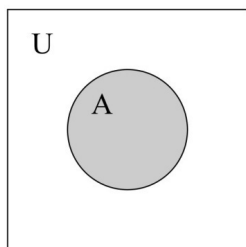


Рис. 16. Множество A в универсальном множестве U

Универсальное множество изображают в виде прямоугольника, а множества, входящие в универсальное множество, – в виде кругов внутри прямоугольника; при этом элементу множества соответствует точка внутри круга. Например, множество A в универсальном множестве U изображается с помощью затемненного круга, расположенного внутри прямоугольника (рис. 16).

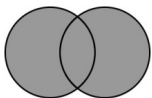
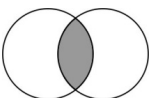
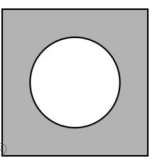
С помощью нескольких кругов Эйлера, расположенных в универсальном множестве, удобно иллюстрировать отношения между множествами и определять характер этих отношений на основе базовых логических операций. Каждый вид отношений между множествами соответствует определенной логической операции. Объединение множеств связано с

операцией дизъюнкции, пересечение множеств характеризуется операцией конъюнкции, а дополнение – инверсией. Примеры простых логических отношений между двумя множествами показаны в табл. 24.

Операции с множествами могут использоваться для решения практических задач, в частности, для обработки данных, полученных из поисковых систем. При поиске по ключевому слову система сообщает, какое количество результатов было найдено. Полученное количество результатов можно представить некоторым множеством web-страниц, на которых найдено ключевое слово. Таким образом, каждому ключевому слову соответствует множество, наполненное конечным значением элементов, а разным ключевым словам соответствуют разные множества. Соотношение и количественную связь между множествами можно найти, используя данные поисковых систем и графическое представление множеств.

Таблица 24

Графическое и логическое представление множеств

Отношения между множествами	Обозначение	Круги Эйлера	Логическое выражение
Объединение двух множеств – новое множество, состоящее из элементов, принадлежащих хотя бы одному из этих множеств	$A \cup B$		Дизъюнкция $A \vee B$
Пересечение двух множеств – новое множество, состоящее из элементов, принадлежащих одновременно обоим множествам	$A \cap B$		Конъюнкция $A \wedge B$
Дополнение множества A до множества U есть множество, состоящее из тех и только тех элементов U, которые не принадлежат A	не A		Инверсия $\neg A$

Пример. Рассчитайте, сколько результатов (в млн шт.) будет найдено поисковой системой по запросу *электрон* или *протон*, если известны результаты запросов, представленные в табл. 25.

Результаты поиска по ключевым словам

Запросы	Найдено результатов (млн шт.)
электрон	155
протон	56
протон и электрон	21

Используем графическое представление множеств в виде кругов Эйлера. Получим два пересекающихся множества, одно из которых относится к слову «электрон», другое – к слову «протон». Вводим обозначения n_1 , n_2 , n_3 для подмножеств пересекающихся множеств. Соотношения между подмножествами определяем из табл. 25 и составляем исходные уравнения, из которых требуется найти объединение множеств, т. е. сумму $n_1 + n_2 + n_3$.



Решаем систему уравнений и получаем, что объединение двух множеств $n_1 + n_2 + n_3 = 190$ млн шт.

История развития вычислительной техники

Создание вычислительных средств началось задолго до появления компьютеров. В истории вычислительной техники принято выделять три основных этапа: *домеханический*, *механический* и *электронно-вычислительный*. В этих этапах укладывается многовековой процесс развития человечества, который начинается со счета на пальцах и завершается созданием суперкомпьютеров.

Вычислительные средства домеханического и механического этапов

История создания счетных инструментов насчитывает более 30 тыс. лет. К числу древних счетных инструментов относятся следующие простые устройства.

1. Абак. Изготовленная из камня или других материалов плита с линиями или желобками, в которые помещались и использовались для счета камешки или шарики (рис. 17). Инструмент появился в Древнем Вавилоне в III тысячелетии до новой эры. Египет узнал об абаке в V в. до н. э., в страны Средиземноморья он пришел в V–VI вв. н. э. и получил распространение в Древней Греции и Древнем Риме.

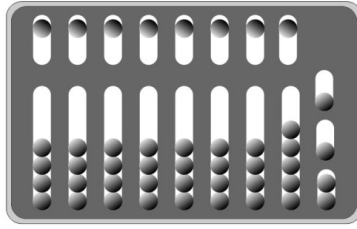


Рис. 17. Абак – древнейший счетный инструмент

2. Суаньпань. Китайская разновидность абака, представляющая собой прямоугольную раму, в которой протянуты параллельные проволоки или веревки с нанизанными на них шариками. Перпендикулярно натянутым проволокам суаньпань разделен перегородкой на два неравных отделения. В большом отделении на каждой проволоке нанизано пять шариков, а в меньшем – два. Время появления этого счетного устройства датируется VI в. Система счисления, используемая в суаньпане, – пятеричная (по числу пальцев на одной руке).

3. Соробан. Появился в Японии в Средние века. Его устройство выполнено по аналогии с китайским суаньпанем. Инструмент известен в настоящее время как японские счеты. Основное отличие соробана от китайского прототипа состоит в количестве спиц (веревек). В Японии соробан и по сей день используется в педагогике для начального обучения арифметическому счету.

4. Русские счеты – простое механическое устройство для выполнения арифметических расчетов. Представляет собой раму, имеющую некоторое количество спиц, на которые нанизаны костяшки. Количество костяшек – 10. Система счисления – десятичная. Мнения историков относительно появления этого устройства на Руси разделились. Одни считают, что счеты – это разновидность китайского суаньпана, пришедшего к нам с золотоордынскими татарами в XIV в. Другие полагают, что счеты произошли от прибора «дощаный щот», возникшего в Московском государстве в XVI в.

5. Логарифмическая линейка. Была сконструирована для выполнения расчетов в XVI–XVIII в. и использовалась в инженерных работах вплоть до 1980 г. Линейка явилась переходным инструментом на пути к механическим счетным устройствам. Логарифмической линейкой можно выполнять деление и умножение чисел, возводить их в квадрат и куб, извлекать корень, решать уравнения и др. Точность вычисления – 3 значащих цифры.

Несмотря на кажущуюся простоту, все инструменты домеханического этапа не только дожили до наших дней, но кое-где используются до настоящего времени.

На следующем механическом этапе развития появились первые счетные механизмы, создание и совершенствование которых продолжалось с XV до XIX в. К наиболее значимым достижениям механического этапа относятся следующие устройства.

1. Первым механическим счетным устройством принято считать тринадцатиразрядное суммирующее устройство на основе зубчатых колес, созданное

Леонардо да Винчи в виде рисунка в начале XV в. При жизни изобретателя это устройство, скорее всего, не создавалось. Уже в XX в. на основе найденного эскиза при участии фирмы ИВМ была создана действующая модель суммирующего устройства.

2. Первая действующая механическая счетная машина была изготовлена в 1623 г. профессором математики Вильгельмом Шиккардом. Машина существовала в единственном экземпляре. В машине Шиккарда впервые была использована принципиальная схема, которая послужила основой для последующих моделей счетных машин.

3. Счетное устройство, на котором можно было выполнять четыре арифметических действия, была создана немецким математиком и физиком Готфридом Лейбницем в 1673 г. Доработанное устройство, имеющее двенадцать рядов, было завершено в 1710 г.

4. Универсальная цифровая вычислительная машина, которая послужила прообразом ЭВМ, была разработана британским математиком и изобретателем Чарльзом Бэббиджем в 1833 г. Машина в автоматическом режиме выполняла сложные вычисления с высокой степенью точности.

5. Арифмометр с зубчаткой и переменным количеством зубцов разработал российский механик шведского происхождения Вильголт Однер. В 1873 г. он изготовил первый прототип, а в 1890 г. наладил промышленное производство арифмометров в Санкт-Петербурге. Позже его арифмометры выпускались в Германии и Швеции.

6. Арифмометр «Феликс», конструкция которого была разработана Однером, с 1925 г. до середины XX в. производился в Москве под торговой маркой «Феликс» и был широко распространенным механическим счетным устройством.

Все вычислительные устройства механического этапа были ручными и не могли функционировать без участия человека в процессе вычислений. Для выполнения каждой операции необходимо было набирать исходные данные, приводить в движение счетный механизм, результаты всех операций записывать. Высокая трудоемкость механических расчетов не позволяла решать вычислительные задачи большого объема.

Предпосылки создания электронных счетных устройств

В первой половине XX в. было положено начало развитию атомной физики и ракетной техники. Для реализации глобальных проектов потребовалось вычислительное устройство, которое по скорости работы значительно превосходило механические средства. Технические предпосылки для разработки нового счетного устройства были заложены в начале XX в.

В 1918–1919 гг. в Советской России и в Англии независимо был создан ламповый триггер – устройство, состоящее из двух ламповых триодов и сохраняющее одно из двух устойчивых состояний (рис. 18). Триггер был использован в качестве элементарной ячейки памяти, которая способна сохранять один бит данных. Возможность быстрого переключения из одного состояния в другое

была основным достоинством триггера, обеспечивающим высокое быстродействие первых вычислительных машин.

Первые образцы электронной вычислительной техники начали создаваться на *ламповой основе*. В середине XX в. группы инженеров независимо занимались разработкой ЭВМ в Германии, Великобритании, США и СССР. К ЭВМ первого поколения, которые оставили след в истории развития вычислительной техники, относятся следующие устройства:

- электронная счетная машина **COLOSSUS**, построенная в Англии в 1943 г. для решения специализированной задачи – расшифровки кодированных немецких сообщений; машина имела 1500 электронных ламп, ее быстродействие позволяло сократить время расшифровки от недель до нескольких часов;

- электронный интегратор и вычислитель **ЭНИАК**, построенный в США в 1945 г., применялся для расчетов режимов ядерных реакций; вычислитель содержал 18 000 электронных ламп, имел тактовую частоту 100 кГц, общий вес вычислителя составлял 27 тонн; вычисления производились в десятичной системе счисления с двадцатиразрядными числами;

- первая в СССР электронно-вычислительная машина **МЭСМ** была построена в 1951 г.; машина содержала 6000 электронных ламп и имела быстродействие 3000 операций в минуту, вычисления производились в двоичной системе счисления с шестнадцатиразрядными числами;

- электронно-вычислительная машина **БЭСМ** была создана в СССР в 1953 г.; по своим характеристикам она значительно превосходила машину МЭСМ и послужила основой для серийного выпуска вычислительных машин, который начался в 1957 г.

Первые вычислительные устройства были использованы для поиска и отработки основных технических решений и формирования принципов построения ЭВМ. Отличительными признаками первых ЭВМ было использование электронных ламп, которые определяли общий облик и технические характеристики систем. Переход с электронных ламп на транзисторы и в дальнейшем на микропроцессоры значительно ускорил прогресс в развитии ЭВМ.

Поколения вычислительных машин

Со второй половины прошлого века для ЭВМ начался период интенсивного развития, который продолжался до конца века и не заканчивается до настоящего времени. За этот относительно небольшой промежуток времени было создано *5 поколений* ЭВМ, каждому из которых отводится от 1 до 2 десятилетий. Краткая характеристика каждого из поколений представлена в табл. 26.

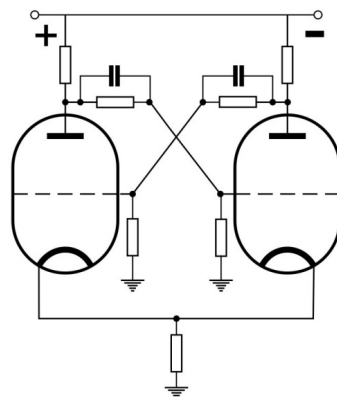


Рис. 18. Ламповый триггер

Первое поколение ЭВМ, созданное в середине 1940-х – начале 1950-х гг., было построено на *электронных лампах*, которые обеспечивали функционирование, но отличались высоким энергопотреблением и низкой производительностью. Существенные ограничения на ЭВМ первого поколения накладывали *перфокарты*, которые применялись в качестве *носителей информации*. Несмотря на ограничения в выборе основных элементов, в рамках первого поколения были созданы десятки опытных образцов, которые послужили основой для создания вычислителей последующих поколений.

Переход с электронных ламп на *транзисторы* в конце 1950-х значительно ускорил прогресс в развитии ЭВМ. Использование *полупроводниковых элементов* позволило существенно сократить энергопотребление, повысить быстродействие и значительно уменьшить размеры устройств. В рамках второго поколения трудоемкие перфокарты и *перфоленты* постепенно вытеснялись более эффективными носителями – *магнитной лентой* и *магнитным барабаном*.

Начало третьему поколению машин было положено изобретением *интегральной микросхемы* – миниатюрного электронного блока, содержащего в одном корпусе совокупность связанных между собой диодов, транзисторов и резисторов, изготовленных на единой полупроводниковой подложке. Появление микросхем сохранило общую тенденцию по уменьшению размеров и величины энергопотребления, а также способствовало бурному росту общего числа вычислительных машин.

Таблица 26

Поколения электронно-вычислительных машин

	Номер поколения				
	I	II	III	IV	V
Годы применения	1940–1950	конец 50-х	60-е	70–90-е	90-е и далее
Вычислитель или процессор	На электронных лампах	На транзисторах	Интегральная схема	Микропроцессор	Многоядерный микропроцессор
Количество ЭВМ в мире (шт.)	~10	~10 ³	~10 ⁴	~10 ⁶	~10 ⁸
Быстродействие (операций в секунду)	10 ³ –14 ⁴	10 ⁴ –10 ⁶	10 ⁵ –10 ⁷	10 ⁶ –10 ⁸	10 ⁸ –10 ¹⁰
Носитель информации	Перфокарта	Магнитная лента	Магнитный диск	Компакт-диск	Флэш-карта

Развитие электронной техники в рамках четвертого поколения привело к созданию *микропроцессора – большой интегральной схемы* (БИС), которая формировалась на одном кристалле кремния и выполняла все функции процессора большой ЭВМ. На базе одного из первых микропроцессоров в 1975 г. в США был разработан первый микрокомпьютер Альтаир 8800, который послужил прототипом персонального компьютера. К концу четвертого поколения в начале 1990-х гг. персональные компьютеры стали доминировать над другими типами вычислительных машин, их общее число в мире выросло до значения 10^8 .

Пятое поколение ЭВМ связано с дальнейшим развитием микропроцессора, в котором на одном кристалле кремния удалось расположить *два и более вычислительных ядер*, каждое из которых способно поддерживать до двух и более вычислительных потоков. Многоядерные процессоры существенно изменили представление об архитектуре ЭВМ, поскольку позволили реализовать компьютеры с *параллельно-векторной* архитектурой, в которой независимо обрабатывается множественный поток команд и множественный поток данных. Параллельно-векторный подход с использованием многоядерных процессоров, разработанный для компьютеров пятого поколения, лежит в основе функционирования *суперкомпьютеров*.

Суперкомпьютер – это мощный компьютер, который в период запуска лидирует по своим вычислительным возможностям среди множества известных компьютеров.

Принципы устройства и структура ЭВМ

Полученный в процессе разработки и эксплуатации ЭВМ опыт был обобщен в 1945 г. американским инженером Джоном фон Нейманом, который сформулировал основные принципы построения вычислительных машин. Краткое изложение принципов, предложенных фон Нейманом, сводится к следующему.

1. Принцип двоичного кодирования. В вычислительной технике необходимо использовать двоичную систему представления данных. В двоичной форме должны быть представлены как численные данные, так и все другие данные, такие как текст или графика.

2. Принцип однородности памяти. Программы и данные должны храниться в одной и той же памяти. При этом с командами можно выполнять такие же действия, как и с данными.

3. Принцип адресуемости памяти. Данные, которые хранятся в памяти вычислительной машины, необходимо делить на части и размещать их в пронумерованных ячейках.

4. Принцип последовательного программного управления. Программа состоит из последовательных команд. Все команды процессор должен выполнять по программе в автоматическом режиме.

5. Принцип жесткости архитектуры. В процессе работы аппаратная составляющая ЭВМ не меняется, а программа является изменяемой частью.

Сформулированные принципы были приняты и успешно реализованы во второй половине XX в. при разработке вычислительных машин последующих

поколений. На основе предложенных принципов была построена структура и сформированы основные правила функционирования ЭВМ. Совокупность структуры с основными правилами функционирования вычислительной системы стала именоваться *архитектурой* ЭВМ. Архитектура, основанная на принципах фон Неймана, получила название классической архитектуры или архитектуры фон Неймана. Отличительным признаком этой архитектуры является использование одного процессора и последовательное выполнение потока команд.

Архитектура вычислительных машин

Архитектура ЭВМ – концептуальная структура вычислительной машины, которая включает порядок обработки информации, методы преобразования информации в данные, а также принципы взаимодействия технических и программных средств.

Архитектура ЭВМ закладывает основные принципы, которые должны соблюдаться при построении вычислительных машин и создании соответствующего программного обеспечения.

Составляющие архитектуры ЭВМ складываются из:

- аппаратных средств, объединенных в некоторую *структуру*;
- системы используемых команд и форматов представления данных;
- программных средств, включая операционную систему и язык программирования.

Структура является одним из основных элементов архитектуры. Структура ЭВМ в классической архитектуре складывается из аппаратных составляющих или блоков, входящих в состав устройства, и связей между этими составляющими. Структурная схема ЭВМ классической архитектуры показана на рис. 19.

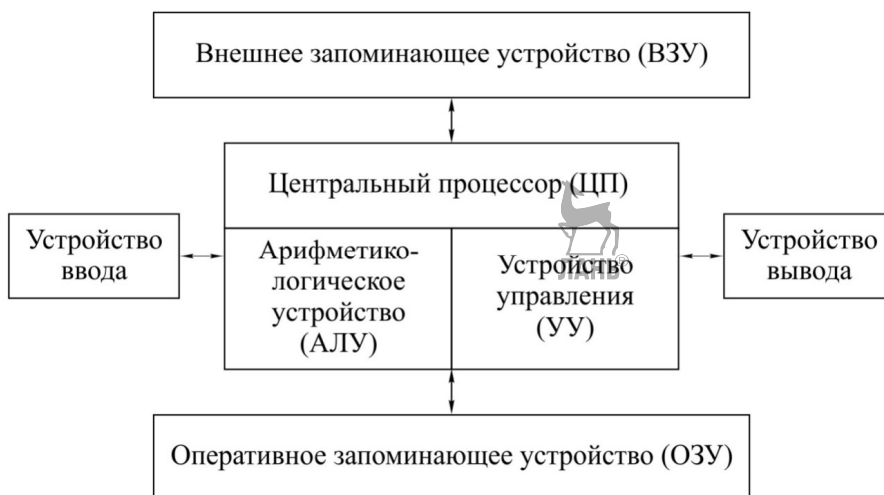


Рис. 19. Структурная схема ЭВМ

В состав ЭВМ, имеющей классическую архитектуру, входят:

- центральный процессор, предназначенный для управления работой всех блоков ЭВМ, а также выполнения арифметических и логических операций;
- арифметико-логическое устройство (АЛУ), выполняющее все операции с числовыми и символьными данными;
- устройство управления (УУ), формирующее управляющие импульсы и синхронизирующее работу всех блоков ЭВМ;
- внешнее запоминающее устройство (ВЗУ), которое используется для долговременного хранения данных;
- оперативное запоминающее устройство (ОЗУ), предназначенное для записи, хранения и считывания данных, участвующих в информационном или вычислительном процессе;
- устройства ввода и вывода, которые обеспечивают загрузку ЭВМ исходными данными и программами, а также вывод результатов обработки на монитор или принтер.

Аппаратные составляющие ЭВМ, показанные на рис. 19, объединены с помощью *магистральной шины*. Магистральная шина является основным структурным элементом, через который осуществляется взаимодействие между блоками вычислительной машины.

Магистральная шина присутствует во всех компьютерах классической архитектуры, в том числе в персональных компьютерах. Персональный компьютер любой категории сложности можно представить в виде простой блок-схемы, в которой все основные устройства, участвующие в обработке данных, соединяются с магистральной шиной через специальные *разъемы*, расположенные на *материнской плате* (рис. 20). Шина представляет собой канал передачи данных в виде многожильного кабеля или *системы проводников на печатной плате*.



Рис. 20. Упрощенная блок-схема персонального компьютера

В зависимости от назначения и количества устройств, подключенных к шине, могут быть реализованы различные схемы компьютера. Сборка каждой из схем выполняется на основе *принципа открытой архитектуры*, который предполагает простоту и доступность добавления, демонтажа или замены аппаратных элементов.

Принцип открытой архитектуры, реализованный в вычислительной технике, позволяет:

- выбирать конфигурацию вычислительного устройства в зависимости от запросов потребителя;
- расширять выбранную конфигурацию устройства по мере необходимости;
- модернизировать вычислительное устройство путем замены устаревших или вышедших из строя элементов.

Разработанные в середине XX в. принципы построения ЭВМ, основанные на классической архитектуре, в полной мере были использованы в последующие десятилетия для создания нескольких поколений вычислительных машин.

Основные характеристики вычислительных машин

К основным характеристикам ЭВМ относятся *быстродействие, емкость памяти, надежность, точность, система команд* и др.

Быстродействие – число команд, выполняемых ЭВМ в единицу времени. Быстродействие зависит от продолжительности процесса поиска нужной информации в памяти. Величина быстродействия может достигать миллиардов операций в секунду.

Емкость оперативной памяти (ОЗУ) и емкость внешней памяти (ВЗУ) определяется количеством данных, которое можно разместить в памяти ЭВМ. Емкость выражается в байтах. Емкость внешней памяти можно наращивать практически неограниченно.

Надежность – свойство ЭВМ выполнять возложенные функции в течение заданного промежутка времени, необходимого для решения задачи. Надежность ограничивается внезапными отказами или постепенной деградацией элементов ЭВМ. Численной характеристикой надежности может быть продолжительность работы на отказ или вероятность безотказной работы за определенное время.

Точность определяется количеством разрядов, необходимых для представления числа. В большинстве современных компьютеров используются 64-разрядные процессоры. Сравнительно недавно появились 128-разрядные процессоры.

Система команд – перечень команд, которые может выполнять процессор. Число команд процессора обычно находится в интервале от 10 до 100.

При выборе и эксплуатации ЭВМ практическое значение имеют и другие характеристики, в том числе *программная совместимость, энергопотребление, стоимость, габаритные размеры, вес*.

Главным показателем ЭВМ остается быстродействие. Для повышения быстродействия современных компьютеров используются разные подходы, одним из которых является поиск более совершенной архитектуры ЭВМ.

Поиск новой архитектуры ориентирован на создание многопроцессорных вычислительных машин. Многопроцессорность – использование в составе вычислительной системы двух и более физических процессоров. Определение многопроцессорности может меняться в зависимости от определения самих процессоров (много ядер в одном кристалле или множество чипов в одном корпусе). Производительность многопроцессорного компьютера равняется сумме производительностей его процессоров. В мощных компьютерах, предназначенных для сложных инженерных расчетов, устанавливают два или четыре процессора. В сверхмощных ЭВМ, предназначенных для решения задач в области ядерной физики или задач по прогнозированию погоды, количество процессоров может достигать нескольких десятков. С архитектурой сложных вычислительных систем, содержащей несколько процессоров, можно ознакомиться в [10].

Быстродействие компьютера зависит также от скорости обмена данными между оперативной памятью и процессором. Для повышения быстродействия постоянно ведутся поиски новых элементов оперативной памяти, которые отличаются высокой скоростью операций чтения и записи. Как правило, повышение быстродействия памяти связано с повышением ее стоимости, поэтому не все технические решения доходят до стадии практической реализации. Одним из реальных решений является построение многоуровневой системы памяти. Многоуровневая память состоит из двух или более частей. Основная и большая часть оперативной памяти строится на медленных и дешевых элементах, а дополнительная (кэш-память) состоит из быстродействующих и более дорогих элементов. Работа процессора строится таким образом, что он в первую очередь обращается к кэш-памяти и быстрее находит там необходимые данные.

В архитектуре современных компьютеров предусмотрены также каналы прямого доступа к оперативной памяти для обмена данными с устройствами ввода-вывода. Прямой обмен осуществляется без участия процессора, что позволяет освободить вычислительную систему от второстепенной работы и повысить тем самым быстродействие системы.

Персональный компьютер

Основным результатом развития вычислительной техники в XX в. можно считать создание и совершенствование высокоэффективной, компактной электронно-вычислительной машины, которую назвали персональным компьютером. Следует подчеркнуть, что этот результат был получен в течение относительно небольшого промежутка времени в противоположность механическому этапу развития, который занял несколько столетий.

Базовая конфигурация

В зависимости от назначения и количества подключенных устройств могут быть реализованы различные конфигурации компьютера. Среди возможных схемных решений выделяется широко распространенная базовая конфигурация компьютера (рис. 21), имеющая в своем составе:

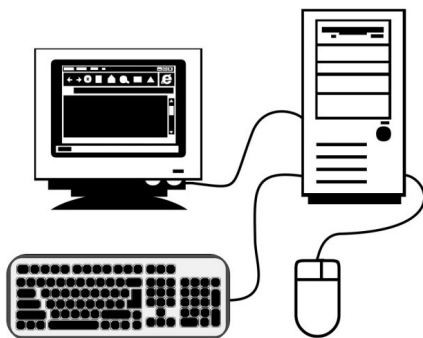


Рис. 21. Набор аппаратных составляющих компьютера

- системный блок – основная составляющая, в которой размещаются важнейшие аппаратные средства компьютера;
- монитор – устройство вывода, предназначенное для визуального отображения текстовых и графических данных;
- клавиатуру – клавишное устройство, предназначенное для ввода алфавитно-цифровых данных и команд управления;
- манипулятор мышь – координатное устройство, предназначенное для перемещения курсора и ввода управляющей информации.

В состав системного блока входят:

- материнская плата, содержащая *процессор*, а также набор элементов, необходимых для функционирования процессора;
- дисковод жесткого диска (винчестер) – устройство *внешней памяти* на магнитном носителе;

-
- контроллеры – электронные блоки, обеспечивающие связь периферийных устройств с материнской платой;
 - адаптеры – устройства, обеспечивающие согласование параметров входных и выходных сигналов;
 - блок питания;
 - органы управления (выключатели, кнопки, индикаторы питания и режимов работы).

Материнская плата имеет в своем составе:

- центральный процессор – основная микросхема, предназначенная для выполнения программного кода и управления работой всех устройств компьютера;
- микропроцессорный комплект (чипсет) – набор микросхем, выступающий в роли связующего элемента (моста), обеспечивающего взаимодействие процессора с различными устройствами памяти, устройствами ввода-вывода, контроллерами и др.;
- оперативную память – комплект микросхем, предназначенных для *временного хранения* данных;
- постоянное запоминающее устройство (ПЗУ) – микросхема для *длительного хранения* данных;
- системную шину – многоканальный проводник, который используется для обмена сигналами между внутренними компонентами компьютера, в том числе для передачи адресов ячеек оперативной памяти, для копирования данных из оперативной памяти в регистры процессора, для трансляции команд из оперативной памяти в процессор;
- разъемы (слоты), обеспечивающие подключение устройств к материнской плате.

Центральный процессор выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера. С конца прошлого века процессоры стали именовать *микропроцессорами*, так как они в процессе своего развития трансформировались в интегральную схему, выполненную на тонкой подложке из кристаллического кремния. Кремниевая подложка имеет прямоугольную форму с площадью несколько квадратных миллиметров, на которой плотно размещены схемы, реализующие все функции процессора.

Основные параметры процессора:

- тактовая частота определяет количество элементарных операций (тактов), выполняемых процессором за единицу времени;
- разрядность показывает количество бит данных, которые может принять и обработать процессор за один такт;
- коэффициент внутреннего умножения тактовой частоты – показатель различия тактовой частоты материнской платы и процессора;
- объем кэш-памяти – объем сверхоперативной буферной памяти, расположенной в процессоре и используемой для ускоренного ввода данных в процессор;

- система команд – совокупность команд, которые может выполнить данный процессор;
- количество транзисторов – параметр, отражающий функциональные возможности микросхемы.

Исторический процесс развития процессоров сопровождался постепенным улучшением всех его параметров. В 1965 г. сотрудник фирмы Intel (США) Гордон Мур высказал предположение о скорости роста производительности процессоров с течением времени. Впоследствии предположение трансформировалось в закон Мура [11], в котором подмечена общая тенденция – число транзисторов в процессоре должно удваиваться каждые два года. Отмеченная тенденция в развитии процессоров сохраняется с 1970 г.

Внутренняя и внешняя память компьютера

Внутренняя память включает все виды запоминающих устройств, расположенных на материнской плате. В состав внутренней памяти входят следующие устройства.

1. Оперативная память, или оперативное запоминающее устройство (ОЗУ), служит для хранения команд и данных, необходимых процессору для выполнения операций. Эта память позволяет обратиться к любой ячейке, поэтому называется также памятью с **прямым доступом (RAM-память)**. Отличается высоким быстродействием. К основному недостатку относится исчезновение данных после выключения электропитания.

2. Кэш-память, или сверхоперативная память, – очень быстрое запоминающее энергозависимое устройство, которое сохраняет текущие данные небольшого объема и предоставляет их процессору при необходимости. К недостаткам относится более сложный процесс изготовления и соответственно большая стоимость.

3. Специальная память имеет несколько составляющих:

- постоянная память, или постоянное запоминающее устройство (ПЗУ), предназначена только для чтения (ROM-память), энергонезависимая, содержание памяти «зашивается» при изготовлении и в процессе эксплуатации не меняется;

- перепрограммируемая постоянная память допускает многократную перезапись, энергонезависимая, содержит базовую систему ввода-вывода (BIOS), которая необходима для автоматического тестирования и загрузки операционной системы при включении компьютера;

- память с питанием от батарейки является разновидностью постоянной памяти и служит для хранения времени, даты и данных о конфигурации системы;

- видеопамять предназначена для хранения видеоданных, которые доступны одновременно процессору и монитору.

Внешняя память включает устройства (накопители), расположенные вне материнской платы и имеющие *носители* с разным принципом действия.

Носитель – это физическая среда или материальный объект, структура которых используется для хранения данных. В дисковом магнитном *накопителе* носителем является ферромагнитный слой на поверхности диска.

Накопители – это запоминающие устройства, предназначенные для длительного хранения больших объемов данных при отсутствии электропитания. В зависимости от принципиальной основы носителя различают накопители *магнитного, оптического и полупроводникового* типа.

Накопитель на магнитных дисках (жесткий диск, винчестер) – это основное устройство длительного хранения данных и программ, основанное на магнитном принципе записи. Магнитный накопитель собирается в герметичном корпусе, внутри которого соосно располагается несколько дисков. Каждый диск с обеих сторон покрыт ферромагнитным слоем, поверхность диска разделена на дорожки и сектора (отформатирована). Диск вращается относительно магнитных головок, с помощью которых производится сохранение и считывание данных (рис. 22). Данные сохраняются в форме дорожки из микроскопических намагниченных участков – доменов, намагниченность которых регистрируется как последовательность логических единиц.

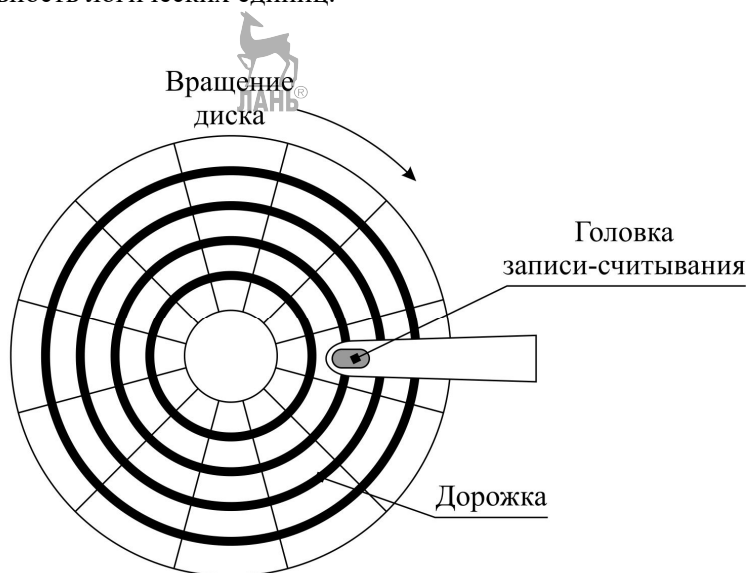


Рис. 22. Запись и считывание данных на магнитном диске

Накопитель на оптических дисках (CD-ROM) – это устройство для длительного хранения больших объемов данных, записанных с более высокой плотностью, чем на магнитном диске. Принцип действия основан на считывании данных с помощью лазерного луча, который отражается от поверхности вращающегося диска. В качестве носителя данных выступает металлизированная поверхность компакт-диска (CD), на которой нанесена спиральная дорожка (рис. 23). Цифровая запись на дорожке компакт-диска сохраняется в виде последовательности участков, которые называются pit (точка, углубление) и land (поверхность). Логическая единица кодируется переходом между углублением и поверхностью. Последовательность углублений, в которой закодирована запись,

наносят либо штамповкой с матрицы, либо прижиганием участков дорожки лучом лазера.

Компакт-диски изготавливаются из полипропилена, на поверхность которого наносится многослойное покрытие, включающее так называемый активный слой. В зависимости от соотношения покрытий, материала активного слоя, ширины дорожки различают компакт-диски разного устройства и назначения:

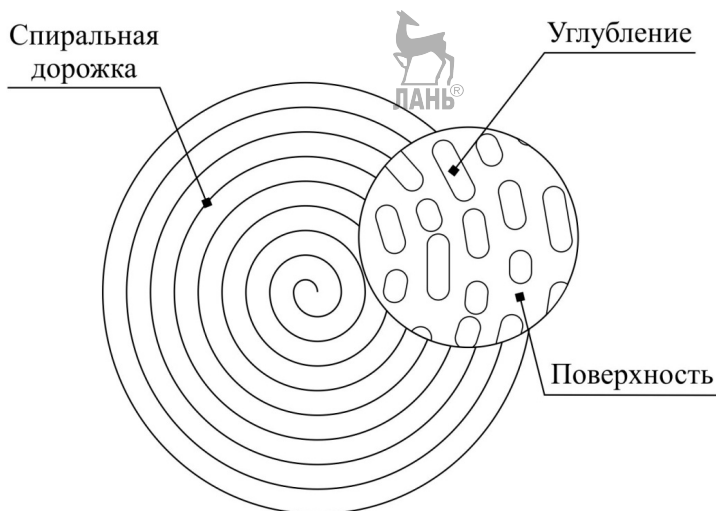


Рис. 23. Структура поверхности компакт-диска

- CD-R – диски, которые позволяют выполнить однократную запись и неограниченное количество считываний;
- CD-RW – диски для многократной записи, перезаписи и чтения данных;
- DVD – диски для многократной записи с повышенной плотностью данных.

Основной недостаток дисковых накопителей выражается в наличии электро-механического привода, который ограничивает надежность, ресурс, вес и размеры устройств.

Твердотельный накопитель – устройство полупроводникового типа для долговременного энергонезависимого хранения данных, которое реализовано на основе микросхемы *флэш-памяти*. В качестве носителя данных выступает массив полупроводниковых ячеек – транзисторов, расположенных внутри микросхемы. Принцип действия полупроводникового накопителя основан на записи и стирании электрического заряда в ячейке полупроводниковой структуры. Твердотельный накопитель в сравнении с дисковыми накопителями имеет существенные преимущества:

- в конструкции накопителя полностью отсутствуют вращающиеся части;
- повышена скорость чтения и записи данных;
- понижено электропотребление и чувствительность к внешним полям;

- полностью отсутствует шум;
- уменьшены габаритные размеры и вес.

Основным недостатком твердотельного накопителя является ограниченное количество циклов перезаписи. Однако ускоренные темпы развития технологии постепенно устраняют имеющиеся недостатки, что позволяет заменять устройства памяти предыдущих поколений на более совершенные твердотельные накопители.



Мониторы

Монитор является одной из основных составляющих базовой конфигурации компьютера. К распространенным типам причисляют следующие разновидности мониторов.

1. Мониторы на основе электронно-лучевой трубки относятся к предыдущему поколению мониторов. Формирование изображения основано на свечении люминофора под воздействием электронного пучка, который создается и управляется электродной системой внутри вакуумной колбы. Основные недостатки выражаются в необходимости поддержания вакуума и высокого напряжения для ускорения электронного пучка, в наличии вредных для организма излучений, а также в больших размерах и весе устройства.

2. Жидкокристаллические мониторы являются плоскпанельными устройствами, которые превосходят электронно-лучевые по многим параметрам, в частности, по толщине, весу, энергопотреблению. Принцип действия ЖК-мониторов основан на повороте плоскости поляризации света, проходящего через ячейку с жидким кристаллом. Угол поворота поляризации определяет интенсивность проходящего света и зависит от напряжения, приложенного к жидкокристаллической ячейке. Совокупность ячеек образует матричную структуру, которая характеризуется *размером и разрешающей способностью*. Размер монитора оценивается по длине его диагонали в дюймах, а разрешающая способность определяется количеством ячеек (пикселей) по горизонтали и вертикали монитора. Для монитора с размером 19 дюймов оптимальное разрешение составляет 1280×960.

3. Сенсорные мониторы – плоскпанельные координатные устройства, сочетающие функции вывода визуальных данных с выбором и вводом отдельных элементов данных или меню путем прикосновения к поверхности монитора. Сенсорные мониторы позволяют создать гибкий интуитивный интерфейс, который широко используется в карманных и планшетных устройствах.

В зависимости от принципа действия различают сенсорные мониторы следующих типов.

1. Резистивные мониторы отличаются многослойной поверхностью экрана, на которой в момент нажатия возникает контакт между слоями с одновременной регистрацией электрического сигнала и координат места контакта. Резистивные сенсорные экраны дешевы и стойки к загрязнению, недостатком является ограниченный срок службы.

2. Проекционно-емкостные мониторы основаны на регистрации координат при изменении емкости между сеткой электродов на экране и телом человека в момент прикосновения. Емкостные мониторы допускают использование толстых защитных стекол, поэтому применяются в уличных автоматах.

3. Поверхностно-акустические мониторы с пьезоэлектрическими излучателями, которые распространяют поверхностно-акустическую волну (ПАВ) по поверхности экрана. Прикосновение к экрану меняет картину распространения акустических колебаний, что позволяет вычислить координаты места возмущения. Экраны на ПАВ отличаются большим ресурсом, применяются в основном в игровых автоматах или в справочных системах.

4. Инфракрасные сенсорные панели покрыты сеткой горизонтальных и вертикальных инфракрасных лучей, которые при касании к монитору прерываются и выдают данные о координатах контакта. Такие панели не допускают загрязнения, поэтому имеют ограниченное применение, например, в образовательных учреждениях в качестве интерактивных панелей большого размера.

В связи с массовым распространением телефонов, смартфонов, планшетных компьютеров сенсорные мониторы являются самым востребованным средством вывода в мобильной компьютерной технике.

Клавиатура и манипулятор мышь

Клавиатура – это стандартное клавишное устройство, предназначенное для ввода алфавитно-цифровых данных и управляющих команд.

Клавиатуры имеют по 100 и более клавиш, которые размещаются по стандарту QWERTY (в верхнем левом углу алфавитной части клавиатуры находятся клавиши Q, W, E, R, T, Y).

Набор клавиш разбит на несколько функциональных групп:

- функциональные занимают блок от F1 до F12, функции этих клавиш определяются выполняемой программой и используемой операционной системой;
- алфавитно-цифровые относятся к блоку клавиш для ввода букв, цифр, знаков пунктуации и арифметических действий;
- управления курсором подают команды на передвижение курсора по экрану монитора;
- дополнительная цифровая панель дублирует клавиши цифрового блока в части ввода цифр и арифметических действий;
- специализированные, или служебные, используются для вспомогательных целей, например для создания комбинаций «горячих» клавиш, режимов вставки, и т. п. К этой группе относятся клавиши Shift, Caps Lock, Enter, Ctrl, Alt и др.

Манипулятор мышь – координатное устройство, предназначенное для преобразования механических движений мыши в перемещения курсора по экрану, а также для выбора данных и управляющих команд. Устройство создано для работы с программами, которые представлены в графическом интерфейсе.

Наибольшее распространение имеют оптические светодиодные мыши, которые в отличие от предыдущих конструктивных вариантов не требуют спе-

циального коврика, работают практически на всех поверхностях, за исключением зеркальных, практически не нуждаются в чистке.

Мышь может подвергаться индивидуальной настройке, к числу настраиваемых параметров относятся:

- чувствительность, характеризующая величину перемещения курсора по экрану при заданном перемещении мыши по горизонтали;
- чувствительность к двойному клику, которая определяет промежуток времени между двумя близкими кликами.

Классификация компьютеров

Категория персональных компьютеров имеет несколько видов классификаций. К числу главных отличительных признаков относится *тип платформы*, который отражает особенности архитектуры компьютера. По типу платформы различают:

- компьютеры на платформе IBM, которые относятся к наиболее распространенным устройствам; компьютеры этой платформы комплектуются микропроцессорами фирмы Intel и программным обеспечением фирмы Microsoft;
- компьютеры на платформе Apple представлены на рынке компьютерами фирмы Macintosh, в которых используют специальное аппаратное и программное обеспечение; в нашей стране имеют ограниченное применение (для подготовки качественных иллюстраций).

Одним из распространенных видов классификации компьютеров является спецификация PC99, которая представляет собой перечень рекомендаций и требований, необходимый для выдачи сертификата на компьютер. В соответствии с этой классификацией рекомендуется соблюдать определенные технические требования и придерживаться заданной конфигурации для каждой из следующих категорий компьютеров:

- пользовательский компьютер предназначен для личного пользования, имеет возможность подключения к сети Интернет;
- офисный компьютер предназначен для офисной деятельности в составе локальной сети, комплектуется из соображений минимальной цены;
- мобильный компьютер (ноутбук) – компактная конструкция, в которой сочетается автономный режим, средства связи, ограниченные размеры и масса;
- рабочая станция имеет расширенные возможности по решению задач, требующих интенсивных вычислений;
- игровой, или развлекательный, компьютер отличается улучшенными средствами воспроизведения графики и звука, в том числе при подключении дополнительных устройств.

Введенное в рамках этой спецификации цифровое кодирование разъемов принято для всех типов компьютеров, предлагаемых на рынке.

На практике используют также классификацию компьютеров *по уровню специализации*, в которой выделяют:

- универсальные компьютеры, которые позволяют построить любую конфигурацию для конкретного применения, например для работы с графикой;

– специализированные компьютеры, которые предназначены для решения конкретных задач, например для создания бортовой вычислительной системы.

Для разделения компьютеров по величине удобно использовать классификацию *по размеру*, в которой выделяются:

– настольные модели наиболее распространены и приспособлены для изменения конфигурации;

– портативные модели предназначены для оперативного перемещения и организации компьютерной связи; к этой категории относятся ноутбуки и планшетные компьютеры;

– карманные модели обеспечивают условия для компактного хранения и быстрого доступа к данным.

К отдельной быстро растущей категории относятся планшетные компьютеры, в состав этой группы входят различные типы компьютеров с сенсорным экраном. Планшетным компьютером можно управлять прикосновениями руки или стилуса – компьютерного пера. При этом клавиатура и мышь не всегда могут быть доступны.

К планшетным компьютерам относят следующие устройства.

1. Планшетный персональный компьютер – разновидность ноутбука, оборудованная сенсорным экраном и позволяющая работать при помощи пальцев как с использованием, так и без использования клавиатуры и мыши.

2. Интернет-планшет – небольшой однокорпусный планшетный компьютер (обычно диагональ не более 11 дюймов), предназначенный для автономной работы, в том числе в режиме постоянного беспроводного подключения к сети Интернет. Для управления интернет-планшетом используется сенсорный экран, взаимодействие с которым осуществляется при помощи пальцев без использования физической клавиатуры и мыши.

3. Мобильное интернет-устройство – компактный ПК с размером экрана до 7 дюймов, предназначенный для работы с web-сервисами.

4. Электронная книга – компактное специализированное планшетное устройство, предназначенное для отображения текстовой информации в электронном виде. Основным отличием данной группы устройств является ограниченная функциональность и большее время автономной работы.

5. Смартфон – карманный персональный компьютер с функциями мобильного телефона. Смартфоны отличаются от обычных мобильных телефонов наличием широких программных и аппаратных возможностей:

а) программная часть находится под управлением мобильной операционной системы, которая значительно расширяет функциональные возможности устройства, в том числе за счет установки дополнительных прикладных программ;

б) в аппаратной части используются современные многоядерные процессоры и устройства оперативной памяти, которые обеспечивают высокое быстродействие системы на уровне компьютеров настольного типа;

с) коммуникационные возможности устройства обеспечивают оперативную связь с глобальной информационной системой по каналам с высокой пропускной

способностью, в том числе подключение и использование систем Глонасс, GPS (системы навигации);

d) имеются возможности установки программных продуктов, которые позволяют работать с файлами разных форматов, в том числе с текстовыми документами, аудио- и видеофайлами, электронными таблицами, презентациями и др.;

e) синхронизация данных с облачными сервисами обеспечивает практически неограниченный по объему внешний накопитель данных;

f) подключение системы электронных платежей позволяет производить расчеты с финансовыми организациями, в частности, с организациями, которые оказывают коммунальные услуги, и др.

Сочетание компактности устройства с возможностью оперативного подключения к сервисам сети Интернет и расширения числа используемых приложений обеспечивает смартфонам широкую популярность и численное превосходство над другими типами компьютеров.



Программное обеспечение компьютера

Программные средства, или программное обеспечение, являются неотъемлемой частью компьютера, это логическое продолжение технического устройства, с помощью которого расширяются возможности и сферы использования компьютера.

Для функционирования компьютера используется комплекс программного обеспечения (ПО), который делится на 3 крупные категории: *системное ПО, системы программирования, прикладное ПО* (рис. 24).



Рис. 24. Структура программного обеспечения компьютера

Каждая категория программного обеспечения предназначена для выполнения специализированного набора функций.

1. Системное программное обеспечение включает комплекс программ, необходимый для создания операционной среды функционирования других программ, обеспечения безопасной и надежной работы компьютера, проведения диагностики и профилактики аппаратуры и др.

2. Системы программирования, или инструментальные системы, объединяют все средства, необходимые для производства программ и формирования их в машинном коде.

3. Прикладное программное обеспечение включает множество программных продуктов, каждый из которых предназначен для выполнения конкретных работ кругом пользователей, объединенных по профессиональному признаку.

Системное программное обеспечение

Программные продукты, которые относятся к категории системного ПО, ориентированы на *квалифицированных компьютерных пользователей*. Системным ПО пользуются системные программисты или администраторы сети. Основное назначение системного ПО – обеспечение эффективной работы аппаратного комплекса.

Системное ПО – это комплекс программ, предназначенный для управления микропроцессором и устройствами, обеспечивающими его функционирование,

а также для организации взаимодействия между аппаратными и программными средствами компьютера.

Системное ПО принято делить на *базовое* и *сервисное*.

Базовое системное ПО включает минимальный набор программных средств, обеспечивающих работу компьютера, в том числе:

- операционные системы (основная и наиболее важная составляющая);
- драйверы в составе операционной системы;
- интерфейсные оболочки, обеспечивающие взаимодействие пользователя с операционной системой;
- системы управления файлами.

Сервисное системное ПО объединяет программы, которые расширяют возможности базового программного обеспечения. Программы сервисного ПО часто называют *утилитами*, или *служебными программами*. Основное назначение служебных программ – организация более удобной среды для работы пользователя. Ниже приведены примеры некоторых сервисных программ.

1. Программы-упаковщики осуществляют архивирование и разархивирование файлов с целью экономии памяти компьютера.

2. Программы создания резервных копий отвечают за копирование данных с одного носителя на другой.

3. Коммуникационные программы обеспечивают обмен данными между сетевыми компьютерами.

4. Программы диагностики осуществляют проверку конфигурации, объема памяти, работоспособности дисков и т. п.

5. Программы оптимизации дисков производят оптимальное распределение файлов на диске в целях рационального использования дискового пространства и ускорения доступа.

Операционные системы

Операционная система (ОС) устанавливается на внешнем носителе компьютера, при включении загружается в оперативную память и выполняет огромный незаметный для пользователя объем работы. Если условно принять все программное обеспечение компьютера за айсберг, то операционную систему можно сравнить с подводной частью айсберга.

Операционная система – комплекс взаимосвязанных программ, предназначенных для тестирования и управления ресурсами компьютера, а также для организации процесса взаимодействия пользователя с программным и аппаратным обеспечением (организации пользовательского интерфейса).

Любая ОС опирается на базовую систему ввода-вывода (BIOS) и одновременно служит опорой для прикладных программных продуктов.

ОС взаимодействует с пользователем в диалоговом режиме. Диалоговый режим основан на исполнении команд, которые подает пользователь. Операционная система способна отреагировать на команду пользователя, прервать текущую работу и переключиться на другой режим работы. Динамичное переключение между режимами обеспечивает гибкий порядок работы.

Основные операции, которые выполняет ОС:

- организация и поддержка интерфейса пользователя;
- загрузка программ в оперативную память и контроль за порядком их выполнения;
- поиск данных на внутренних и внешних носителях;
- реализация штатных и нештатных ситуаций, например контроль процесса окончания работ или восстановление данных в случае ошибочных действий пользователя;
- выполнение функций, обеспеченных сервисными программами, в том числе копирование, архивирование и многое другое.

Различают текстово-символьные и графические операционные системы.

Текстово-символьные системы (неграфические) основаны на ручном вводе командной строки в текстовом режиме. Система построена таким образом, что после ввода командной строки и нажатия клавиши Enter начинается исполнение команды. Этот вид интерфейса использован в семействе операционных систем MS-DOS.

Графические системы в дополнение к клавиатуре в качестве управляющего устройства имеют *манипулятор мышью*.

Указатель мыши является активным элементом, с помощью которого осуществляется управление компьютером. Этот вид интерфейса реализован для семейства операционных систем типа Windows и используется во всех современных системах.

В структуре ОС можно выделить следующие основные компоненты.

1. Ядро – группа основных системных программ, которая управляет аппаратными средствами компьютера и ходом выполнения программ. В состав ядра входят основные программные средства:

- модуль управления системой прерываний;
- средства управления оперативной памятью;
- средства кэширования дисков;
- средства резервного копирования.

2. Система управления файлами – программа, которая обеспечивает размещение сохраняемых данных на запоминающих устройствах и организацию удобного доступа к созданным файлам.

3. Интерфейсная оболочка ОС – программа, которая организует удобное взаимодействие пользователя с компьютером.

4. Служебные программы – программы, входящие в состав ОС, такие как программы-драйверы, управляющие внешними устройствами; при функционировании компьютера эти программы постоянно находятся в оперативной памяти.

Начало развитию операционных систем было положено в 80-х гг. прошлого века. В первых компьютерах операционная система отсутствовала, и управление осуществлялось на основе простейших языков программирования, которые содержались в ПЗУ, позволяли загружать программы и управлять программой с клавиатуры.

В начале 1980-х гг. с появлением магнитных дисков была написана первая программа, в которой каждому названию файла присваивались определенные номера секторов. Так появилась *дискровая операционная система* – DOS. Далее дискровой системе поручили решать более сложные задачи, например записывать, копировать и удалять файлы, исключать повторяющиеся названия и т. п. В связи с необходимостью структурирования возрастающего объема данных появились каталоги файлов, для которых потребовалось создать систему управления файлами.

Фирма Microsoft (США) с 1981 по 1995 г., постепенно усложняя, выпустила несколько версий дискровой операционной системы, начиная с MS-DOS 1.0 и завершая версией MS-DOS 6.22. Дальнейшее развитие дискровых операционных систем сдерживалось существенными недостатками этих систем:

- системы были построены на текстово-символьном интерфейсе (набор команд с клавиатуры);
- на изучение последних версий ОС и решение компьютерных задач требовалось много времени.

Для устранения недостатков ОС была создана *программа-оболочка* – специальная надстройка операционной системы, которая предназначена для упрощения работы пользователя. Упрощение заключалось в выборе команд или файлов из имеющегося списка вместо набора символов с клавиатуры. Самая известная программа-оболочка – Norton Commander, которая наглядно показывает всю файловую структуру, позволяет не запоминать команды, а работать с ними через строку меню. Несмотря на появление более совершенных программ, Norton Commander до настоящего времени используется на многих компьютерах.

На смену текстовой программе-оболочке типа Norton Commander пришла графическая оболочка операционной системы. Фирмой Microsoft сначала была создана графическая оболочка в системе Windows 1.0, затем появились ее версии с номерами 2.0; 3.0; 3.1; 3.11. Оболочка Windows, которую называли *средой*, работала под управлением системы MS-DOS и не являлась самостоятельной операционной системой.

Среда Windows изначально отличалась следующими признаками:

- графический интерфейс;
- многозадачность;
- единый программный интерфейс;
- единый интерфейс пользователя;
- единый аппаратно-программный интерфейс.

Распространение графических принципов построения программ на всю операционную систему в целом привело к созданию графической операционной системы. Лидером в разработке графических систем признана фирма Microsoft, которая с начала 1990-х гг. создала целый ряд ОС, среди которых можно выделить наиболее известные версии (табл. 27).

Версии операционной системы Windows

Наименование	Основное назначение
Windows NT	первая графическая система начала 1990-х гг., которая дала начало большому семейству операционных систем
Windows 95	широко распространенная версия 1990-х гг., отличается рядом удачных решений
Windows 98	доработанная и усовершенствованная 95-я версия
Windows 2000	система с широкой областью применения в серверном и клиентском исполнении, для корпоративного клиента
Windows XP	наиболее распространенная версия для настольных компьютеров с 2002 г.
Windows 7	популярная среди пользователей система, выпущенная на рынок в 2009 г.
Windows 8	внедряется с 2012 г., включает возможности работы с сенсорными экранами, имеет плиточный интерфейс
Windows 10	внедряется с 2015 г. в качестве единой системы для персональных компьютеров, планшетов и смартфонов

Все версии операционных систем фирмы Microsoft относятся к категории лицензионных и продаются на рынке программных продуктов по коммерческим ценам.

Операционные системы Windows и Linux

Среди графических операционных систем самыми распространенными для офисных и домашних компьютеров являются операционные системы Windows фирмы Microsoft. К *достоинствам* ОС Windows относятся, в частности:

- распространенность и доступность;
- широкие функциональные возможности;
- наличие большого количества приложений и дополнений.

Наряду с достоинствами ОС Windows имеет ряд *недостатков*, к числу которых относятся:

- высокая коммерческая цена;
- закрытый исходный код программы;
- низкий уровень защиты от вирусов;
- возможность зависания программ и замедления в работе;
- высокие требования к аппаратной части компьютера.

Альтернативой ОС Windows выступает распространяемая бесплатно операционная система Linux. Автором исходного варианта ОС Linux является Линус Торвалдс (Финляндия), который опубликовал основные принципы построения своей системы в сети Интернет. После публикации к развитию системы подклю-

чилось большое число заинтересованных разработчиков и пользователей. Коллективные усилия независимых разработчиков превратили ОС Linux в современную, устойчивую и быстроразвивающуюся систему, которая имеет статус свободного ПО и распространяется бесплатно. ОС Linux отличается рядом *преимуществ*:

- обладает многими возможностями, которые присущи современным полнофункциональным операционным системам;
- предоставляет пользователю исходные тексты программ;
- имеет высокое быстродействие и надежность;
- практически не поддается вирусному заражению;
- имеет хорошую учебную документацию.

Основным *недостатком* ОС Linux является некоторое отставание от полноценных операционных систем в части функциональных возможностей. Несмотря на имеющиеся недостатки, число пользователей ОС Linux быстро увеличивается. В нашей стране решение об использовании свободного программного обеспечения принято на уровне правительства. В декабре 2010 г. подписано Распоряжение Правительства РФ о переходе государственных бюджетных организаций, а также органов федеральной исполнительной власти на свободное программное обеспечение. Переход заключается в постепенной замене имеющегося программного обеспечения с закрытым исходным кодом на свободные аналоги с целью повышения безопасности и снижения зависимости от производителя-разработчика.

Системы управления файлами

Система управления файлами входит в состав ОС и обеспечивает размещение сохраняемых данных на внешних носителях. Система построена в форме таблицы, в которую заносятся имя файла и место его размещения. Для обозначения места память накопителя разбивается на *сектора*, каждый сектор нумеруется, группа секторов объединяется в *кластер*, размер которого строго не фиксирован. Каждый кластер получает свой *адрес*, который используется при обращении к данным. Данные, содержащие имя файла и адреса кластеров, на которых размещается файл, собираются в таблицах файловой структуры.

Для операционных систем типа MS-DOS таблица размещения файлов состоит из 16-разрядных полей и называется FAT16. Длина кластера в этой системе обычно составляет 64 сектора, или 32 Кбайт. Основные недостатки системы FAT16 – нерациональный расход дискового пространства и невозможность работы с большими объемами данных.

Для операционных систем типа Windows таблица размещения файлов была заменена на 32-разрядную и получила название FAT32. Длина кластера в этой системе сократилась до 8 секторов или 4 Кбайт, что позволяло более рационально использовать дисковое пространство.

При переходе на операционную систему Windows NT файловая система FAT32 была заменена на 64-разрядную систему NTFS. В новой системе появи-

лась служебная информация, которая собрана в главной таблице файлов, а размер кластера по умолчанию составляет от 512 байт до 64 Кбайт. Система NTFS предоставляет пользователю возможности по поиску, созданию, копированию, перемещению, переименованию и удалению файлов и папок, содержащих группы файлов.

Поиск одного файла осуществляется по его имени, для поиска и выделения группы файлов, имеющих общие фрагменты в наименовании, используется *маска имени*. В маске указываются символы, которые присутствуют во всех файлах группы, а для обозначения оставшейся части используют специальные символы:

– символ * предназначен для обозначения любого количества символов, в том числе нулевого;

– символ ? предназначен для обозначения единственного символа.

Использование маски позволяет выделять однотипные группы файлов:

– маска *.txt выделит все файлы с расширением .txt;

– маска ???.* определит все файлы, имена которых состоят из трех символов.

Создание файлов сопровождается обязательным присвоением собственного *имени*. Имя файла может содержать до 260 латинских или русских символов и состоит из двух частей, разделенных точкой. Первая часть имени присваивается пользователем и несет, как правило, смысловую нагрузку. Вторая часть называется расширением, состоит из 3–4 символов и указывает на принадлежность файла к определенной программе. Расширение имени файла приписывается системой автоматически. Кроме расширения система фиксирует некоторые свойства файлов, такие как место расположения на диске, объем файла в байтах, дату и время создания файла.

В дополнение к основным свойствам система управления файлами сохраняет *атрибуты* файла или его дополнительные параметры. Атрибуты файла отражают особенности его использования.

1. Только для чтения. Атрибут означает, что файл не предназначен для внесения изменений.

2. Скрытый. Операционная система не отображает на экране файлы с данным атрибутом.

3. Системный. Операционная система не может изменить файлы с данным атрибутом.

Система управления файлами предоставляет также возможности по удалению файлов, различают три режима удаления данных.

1. Удаление файла. В этом режиме файл перемещается в папку Корзина, но на уровне физической файловой структуры не происходит изменение, меняется только путь к файлу (через папку Корзина).

2. Уничтожение файла. Выполняется при очистке Корзины. В таблице файлов помечается как удаленный, но физически остается на прежнем месте. При этом в свободные кластеры можно записать новый файл.

3. Стирание файла. Выполняется специальными программами путем занесения в свободные кластеры новых данных.

Служебные программы и стандартные приложения ОС

В состав современной операционной системы включаются служебные программы, которые выполняют важные и полезные функции.

Проверка дисков используется для устранения двух видов ошибок.

1. Логические ошибки, которые выражаются в нарушениях файловой структуры и устраняются средствами операционной системы.

2. Физические ошибки, которые заключаются в физическом повреждении носителя, например ферромагнитного слоя диска, в определенных секторах. Поврежденные сектора операционная система исключает из активной работы.

Управление оперативной памятью используется в случае, когда емкости оперативной памяти не хватает для работы некоторой программы. Для расширения памяти операционная система создает на жестком диске дополнительную виртуальную память, которая реализуется в виде файла подкачки, и организует обмен данными между оперативной памятью и файлом подкачки. В конечном итоге это расширяет возможности и повышает производительность системы. Размером файла подкачки можно управлять.

Кэширование дисков обеспечивает сохранение части использованных данных в блоке сверхоперативной кэш-памяти и выдачу сохраненных данных процессору при повторном обращении. Обращение к сверхоперативной памяти повышает быстродействие системы.

Кроме перечисленных функций служебные программы современной операционной системы обеспечивают широкий спектр дополнительных возможностей, в частности:

- резервное копирование;
- поддержку функционирования локальной сети;
- обеспечение доступа к службам сети Интернет;
- возможность создания сервера сети Интернет;
- защиту данных от несанкционированного доступа;
- возможность работы различных пользователей на одном компьютере;
- поддержку функционирования мобильных устройств и др.

Современная операционная система содержит также определенный набор прикладных программ для решения простейших задач, которые не требуют привлечения мощных программных средств.

Простые прикладные программы в составе операционной системы называют стандартными приложениями. В силу относительной простоты эти программы принято использовать в качестве учебных, поскольку знание приемов работы со стандартными приложениями способствует освоению специализированных программных средств. В операционную систему Windows включены следующие стандартные приложения.

Калькулятор – компьютерная версия обычного карманного или настольного электронного калькулятора. Отличительная черта этой программы – возможность безбумажной технологии выполнения расчетов за счет выведения на экран неограниченного количества калькуляторов (для параллельного выполнения расчетов или хранения промежуточных результатов вычислений). Наряду

с простыми вычислениями с помощью калькулятора можно выполнять более сложные расчеты, в том числе:

- вычисление основных и обратных алгебраических, тригонометрических и гиперболических функций;
- проведение статистических расчетов;
- вычисления в двоичной, восьмеричной, десятичной, шестнадцатеричной системах счисления;
- выполнение логических операций и др.

Блокнот – простейший текстовый редактор, предназначенный для создания и редактирования текстовых файлов с расширением .txt, которые не требуют форматирования и не превышают по объему 64 КБайт. Блокнот удобно использовать для ведения коротких записей или пометок, редактирования командных файлов, создания web-страниц.

Paint – средство для рисования или однооконный редактор растровой графики. Программа предназначена для создания и редактирования графических файлов различной сложности с расширением .bmp, .jpg, .gif. Paint можно использовать для создания цветных и черно-белых рисунков, схем, чертежей, диаграмм, надписей, изображений, используемых в качестве фона рабочего стола, а также для просмотра и правки отсканированных изображений и объектов, созданных в других приложениях.

WordPad – базовый однооконный текстовый редактор, позволяющий управлять содержимым документа (создавать, редактировать), а также проводить простейшие операции форматирования, связывания и внедрения данных из других документов. WordPad поддерживает форматы rtf, doc, txt. WordPad можно использовать для создания и редактирования простых текстовых документов, создания и редактирования документов большого объема со сложным форматированием и рисунками, обмена данными между приложениями и т. д.

Проводник – служебная программа с графическим интерфейсом для доступа к файловой структуре в операционной системе Windows, используется для навигации по файловой структуре компьютера и проведения операций с файлами. Проводник удобно использовать для просмотра иерархической структуры файлов, перемещения файлов по папкам, копирования и перемещения файлов и папок между разными накопителями.



Системы и технологии программирования

История развития технологий программирования

В истории развития языков программирования выделяется несколько этапов [12].

Этап 1 (от появления первых ЭВМ до середины 1960-х) характеризуется полным отсутствием правил программирования. Первые программы имели простую структуру и были написаны на машинном языке (в двоичном коде). Перед программистом возникали большие сложности по формированию двоичных кодов и отслеживанию порядка выполняемых операций. Дополнительная трудность была связана с определением местонахождения данных. Для упрощения процесса программирования отдельные группы кода были заменены на символические имена, так появился *первый язык* программирования, который назвали Ассемблер (от *англ.* assembler – сборщик). Ассемблер относится к машинно-ориентированным языкам низкого уровня и отличается привязкой к системе команд конкретного процессора.

Дальнейшее снижение трудоемкости процесса программирования произошло после появления языков высокого уровня, первыми из которых были языки Fortran и Algol. Языки высокого уровня были построены на основе алгоритмических конструкций, которые удобны для записи, более универсальны и не связаны с системой команд конкретного процессора. Все это снизило уровень детализации и повысило сложность разрабатываемых программ.

В языках высокого уровня появилось средство, которое позволяло оперировать подпрограммами. Подпрограммы можно было сохранять и использовать при создании других программ. Для поддержки этой технологии были созданы библиотеки расчетных и служебных подпрограмм. Типичная программа того времени состояла из основной программы с набором глобальных данных и ряда подключенных к ней библиотечных подпрограмм. Первоначально все глобальные данные собирались в основной программе, но со временем стало ясно, что более удачным решением будет введение локальных данных для каждой подпрограммы.

При использовании подпрограмм разработка велась по принципу «снизу – вверх», т. е. сначала проектировали подпрограммы, а потом их использовали для создания сложной программы. Недостатки этого подхода проявлялись в сложности стыковки созданных подпрограмм и в значительном затягивании работ по реализации больших проектов.

Этап 2 (1960–1970-е гг.) связан с появлением структурного подхода к программированию. В основе структурного подхода лежит разбиение сложных программ на части (декомпозиция) и реализация отдельных частей в виде подпрограмм. При этом процесс проектирования строился под общей идеей и формировался по принципу «сверху – вниз». Структурный подход позволял представить решаемую задачу в виде иерархии подчиненных задач. Одновременно с этим вводились формальные модели алгоритмов, правила описания алгоритмов, а также метод пошаговой детализации алгоритмов.

Принципы структурного программирования были реализованы при создании процедурных языков программирования, например языка Pascal. В таких языках

появились операторы, которые обеспечивали перемещение по структуре, в частности, оператор передачи управления.

В результате развития структурного программирования появилась технология модульного программирования. Архитектура модульного программирования предполагает создание в составе программы модулей, состоящих из групп подпрограмм, использующих одни и те же глобальные данные. Этот подход позволил упростить процесс программирования, поскольку позволял каждому из программистов разрабатывать свои модули независимо от других. Сочетание структурного подхода с модульным программированием позволило получать достаточно сложные и надежные программы. Но при значительном увеличении числа проектируемых модулей появлялись ошибки, связанные с функционированием межмодульного интерфейса. Для разработки программного обеспечения с большим числом модулей потребовалось сделать переход к третьему этапу развития программного обеспечения.

Этап 3 (середина 1980-х – конец 1990-х) характеризуется появлением объектно-ориентированного программирования. Технология объектно-ориентированного программирования основана на представлении программы в виде совокупности объектов, каждый из которых является *экземпляром определенного класса*, а классы в свою очередь образуют *иерархическую структуру* с возможностью *наследования свойств объектов*. При этом взаимодействие между объектами основано на передаче сообщений.

Выбранный подход обеспечил более полную декомпозицию программного обеспечения и существенно *упростил* работу, поскольку позволил практически независимо вести разработку отдельных подпрограмм. Дополнительное упрощение было достигнуто путем введения новых способов организации программ, основанных на *механизмах наследования, полиморфизма, композиции и наполнения*, которые позволили конструировать сложные объекты из сравнительно простых.

В рамках этого подхода был разработан графический интерфейс используемых средств, с помощью которого была реализована технология визуального программирования, в частности, в системах Delphi, Visual C++. Привлечение визуальных средств позволило по упрощенной схеме создавать заготовку будущей программы, что привело к дополнительному уменьшению трудоемкости проектирования.

Основные недостатки объектного подхода связаны со сложностью компиляции (перевод в двоичные коды) созданных объектов и последующей компоновки скомпилированных блоков-модулей в единое целое. Проблема компоновки в свою очередь связана с недостаточно отработанным механизмом обмена данными между модулями.

Этап 4 (с середины 1990-х до нашего времени) связан с переходом на компонентный подход и возникновением CASE-технологий. Основу компонентного подхода составляет построение программного обеспечения из отдельно существующих частей, взаимодействие между которыми организовано на основе двоичных интерфейсов стандартного типа. Понятие объектов при этом трансформируется в *объект-компоненты*, которые сохраняются в двоичном коде и могут использоваться в языках программирования, которые поддержи-

вают эту технологию. Сформированный подход позволяет собирать библиотеки разработанных компонентов и использовать наработанные результаты при разработке новых продуктов. На сегодняшний день местом сбора компонентов являются серверы сети Интернет.

Компонентный подход лежит в основе двух типов технологий, первая из которых разработана на базе компонентной модели объекта COM (Component Object Model), а вторая связана с созданием распределенных приложений CORBA (Common Object Request Broker Architecture – общая архитектура с посредником обработки запросов объектов). В этих технологиях используются общие принципы компонентного подхода и различаются некоторые детали реализации.

Современный этап развития программирования характеризуется активным привлечением средств автоматизации разработки, без которых трудно создавать сложное программное обеспечение. Комплекс автоматизированных технологий, которые обеспечивают разработку и сопровождение программного обеспечения, называют CASE-технологиями (Computer-Aided Software/System Engineering – разработка программного обеспечения программных систем с использованием компьютерной поддержки). Областью применения CASE-технологий в первую очередь являются программы высокой сложности, при разработке которых программисты не могут фиксировать все особенности создаваемого продукта и вынуждены использовать средства автоматизации.

Этапы решения задач на программной основе

Решение сложной задачи с использованием компьютерных программ предполагает выполнение определенного круга действий. Эти действия можно разложить на этапы.

Этап 1. *Определение цели и постановка задачи.* На первом этапе определяется основная цель предстоящей работы и перечень задач, которые необходимо решить для достижения поставленной цели. Здесь же формируются известные исходные данные, а также определяется формат и способ представления конечных данных.

Если работа укладывается в рамки технического проекта, то постановка задачи формулируется в специальном документе, который называется *техническим заданием*.

Правильная постановка задачи позволяет последовательно выполнить все последующие этапы. Напротив, допущенные ошибки могут привести к тяжелым последствиям с повторным выполнением всей работы.

Этап 2. *Моделирование.* Этап выполняется при условии, когда прямое решение задачи невозможно из-за отсутствия реального объекта или по причине экономической нецелесообразности известных подходов. В этом случае реальный объект заменяется формальным описанием или моделью, которая сохраняет лишь некоторые свойства реального объекта. Например, для сбора данных о состоянии здоровья населения целесообразно от изучения каждого человека перейти к обработке медицинских карт.

Этап завершается описанием выбранной модели и перечнем свойств модели, которые необходимо учитывать в процессе решения задачи.

Этап 3. *Построение алгоритма.* На этом этапе выстраивается логическая последовательность действий, направленных на решение конкретных задач. Последовательность действий выстраивается в форме алгоритма, который задает некоторый набор предписаний и правил, определяющих процесс преобразования исходных данных в результат решения задачи.

Сложные алгоритмы строят на основе модельных представлений, заложенных на предыдущем этапе.

Этап 4. *Разработка программы.* Полученный алгоритм используется для разработки программы, которая позволяет передать дальнейшее решение задачи от человека к машине. Запись программы производится на языке программирования. Этап завершается созданием текста программы, которая передается на этап трансляции.

Этап 5. *Трансляция программы.* Текст созданной программы подвергается автоматическому преобразованию использованных в программе переменных и операций в машинные команды. Перевод программы на машинный язык осуществляется с помощью транслятора, который привязан к конкретному языку программирования. При трансляции программы исходными данными является текст программы, а результатом – запись программы на машинном языке. Для каждого языка программирования существуют свои программы-трансляторы.

Этап 6. *Отладка программы.* Цель отладки заключается в выявлении и устранении ошибок, допущенных на предыдущих этапах. Суть отладки сводится к тестированию, которое включает стадии *альфа-тестирования* (с использованием тестов разработчика) и *бета-тестирования* (в реальных условиях эксплуатации). При возникновении сложностей отладка по длительности процесса может значительно превышать этап разработки программы.

Этап 7. *Эксплуатация программы.* Этап включает доработку программы по предложениям заказчика, установку программы на аппаратную базу заказчика, обучение пользователей программы, разработку эксплуатационной документации. Для отдельных программ, требующих постоянного контроля и обновления, этап эксплуатации по длительности приближается к продолжительности жизненного цикла созданной программы.

Моделирование как процесс упрощения задачи

Моделирование относится к одному из начальных этапов работы по созданию компьютерной программы.

Моделирование – это процесс построения модели в целях упрощения задачи и последующего изучения оригинала на основе построенной модели.

Модель – материальный или мысленно представляемый объект (процесс, явление), который в процессе исследования замещает оригинал и сохраняет при этом важные свойства оригинала. Классическим примером модели является географический глобус, который замещает оригинал, планету Земля, сохраняя ее форму.

Замена оригинала моделью может быть вызвана разными причинами, в частности:

- из-за отсутствия или недоступности оригинала;
- в связи с возможным разрушением оригинала в процессе исследования;
- по причине высокой стоимости процесса исследования.

Модель отличается от оригинала, поэтому содержит не все, а только существенные сведения об объекте. Эти сведения выбираются с учетом цели, для достижения которой модель создана. Модель всегда связана с определенным набором упрощений. Если модель при наличии всех упрощений дает удовлетворительные результаты, то говорят, что модель *адекватна* рассматриваемому объекту (процессу или явлению).

Для создания моделей используют следующие методы.

1. Предметное моделирование заключается в создании материальной модели, имеющей некоторые свойства оригинала (модели самолетов, кораблей, автомобилей).

2. Физическое моделирование выражается в экспериментальном исследовании модели, имеющей общие свойства с оригиналом (изучение молнии как электрического разряда между проводниками).

3. Аналоговое моделирование основано на аналогии (изоморфизме) различных явлений, для которых можно использовать одинаковое описание (форма самолета и летящей птицы).

4. Знаковое моделирование использует для построения модели в форме схемы, графика, чертежа или других знаковых систем (периодическая таблица Д. И. Менделеева).

5. Мысленное моделирование – разновидность знакового моделирования, при котором реальное построение заменяется мысленным представлением знаков или операций над ними (игра в шахматы вслепую).

Большое разнообразие моделей позволяет проводить их классификацию по разным признакам. В частности, в классификации по *форме представления* выделяют:

– материальные (предметные) – модели, которые воспроизводятся в форме реального объекта и используются в экспериментальных методиках познания окружающей среды; к ним относятся игрушки для детей, учебные пособия, экспериментальные исследовательские установки;

– информационные (теоретические) – модели, которые заменяют реальные данные об объекте формальным описанием, например компьютерной программой.

Информационная модель – совокупность формальных данных, которая характеризует существенные свойства объекта и характер его взаимосвязей с внешней средой (электронная карта местности характеризует только состояние земной поверхности).

Множество существующих информационных моделей может быть классифицировано по *видам* моделей:

– вербальные – модели, выраженные с помощью естественных языков в речевой или текстовой форме, например текст компьютерной программы;

– абстрактные, или мысленные, – модели, которые используют не существующие в жизни понятия, такие как абсолютно черное тело или идеальный газ;

– знаковые – модели, которые основаны на записи специальных символов, связанных с системой представления данных в изучаемой области знаний, например запись музыкального произведения с помощью нот на нотном стане;

– графические – модели, основанные на представлении объекта в виде схемы, графика, диаграммы, карты, чертежа, таблицы, диаграммы и т. п.

По способу *организации данных* выделяются информационные модели:

– табличные – модели, в которых данные оформляются в виде пересекающихся строк и столбцов (электронные таблицы);

– иерархические – модели с распределением объектов по разным уровням, сгруппированным в форме дерева (файловая структура);

– сетевые – модели, в которых объекты взаимодействуют между собой через множество связей (локальная компьютерная сеть); сетевые модели можно представлять в форме графа – математического объекта, состоящего из конечного множества вершин или узлов, связанных между собой ребрами или дугами.

Моделирование может выступать в качестве отдельного этапа работы в процессе создания компьютерной программы или служить средством для получения основного результата. Во втором случае в качестве основной цели выбирается разработка компьютерной модели и проведение на базе созданной модели исследовательской работы, которую невозможно провести с реальным объектом. Разработка и исследование такой модели выполняется на компьютере, а сама модель в этом случае называется *имитационной моделью*.

Имитационное моделирование – это процесс создания модели реальной системы и постановки компьютерного эксперимента на этой модели для изучения и прогнозирования ее поведения в целях улучшения характеристик рассматриваемой системы [13].

Имитационное моделирование применяется к исследованию сложных моделей, которые невозможно построить другими способами. При этом создание модели в виде компьютерной системы и проведение исследований на компьютере является обязательным условием.

Необходимость в проведении имитационного моделирования возникает в случаях, когда известные подходы не позволяют решить поставленную задачу:

– не существуют аналитические методы решения созданных математических моделей или эти методы связаны с высокой трудоемкостью;

– реальные условия, в которых находится объект исследования, трудно или невозможно обеспечить;

– нет физической возможности для отслеживания параметров процесса, например, в случае быстропротекающих или опасных процессов.

Имитационное моделирование целесообразно применять при решении численных задач в области математики и естественных наук или при решении задач организационного управления в разных сферах деятельности, например при проектировании производственных систем. Возможной областью применения имитационного моделирования является сфера образования, науки и профессиональной подготовки. Замена реального эксперимента на исследование модели зачастую дает более глубокие и порой неожиданные результаты.

Сфера применения имитационных моделей ограничена существующими недостатками:

- на решение задачи требуется много времени, необходимо также привлечение высококвалифицированных специалистов;
- полученные результаты выражаются только в численном виде;
- возникают трудности с получением исходных данных, необходимых для формирования модели.

Алгоритм и его основные свойства

Построение алгоритма относится к этапу работ, предшествующему разработке программы.

Понятие «алгоритм» связано, как правило, с решением сложных задач, требующих привлечения вычислительной техники. Вместе с тем это понятие можно использовать и при описании простых операций и решений. Примеры простых решений, в которых используются алгоритмические подходы: рецепты кулинарной книги, порядок автоматической стирки белья, кипячение воды в чайнике.

Простые подходы связаны с произвольным толкованием последовательности и содержания действий, в вычислительной технике последовательность действий тесно привязана к исполнителю алгоритма и перечню операций, которые может выполнять исполнитель.

Алгоритм – определенная последовательность действий, которая описывает порядок перехода объекта из начального состояния в конечный результат с использованием понятной исполнителю системы команд.

Из определения следует, что алгоритм можно представить как некоторую совокупность предписаний, которая выбирается из набора правил и инструкций конкретного исполнителя и позволяет решать задачу из некоторого класса однотипных задач.

Различают процессы *создания* и *реализации* алгоритмов.

Создание алгоритма – творческий процесс, выполняемый *специалистом* в области разработки алгоритмов.

Реализация – процесс выполнения предписанных команд *формальным исполнителем*, к которым в первую очередь относятся различные автоматические устройства, в том числе вычислительная техника.

Формальный исполнитель не вникает в смысл того, что он делает, но получает при этом необходимый результат. Строгое выполнение последовательности операций с отвлечением исполнителя от содержания поставленной задачи выражается в особенности, которая называется формальностью алгоритма. Наряду с формальностью алгоритм характеризуется и другими определенными свойствами.

1. Дискретность. Одно из свойств алгоритма, которое выражается в разбиении описываемого процесса на последовательность отдельных шагов или команд. Совокупность отдельных шагов образует дискретную структуру алгоритма.

2. Понятность. Для создания алгоритма могут быть использованы только те команды, которые исполнитель понимает и может выполнить. Другими словами, алгоритм должен состоять из команд, которые имеются в системе команд исполнителя.

3. Определенность, или детерминированность. При разработке алгоритма не могут быть использованы команды, смысл которых воспринимается исполнителем неоднозначно. Иначе говоря, алгоритм не должен оставлять места для произвола исполнителя.

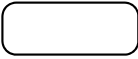
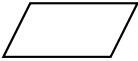
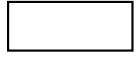
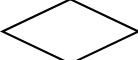
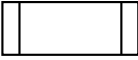
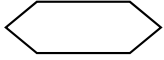
4. Результативность. Процесс, описываемый алгоритмом, должен прекратиться за конечное число шагов с получением определенного результата.

5. Массовость. Чаще всего алгоритмы обеспечивают решение не одной конкретной задачи, а некоторого класса задач данного типа. Это свойство позволяет выделять область применимости алгоритма.

Каждый алгоритм формируется из некоторого определенного набора типовых действий. Набор этих действий получается довольно наглядным в случае графического описания алгоритма, которое основано на представлении каждого действия в виде геометрической фигуры (блока). Графическое описание позволяет представлять алгоритм как систему связанных между собой функциональных *блоков*. Примеры графического изображения блоков, из которых складывается описание алгоритма, представлены в табл. 28.

Таблица 28

Графическая запись алгоритмических действий

Наименование действия	Обозначение	Функция
Пуск – останов		начало или конец действия
Ввод – вывод		ввод или вывод данных
Процесс (действие)		вычислительное действие
Условие (ветвление)		проверка условия и выбор направления
Предопределенный процесс		расчет по подпрограмме
Модификация		начало цикла
Документ		вывод на печать

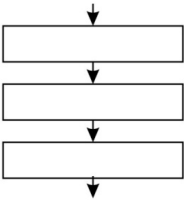
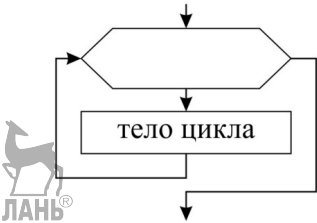
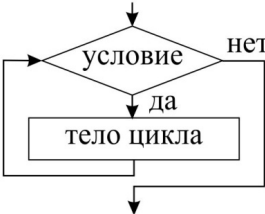
Графическое представление алгоритма называют *блок-схемой* алгоритма.

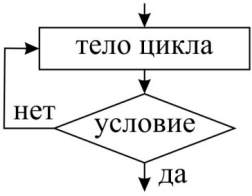
Блок-схема состоит из связанных между собой блоков, которые располагаются сверху вниз и соединяются между собой стрелками перехода. Взаимное расположение блоков показывает очередность выполнения процесса, а стрелки – направление вычислительного процесса. Внутри каждого блока вводится надпись, которая уточняет функцию блока или описывает математическое действие.

Алгоритмы в зависимости от цели, начальных условий, путей решения задачи, последовательности действий исполнителя подразделяются на разные типы. К основным наиболее часто используемым типам относятся *линейные* и *циклические* алгоритмы. Блок-схемы основных типов алгоритмов представлены в табл. 29. Кроме этого выделяют и другие типы, механические или жесткие алгоритмы, которые описывают работу механизмов, например двигателя внутреннего сгорания.

Таблица 29

Разновидности алгоритмов

Тип алгоритма	Блок-схема
<p><i>Линейные</i> алгоритмы состоят из набора команд, которые выполняют последовательно друг за другом</p>	
<p><i>Циклические</i> алгоритмы связаны с многократным заранее определенным количеством повторяемых действий и обновлением расчетных данных на каждом действии</p>	
<p><i>Циклические с предусловием</i> алгоритмы содержат предварительную проверку возможности выполнения цикла</p>	

Тип алгоритма	Блок-схема
<p><i>Циклические с постусловием</i> алгоритмы основаны на проверке результатов циклических вычислений и остановке цикла после выполнения постусловия</p>	

Приведенные в табл. 29 блок-схемы могут использоваться для построения различных алгоритмических конструкций. На рис. 25 в качестве *примера* показана схема алгоритма для расчета значения функции $f(n) = 3^n$ при $n = 5$.

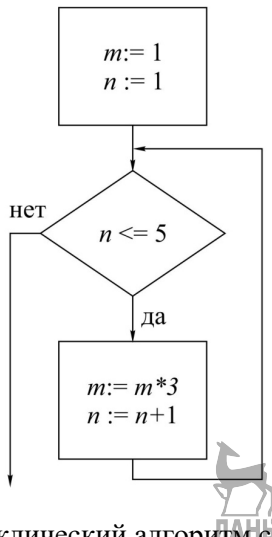


Рис. 25. Циклический алгоритм с предусловием

Наряду с графическим представлением существуют и другие формы записи алгоритмов. Для сравнения разных подходов рассмотрим пример составления алгоритма в разных формах записи.

Пример. Составим алгоритм начисления заработной платы, используя следующие правила:

- если стаж работы менее 1 года, то заработная плата равна А;
- при стаже от 1 до 3 лет – заработная плата возрастает в два раза;
- более 3 лет – каждый год идет повышение на величину В.

Введем обозначения: заработная плата – ZP , стаж работы – ST ; запишем математическую формулировку задачи:

$$ZP = \begin{cases} A & ST < 1 \\ 2 \cdot A & 1 \leq ST \leq 3 \\ 2 \cdot A + (ST - 3) \cdot B & ST > 3 \end{cases}$$

1. Словесно-формульное описание основано на представлении алгоритма в форме последовательности слов, предложений и формул с использованием естественного языка. Решение задачи с использованием этого описания складывается следующим образом.

- 1) Ввести ST, перейти к п. 2.
- 2) Если $ST < 1$, то $ZP = A$, перейти к п. 4, иначе перейти к п. 3.
- 3) Если $ST \leq 3$, то $ZP = 2 \cdot A$, иначе $ZP = 2 \cdot A + (ST - 3) \cdot B$, перейти к п. 4.
- 4) Вывести (на печать) значение ZP.
- 5) Вычисления прекратить.

Распространение словесно-формульного описания на практике ограничивается отсутствием строгой формализации, наличием многословности, а также неоднозначностью толкования отдельных действий алгоритма.

2. Графическое представление этого же алгоритма на основе структурной блок-схемы показано на рис. 26.

Графический подход отличается простотой и наглядностью, дает хорошие результаты при решении относительно небольших задач, но занимает значительно больше места. Объемная задача раздвигает границы блок-схемы до такой степени, что ее визуальное восприятие становится трудным или невозможным. По этой причине использование графического подхода при всех его преимуществах ограничено относительно небольшими программами.

3. Описание на основе псевдокодов выражается в полуформализованном представлении алгоритмов на условном алгоритмическом языке, который содержит элементы языка программирования, фразы естественного языка, общепринятые математические обозначения. Сочетание естественного языка с математической записью и общепринятыми служебными словами выводит псевдокод на промежуточное место между естественным и формальным языками. Запись алгоритма для начисления заработной платы на псевдокоде производится следующим образом:

```

алг Зарплата (цел ST, вещ ZP)
  арг ST, рез ZP
  нач
    если ST < 1    то ZP = A
  иначе
    если ST <= 3  то ZP = 2 * A
  иначе ZP = 2 * A + (ST - 3) * B
  все
все
кон.
```

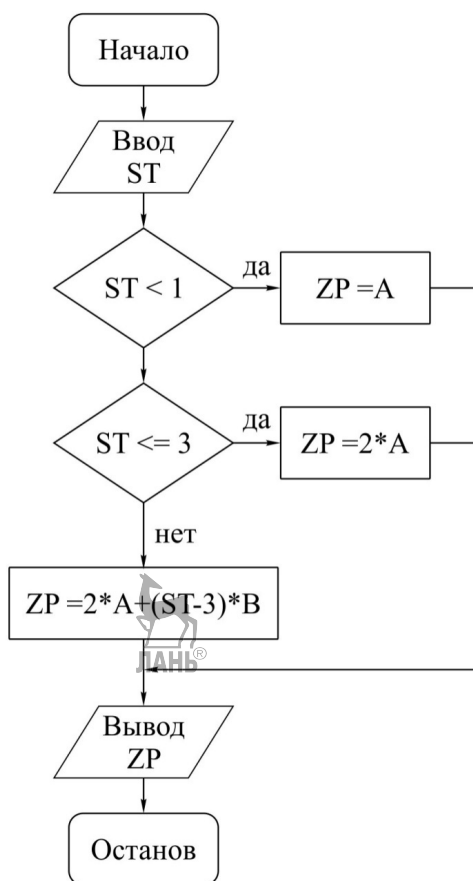


Рис. 26. Графическое представление алгоритма

Отсутствие строгих синтаксических правил и определенный произвол в записи команд не позволяют поставить псевдокод в один ряд с алгоритмическими языками программирования.

Представление алгоритма на языке программирования

Следующим этапом после создания алгоритма является написание текста программы. О программе, выполняющей действия, предписанные алгоритмом, говорят, что она реализует данный алгоритм на языке программирования.

Основная сложность создания программы заключается в том, что алгоритм формулируется на естественном человеческом языке, а программа, загружаемая в компьютер, представляет собой набор двоичных кодов – нулей и единиц. Понятно, что перевести текст напрямую с естественного языка в набор двоичных цифр довольно сложно. В связи с этим в данный процесс вводится промежу-

точный этап – разработка текста программы на специальном языке, который называется языком программирования.

Введение промежуточного этапа позволяет представить первую часть процесса создания программы как постепенный переход от естественного языка к языку более высокого уровня, чем машинный.

Уровень языка программирования характеризуется степенью близости языка программирования и машинного языка. За начало отсчета уровней принимается машинный язык, уровень которого равен нулю. Язык человека рассматривается как язык наивысшего уровня, алгоритмические языки относятся к языкам высокого уровня.

В основе алгоритмического языка программирования заложены три базовых составляющих:

алфавит – набор символов, используемый для написания программы;

синтаксис – правила соединения символов в конструкции языка;

семантика – принципы обозначения и толкования конструкций языка.

Язык программирования – искусственный язык, в котором символы алфавита соединяются в соответствии с синтаксическими и семантическими правилами.

Программирование на языках высокого уровня, очевидно, проще, чем на языках машинного уровня. Оно не требует глубоких знаний устройства процессора и поэтому доступно людям, которые не являются специалистами в аппаратной части. В отличие от языка человека, в котором допускается некоторый произвол в выборе и образовании слов и предложений, языкам программирования свойственна строгость синтаксических и семантических правил. Алфавит, синтаксис и семантика языков программирования определяются, исходя из стремления приблизить их к языку человека, но вместе с тем обеспечить возможность программного перевода алгоритма с языка программирования на машинный язык.

Для языков высокого уровня создание текста программы, который называется исходным модулем (рис. 27), выполняется вручную или с использованием автоматизированных средств, а перевод в машинные двоичные коды – в автоматическом режиме с помощью *транслятора* [14].

Трансляция (от *англ.* translation – перевод) – преобразование программы, представленной на исходном языке программирования, в программу на другом языке.



Рис. 27. Процесс создания программы в машинном коде

Для перевода исходного модуля в машинный код можно использовать два типа трансляторов:

- интерпретаторы – программы, которые построчно переводят и исполняют строки исходного кода (удобно для отладки и тестирования);

- компиляторы – программы, которые на первом этапе осуществляют перевод кода целиком с созданием объективного модуля – промежуточной программы, в которой отсутствуют реальные адреса данных и встроенные подпрограммы; на втором этапе к объективному модулю с помощью программы-сборщика подключаются необходимые адреса и подпрограммы.

Результатом трансляции является исполняемый модуль – программа, которая готова к загрузке, запуску и выполнению работы.

Для создания программы на выбранном языке используется так называемая интегрированная система программирования, которая включает связанную совокупность программных продуктов для разработки программного обеспечения. В состав интегрированной системы программирования входят следующие средства.

1. Текстовый редактор для написания программы. Текст программы состоит из последовательности символов или ключевых слов на английском языке. Готовая программа называется исходным текстом, или исходным модулем. Для написания программы можно использовать любой текстовый редактор, но лучше – специальный. Специальные текстовые редакторы созданы для всех популярных языков программирования, они позволяют проверять синтаксис языка и упрощать процесс разработки программы.

2. Программа-компилятор (или интерпретатор), которая переводит исходный текст в машинный код. В результате компилирования получается объективный модуль, который требует дальнейшей обработки. Для каждого модуля исходного текста создается отдельный объективный модуль.

3. Редактор связей (сборщик) обрабатывает все модули, созданные компилятором, связывает их между собой, при необходимости добавляет стандартные программы. В результате сборки на выходе появляется *исполняемый модуль*, который предназначен для работы на конкретном процессоре. Исполняемый модуль – это законченная программа, которая запускается на компьютере под контролем операционной системы. Файл исполняемого модуля имеет расширение .exe.

4. Библиотека стандартных программ – совокупность программ, разработанных на одном из языков программирования для решения типовых задач и расширения функциональных возможностей языка программирования.

5. Средства отладки программ – программные инструменты, предназначенные для наблюдения за пошаговым выполнением исследуемой программы с целью обнаружения и устранения ошибок.

Перечисленные средства, входящие в состав интегрированной системы программирования, образуют так называемую интегрированную среду разработки программ. В рамках этой среды используется весь комплекс имеющихся средств и обеспечиваются условия для создания новых программ в сжатые сроки

с высоким качеством конечного продукта. Интегрированная среда программирования поддерживает полный цикл проектирования сложных программных продуктов от начального исследования объекта до оформления технической документации на завершающем этапе.

Интегрированная среда программирования, усиленная технологиями автоматизированного проектирования программ, образует современное средство разработки – CASE-технологии. CASE-технология – это программный комплекс, который позволяет автоматизировать технологический процесс анализа, проектирования, разработки и сопровождения программных систем. Преимущества CASE-технологии выражаются в коллективной разработке программных систем путем подключения к локальной компьютерной сети и организации процесса экспорта и импорта программных фрагментов. Другим преимуществом новой технологии является возможность организационного управления всем проектом по созданию программных систем.

Разновидности языков программирования

Последовательное развитие технологий программирования привело к созданию несколько сотен языков программирования всевозможных принципов и структур. Одна из классификаций языков программирования, в которой отражается история развития и степень связи языка с вычислительной системой, представлена на рис. 28.

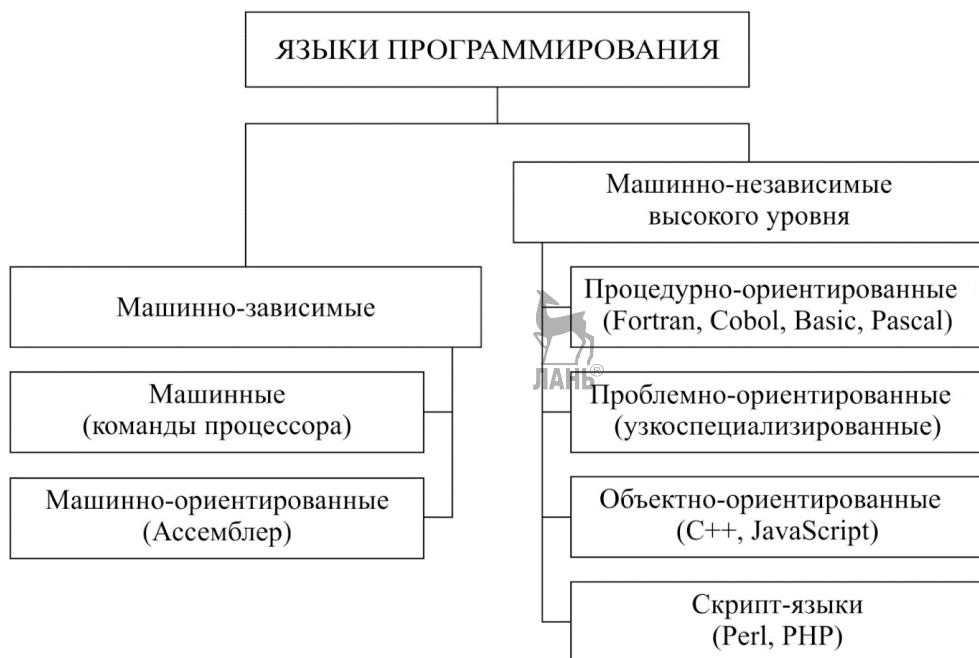


Рис. 28. Классификация языков программирования

Представленные в данной классификации языки высокого уровня делятся на группы, деление на группы и выбор языка внутри группы зависят от специфики и сложности решаемой задачи.

Группа процедурно-ориентированных языков относится к компилируемым языкам высокого уровня, основанным на принципе последовательных действий в процессе решения задачи. Основное преимущество языков этой группы выражается в простоте их освоения, поскольку они достаточно отдалены от машинного языка и близки по своей сути к естественному языку.

Fortran – процедурно-ориентированный язык, созданный в середине 1950-х гг. На этом языке впервые было показано, что можно создать язык высокого уровня, который поддается компилированию. Принципы, заложенные при создании этого языка, позднее были использованы в других языках. Язык широко используется для инженерных и научных расчетов и остается одним из самых востребованных.

Cobol – язык начала 1960-х гг. для решения задач в области финансов. Отличается многословностью и громоздкостью, предназначен для работы с большими объемами данных.

Basic – создан в 1960-х гг. в качестве учебного языка. Остается одним из самых простых языков и широко используется в образовательных целях.

Pascal – относится к процедурным языкам структурного типа, в котором одним оператором можно обозначать целую структуру действий. Разработан в 1970-х гг. как язык, предназначенный для обучения программированию, но получил широкое распространение не только в системе обучения. По причине простоты и доступности является одним из самых распространенных языков.

C (Си) – также относится к языкам структурного типа. Разработан в 1970-х гг. как язык программирования общего назначения, в том числе с целью замены Ассемблера. Широко используется для создания операционных систем, баз данных и других сложных программ.

Группа объектно-ориентированных языков программирования начала формироваться в 1980-х гг., когда объем работ по программированию уже приближался к пределу человеческих возможностей. В объектно-ориентированном программировании (ООП) использован новый подход, который позволил расширить возможности программирования в части производительности. К объектно-ориентированным относятся следующие языки программирования.

C++ – объектно-ориентированное расширение языка C, созданное в 1980-х гг. Использование этого языка повышает производительность программирования, но требует высокой профессиональной подготовки. Язык используется для разработки различных платформ и систем, в том числе для создания операционных систем, серверных систем, прикладных программ. Благодаря своим достоинствам язык C++ является лидером по использованию среди универсальных языков высокого уровня.

JavaScript – объектно-ориентированный язык, созданный в 1995 г. и предназначенный для описания последовательности операций. Один из основных языков, применяемых для создания web-сайтов.

C Sharp (C#) – объектно-ориентированный язык программирования, созданный в конце 1990-х гг. При создании языка использованы лучшие идеи известных языков программирования. Язык привязан к системе Windows, используется для разработки приложений операционной системы. К достоинствам языка C# относятся:

- сохранение основных принципов и идей, заложенных в популярных языках программирования C и C++, что облегчает переход программистов на новый язык;
- упрощение и повышение надежности основных операций, выполняемых на языке C++;
- наличие мощной библиотеки, которая поддерживает возможности построения приложений различного типа.

Группа скриптовых языков программирования предназначена для создания программ, описывающих последовательность операций или сценариев, распространяемых в компьютерных сетях. Все скриптовые языки являются интерпретируемыми, а сами программы распространяются в исходных текстах. К числу скриптовых относятся следующие языки.

HTML – первый язык для оформления web-документов. Содержит элементарные команды форматирования (разметки) и вставки объектов. Получил широкое использование в период становления Интернета, со временем получил значительное обновление и дополнение.

Perl – высокоуровневый интерпретируемый язык общего назначения с широкими возможностями для обработки больших текстовых файлов. Используется для решения широкого круга задач, в том числе в области web-разработок, сетевого программирования, графических интерфейсов.

PHP – сценарный язык общего назначения, имеет большой набор средств для разработки web-приложений, является лидером среди языков, которые используются для создания динамических сайтов.

Каждый язык программирования имеет свои особенности и специфические области применения. Универсальные языки программирования наиболее широко используются в разных областях деятельности, в частности:

- системное программирование – C++, Java;
- обработка данных – C++, Cobol, Java;
- научные исследования – C++, Fortran, Java;
- обработка сетевой информации – Perl, PHP, C++, Java.

Степень использования и актуальность языков программирования меняется с течением времени, что связано с рядом факторов:

- появление новых более удобных и простых технологий, использующих средства автоматизации;
- удобство разработки и тестирования программ;
- стоимость разработки программного обеспечения и др.

Прикладное программное обеспечение

Прикладные программные продукты предназначены для решения конкретных задач, решаемых пользователем, например задач по обработке числовых, текстовых или графических данных. Прикладные программы функционируют под управлением операционной системы и поэтому иногда называются приложениями операционной системы.

Прикладные программы разрабатываются специалистами как для широкого, так и для узкого круга пользователей. Небольшие прикладные программы создаются пользователями для собственных нужд. Тематика прикладных программ чаще всего тесно связана с профессиональной деятельностью пользователей, а общее число прикладных программ постоянно растет и постепенно приближается к числу профессий, которыми владеют пользователи компьютера.

История создания и развития прикладных программ началась с 70-х гг. XX в., когда были написаны первые программы для создания и обработки *текстовых файлов*. Спустя десятилетие появились программы для создания и редактирования *рисунков*, далее до конца XX в. были созданы приложения для упорядоченного хранения и обработки *больших объемов данных*, для записи и прослушивания звуковых данных, для разработки и воспроизведения *анимации, видео* и многое другое. Число областей применения прикладных программ с тех пор постоянно растет, в рамках каждого применения наблюдается выпуск все более совершенных версий программных продуктов.

С конца XX в. наиболее часто используемые программные продукты стали объединять в интегрированные пакеты. Программы, входящие в интегрированные пакеты, характеризуются общим интерфейсом и возможностью переноса данных из одной программы в другую через буфер обмена – промежуточное хранилище данных. К числу программ, которые включаются в число интегрированных пакетов, чаще всего относятся текстовые редакторы, электронные таблицы, системы для управления базами данных, программы для создания презентаций и др. Наибольшее распространение получили пакет Microsoft Office под операционной системой Windows и пакет OpenOffice под операционными системами Windows и Linux.

Другое направление развития прикладного программного обеспечения выражается в совместной разработке *операционных систем* и приложений, входящих в состав операционной системы. Приложения в составе операционной системы предназначены для выполнения работ, которые пользуются наибольшим спросом пользователей, например работ по поиску данных в сети Интернет. Для работы с сетевыми данными в состав операционной системы включается браузер – программа для поиска и просмотра *web-страниц*. В состав операционной системы могут включаться также текстовые или графические редакторы ограниченной функциональности, предназначенные для решения простых задач или для начального обучения пользователей.

Множество прикладных программ можно разделить на три категории и классифицировать по видам деятельности (рис. 29).



Рис. 29. Классификация прикладного программного обеспечения по видам деятельности

В представленной классификации отражен лишь принцип деления программ по категориям деятельности и показаны далеко не все прикладные продукты. В категории общего назначения представлены основные программы, которые находят самое широкое применение и установлены практически на каждом компьютере. В каждой из программ общего назначения предусмотрены широкие возможности для работы специалистов различных направлений. Остановимся на некоторых прикладных программах и подробнее рассмотрим их возможности для работы специалистов технического профиля.

Редакторы текстов

До появления компьютеров основным средством создания нетиражируемых текстовых документов являлась механическая печатная машинка. Технология механического печатания имела много недостатков:

- печатный текст практически не мог корректироваться, поскольку выводился непосредственно на бумажный носитель;
- текст не сохранялся в памяти, для создания ограниченного числа копий применялась печать через копировальную бумагу;
- печать выполнялась одним шрифтом, для изменения шрифта приходилось менять печатную машинку;
- процесс ввода и изменения текста отличался высокой трудоемкостью.

С появлением компьютеров для редактирования текстов были разработаны специальные программы, которые называли *текстовыми редакторами*.

Текстовые редакторы позволяли редактировать набранный текст в электронном виде и тем самым значительно упрощать работу с документами. Первоначально текстовые редакторы использовались для создания компьютерных программ, а впоследствии стали широко применяться при наборе различных текстов.

Вслед за простыми текстовыми редакторами появились программы, которые предназначались для создания и внешнего оформления (форматирования) текста. Более совершенные программы называли текстовыми процессорами, которые позволяли не только форматировать текст, но и вставлять нетекстовые объекты (таблицы, рисунки и пр.).

В России по мере развития компьютерной техники в 1980–1990-х гг. появились и были популярны различные текстовые редакторы:

- Лексикон – созданный в СССР в конце 1980-х гг. достаточно функциональный текстовый редактор полностью на русском языке;
- Фотон – текстовый редактор, созданный в 1989 г. в СССР на русском языке, получил в свое время широкое распространение;
- Слово и Дело – российский текстовый редактор, популярный в начале 1990-х и созданный для редактирования несложных текстов;
- Multi-Edit – текстовый редактор (США), распространенный в 1990-х по причине богатых возможностей по редактированию текстов, в среде DOS был так же популярен, как MS Word в среде Windows;
- ChiWriter – текстовый редактор (США), русская версия которого широко использовалась для создания научных текстов в 1990-х гг.;
- MS Word – текстовый процессор фирмы Microsoft с широкими функциональными возможностями, получивший широкое распространение в России и во всем мире с середины 1990-х.

Общее число используемых текстовых редакторов значительно превышает приведенный перечень и обеспечивает разнообразные функциональные возможности. Множество компьютерных программ для обработки текста принято классифицировать по группам. По типу обрабатываемого объекта текстовые редакто-

ры делятся на четыре группы: простые редакторы, редакторы форматированных текстов, научные редакторы, издательские системы.

1. Простые текстовые редакторы используются для написания простого текста (*plain text*) и относятся к самым используемым программам для создания заметок или небольших документов, небольших программ, web-страниц. Одним из известных в этой группе является редактор Блокнот в составе операционной системы Windows. Как правило, простые редакторы отличаются невысокой ценой, бесплатным распространением, ограниченным объемом, мультиязычным интерфейсом.

2. Редакторы форматированных текстов (текстовые процессоры) предназначены для работы со сложными документами, структура которых может содержать вложенные документы, страницы, абзацы и т. д. Документ может включать также таблицы, диаграммы, рисунки, фотографии, звук и видео. При этом таблицы, диаграммы и рисунки могут быть созданы как в самом редакторе, так и в других приложениях. Существует большое количество редакторов форматированных текстов различной сложности. Наиболее распространены в среде компьютерных пользователей редакторы Microsoft Word, OpenOffice Writer.

3. Редакторы научных текстов обеспечивают подготовку и редактирование текстов, содержащих большое количество математических формул, графиков, специальных символов и т. п. Распространенными и известными среди пользователей являются научные редакторы MatLab, TEX.

4. Издательские системы используются для подготовки больших и сложных документов, предназначенных для тиражирования, в том числе полиграфическими способами (книги, альбомы, журналы, газеты). К распространенным издательским системам относятся программы MS Publisher, Adobe PageMaker.

Создание и вставка объектов в текстовом редакторе

Среди перечисленных разновидностей редакторов по распространенности на первом месте находятся текстовые процессоры, которые в числе прочих применений используются для создания технических документов: отчетов, технических заданий, планов, инструкций и т. п. Текстовые процессоры позволяют создавать и вставлять в текст характерные для технического документа объекты, такие как таблицы, рисунки, диаграммы, формулы и др. В качестве примера рассмотрим, как решаются задачи по созданию нетекстовых объектов с помощью редактора MS Word.

Пример. Построим таблицу, содержащую данные о расстоянии между Солнцем и планетами Солнечной системы. По данным, представленным в таблице, построим гистограмму.

1. Для построения таблицы в текстовом редакторе MS Word используем инструмент *Вставка/Таблицы*, выберем вариант таблицы с десятью строками и двумя столбцами. Данные, необходимые для заполнения таблицы, найдем в сети Интернет.

2. Рядом с таблицей развернем рамку *Надпись* (*Вставка/Фигуры/Надпись*), в которой будем строить гистограмму.

3. После команды *Вставка/Диаграмма/Гистограмма* откроется окно электронной таблицы Excel для ввода данных, необходимых при построении гистограммы. В окне Excel заполним два столбца: первый для названия планет, второй для численных данных о расстоянии. После занесения данных в текстовом редакторе рядом с нашей таблицей появится гистограмма.

4. Останется ввести название гистограммы и обозначение оси (*Макет/Название диаграммы* и *Название осей*). В результате получится изображение, представленное на рис. 30.

Название планеты	Расстояние до Солнца (а.е.)
Меркурий	0,39
Венера	0,72
Земля	1
Марс	1,52
Юпитер	5,2
Сатурн	9,5
Уран	19,2
Нептун	30
Плутон	39



Рис. 30. Гистограмма, построенная по табличным данным

Другой пример связан с вставкой математической формулы, которые часто используются при создании технических документов и научных текстов.

Пример. Введем по центру страницы формулу и ее текущий номер N .

$$f(x) = n \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} \cdot e^{-\frac{mx^2}{2kT}}.$$

1. Для ввода формулы создаем таблицу 2*1. В 1-м столбце отступ слева 1,5 см, выравнивание по центру, интервал до и после абзаца 6 пт. Ширина 2-го столбца 1,5 см, выравнивание по правому краю. В 1-й столбец вводим формулу. Открываем *Конструктор* формул (*Вставка/Формула/Вставить новую формулу*) и собираем структуру формулы. Во 2-й столбец вводим номер формулы *N*.

$f(x) = n \left(\frac{\square}{\square} \right)^{\square} \cdot \square$	(N)
---	-----

2. Заполняем структуру символическими данными с клавиатуры, символы π и «» вставляем из набора символов в *Конструкторе*. Границы таблицы делаем невидимыми. Получаем результат:

$$f(x) = n \left(\frac{m}{2\pi kT} \right)^{\frac{3}{2}} \cdot e^{-\frac{mx^2}{2kT}}. \quad (N)$$

Ряд технических документов связан с созданием и вставкой небольших графических объектов. В качестве примера рассмотрим решение следующей задачи.

Пример. Создадим изображение кристаллической решетки оксида меди (рис. 31) с перечнем его физических и химических свойств.

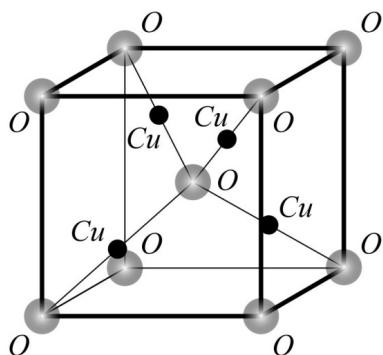


Рис. 31. Кубическая кристаллическая решетка

ОКСИД МЕДИ

Получение:

- $2\text{CuO} + \text{O}_2 \rightarrow 2\text{CuO}$
- $\text{Cu}(\text{OH})_2 \rightarrow \text{CuO} + \text{H}_2\text{O}$

Физические свойства:

$$\rho = 6,48 \text{ г/см}^3$$

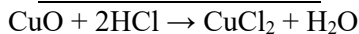
$$t_{\text{пл}} = 1335^\circ\text{C}$$

Растворимость – 0,15 мг/л H_2O

От темно-коричневого до черного цвета

Полупроводник

Химические свойства:



1. Используя элементарные фигуры и линии (*Вставка/Фигуры*), строим каркас кристалла. Видимые и невидимые линии в кубе должны отличаться по толщине.

2. Круглую заготовку атома кислорода (*Вставка/Фигуры/Овал*) окрашиваем градиентной заливкой (*Формат/Заливка фигуры/Градиент/Темные варианты/Из центра*).

3. Копируем заготовку и размещаем атомы кислорода в вершинах кубической структуры и в ее центре.

4. Отрезками тонких линий соединяем центр куба с его четырьмя вершинами. В середине каждого отрезка помещаем изображение атома меди.

5. Создаем заготовку для надписей. Вводим название химического элемента атома в рамку *Надпись (Вставка/Фигуры/Надпись)*, отказываемся от *Контура* и *Заливки* рамки (*Формат/ Контур фигуры/Нет контура* и *Заливка фигуры/Нет заливки*). Копируем рамку и расставляем надписи вблизи каждого атома.

6. Обводим прямоугольной рамкой и выделяем все созданные графические элементы (*Главная/Выделить/Выбор объектов*) и группируем их в единый объект (*Формат/Группировать*).

7. Смещаем созданное изображение в левую часть листа, рядом с ним разворачиваем рамку *Надпись* и вводим в эту рамку перечень физических и химических свойств.

Электронные таблицы

Электронными таблицами, или табличными процессорами, называются компьютерные программы, предназначенные для хранения и обработки численных данных, представленных в табличном виде.

Электронные таблицы являются универсальными средствами для автоматизации расчетов, они позволяют существенно уменьшить затраты времени на вычислительную работу. Электронные таблицы можно назвать также программными средствами, предназначенными для управления численными массивами данных.

До появления компьютеров таблицы составлялись вручную или печатались на механической машинке. Математические расчеты производились отдельно, результаты расчетов вставлялись в таблицу в готовом виде. Все ручные операции по расчету и выводу данных отличались высокой трудоемкостью.

В 1981 г. фирмой VisiCorporation была выпущена первая электронная таблица с названием VisiCalc. С этого момента началось развитие электронных таблиц как самостоятельного вида программного обеспечения.

Опыт эксплуатации специализированной программы, выполняющей операции с табличными данными, оказался востребованным. В 1980-х гг. электронные таблицы уже разрабатывали многие фирмы, а уже в 1990-х гг. программный продукт был значительно усовершенствован целым рядом дополнительных функций. С тех пор электронные таблицы заняли одно из первых мест в деловом программном обеспечении, они входят в интегрированные пакеты программ в качестве обязательного приложения.

Среди электронных таблиц наибольшее распространение получил табличный процессор Excel, который разработан фирмой Microsoft и входит в состав программного пакета MS Office, а также электронная таблица Calc, входящая в состав программного пакета OpenOffice. Позже наряду с электронными таблицами универсального назначения стали появляться специализированные программы, предназначенные для выполнения расчетов в рамках отдельного вида деятельности, например бухгалтерского учета.

Области применения электронных таблиц чаще всего связаны с финансовыми и техническими расчетами:

- бухгалтерский учет – подготовка отчетных финансовых документов на предприятиях на основе специализированной электронной таблицы;
- банковский учет – взаиморасчеты с клиентами и подведение итогов деятельности;
- проектно-сметные работы – формирование сметных ведомостей, расчет конструкций в строительстве;
- инженерно-технические расчеты – выполнение технических расчетов в процессе разработки и производства изделий;
- научные исследования – статистическая обработка результатов измерений;
- обработка больших массивов данных в метеорологии, экологии, экономике и др.

Использование электронных таблиц зачастую позволяет решать расчетные задачи без разработки алгоритма и отладки программы, что значительно упрощает процесс решения. Вместо создания программы в таблицу вписываются исходные данные и математические формулы, которые обеспечивают быстрое получение результата. Если при решении задачи формула используется многократно, то электронная таблица позволяет скопировать формулу в другие ячейки и получить автоматический пересчет результатов с новыми данными. Возможность копирования формулы является принципиальным отличием электронных таблиц.

Разнообразие функций и способов программной обработки в электронных таблицах обеспечивают широкий спектр операций с данными, в том числе:

- расчет и сравнительный анализ численных данных;
- использование нескольких категорий функций;
- создание файла из тематически связанных таблиц;
- создание математических моделей и расчет их эффективности;
- статистическую обработку массивов данных;
- построение диаграмм различного вида;
- определение вида математической зависимости и прогнозирование ожидаемых результатов;
- создание и обработку баз данных;
- выполнение операций с логическими функциями;
- вывод результатов на печать и др.

Перечисленные операции свидетельствуют о широких функциональных возможностях электронных таблиц в части решения математических и логических задач. Более детально возможности электронной таблицы можно изучить на примере *Excel* – программы-приложения ОС Windows.

Excel – одна из самых популярных электронных таблиц. Файлы, созданные в программе *Excel*, имеют расширение *.xls* и могут копироваться через буфер обмена в рамках программного пакета MS Office.

Каждый документ табличного процессора Excel называют *Книгой*, которая состоит из *рабочих листов*. На каждом рабочем листе расположена *электронная таблица*.

Электронная таблица состоит из *столбцов* и *строк*. Столбцы помечаются латинскими буквами от А до Z и комбинациями букв AA, AB и т. д. Строки пронумерованы целыми числами, общего количества строк достаточно для создания больших объемов данных. Элемент, находящийся на пересечении столбца и строки, называется *ячейкой*.

Ячейка, с которой выполняется работа, называется *активной*. Активная ячейка выделяется жирной прямоугольной рамкой. В правом нижнем углу рамки расположена крупная точка квадратной формы, которая называется маркером заполнения. Протягивание маркера заполнения с помощью мыши приводит к копированию данных активной ячейки в другие рядом расположенные ячейки.

Каждая ячейка имеет следующие характеристики.

1. Адрес – состоит из названия столбца и номера строки, например С3. При перемещении или сдвиге ячейки с помощью маркера заполнения Excel автоматически меняет номера столбцов и строк в зависимости от количества столбцов или строк, на которые произошел сдвиг. Адрес ячейки, который изменяется при сдвиге, называют *относительным*.

Если после перемещения формулы необходимо оставить в ней прежние адреса ячеек, то относительные адреса ячеек в формуле меняют на *абсолютные*. Для замены или фиксирования адреса используют символ \$, который размещают перед номером столбца и строки. Так, относительный адрес С3 после фиксирования приобретает вид \$С\$3. Для быстрого перехода к фиксированному адресу используют клавишу F4. При перемещении по таблице фиксированные адреса в формулах *не меняются*.

2. Содержимое – вносится в ячейку в следующих видах:

– число – целое, дробное или дробное с плавающей запятой; при внесении больших или очень малых чисел форма записи числа может измениться, например число $1,2 \cdot 10^{10}$ Excel запишет в виде 1,2E+10;

– текст – вносится и может изменять внешний вид путем использования элементов форматирования, которые аналогичны элементам текстового редактора;

– формула – запись формулы начинается со знака «=»; вводить формулу можно с помощью кнопки *Вставить функцию* или путем написания названия функции в *Строке формул*; в первом случае следует заполнять диалоговое окно для каждой функции, а во втором – написать первые символы и далее использовать подсказку, которую выводит программа.

3. Изображение – разное представление одного и того же числа в зависимости от выбранного формата, например формата даты, процента, или денежный. Формат числа связан с единицей измерения численной величины, которую добавляет пользователь.

4. Формат ячейки – включает параметры внешнего оформления, в частности, формат чисел, шрифт, цвет символов, вид рамки, цвет фона и др.

5. Имя – присваивается ячейке и используется для ссылок в формулах.

6. Примечание – сопроводительный текст, расположенный в рамке рядом с ячейкой, отмеченной красным треугольником в правом верхнем углу ячейки.

Прямоугольная двумерная область таблицы, состоящая из нескольких ячеек, называется *блоком*, или *диапазоном*, *ячеек*. Блок задается адресами верхней левой и правой нижней ячеек блока, перечисленными через двоеточие. Блок B2:D5, например, содержит 12 ячеек. Одинаковый блок ячеек, выбранный на разных рабочих листах, называется *трехмерным блоком* или *диапазоном*. Трехмерные блоки используются для одновременного открытия нескольких листов и внесения изменений на всех открытых листах одновременно.

Табличный процессор Excel содержит более 400 встроенных функций, которые разбиты на 11 категорий. Ввод функций осуществляется по определенным правилам, при вводе функции с нарушением принятых правил возникают ошибки. Типичные ошибки пользователя программа регистрирует и выдает сообщение о виде ошибки. Сведения об ошибке выводятся в форме короткого слова.

#ИМЯ? – допущена ошибка в отображении имени функции или имени аргумента, например вместо функции $=\sin(A1)$ написано $=\text{cin}(A1)$.

#ЧИСЛО! – отображается в том случае, если в качестве числового аргумента используется неприемлемый аргумент. Например, расчет $=\text{КОРЕНЬ}(A1)$ при $A1 = -2$ невозможен.

#ЗНАЧ! – отображается тогда, когда используется недопустимый тип значений аргумента, например формула $=2*A1$ при $A1=x$.

#ДЕЛ/0! – деление на 0 невозможно, например $=2/A1$ при $A1=0$ или $A1$ – пустая ячейка.

#ССЫЛКА! – ссылка на ячейку недействительна, например, когда используемая в вычислениях ячейка была удалена.

Обработка данных в электронных таблицах

Рассмотрим особенности обработки данных в электронной таблице Excel с использованием функций различных категорий.

Математические функции

Название функции представлено в виде сокращенного набора символов, после которых в скобках следует перечень аргументов. В качестве примера рассмотрим некоторые функции из категории математических функций.

СЛЧИС() – генерирует случайное 17-разрядное число в интервале от 0 до 1. Клавиша F9 – генерация нового случайного числа. Копированием через маркер заполнения можно получить множество случайных чисел. СЛЧИС() относится к категории функций, не имеющих аргумента. В соответствии с принятым форматом при записи такой функции сохраняются скобки, не содержащие аргумента.

СЛУЧМЕЖДУ (нижняя граница; верхняя граница) – генерирует случайное число между двумя заданными числами – верхней и нижней границей. Клавиша F9 – генерация нового случайного числа в заданном интервале. Копированием

через маркер заполнения можно получить множество случайных чисел в этом же интервале.

ОКРУГЛ (число; число _ разрядов) – округляет выбранное число до заданного количества разрядов после запятой.

ОТБР (число; число _ разрядов) – оставляет заданное число разрядов после запятой.

Разницу между функциями **ОКРУГЛ** и **ОТБР** можно показать на округлении числа π , которое в Excel определяется как функция ПИ(). Значение π является константой ($\pi = 3,141592\dots$).

ОКРУГЛ (ПИ(),4) = 3,1416

ОТБР (ПИ(),4) = 3,1415

Логические функции

В алгебре логики логические функции принято обозначать словами естественного языка или специальными символами. В электронных таблицах принятые обозначения сохраняются и дополняются возможностью проведения логических расчетов. К категории логических относятся функции, представленные в табл. 30.

Таблица 30

Логические функции в программе Excel

Наименование	Определение
И	принимает значение ИСТИНА, если все аргументы истинны
ИЛИ	принимает значение ИСТИНА, если хотя бы один аргумент равен истине, и значение ЛОЖЬ, если все аргументы ложны
НЕ	производит замену значения ИСТИНА на ЛОЖЬ и значения ЛОЖЬ на ИСТИНА
ЕСЛИ	выполняет логическую проверку и выбирает одно из двух возможных действий

Каждая из логических функций имеет *аргументы*, которые отражаются в скобках, следующих за названием. Например, функция И с аргументами записывается в виде И (логическое значение 1; логическое значение 2; ...). При расчетах вместо логических значений вставляются *адреса конкретных ячеек*, в которых находятся эти значения. Расчетное значение функции И можно записать так: =И(A1;B4). Используя расчетные значения логических функций, можно составить таблицу истинности для основных логических операций (табл. 31).

Таблица 31

Значения логических функций аргументов X и Y

X	Y	X И Y	X ИЛИ Y	НЕ X И Y	НЕ X ИЛИ Y
1	1	ИСТИНА	ИСТИНА	ЛОЖЬ	ЛОЖЬ
1	0	ЛОЖЬ	ИСТИНА	ИСТИНА	ЛОЖЬ
0	1	ЛОЖЬ	ИСТИНА	ИСТИНА	ЛОЖЬ
0	0	ЛОЖЬ	ЛОЖЬ	ИСТИНА	ИСТИНА

Логическую функцию ЕСЛИ можно использовать для отбора чисел, значения которых меньше некоторого порогового. В табл. 32 в качестве примера показано, каким образом можно отобрать числа, значения которых меньше 40.

Таблица 32

Выделение чисел с превышением порогового значения

	A	B
1	55	=ЕСЛИ(A1>40; «Превышение порога»;A1)
2	38	=ЕСЛИ(A2>40; «Превышение порога»;A2)

После выполнения расчета в ячейке B1 будет записано «Превышение порога», а в ячейке B2 – число 38.

Функции обработки текста

При обработке текстовых данных можно использовать оператор & (*ampersand*), который аналогичен союзу И. Этот оператор может использоваться для объединения последовательности текстовых символов. В табл. 33 показано, каким образом можно разрозненное по ячейкам полное имя представить в одной ячейке.

Таблица 33

Объединение текстовых символов в одной ячейке

	A	B	C	D
1	Ф	И	О	=A1&B1&C1

СЦЕПИТЬ(текст 1;текст 2;...) – функция аналогичного действия, объединяет до 255 текстовых строк в одну строку в одной ячейке.

Обратная операция – разделение текста из одной ячейки по разным ячейкам выполняется с помощью *Мастера текстов*. В диалоговом окне Мастера текстов программа предложит вариант своего деления фразы, который может быть изменен пользователем в части формата или расстановки делителей. Мастер представит также предварительный просмотр выбранных данных. Фразу, помещенную в ячейку А1 (табл. 34), с помощью команды *Данные/Текст по столбцам/Мастер текстов(разбор)* можно разделить на три ячейки

Таблица 34

Деление текста по ячейкам

	А	В	С	Д
1	казнить нельзя помиловать	казнить	нельзя	помиловать

С помощью функций обработки текста можно также определить день недели для любой даты календаря.

ТЕКСТ(значение из ячейки; формат «ДДДД») – функция, которая преобразует дату, указанную в ячейке, в соответствующий день недели. Например, если в ячейку А1 занести дату рождения (в формате Дата), а в ячейку В1 функцию ТЕКСТ с аргументами, показанными в табл. 35, то результат вычисления будет представлен в виде соответствующего дня недели.

Таблица 35

Определение дня недели для конкретной даты

	А	В
1	00.00.00	=ТЕКСТ(А1; «ДДДД»)

Функции категории Дата и время

В категории Дата и время содержится примерно два десятка функций.

СЕГОДНЯ() – функция без аргумента, которая выдает текущую дату (на день ввода). Используя эту функцию можно произвести расчет количества прожитых дней (табл. 36), если в ячейку В1 занести дату рождения (в формате Дата), в ячейку В2 – функцию СЕГОДНЯ(), а в ячейку В3 – формулу для расчета разности (в общем формате).

Таблица 36

Промежуток времени между датами

	А	В
1	Дата рождения	00.00.00
2	Сегодня	=СЕГОДНЯ()
3	Прожито дней	=В2-В1

Функции этой категории позволяют также рассчитать промежуток времени между событиями с точностью до секунды. Пример расчета показан в табл. 37.

Таблица 37

Определение продолжительности события

	А	В
1	1 Дата, время	30.03.18 6:00:00
2	2 Дата, время	01.04.18 12:00:00
3	Промежуток времени	=B2-B1 (формат Время)

Средства анализа данных



В составе приложения Excel имеются специальные средства для анализа данных. К таким средствам относятся инструменты: *подбор параметра* и *поиск решения*.

Подбор параметра

Задачи по отысканию параметра, при котором некоторая функция $y=f(x)$ принимает определенное значение, возникают в математике, технике, экономике, бухгалтерии, банковском деле и многих других областях. В Excel решение подобных задач реализовано как поиск параметра или аргумента, удовлетворяющего конкретному значению функции. Если известно, какой результат необходимо получить в расчете по известной формуле $y=f(x)$, но не известно значение аргумента x , при котором достигается этот результат, то используется средство *Подбор параметра*. Excel ищет решение *методом последовательных приближений* (итераций).

Чтобы воспользоваться средством Подбор параметра, необходимо выполнить следующие операции.

1. Заполнить ячейки формулами и данными, которые будут использоваться для расчета. Выделить и оставить пустой ячейку, предназначенную для параметра (аргумента).

2. Установить курсор в ячейке с формулой.

3. Выбрать команду *Данные/Анализ «что-если»/Подбор параметра* и заполнить поля диалогового окна *Подбор параметра*:

– в поле *Установить в ячейке* должна находиться ссылка на выделенную ячейку;

– в поле *Значение* нужно ввести требуемое числовое значение функции;

– в поле *Изменяя значение ячейки* необходимо ввести ссылку на ячейку, зарезервированную для аргумента.

4. После нажатия кнопки *ОК* программа автоматически найдет решение.

Пример. С помощью подбора параметра найдем корень уравнения:

$$(2x + 10)/2 + (4x + 20) = 60.$$

Первоначально занесем расчетную формулу в электронную таблицу. При этом будем считать, что неизвестный параметр x находится в ячейке A1 (табл. 38).

Таблица 38

Ввод расчетной формулы

	A	B
1		$= (2 * A1 + 10) / 2 + (4 * A1 + 20)$

Выберем команду *Данные/Анализ «что-если»/Подбор параметра*. В окно *Подбор параметра* введем данные (табл. 39).

Таблица 39

Окно инструмента Подбор параметра

Установить в ячейке	B2
Значение	60
Изменяя значение ячейки	A1

После нажатия кнопки ОК программа выведет в ячейке A1 значение 7. Это означает, что корень уравнения $x = 7$.

Пример. Составим штатное расписание и определим размер месячной заработной платы работников станции технического обслуживания, если фонд месячной заработной платы равен 800 тыс. руб., при этом для расчетов заработной платы используем следующие соотношения:

- слесарь получает в 1,6 раза больше рабочего;
- станочник – в 1,3 раза больше слесаря;
- мастер – на 1000 руб. больше, чем станочник;
- менеджер – в 1,5 раза больше слесаря;
- зав. складом – на 2500 руб. меньше мастера;
- главный бухгалтер – в 1,5 раз больше менеджера;
- директор – в 1,4 раза больше главного бухгалтера.

Используем приведенные соотношения и заполняем таблицу исходными формулами, ячейку B2 оставляем под параметр, который нужно определить (табл. 40).

Ввод расчетных формул

	А	В	С	Д
1	Должность	Зароботная плата	Кол-во персонала	Сумма выплат
2	рабочий		2	=B2*C2
3	слесарь	=B2*1,6	8	=B3*C3
4	станочник	=B3*1,3	3	=B4*C4
5	мастер	=B4+1000	3	=B5*C5
6	менеджер	=B3*1,5	4	=B6*C6
7	зав. складом	=B5-2500	1	=B7*C7
8	главный бухгалтер	=B6*1,5	1	=B8*C8
9	директор	=B8*1,4	1	=B9*C9
10		Всего:		=СУММ(D2:D9)

В окно *Подбор параметра* вводим данные (табл. 41).

Таблица 41

Окно инструмента Подбор параметра

Установить в ячейке	D10
Значение	800000
Изменяя значение ячейки	B2

После нажатия клавиши ОК программа заменит все формулы в табл. 40 численными данными, в том числе данными о заработной плате всех работников в текущем месяце.

Следует отметить, что совокупность использованных расчетных формул образует небольшую программу. Составленная таким образом программа может использоваться для ежемесячной корректировки заработной платы в зависимости от полученного дохода.

Подбор параметра используется в случае, когда необходимо отыскать единственный возможный аргумент функции. При увеличении числа параметров следует пользоваться инструментом *Поиск решения*.

Поиск решения

В экономике и технике широкое применение находят оптимизационные модели. Эти модели используются для решения задач подбора сбалансированного

рациона питания, оптимизации ассортимента продукции, транспортной логистики и пр.

Надстройка программы *Поиск решения* предназначена для быстрого поиска оптимальных решений с использованием сложных моделей. Поиск решения позволяет использовать одновременно большое количество изменяемых ячеек (до 200) и задавать ограничения для изменяемых ячеек.

Общие свойства, которые характерны для задач, решаемых с помощью надстройки Поиск решения.

1. Существует единственная целевая ячейка, значение в которой должно быть сделано максимальным, минимальным или же равным какой-то конкретной величине.

2. Значение в целевой ячейке связано формулами с рядом изменяемых ячеек. Поиск решения заключается в том, чтобы подобрать такие значения переменных в изменяемых ячейках, которые обеспечили бы оптимальное значение для целевой ячейки.

3. Дополнительно может быть задано некоторое количество ограничений – условий или соотношений, которым должны удовлетворять некоторые из изменяемых ячеек.

Кнопка надстройки *Поиск решения* расположена в разделе меню *Данные*. Если кнопка отсутствует, то для её установки следует выполнить последовательность действий:

Кнопка Office/Параметры Excel/Надстройки/Поиск решения/Перейти/Поиск решения/ОК.

В качестве примера рассмотрим решение задачи, в которой необходимо отыскать три неизвестных параметра.

Пример. Решим известную старинную задачу из арифметики Л. Ф. Магницкого.

Крестьянин на базаре за 100 руб. купил 100 голов скота. Бык стоит 10 руб., корова 5 руб., теленок 50 коп. Сколько быков, коров и телят купил крестьянин?

Воспользуемся инструментом Поиск решения. Заносим исходные данные и расчетные формулы, которые потребуются для решения задачи, в таблицу (табл. 42).

Таблица 42


Ввод исходных данных и расчетных формул

	A	B	C	D
1		Кол-во голов	Цена, руб.	Сумма, руб.
2	Бык		10	=B2*C2
3	Корова		5	=B3*C3
4	Теленок		0,5	=B4*C4
5	Всего	=СУММ(B2:B4)		=СУММ(D2:D4)

Ячейки В2, В3, В4 отведены под определяемые параметры или изменяемые ячейки – количество голов скота, все остальные ячейки таблицы заполнены данными и формулами. Далее открываем окно Поиск решения, вводим имена изменяемых ячеек и ограничения, заложенные в условиях (табл. 43).

Таблица 43

Окно надстройки Поиск решения

Установить в целевую ячейку D5 значение	100
Изменяя ячейки	В2:В4
Ограничения	 В5=100 В2>=1 В3>=1 В4>=1 В2= целое В3= целое В4= целое

После нажатия клавиши *Выполнить* программа выполнит расчеты и найдет одновременно три параметра: В2 = 1, В3 = 9 и В4 = 90.

Базы данных и системы управления базами данных

С появлением компьютера на рабочем столе каждый специалист в той или иной мере стал заниматься сбором, накоплением и обработкой систематизированных цифровых данных, которые со временем выделились в отдельный объект исследования и получили название «База данных». Первое оборудование и программное обеспечение для обработки структурированных цифровых записей появилось в 50-х гг. прошлого века. Термин «база данных» стал общепринятым в 1960-х, а в 1970-х гг. была сформирована модель представления данных, основанная на математических и логических подходах.

Для первых баз данных был создан структурированный язык запросов (*SQL*), который позволял манипулировать в базе отдельными записями и группами записей. Позднее для управления большими базами данных были созданы системы управления базами данных – СУБД. Все эти системы поддерживают язык запросов SQL наравне со своим уникальным языком.

С начала 1980-х базы данных стали широко использоваться для удовлетворения потребностей многих предприятий и организаций. Первоначальное представление об упорядоченном наборе записей сменилось на более точное понятие, которое продолжает дополняться и корректироваться. В зависимости от источника данных это понятие выражается в разных формулировках. Общеизвестная формулировка баз данных (БД) пока отсутствует, для раскрытия понятия удобно пользоваться определениями из разных источников.

1. Базой данных является совокупность самостоятельных материалов (статей, расчетов, нормативных актов и подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью ЭВМ [15].

2. База данных – это организованная структура, предназначенная для хранения информации [1]. Это определение означает, что в современных базах может храниться информация, хотя их по-прежнему называют базами данных.

На основе известных формулировок можно дать следующее короткое определение.

База данных – совокупность структурированных логически связанных материалов, которые собраны в одном или нескольких файлах.

Для уточнения понятия БД используют также *отличительные признаки*, которые выражаются в следующем.

1. БД хранятся в *компьютерах* и обрабатываются с помощью компьютерных программ. Этот признак исключает из числа баз данных некомпьютерные хранилища, например бумажные архивы или библиотеки.

2. Данные в базе *логически структурированы* или *систематизированы* для удобства пользования, выполнения операций поиска, дополнения, выборки данных и др.

3. В БД *встраивается логическая структура* данных в соответствии с моделью, которая закладывается при разработке базы.

Наряду с определениями и отличительными признаками существует общепринятая практика, в соответствии с которой не называют базами данных файловые архивы, интернет-порталы, электронные таблицы, хотя они обладают некоторыми признаками БД.

Определения и отличительные особенности БД свидетельствуют о том, что они созданы для хранения огромных массивов данных и автоматической обработки этих массивов с помощью вычислительной техники. Для надежного хранения информации, которая может иметь большую ценность, в базе данных в отличие от других программ используют *двойной подход* к изменению данных.

1. Операции по созданию или изменению структуры базы данных производят *только с копией* и никогда не производят с базой данных, находящейся в эксплуатации.

2. Все изменения с данными сохраняются *немедленно и автоматически*. Базу данных нельзя закрыть без сохранения, так как все сохранения бывают завершены к моменту закрытия.

Классификация и структура баз данных

Множество существующих баз данных поддается различным видам классификаций [16]. Одной из распространенных является классификация *по модели данных*, в которой выделяются следующие категории.

1. Иерархические базы данных могут быть представлены как *дерево*, состоящее из объектов различных уровней. Примером иерархической структуры является файловая структура.

2. Сетевые базы данных состоят из *элементов (узлов)*, каждый из которых находится на определенном уровне и может быть связан любым другим элементом.

3. Реляционные базы данных формируются на основе таблиц, *связанных* между собой определенными отношениями.

4. В классификации по содержанию название категории определяется содержанием БД: исторические, экологические, экономические и т. п.

В классификации по степени распределенности отмечают место расположения и выделяют *централизованные* (сосредоточенные) и *распределенные* базы данных. Распределенные базы данных, как правило, обрабатываются разными удаленными друг от друга пользователями или организациями.

Относительно недавно распределенные базы данных получили мощный импульс в своем развитии. Этот импульс связан с возникновением технологии *блокчейн* (англ. blockchain – цепочка блоков). Суть технологии или платформы блокчейн выражается в формировании данных в виде связанных между собой блоков, которые хранятся на некотором множестве компьютеров. Другими словами, блокчейн – это распределенная база данных, в создании и поддержании которой принимает участие множество пользователей.

Создание записей или блоков в этой базе подчиняется определенным правилам. Новые записи постоянно создаются пользователями, каждая вновь созданная запись проверяется участниками сети. После подтверждения запись присоединяется к уже существующему блоку, после этого блок сохраняется и автоматически обновляется на всех сетевых компьютерах. Обновленный блок хранится в зашифрованном виде, доступ к нему возможен с помощью электронной цифровой подписи.

Технология блокчейн имеет свои специфические преимущества.

1. Безопасность базы данных обеспечивается ее распределенным функционированием и многократным копированием. Взломать или изменить такую базу невозможно, поскольку задача по отысканию тысяч компьютеров и дешифрованию данных каждого относится к практически нерешаемым.

2. Децентрализация заложена одним из принципов существования этой сети. Технология позволяет любым пользователям напрямую обмениваться данными. Подлинность операций проверяет только сообщество пользователей. Централизованный надзор за процессом функционирования не предусмотрен.

3. Прозрачность всех операций обеспечивается публичным доступом к сохраненным записям.

Первую проверку технология блокчейн получила в 2009 г., когда на ее основе была создана криптовалюта биткойн. Опыт эксплуатации новой платформы, полученный в ходе развития этой и других криптовалют, показал жизнеспособность и перспективы существования технологии блокчейн. Области применения блокчейна в дальнейшем связаны с учетом и хранением больших объемов данных, которые представляют ценность для пользователей. К возможным областям применения относятся формирование систематизированных записей в форме реестра (электронные библиотеки, карта звездного неба), облачные системы

хранения данных, финансовые операции, контроль сделок с недвижимостью и многое другое.

БД имеет *табличную структуру*, в которой в отличие от обычной таблицы столбец называют *полем*, а строку – *записью*. Поля образуют структуру, а записи относятся к данным, которые хранятся в базе (табл. 44).

Структура – это сетка с наименованием полей, которая может быть построена и без записей. Например, ежедневник – это определенная база данных, которая может существовать в заполненном и незаполненном состоянии. Структура базы данных создается в процессе ее проектирования и не может быть изменена пользователем при занесении данных.

Таблица 44

Структура базы данных

	Поле 1	Поле 2	Поле 3
Запись 1			
Запись 2			
Запись 3			

Каждому полю при создании базы данных присваиваются определенные свойства. Свойства полей зависят от типа данных. Различаются свойства текстовых и графических данных. В приложении к текстовым данным могут быть варианты, в частности, следующие свойства.

Имя поля – по умолчанию именем поля является заголовок столбца.

Тип поля – определяет тип данных, которые могут содержаться в поле.

Размер поля – определяет предельную длину данных (в символах), которые могут размещаться в поле.

Формат поля – определяет способ форматирования данных в ячейках, принадлежащих полю.

Маска ввода – определяет форму, с помощью которой вводятся данные в поле (средство автоматизации ввода).

Значение по умолчанию – значение, которое вводится в ячейки автоматически (средство автоматизации ввода).

Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при вводе ошибочных данных.

Обязательное поле – свойство, определяющее обязательность заполнения данного поля.

Индексированное поле – полю придается дополнительное свойство для ускорения поиска или сортировки записей.

Перечень и названия свойств полей могут быть значительно расширены в зависимости от специфики базы данных. Наряду со свойством полей в БД необходимо учитывать *тип данных*. Таблицы БД могут иметь значительное число типов данных.

Текстовый – тип данных для хранения текста ограниченного размера (до 255 символов).

Поле Мемо – специальный тип данных для хранения больших объемов текста.

Числовой – для хранения действительных чисел.

Дата/время – для хранения календарных дат и текущего времени.

Денежный – для хранения данных о денежных суммах.

Счетчик – для натуральных чисел с автоматическим наращиванием, например для организации порядковой нумерации.

Логический – для хранения логических данных (Да или Нет).

Гиперссылка – специальное поле для хранения адресов Web-страниц.

Мастер подстановок – тип данных, которые вводятся из раскрывающегося списка.

Тип данных не ограничивается приведенным списком и может быть значительно расширен.

Базу данных, содержащую большое количество полей и записей, чаще всего представляют в виде нескольких таблиц, которые называют *связанными таблицами*, а саму базу данных при этом именуют реляционной базой данных. Название «реляционная» подчеркивает наличие определенных отношений между полями связанных таблиц.

Связанные таблицы состоят, как правило, из одной базовой и нескольких подчиненных таблиц или нескольких таблиц, между которыми обозначены связи. Базовая таблица и подчиненные таблицы связаны потому, что они хранятся в одной базе данных. Графическая связь между таблицами отражается с помощью схемы данных. Пример схемы данных представлен в табл. 45.

Программная связь между таблицами осуществляется с помощью полей с одинаковыми типами данных. Каждая из таблиц – базовая и подчиненная – имеют в своем составе *уникальное поле* или поле, значения в котором не могут повторяться.

Таблица 45

Схема данных в связанных таблицах



Ключевым полем, или первичным ключом, называется уникальное поле *базовой таблицы*. В ключевом поле записи не могут повторяться. Любому значению первичного ключа соответствует единственная запись в таблице.

Внешним ключом называется ключевое поле в *подчиненной таблице*. Внешний ключ *однозначно связан* с первичным ключом, поскольку это столбец

таблицы, значения которого полностью соответствуют значениям первичного ключа другой таблицы. Наличие ключевых полей позволяет осуществлять корректный переход от одной таблицы к другой. Из приведенных определений связанных таблиц и ключей следует несколько правил работы с реляционной базой данных.

1. В подчиненную таблицу нельзя добавить запись с несуществующим в базовой таблице ключом.
2. В базовой таблице нельзя удалить запись, если не удалены соответствующие записи в подчиненной таблице.
3. Изменения ключа базовой таблицы должны сопровождаться изменениями соответствующих записей подчиненной таблицы.

Системы управления базами данных

Программные средства, с помощью которых создаются, наполняются и используются базы данных, называются *системами управления базами данных* (СУБД). Одной из самых распространенных СУБД является программа Access, которая поддерживает все средства и возможности по обработке данных в рамках реляционной модели. Access входит в состав программного пакета Microsoft Office и позволяет через буфер обмена поддерживать динамический обмен данными из других программ пакета, такими как Word или Excel.

Программа Access дает возможность выполнять различные операции со связанными таблицами, в том числе создание таблиц и установление связи между ними, наполнение таблиц полями, индексирование записей, создание запросов на выборку и др. Созданные таблицы могут быть скопированы, переименованы или удалены. К каждой из таблиц могут применяться операции сортировки или фильтрации данных.

Перечисленные и многие другие операции, которые выполняет программа Access, относятся к категории управления упорядоченными записями. Управление заключается в программной обработке отдельных объектов базы данных. В качестве объектов в программе Access могут выступать данные, сгруппированные в таблицы, запросы, формы, отчеты, страницы и др.

1. Таблицы – это основные объекты любой базы данных. Таблица хранит структуру базы и все ее данные. Таблицы создаются пользователями для хранения данных.
2. Запросы – объекты для извлечения данных и представления их в удобном виде. С помощью запросов пользователь может делать отбор, сортировку и фильтрацию данных. При запросах извлекаются данные из базовой таблицы и создаются новые результирующие или подчиненные таблицы. Пример извлечения данных представлен в табл. 46.

3. Формы – это табличные средства, упрощающие процесс ввода данных в базу. Формы облегчают работу с базой данных и позволяют частично автоматизировать процесс ввода.

Базовая таблица

Комплектующие			
Микропроцессор	Модель	Цена	Поставщик

Подчиненная таблица 1

Цена	
Модель	Цена

Подчиненная таблица 2

Поставщики	
Поставщик	Модель

4. Отчеты – это средства вывода данных на печать. Данные выводятся в форме таблицы и могут иметь некоторые оформительские элементы, например колонтитулы.

5. Страницы доступа данных предназначены для управления доступа к данным, расположенным в базе. Страницы определяют тип данных, которые предоставляются конкретному пользователю.

Управление перечисленными объектами лежит в основе работы программы Access. Порядок работы с программой подробно описан в учебной литературе, с одной из версий программы можно ознакомиться в [17].

Системы автоматизированного проектирования

Система автоматизированного проектирования (САПР) – это организационно-техническая система, которая предназначена для автоматизации процесса проектирования и подготовки производства изделий различной степени сложности. САПР относятся к прикладному программному обеспечению *профессионального назначения*. В отечественной литературе эти системы принято обозначать аббревиатурой САПР, в англоязычной – CAD/CAM-системы. В англоязычной аббревиатуре заложено уточненное толкование этого понятия:

- CAD – Computer-Aided Design – система автоматизированного проектирования;
- CAM – Computer-Aided Manufacturing – система автоматизированного производства.

Ни одно конструкторское бюро или промышленное предприятие не обходится без систем автоматизированного проектирования, которые помогают создавать товары для повседневной жизни, продукцию машиностроительного профиля, строительные объекты и многое другое.

Программы САПР, предназначенные для графического представления объектов, имеют широкие возможности, в том числе:

- компьютерное конструирование деталей и изделий;

-
- трехмерное моделирование, позволяющее поворачивать, вращать, добавлять цветовые и теневые эффекты;
 - каркасное моделирование для отражения контуров деталей;
 - формирование технологии обработки деталей;
 - проектирование процессов разработки и подготовки изделия к производству.

Первая САД-система появилась в 1960 г. на фирме General Motors (США) под названием «Интерактивная графическая система подготовки производства». Уже в 1971 г. была разработана основа современных САПР. В рамках жизненного цикла изделия САПР занимает стадии проектирования и подготовки производства. Основная цель системы – повышение эффективности труда на стадиях проектирования и производства, в том числе:

- сокращение трудоемкости, себестоимости и сроков проектирования;
- повышение технико-экономического уровня проектирования;
- сокращение затрат на моделирование и испытания.

Историческое развитие систем привело к делению рынка программных продуктов, связанных с проектированием и автоматизацией производства, на несколько сегментов.

Сегмент 1 – САПР верхнего уровня (тяжелый). Сложные системы, которые применяются для решения глобальных специфических задач, например для прокладки нефтепроводов. Сегмент объединяет программы, которые отличаются обширными функциональными возможностями, высокой производительностью и значительной стоимостью. Используются в крупных производствах или отдельных отраслях. Ценовой диапазон – от 7 до 20 тыс. долл.

Сегмент 2 – САПР среднего уровня (средний). Продукты этого сегмента предназначены для поверхностного и твердотельного моделирования. Применяются при решении проектных задач в области машиностроения, архитектуры, строительства на отдельных производствах. Аппаратной основой средних систем является персональный компьютер, использование которого позволило снизить цену и сохранить при этом основные возможности тяжелых систем. Ценовой диапазон – от 4 до 7 тыс. долл.

Сегмент 3 – САПР нижнего уровня (легкий). Системы применяются для выпуска конструкторской документации в проектах ограниченной степени сложности, связанных с машиностроением, архитектурой, строительством, геодезией и др. Продукты этого сегмента *ориентированы преимущественно на двумерное конструирование*, отличаются *невысокой ценой* и имеют самое широкое распространение среди разработчиков изделий. Ценовой диапазон – до 4 тыс. долл. Сюда же можно отнести персональные системы – самые простые САПР, включающие базовые средства черчения и каркасного моделирования. Персональные системы чаще всего используются индивидуальными пользователями, ценовой диапазон этих систем в пределах 1 тыс. долл.

Программные продукты систем проектирования можно классифицировать по разным признакам. В классификации *по назначению* выделяются проектирующие и обслуживающие системы:

- проектирующие системы предназначены для решения проектных задач;
- обслуживающие системы предназначены для поддержания функционирования проектирующих систем, в состав обслуживающих систем входят:
 - системы управления проектными данными;
 - обучающие системы;
 - системы графического ввода-вывода;
 - системы управления базами данных.

Классификация *по целевому* назначению основана на представлении англоязычных аббревиатур в приложении к направлению деятельности:

- CAD (Computer-Aided Design/Drafting) – средства автоматизированного проектирования и создания чертежей;
- CAE (Computer-Aided Engineering) – средства автоматизации инженерных расчетов и анализа физических процессов;
- CAM (Computer-Aided Manufacturing) – средства технологической подготовки производства (в РФ используется аналогичный термин АСТПП – автоматизированная система технологической подготовки производства);
- CAPP (Computer-Aided Process Planning) – средства автоматизации планирования технологических процессов;
- PLM (Product Lifecycle Management) – программное обеспечение для управления жизненным циклом продукции;
- MDM (Master Data Management) – совокупность процессов и инструментов для постоянного определения и управления основными данными компании.

Если система допускает совмещение различных аспектов проектирования, то она относится к *комплексным* системам типа CAD/CAM или CAD/CAE.

В классификации *по отраслевому* назначению выделяют следующие системы:

- MCAD (Mechanical Computer-Aided Design) – система автоматизированного проектирования механических устройств;
- EDA (Electronic Design Automation) – проектирование электронных устройств, интегральных схем, печатных плат;
- AECCAD (Architecture, Engineering and Construction Computer-Aided Design) – система автоматизированного проектирования в области архитектуры и строительства.

Среди производителей систем проектирования в нашей стране к наиболее известным и востребованным относятся фирма Autodesk (США) и фирма Аскон (РФ).

Фирма Autodesk выпускает САПР® AutoCAD, начало выпуска – 1982 г. Autodesk – крупнейший поставщик САПР для машиностроения и строительства, имеет более 9 млн пользователей по всему миру. Штаб-квартира фирмы нахо-

дится в Калифорнии. Фирма имеет 80 учебных центров. Российское представительство фирмы находится в Москве. ПО фирмы Autodesk отличается обширной областью применения и широкими функциональными возможностями.

1. Архитектура и строительство. Системы проектирования позволяют до начала строительства исследовать важные характеристики объекта: стоимость, план-график работ, влияние на окружающую среду.

2. Автомобилестроение. С помощью САПР до начала производства можно подробно изучить стадии дизайна, моделирования, испытания и маркетинга.

3. Промышленное производство. САПР обеспечивают ускоренный вывод на рынок промышленного оборудования и потребительских товаров.

4. Графика и анимация. В этой области предоставляются средства для создания фильмов, видеоигр и телепрограмм.

5. Коммунальные предприятия. Системы проектирования обеспечивают решение задач по моделированию инженерных систем зданий.

Основным недостатком САПР AutoCAD является привязка к западной технологической базе. Особенности отечественной технологической базы ориентируют проектировщиков на отечественные программные продукты. Отечественным лидером разработки программного обеспечения в области автоматизированного проектирования является фирма Аскон, которая представляет на рынок систему автоматизированного проектирования КОМПАС.

Отечественная САПР КОМПАС разрабатывается и поставляется с 1989 г. Разработка системы начата в СССР и успешно продолжена в рамках фирмы Аскон. В системе использовано математическое ядро и параметрические технологии собственной разработки. Продукты Аскон применяют более 10 000 промышленных предприятий и проектных организаций в России и за рубежом, 75 000 инженеров включено в число пользователей системы. Аскон реализует комплексные проекты в *машиностроении, приборостроении и радиоэлектронике, оборонно-промышленном комплексе, атомной, нефтегазовой, химической промышленности и металлургии, промышленно-гражданском строительстве* и других отраслях.

Для машиностроения фирма разрабатывает системы проектирования, системы управления жизненным циклом изделия, системы управления нормативно-справочной информацией и системы поддержки качества продукции; для строительства – системы проектирования и управления проектными работами.

Система КОМПАС включает несколько продуктов.

1. КОМПАС-3D – система трехмерного проектирования с возможностями подготовки электронной модели изделия, здания или сооружения. Охватывает процесс разработки от стадии формирования идеи до подготовки полного комплекта документации.

2. КОМПАС-График – автоматизированная система разработки и оформления конструкторской и проектной документации. Позволяет выпускать чертежи изделий и конструкций, планы зданий, схемы, спецификации, различные ведомости и инструкции, расчетно-пояснительные записки, технические условия и др.

3. ЛОЦМАН:КБ – автоматизированная система управления проектированием и электронным архивом конструкторской документации. Позволяет организовать коллективную разработку конструкторской документации и поддерживать в актуальном состоянии электронный архив технической документации.

4. ЛОЦМАН:PLM – автоматизированная система с функциями хранения данных, управления структурой и конфигурацией изделия, календарного планирования, управления проектами и др.

5. ВЕРТИКАЛЬ – система автоматизированного проектирования технологических процессов, решающая задачи в области автоматизации технологической подготовки производства.

6. СПРАВОЧНИКИ – комплекс программ, которые обеспечивают пользователей доступной информацией о стандартных и типовых изделиях, о конструкторских материалах и сортаментах и др.

Отличительная особенность САПР КОМПАС – наличие подробных библиотек. Библиотеки – специальные приложения для расширения возможностей системы КОМПАС. Библиотека может содержать, например, функцию построения часто встречающихся геометрических фигур, в частности, резьбовых отверстий. Библиотека стандартных машиностроительных элементов значительно ускоряет процесс проектирования. Система допускает одновременную работу с несколькими подключенными библиотеками. При этом библиотеки могут быть представлены через меню или диалоговое окно.

Программные продукты автоматизированного проектирования находятся в процессе постоянного развития, ориентируясь на все более сложные объекты и используя возрастающие возможности аппаратных средств.



Компьютерные сети

Предшественниками современных компьютерных сетей были *сети терминалов*, состоящие из одной вычислительной машины, к которой подключалось несколько рабочих мест. Каждое рабочее место содержало монитор и клавиатуру. Передача данных по терминальным сетям ограничивалась, как правило, рамками одного предприятия или учреждения. Для передачи данных на большие расстояния использовались переносимые дискеты.

Начало работам по созданию современных компьютерных сетей было положено в 1960-х гг., когда возникла потребность в передаче больших объемов данных, а несколько позднее – в подключении к сетевым информационным ресурсам. В течение последующих десятилетий (1970–1980-е гг.) были разработаны и успешно опробованы основные принципы функционирования программной и аппаратной составляющей компьютерных сетей.

Аппаратное и программное обеспечение сетей

В основу современной компьютерной сети заложены следующие понятия и определения [18].

Компьютерная сеть – это совокупность соединенных между собой вычислительных машин, компьютеров, мобильных и периферийных устройств, обеспечивающих передачу и прием данных с использованием специального коммуникационного оборудования и программного обеспечения.

Каждая компьютерная сеть делится на составляющие, которые называются *компонентами*, или *слоями*, компьютерной сети:

- аппаратная компонента включает все типы компьютеров и вычислительных устройств, входящих в состав сети;
- коммуникационное оборудование складывается из совокупности кабельных систем и сетевых компьютеров, участвующих в процессе передачи данных;
- сетевая операционная система является программной платформой сети, которая управляет процессом передачи и поддерживает функционирование сети;
- сетевые приложения включают программные средства (сетевые базы данных, почтовые системы) и аппаратные средства (сетевой принтер, накопитель данных).

В зависимости от способа подключения и способа передачи данных сети бывают *однорангового* и *иерархического* типа (рис. 32).

Одноранговая схема отличается тем, что каждый пользователь может предоставить другим пользователям сети свои данные и получить доступ к данным другого пользователя. Такие сети основаны на принципе взаимоотношения «равный к равному».

Иерархическая схема относится к клиент-серверной сети, в которой пользователи или клиенты обмениваются данными через сервер, управляющий процессом обмена информацией. Отношения между клиентами в такой сети регули-

руются *сервером*. Каждый сеанс связи начинается с запроса на передачу или прием данных и продолжается после проверки и получения разрешения со стороны сервера.

Сервер сети – это специальная система управления сетевыми ресурсами общего доступа, а также процессом предоставления ресурсов сетевым пользователям. Сервер является комбинацией *аппаратного* и *программного* обеспечения.

Компьютерную сеть можно представить в виде *графа* – математической модели, в которой на вершинах или в *узлах сети* расположены вычислительные средства, соединенные между собой ребрами или средствами связи. На компьютерах, расположенных в узлах сети и называемых *хост-машинами*, хранятся сетевые ресурсы, к которым могут обращаться пользователи сети.

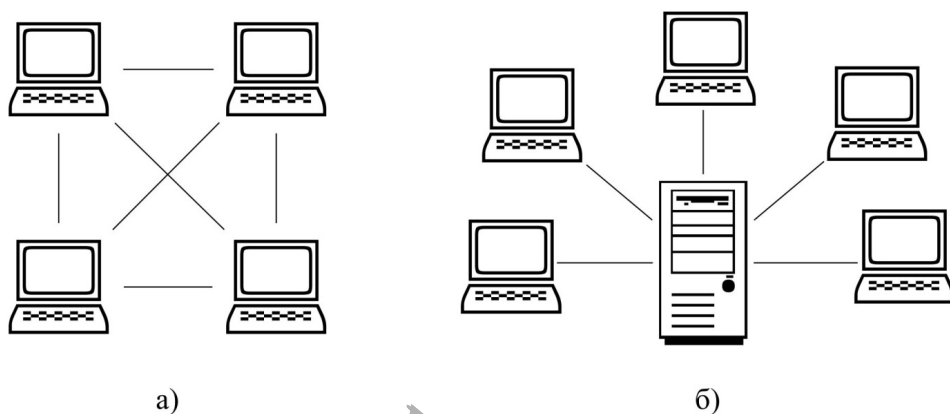


Рис. 32. Типы компьютерных сетей:
а) одноранговая, б) иерархическая.

Средства обеспечения связи между узлами называются *каналами связи*, каналы состоят из *каналообразующей аппаратуры* и *среды передачи сигнала*.

Среда передачи сигнала – физическая среда, в которой происходит распространение электрических, оптических или радиосигналов, используемых для передачи данных между компьютерами. В зависимости от типа используемой среды различают *кабельные* и *беспроводные соединения*.

К кабельным соединениям, в которых электрический сигнал передается по медному проводу, относятся *коаксиальный кабель* (медный проводник, окруженный изолятором и металлическим экраном) и *витая пара* (несколько пар изолированных проводников, скрученных между собой и покрытых пластиковой оболочкой). В *оптоволоконном кабеле* оптический сигнал передается по *кварцевым волоконным световодам*, собранным в компактный жгут.

В беспроводных соединениях передача сигнала осуществляется с помощью *радиоволн* в воздушной или безвоздушной среде.

Поток данных, передаваемых по сети, называется *сетевым трафиком*. Трафик заполнен полезными и служебными данными. Полезные данные вырабатываются и передаются пользователем. Служебные данные формируются на основе

принятых в сети *правил передачи данных*. Правила предусматривают деление служебных данных на стандартные уровни, с помощью которых организуется взаимодействие компьютеров в процессе передачи.

В состав каналообразующей аппаратуры входят (рис. 33):

- сетевой коммутатор – устройство, используемое для передачи данных между отдельными пользователями или сегментами сети; коммутатор функционирует как повторитель, который получает сообщение, определяет адрес получателя и отправляет сообщение по назначению;

- сетевой концентратор, или хаб (от *англ.* hub – центр), – устройство для усиления сигналов в процессе передачи; используется при объединении компьютеров в сеть локального типа для увеличения числа компьютеров в сети и расстояния между компьютерами и сервером;

- маршрутизатор – специализированный сетевой компьютер, который определяет маршрут следования сообщения на основе встроенного протокола маршрутизации, используется для пересылки данных между различными сегментами сети.

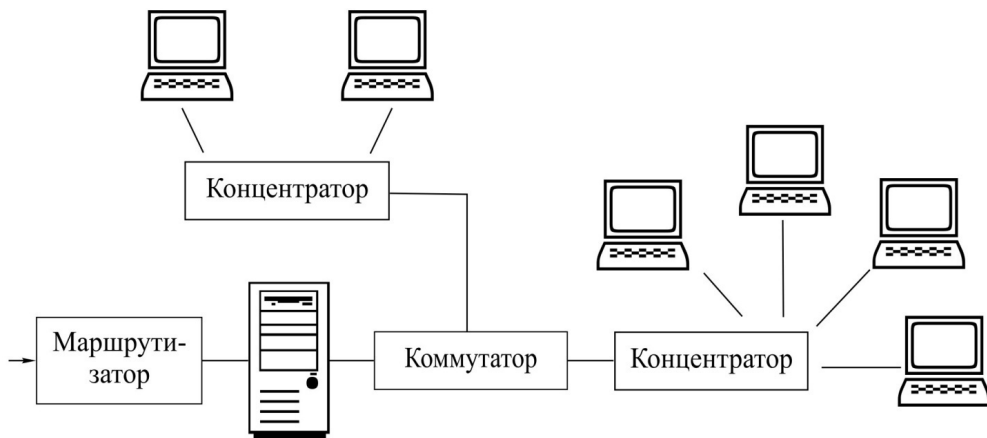


Рис. 33. Состав каналообразующей аппаратуры

Пропускная способность сети определяется количеством данных, проходящих через линию связи за единицу времени, за единицу измерения пропускной способности принимается бит/сек, Кбит/сек и т. д.

Программа-сервер – специальная программа, предназначенная для обслуживания запросов на доступ к ресурсам данного компьютера.

Программа-клиент – специальная программа, предназначенная для формирования и отправки запросов на доступ к удаленным ресурсам, а также получения и отображения полученных данных на компьютере пользователя.

Сетевая служба – пара программных модулей «клиент-сервер», обеспечивающая совместный доступ пользователей к определенному типу ресурсов, например к службам *e-mail* или *www*.

Порядок передачи данных в сетях

В основу функционирования компьютерной сети заложен принцип *пакетной коммутации*, который первоначально был разработан для передачи сообщений между внутренними сетями, а впоследствии был использован в качестве технической основы глобальной сети Интернет.

Пакетная коммутация – это способ передачи данных с *разделением* отправляемого сообщения на части (пакеты), которые передаются по сети независимо, а в конечной точке собираются в исходный вид.

Каждый из пакетов снабжается *идентифицирующими данными (адресами)* начальной и конечной точек соединения. Адресная информация пакетов обеспечивает их прохождение по сети общего пользования (возможно по разным трактам в зависимости от загруженности или работоспособности) до конечной точки соединения, где они собираются в нужной последовательности (рис. 34).

Использование пакетной коммутации имеет очевидные преимущества:

- один канал используется для одновременной передачи множества сообщений;
- сокращается время непроизводительного простоя канала;
- повышается надежность и скорость передачи;
- отсутствуют приоритеты в порядке передачи.

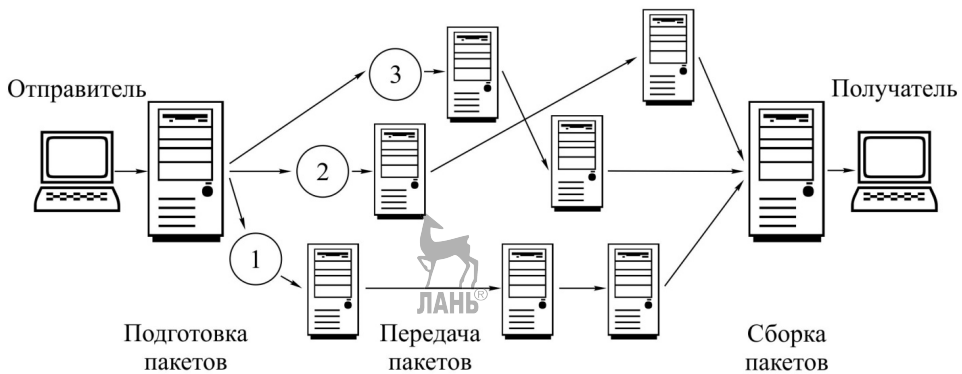


Рис. 34. Подготовка и передача сообщения в пакетной форме

Для реализации пакетной коммутации разработан специальный *протокол передачи данных*.

Протокол передачи – это совокупность правил, регламентирующая формат и процедуру обмена данными между компьютерными сетями. Протокол, обеспечивающий совместимость внутренних сетей различного устройства, называют *стандартным*.

Стандартный протокол – это общепринятый протокол или набор протоколов, обеспечивающий связь между различными компьютерными сетями. Примером стандартного протокола является взаимозависимый набор протоколов TCP/IP

(*Transmission Control Protocol/Internet Protocol*), который был разработан для передачи сообщений между внутренними компьютерными сетями различных систем.

В основу протокола TCP/IP заложены четыре принципа:

1. Каждая сеть должна сохранять свою индивидуальность и не должна подвергаться переделкам при подключении к другим сетям.
2. Передача данных должна выполняться до тех пор, пока все пакеты одного сообщения не прибыли в пункт назначения, иначе источник должен сделать повторную передачу.
3. Для связывания сетей должны использоваться компьютеры, которые пересылают пакеты, но не хранят информацию о протекающих потоках.
4. На эксплуатационном уровне в сетях не должно существовать глобальной системы управления.

Классификация и топология компьютерных сетей

Существуют различные системы классификации компьютерных сетей. В классификации по принципу *территориального размещения* выделяют следующие категории.

1. Локальные сети. Объединяют компьютеры, расположенные в пределах здания или некоторой ограниченной территории (обычно в радиусе 1–2 км) и соединенные между собой кабелями (чаще всего витой парой). Для обозначения локальных сетей используются аббревиатуры ЛВС (локальная вычислительная сеть) или LAN (*Local Area Network*) – объединение компьютеров, сосредоточенных на небольшой территории.

2. Городские сети. Охватывают группу зданий, реализуются обычно на оптоволоконных кабелях. Эти сети предназначены для обслуживания территорий крупного города, для обозначения городских сетей используется аббревиатура MAN (*Metropolitan Area Network*). В число городских сетей включаются:


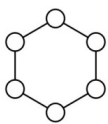
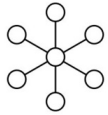
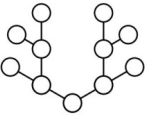
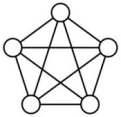
- кампусная сеть – объединяет сети в близко расположенных зданиях: в разных корпусах предприятия или в учебных зданиях университета;
- сеть городского масштаба – объединяет локальные сети предприятий и организаций в пределах городского района;
- широкомасштабная сеть – объединяет сети в пределах крупного города (мегаполиса).

3. Глобальные сети. Охватывают значительные по величине территории или регионы земного шара, используют разнообразные каналы связи, включая спутниковые. Для обозначения глобальных сетей используется аббревиатура WAN (*Wide Area Network*) – это сети, объединяющие территориально рассредоточенные локальные сети.

В пределах одной категории сети могут различаться по типу *топологии*.

Топология сетей – это геометрические принципы прокладки сети или способы соединения компьютеров в сеть. Любая компьютерная сеть строится на основе стандартных видов топологий: общая шина, звезда, кольцо, древовидная (табл. 47).

Стандартные компьютерные топологии

Наименование топологии	Схематическое изображение
Общая шина	
Кольцо	
Звезда	
Древовидная	
Полносвязанная	

Каждая из стандартных топологий может использоваться при проектировании сети, обычно вид топологии определяется схемой распределения компьютеров в здании. В чистом виде стандартные топологии используются редко, большинство компьютерных сетей построено на основе смешанной топологии. Смешанная топология строится путем комбинации стандартных, например несколько «звезд» объединяются «общей шиной».

Модель компьютерной сети

Модель сети предназначена для стандартизации подходов к разработке компьютерных сетей и унификации процесса передачи и приема данных. Модель определяет правила, которые можно использовать для обмена данными в сети и для разработки сетевого оборудования. Первая версия модели была разработана в 1977 г. под названием *Базовая модель взаимодействия открытых систем*, или *семиуровневая модель OSI (Open System Interconnection)*. Доработанная в 1984 г. модель OSI была утверждена на уровне *международного стандарта*.

Модель OSI основана на *уровневых протоколах*, что позволяет упростить программу сложной сети путем ее деления на простые части (уровни), ввести

стандартные интерфейсы и протоколы на каждом уровне, а также использовать общий язык для взаимопонимания разработчиков.

Для любого узла сети вводится 7 уровней обработки данных: *прикладной, представления данных, сеансовый, транспортный, сетевой, канальный, физический*.

За каждым уровнем закреплена вполне определенная функция обработки и набор протоколов, которые используются для поддержки уровня. Каждый из уровней отвечает за обработку данных по определенным правилам.

1. Прикладной уровень обеспечивает пользователю доступ к сетевым ресурсам и определяет перечень протоколов, которые используются для формирования запросов.

2. Уровень представления данных обеспечивает стандартные способы кодировки, форматирования, сжатия или распаковки данных.

3. Сеансовый уровень отвечает за установку и поддержку сеанса связи между пользователями сети, определяет правила формирования и передачи пакетов.

4. Транспортный уровень отвечает за передачу и сборку пакетов на стороне получателя, определяет пути следования пакетов, следит за качеством доставки.

5. Сетевой уровень обеспечивает маршрут следования пакетов и организует связь между разными сетями.

6. Канальный уровень предназначен для поиска возможных ошибок и восстановления данных в случае потери или сбоя.

7. Физический уровень обеспечивает преобразование двоичного кода в электрические импульсы для передачи сигнала и отвечает за процедуры управления аппаратной частью.

Существование модели компьютерной сети упрощает процесс передачи данных между существующими сетями и обеспечивает возможность подключения новых сетей к глобальной системе.



Глобальная компьютерная сеть Интернет

Интернет относится к глобальным компьютерным сетям. В топологии глобальной сети основной элементарной единицей является локальная сеть, поэтому Интернет иногда называют *сетью сетей*, или *Всемирной паутиной*. С понятием Интернет тесно связано английское слово *web*-сеть, сплетение, паутина, от которого происходит сетевая терминология: *web*-дизайн, *web*-сайт, *web*-камера.

Интернет – всемирная компьютерная сеть, состоящая из разнообразных компьютерных сетей, объединенных стандартными соглашениями о способах обмена информацией (протоколами) и единой системой адресации.

Топология сети Интернет аналогична сети транспортных магистралей, схема которых изображается на карте автомобильных или железных дорог. Используя аналогию, каналы связи между сетями можно представить как транспортные магистрали, систему адресации глобальной сети – как почтовые адреса, а протоколы обмена данными – как правила перевозки.

История Всемирной паутины

В истории сети Интернет выделяется 3 периода развития: этап *экспериментальных работ*, этап *становления*, этап *коммерческого применения и совершенствования* [19].

Этап экспериментальных работ (60–70-е гг. XX в.) был начат в США с экспериментов по созданию нового средства связи, которое может сохранять функционирование при частичном разрушении каналов связи. Цель экспериментальных работ заключалась в создании коммуникационной структуры, которая обеспечивает оперативный обмен данными, в том числе в условиях непредвиденного выхода из строя отдельных линий связи. Экспериментальные работы выполнялись в Департаменте техники обработки информации Министерства обороны США (ARPA). Основные результаты работ, выполненных с целью проверки теоретических предположений, включают этапы:

- создание в 1969 г. первой компьютерной сети под названием ARPANET, которая состояла из четырех компьютеров, расположенных в разных университетах США;
- совершенствование сети ARPANET, увеличение числа сетевых компьютеров до 40 штук, демонстрация работы сети, в том числе с подключением иностранных узлов в Великобритании и Норвегии;
- демонстрация в 1977 г. возможности объединения четырех различных компьютерных сетей, включая сеть ARPANET, на основе правил обмена данными, которые в дальнейшем были использованы в качестве основы для базовых протоколов сети Интернет.

В ходе экспериментальных работ были разработаны и проверены основные теоретические предположения, в том числе:

- возможность передачи сообщений по обходным маршрутам при случайном разрушении используемых каналов связи;

– возможность повышения надежности передачи данных за счет разбиения отправляемого сообщения на части (пакеты), которые передавались независимо и собирались в пункте назначения.

Этап становления (1980-е гг.), на котором были подтверждены и усовершенствованы результаты первого этапа, включает следующие основные события:

– разработка правил *межсетевого взаимодействия* в форме протокола TCP/IP и перевод в 1982 г. сети ARPANET на этот протокол;

– увеличение числа компьютеров в этой сети до 1000 шт.;

– появление термина «Интернет» (1982 г.) как глобальной компьютерной сети;

– объединение 5 компьютерных центров США в новую сеть под названием NSFNET (NSF – *National Science Foundation*) с увеличением числа компьютеров в этой сети к 1989 г. до 100 тыс. шт.

На этапе становления к концу 1980-х гг. в США была создана компьютерная сеть национального масштаба, которая впоследствии стала базовой основой сети Интернет.

Этап коммерческого применения и совершенствования (конец 1980-х – 1990-е гг.), в течение которого завершается формирование основных элементов сети Интернет и правил обработки сетевых документов. Принципы обработки сетевых документов были разработаны в Церне – Европейском совете по ядерным исследованиям (Швейцария). Основные результаты этапа:

– разработка в 1989 г. технологии публикации гипертекстовых документов, включая создание сетевого протокола HTTP, языка разметки сетевых документов HTML и системы адресации URL, которые со временем стали основой Всемирной паутины;

– реализация в 1991 г. протокола WWW (World Wide Web) и создание первого сайта, который был посвящен описанию устройства сети Интернет;

– создание в 1994 г. Консорциума Всемирной паутины (W3C) для разработки и внедрения технологических стандартов Интернета;

– мощное развитие глобальной сети во второй половине 1990-х, в том числе появление сетевой операционной системы Windows 95, начало массовой разработки корпоративных сайтов, развитие электронной коммерции и интернет-магазинов.

Этап характеризуется также лавинообразным ростом числа компьютеров подключенных к сети Интернет. В 1992 г. их общее количество превысило 1 млн. Лидером по производству компьютеров становится фирма IBM. Интернет получает признание на уровне власти, бизнеса и средств массовой информации. Между ведущими производителями программного обеспечения для Интернета возникает острая конкурентная борьба, лидером в этой борьбе выступает корпорация Microsoft.

Протоколы и система адресации

Одна из основных особенностей глобальной сети – единые правила передачи и приема данных или единый сетевой протокол, который обеспечивает подключение к сети разнотипных компьютеров, работающих под управлением разных

операционных систем. Первоначально каждая локальная сеть разрабатывалась по собственным правилам, которые создавались в процессе самой разработки. В результате для функционирования сетей было создано множество различных протоколов. Для связи сетей с разными протоколами был разработан основополагающий протокол TCP/IP, который обеспечивает взаимодействие почти 100 протоколов различного уровня.

Протокол TCP/IP является базовым протоколом глобальной сети. Протокол состоит из двух частей, первая из которых (TCP) отвечает за разбиение сообщения на блоки, добавление служебных данных и формирование пакета, сборку пакетов после передачи. Вторая часть протокола (IP) отвечает непосредственно за проводку пакетов по сети с соблюдением системы адресации.

Наряду с основополагающим протоколом используются и другие протоколы, которые строятся на основе базового. Некоторые из этих протоколов приведены ниже.

Протокол UDP – стандартный упрощенный протокол, обеспечивающий упаковку данных в один пакет, который называется *дейтограммой*. Протокол не содержит средств обнаружения ошибки, не гарантирует доставку и может использоваться для передачи коротких сообщений.

Протокол HTTP – создан для передачи гипертекста и является основой системы поиска и передачи мультимедийных документов (World Wide Web).

Протокол FTP – специально разработан для передачи файлов, которые находятся на компьютере пользователя или хранятся на специальном сервере.

Протокол WAP – беспроводной протокол передачи данных, используется для доступа к сети мобильных телефонов, оснащенных программами просмотра – браузерами. Протокол ориентирован на малый размер экрана, на ограниченный объем памяти и малую пропускную способность.

Другой базовой основой Интернета является единая система адресации компьютеров и документов. Каждый компьютер в сети имеет два функционально равноценных адреса, первый из которых формируется на основе IP-адресации, а второй вырабатывается по системе DNS.

В основе *системы IP-адресации* заложено присвоение каждому сетевому компьютеру уникального адреса в числовом формате, который содержит 4 блока чисел (по одному байту), при этом каждое число заключено в интервале от 0 до 255 и не может иметь больше трех разрядов. IP-адрес недействителен, если хотя бы один из блоков превышает значение 255. При записи IP-адреса блоки чисел разделяются точками и приводятся к виду, структура которого показана на рис. 35.

Каждый блок адреса имеет определенное значение. Начало адреса определяет *номер большой сети*, в которой находится компьютер, а крайний правый – *номер компьютера в локальной сети*. В результате каждый компьютер получает единственный в своем роде номер, на



Рис. 35. Структура сетевого IP-адреса

основе которого можно найти компьютер в сети. Для разделения адреса на структурные составляющие используют *маску сети*.

Маска сети строится на битовой основе и используется для того, чтобы определить, какая часть IP-адреса относится к адресу *сети*, а какая – к адресу *подсети*. Чтобы получить адрес сети, нужно применить операцию поразрядной конъюнкции (логическая операция И) к IP-адресу и маске подсети. В табл. 48 показан переход к адресу сети с помощью битовой маски.

Таблица 48

Поразрядная конъюнкция IP-адреса и маски сети

IP-адрес	192.168.1.2	11000000 10101000 00000001 00000010
Маска подсети	255.255.255.0	11111111 11111111 11111111 00000000
Адрес сети	192.168.1.0	11000000 10101000 00000001 00000000

Система IP-адресации делится на *классы* в зависимости от того, сколько бит в адресе отводится под номер *сети* и сколько под номер подсети (табл. 49).

Класс А содержит адреса сетей, которые имеют в первом байте значения от 1 до 127 и большие возможности по количеству сетевых компьютеров, которые можно использовать в качестве хостов. Адреса этого класса назначаются *крупным организациям*, которые имеют большое количество сетевых устройств.

Класс В имеет в первом байте значения от 128 до 191, что позволяет подключить к нему до 16384 сетей. Адреса класса В назначаются сетям *большого и среднего* размера.

Класс С объединяет адреса со значениями в первом байте от 192 до 223, что позволяет увеличить количество возможных сетей до 2 млн и более. Адреса этого класса назначаются *малым сетям*.

Таблица 49

Классификация IP-адресов

Класс сетей	Значение первого байта адреса	Количество сетей	Количество хостов в сети
А	001-126	126	16777214
В	128-191	16384	65534
С	192-223	2097152	254

На практике пользоваться длинными численными адресами неудобно, поэтому наряду с IP-системой разработана и существует другая система регистрации.

Доменная система адресации, или доменная система имен (DNS – *Domain Name System*), основана на делении сети на *части* или *зоны* с присвоением каждой зоне имени в виде последовательности символов, имеющей смысловую окраску.

В этой системе адресации используется термин «домен», обозначающий зону, сообщество или участок сети, к которому принадлежит данный компьютер.

В доменной системе имен каждому IP-адресу присваивается собственное имя, которое называется DNS-адрес.

Доменное имя – это буквенный адрес компьютера. DNS-адрес формируется из нескольких доменов, домены разделяются точками и считаются по порядку справа налево. При этом адрес выглядит, например, так – miass.city.ru. Структура доменного имени показана на рис. 36.



Рис. 36. Структура доменного имени

В этой системе адресации справа находится домен *первого уровня*, объединяющий самую крупную группу компьютеров. Чаще всего домен первого уровня состоит из двух букв латинского алфавита и формируется на основе буквенной кодировки государств, примеры которой приведены в табл. 50.

Таблица 50

Примеры доменов первого уровня в кодировке государств

Государство	Домен	Государство	Домен
Россия	su, ru	Великобритания	uk
Белоруссия	by	Германия	de
Украина	ua	Франция	fr
Казахстан	kz	Италия	it

Наряду с буквенной кодировкой государств существует *наднациональная кодировка* доменов, примеры которой приведены в табл. 51.

В 2011 г. в правила наименования доменов внесены изменения, на основе которых для кодировки государств можно использовать не только латинские буквы, но и буквы национального алфавита. На основе нового правила в России

в дополнение к существующим доменам **su** и **ru** введен русскоязычный домен **рф**.

Таблица 51

Наднациональные домены первого уровня

Назначение или зона охвата	Домен
Европейский союз	eu
правительственные	gov
военные	mil
коммерческие	com
образовательные	edu
сетевые	net
прочие организации	org

Двойная система адресации компьютеров может существовать только при полном соответствии двух функциональных сетевых адресов. Для соответствия числового и доменного адреса в Интернете имеется единый *электронный каталог*, который ведется в США. Для преобразования доменного адреса в числовой формат посылается запрос на DNS-сервер. Сервер обрабатывает доменное имя, определяет IP-адрес и отправляет ответ на запрос.

Единую систему адресации компьютеров сети Интернет дополняет единое адресное пространство всех сетевых документов. Каждый документ, размещенный на серверах сети, имеет свой адрес, который формируется на основе *унифицированного определителя адресов сетевых ресурсов URL*, или *стандарта URL*.

URL (*Universal Resource Locator*) – общий формат представления адреса сетевого ресурса. Формат URL содержит сведения о компьютере и сведения о документе, которые объединяются в единую строку.

имя протокола://DNS-адрес компьютера/путь к документу/ имя файла.

Стандарт URL используется для обозначения и поиска всех ресурсов в сети Интернет.

Службы Интернета

Службами Интернета принято называть средства предоставления услуг для пользователей глобальной сети в форме доступа к определенным информационным ресурсам или предоставления возможности для обмена сообщениями между пользователями [19].

WWW (*World Wide Web*) – это наиболее распространенная служба, которая представляет собой сеть гипертекстовых документов (рис. 37), связанных между собой *гиперссылками*.

Наименьшей информационной единицей в этой службе является web-страница, представляющая собой совокупность текстовых, графических и мультимедийных файлов. Группа web-страниц, принадлежащих одному владельцу и связанных между собой по содержанию, образует *web-сайт*.

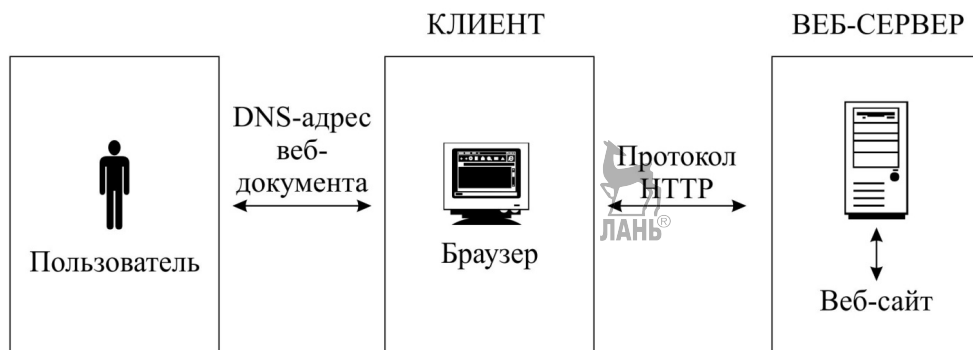


Рис. 37. Просмотр гипертекстовых документов

Web-сайт – это множество информационных ресурсов, к которым открыт доступ в службе WWW. Сайты можно просматривать с помощью браузеров – программ-обозревателей. К ним относятся Internet Explorer, Opera, Mozilla Firefox, Google Chrome. Материалы, представленные на сайте, или содержательное наполнение сайта, называют *контентом*.

Электронная почта (*E-mail*) – наиболее старая и одна из самых массовых служб в сети. Служба предназначена для обмена электронными письмами между пользователями в режиме отложенного общения (*off-line*). В электронной почте каждый пользователь получает персональный почтовый адрес, который состоит из двух частей, разделенных символом @. Справа от символа располагается DNS-адрес компьютера, на котором размещается *почтовое отделение* абонента. Слева от символа @ расположено *имя абонента*.

Служба передачи файлов основана на протоколе передачи файлов FTP (*File Transfer Protocol*). Хранение файлов обеспечивают так называемые FTP-серверы, а пользователи выступают в роли FTP-клиентов, имеющих доступ к файлам. На серверах чаще всего находятся программные файлы, но могут храниться файлы и любых других форматов. Данные на серверах образуют *иерархическую структуру папок*, которая удобна для поиска и хранения.

Социальные сети пришли на смену популярным службам, предназначенным для организации процесса общения между пользователями. Социальные сети реализуются в форме сайта или постоянно подключенного к сети сервиса, который позволяет строить социальные связи, устанавливать взаимоотношения, обмениваться разнообразной информацией. Особенностью социальных сетей являются неограниченные возможности по размещению личной информации, по

обмену информацией с другими пользователями, по созданию групп или сообществ, имеющих близкие интересы.

Кроме перечисленных к службам сети Интернет относится целый ряд известных систем: видеоконференции, системы поиска, электронные средства массовой информации, системы электронных платежей, online-переводчики, словари и др.

Публикация web-документов

Web-документ – это электронный документ, содержащий текстовые, графические и другие данные, которые отформатированы с учетом особенностей приема и передачи информации в сети Интернет. Для создания web-документов можно использовать обычные текстовые и графические редакторы, но более эффективными являются *web-редакторы*, которые применяют на профессиональном уровне.

Форматированный web-документ называют также HTML-документом, или HTML-кодом, поскольку форматирование осуществляется на основе инструкций (команд) языка HTML (Hypertext Markup Language), который называют языком *гипертекстовой разметки*.

Гипертекст – это система текстовых страниц, имеющих перекрестные гиперссылки. Гипертекст составляет основу любого web-документа.

HTML – стандартизованный язык разметки документов, предназначенных для *публикации* во Всемирной паутине.

Стандарт представления информации в сети Интернет основан на использовании языка HTML. Первоначально различные компании-разработчики предлагали свои варианты представления данных в сети, но со временем возникла необходимость в создании стандартного языка, который выполняет эту функцию. Первый стандарт языка HTML был сформирован в середине 1990-х гг., а в 1997 г. была утверждена спецификация HTML 3.2, которая принята разработчиками в качестве официальной версии. Быстрое развитие технологии публикации web-документов приводит к постоянному дополнению и обновлению официальной версии [20].

HTML представляет собой набор команд или инструкций, которые называются *тегами*. Файлы, содержащие инструкции языка HTML, имеют расширение .htm или .html.

Теги используются для описания структуры документа и представляют собой определенные ключевые слова. Теги позволяют выделить в документе отдельные логические части – заголовки, абзацы, таблицы, списки-перечисления, а также могут использоваться для связи web-страницы с блоками изображения, видео и звука. Процедура форматирования с использованием тегов похожа на процесс верстки типографического издания, поэтому ее иногда называют HTML-*версткой*. Теги не отображаются на экране или при печати, но при этом управляют отображением.

Просмотр размеченных тегами документов осуществляют с помощью браузеров – программных средств, предназначенных для работы с web-доку-

ментами. Однако может возникнуть ситуация, когда разные браузеры по-разному отображают один и тот же HTML-код. Иногда разработчик web-страниц предлагает пользоваться конкретным браузером для наилучшего отображения web-документа на экране. Разница в отображении возникает в случае, когда браузеры (Internet Explorer или Mozilla Firefox), с помощью которых пользователи просматривают web-страницы, не полностью соответствуют обновленным версиям HTML.

С появлением *мобильных сенсорных устройств* возникли новые требования к HTML-верстке. В новых условиях сайт должен корректно отображаться не только при обычных разрешениях монитора, но и быть адаптированным к мобильным устройствам или иметь мобильную версию.

Теговая модель описывает документ как совокупность *контейнеров*, каждый из которых начинается и заканчивается тегами. Теги HTML-документов в большинстве своем просты и понятны, так как они образованы с помощью общеупотребительных слов английского языка, понятных сокращений и обозначений. Текст тега заключается в угловые скобки (" $<$ " и " $>$ "). Простейший вариант тега – имя, заключенное в угловые скобки, например `<HEAD>`. Общая схема построения контейнера в формате HTML может быть записана в следующем виде:

```
"контейнер"= <"имя тега" "список атрибутов">  
содержание контейнера  
</"имя тега">
```

Исходя из этой схемы простой текстовый HTML-документ, содержащий заголовков и текстовый блок, в контейнерной форме выглядит следующим образом (рис. 38):

```
<HTML>  
<HEAD><TITLE> Заголовок документа</ TITLE></ HEAD>  
<BODY>  
Текст документа  
</BODY>  
</ HTML>
```

HTML-код простого документа можно создать с помощью любого текстового редактора, например редактора Блокнот. Если сохранить созданный файл с расширением .htm или .html, то полученная страница может быть просмотрена любым браузером. Для создания сложного HTML-документа необходимо использовать специальные *редакторы* Dreamweaver или Front Page.

В промышленных масштабах при создании крупных сайтов с сотнями или тысячами web-страниц необходимо создавать *программу*, которая управляет системой переходов между страницами. Для составления программы используют *языки программирования (скрипт-языки)*, которые позволяют также автоматизировать создание HTML-кода и управлять элементами кода. Для управления

сложными системами используются языки программирования Java, JavaScript, PHP и др.

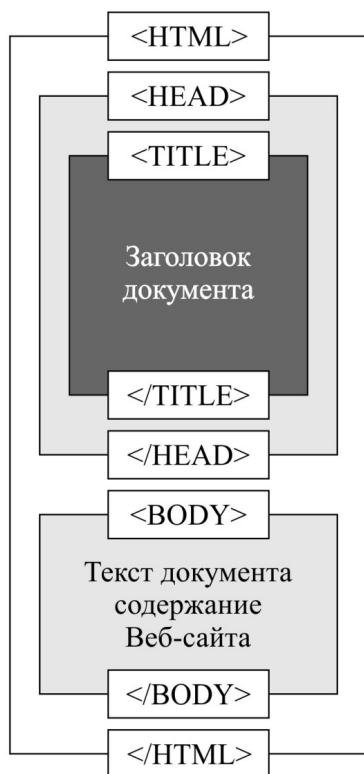


Рис. 38. Структура HTML-документа

Следует также отметить, что современный язык HTML дополнен технологией CSS (Cascading Style Sheets – каскадные таблицы стилей) [21]. Необходимость дополнения языка связана с тем, что HTML используется для описания состава и логической структуры web-документа, а основное назначение технологии CSS – описание внешнего вида этого документа или его форматирование.

Каскадные таблицы стилей CSS – это формальный язык описания стиля документа, который создан с использованием языка разметки HTML, другими словами, это набор *правил форматирования* тегов.

Таблицы стилей предназначены для поддержания единства стиля, которое не обеспечивается технологией теговой разметки. Данные о стиле хранятся отдельно и включают в себя различные элементы дизайна документа, в частности, шрифт текста, цвет фона, цвет ссылок, расположение встроенных объектов. Термин «каскадные таблицы» означает, что на одной веб-странице могут использоваться разные типы стилей в зависимости от выбранного метода применения. При использовании каскадных таблиц стилей может быть выбрано три метода применения:

- встроенный – обеспечивает выбор стиля для любого тега, например для одного абзаца;
- внедренный – позволяет присваивать выбранный стиль всей HTML-странице;
- связанный – предназначен для создания образцов стилей, которые используются в качестве единых для любого документа, размещенного на веб-узле.

При просмотре web-документа браузер, который поддерживает таблицы стилей, будет соблюдать определенный порядок интерпретации стилей. В первую очередь будут реализованы связанные стили, затем – внедренные и, наконец, встроенные.

Принципы построения сайтов в сети Интернет

Каждый сайт в сети предназначен для решения какой-либо *конкретной задачи*: обучение, повышение уровня продаж, создание определенного имиджа

и др. Несмотря на большое разнообразие сайтов, можно выделить общие закономерности и принципы формирования сайтов.

1. Контент сайта должен иметь *эксклюзивный* характер и отличаться от сайтов похожих тематик уникальностью представленных материалов.

2. Все размещенные на сайте материалы должны быть *достоверными* и регулярно *обновляемыми*.

3. Документы сайта, как правило, выполняются в едином *стиле* и снабжаются средствами *навигации*, которые обеспечивают удобство пользования и высокую посещаемость.

4. В сетевых документах необходимо сочетать ограничение по объему с высоким качеством представленных материалов.

Порядок построения сайтов также имеет общие закономерности и включает 3 основных этапа (рис. 39).



Рис. 39. Построение сайта в сети Интернет

На первом этапе выполняется разработка структуры сайта, которая включает следующие виды работ.

1. Составление технического задания, в котором заказчик согласует свои намерения с разработчиком сайта. Для рекламного сайта, например, работа начинается с брифа – краткой письменной формы согласия между рекламодателем и рекламистом.

2. Разработка основной и типовых страниц сайта – выполняется веб-дизайнером в соответствии с техническим заданием. Результат работы состоит из нескольких электронных страниц, каждая страница представляет собой многослойный рисунок.

3. HTML-верстка – созданные дизайнером страницы дополняются кодами разметки, что позволяет просматривать и анализировать материалы в браузере.

Второй этап включает работы по созданию и проверке системы управления сайтом.

4. Программирование – создание управляющей программы, которая формирует так называемый «движок», отражает структуру навигации и управляет переходами между страницами.

5. Тестирование – проверка функционирования сайта в разных браузерах. Обнаружение и исправление ошибок, допущенных ранее.

6. Тестирование в сети Интернет – отправка материалов на сервер провайдера, адресная настройка и тестирование с удаленных компьютеров. На этом этапе сайт закрыт для посетителей.

На третьем этапе работа с сайтом завершается публикацией в глобальной сети и продвижением в поисковых системах.

7. Наполнение контентом и публикация – в структуру сайта встраивают заготовленные текстовые блоки, изображения, файлы. Готовый сайт открывается для пользователей.

8. Продвижение сайта или оптимизация – выполняется с целью увеличения посещаемости и состоит из двух стадий. На первой стадии выполняется внутренняя оптимизация, в процессе которой определяются и вставляются в контент ключевые слова, которые привлекут наиболее заинтересованных посетителей. Одновременно тексты и ссылки подстраиваются под работу поисковых систем. На второй стадии осуществляется внешняя оптимизация или «раскрутка» сайта – длительный и трудоемкий процесс совершенствования, направленный на повышение конкурентоспособности сайта. Стоимость внешней оптимизации может превысить расходы на создание сайта.

9. Сдача проекта – ознакомление заказчика с результатами работы и подписание акта приемки-сдачи. Сдача проекта может сопровождаться обучением представителя заказчика навыкам работы в администраторской зоне сайта.

Множество представленных в сети сайтов имеет разную степень восприятия. На одних сайтах все понятно с первого взгляда, на других пользователь испытывает затруднения при работе. Степень восприятия сетевых материалов принято оценивать понятием «юзабилити» – мерой дружелюбности сайта. В переводе с английского это слово означает удобство пользования. Понятие «юзабилити» складывается из комбинации факторов, влияющих на восприятие экранной компьютерной информации. К числу основных относятся следующие факторы юзабилити.

1. Простота обучения. Отражает время, необходимое пользователю для решения простых задач при условии, что интерфейс сайта он видит впервые.

2. Эффективность использования. Отражает время, необходимое для решения задач после прохождения обучения на сайте.

3. Запоминаемость. Характеризует часть материалов, которую пользователь не смог запомнить после первичного обучения и пользования материалами сайта.

4. Количество и серьезность ошибок. Включает число простых и серьезных ошибок, допущенных пользователем при работе с сайтом.

5. Субъективное удовлетворение. Формируется на основе индивидуальной оценки пользователя.

Для численной оценки факторов применяют юзабилити-тестирование, которое заключается в наблюдении и фиксации вышеперечисленных факторов для

нескольких простых пользователей. Цель юзабилити-тестирования заключается в анализе полученных данных и выработке пакета рекомендаций по улучшению оформления, оптимизации информационной структуры, системы навигации и технологических решений сайта.

Способы соединения с глобальной сетью

Подключение к сети производится через организации, которые являются поставщиками интернет-услуг и называются *провайдерами*. Провайдеры предоставляют разные варианты подключения, отношения с провайдером строятся на договорной основе.

Модемное соединение с поставщиком услуг осуществляется по телефонным сетям с помощью *модема*, обеспечивающего *модуляцию-демодуляцию* сигналов. Это один из первых доступных способов подключения компьютера к сети Интернет через провайдера. Первые образцы модемов (низкочастотные) позволяли подключать компьютер только с отключенным телефоном. Высокочастотные модемы в отличие от низкочастотных обеспечивают одновременную передачу интернет-трафика и телефонного разговора. Основным недостатком этого соединения является низкое качество телефонных сетей и ограниченная скорость передачи данных, связанная с низкой пропускной способностью телефонных сетей. По этой причине первоначально распространенное модемное соединение уступает место более совершенным способам.

Соединение по выделенной линии основано на передаче сигнала по кабелю, соединяющему провайдера с пользователем услуг. Выделенная линия обеспечивает прямую постоянную высокоскоростную связь, которая используется только для соединения с сетью. Способ широко используется в условиях городской застройки. К недостаткам относится высокая стоимость подключения и абонентской платы, особенно по тарифам высоких скоростей.

Радиодоступ относится к беспроводному способу соединения, основанному на использовании приема-передающего радиооборудования, в том числе на основе беспроводных модемов. Радиодоступ обеспечивает высокую мобильность пользователя, например, в полевых условиях. Одним из недостатков радиодоступа является ограничение зон охвата радиосигналом. Надежность радиодоступа уступает соединению по выделенной линии.

Спутниковое соединение чаще всего основано на комбинированном способе доступа, при котором запрос на связь отправляется по телефонной линии, а ответ – через спутниковую антенну. Способ доступен по цене и имеет высокую скорость передачи данных. Одновременно, как правило, можно принимать спутниковые телевизионные каналы. Каждый передающий спутник имеет свою карту покрытия, которая ограничивает зоны обслуживания спутниковой связью.

Все способы соединения с глобальной сетью отличаются между собой по ряду показателей: принцип работы, скорость передачи данных, надежность, сложность настройки и поддержки, цена установки и эксплуатации. Выбор способа подключения определяется конкретными условиями, в которых находится пользователь.

Защита компьютерных данных

Вопросы по защите информационных систем возникают в связи с существованием угроз компьютерной безопасности [22], которые могут привести к потере, разрушению или искажению данных, а также к потере программных или аппаратных средств. Все угрозы компьютерной безопасности принято делить на случайные (непреднамеренные) и умышленные.

Случайные угрозы вызваны произвольным воздействием, таким как ошибочные действия пользователей, просчеты в программном обеспечении, выход из строя аппаратных средств и т. п.

Умышленные угрозы нацелены на нанесение ущерба и подразделяются в свою очередь на пассивные и активные.

Пассивные угрозы не оказывают влияния на функционирование информационного ресурса. Получение информации путем считывания данных, например, не нарушает функционирование каналов связи или носителей информации.

Активные угрозы направлены на нарушение процесса функционирования системы путем целенаправленного воздействия на информационные ресурсы. Примером такого рода угрозы является умышленное искажение сведений в базе данных или разрушение программного обеспечения. Источниками активных угроз чаще всего бывают действия злоумышленников с использованием вредоносных программ – *вирусов*, которые создают злоумышленники.

К основным угрозам компьютерной безопасности относят:

- хищение конфиденциальных данных или пароля доступа к данным;
- хищение электронного носителя;
- внесение несанкционированных изменений (компрометация);
- несанкционированное использование информационных ресурсов с целью нанесения ущерба владельцу;
- уничтожение информационных массивов или аппаратных средств.

Перечисленные угрозы могут не только приостановить работу компьютера, но и привести к более серьезным последствиям, таким как нанесение финансового ущерба или сбой в деятельности предприятия, организации, отрасли.

Одним из надежных способов защиты компьютерных данных является система управления доступом к информационным ресурсам.

Системы управления доступом

Система управления доступом или контроля доступа основана на делении пользователей информационного ресурса на категории, каждая из которых подчиняется своим правилам доступа или принятому в сети протоколу доступа. Протокол доступа регламентирует права каждого пользователя ресурса и ведет учет всех пользователей, которые знакомятся с информацией или изменяют содержание ресурса. Деление на категории пользователей, у которых имеется доступ, и предоставление ресурсов в пользование осуществляется с помощью системы авторизации. Авторизация – это процедура входа в информационную систему,

в процессе которой устанавливается имя пользователя и пароль, которые обеспечивают доступ.

Имя пользователя или идентификатор является несекретной последовательностью символов, которая присваивается каждому пользователю при регистрации в системе. Идентификатор может обозначаться термином login (авторизоваться), после которого следует имя – инициалы, фамилия, уникальный набор букв или цифр. Предъявленное имя сравнивается со списком допущенных имен и получает статус правомочного (или неправомочного). После идентификации имени следует проверка заявленного пароля и его принадлежности к обозначенному имени.

Пароль – секретная последовательность символов, которую получает или заявляет пользователь при регистрации в системе и предъявляет при каждом входе в информационный ресурс. При совпадении имени и пароля пользователь получает доступ к информационному ресурсу. Пароль используется для деления пользователей по категориям. В зависимости от категории пользователь может получить право на ознакомление с ресурсами, на просмотр каталогов или папок, на внесение изменений, на установку дополнительных программ и др. После окончания работы с ресурсом пользователь обязан завершить сеанс, чтобы предотвратить возможность использования его имени и пароля другими пользователями.

В процессе регистрации, кроме получения имени и пароля, каждый пользователь в бумажной или электронной форме заполняет дополнительные учетные данные, в которых раскрываются более подробные сведения о пользователе, в частности, место учебы или работы, принадлежность к группе или организации, номер телефона и т. п. Перечень учетных данных определяется правилами эксплуатации информационного ресурса, наличие учетных данных помогает при необходимости осуществлять непосредственное взаимодействие с пользователями.

В информационных системах, имеющих повышенную степень защиты, для подтверждения подлинности наряду с именем и паролем могут использоваться дополнительные средства защиты, в частности, банковские смарт-карты, системы биометрического контроля.

Банковские смарт-карты применяются для дополнительной защиты платежных систем от злоумышленников. В состав смарт-карты вводится микросхема (чип), которая обеспечивает дополнительную степень защиты. Для проверки подлинности используется сочетание пароля (кода доступа) с наличием смарт-карты, данные с которой считываются и проверяются банком при каждом обращении к системе.

Биометрические системы контроля используются для идентификации личности (биометрические паспорта) или для охраны помещений от проникновения злоумышленников (банки, исследовательские лаборатории, военные сооружения). Дополнительным параметром контроля в этом случае являются уникальные биометрические характеристики человека, которые невозможно потерять или изменить. К таким характеристикам относятся отпечатки пальцев, сетчатка

и радужная оболочка глаз, особенности строения лица, геометрия кистей рук. Контроль биометрических характеристик осуществляется с помощью специальных оптических или электронных устройств. Устройства контроля могут быть встроены в рабочие инструменты, в частности, в манипулятор мышь или в клавиатуру компьютера.

Компьютерные вирусы и методы защиты от вирусов

Угрозы безопасности чаще всего реализуют с помощью компьютерных вирусов.

Компьютерный вирус – это специально написанная вредоносная программа, которая способна самостоятельно распространяться, многократно копироваться, производить свои дубликаты и изменять компьютерные данные или программы, к которым имеется доступ. Дубликаты не всегда совпадают с оригиналом, копии и дубликаты вируса могут сохранять способность к дальнейшему распространению.

Основные каналы распространения вирусов – это электронная почта и сайты сети Интернет. Источником заражения могут быть также непроверенные флэш-носители. Основные объекты для внедрения вирусов – операционные системы или прикладные программы и файлы.

Вирусы могут выполнять различные деструктивные и несанкционированные действия, а именно:

- нарушать нормальное функционирование прикладных программ;
- воспроизводить себя неограниченное число раз, замедляя или останавливая работу компьютера;
- неожиданно менять экранные сообщения;
- собирать конфиденциальные данные;
- осуществлять вирусные атаки с целью блокировки серверов.

Все компьютерные вирусы являются результатом деятельности человека или группы злоумышленников, которых можно считать преступниками в интеллектуальной сфере деятельности. Общее число вирусов исчисляется десятками тысяч и характеризуется постоянным ростом числа разновидностей, несмотря на систему мер, предпринимаемых для обеспечения компьютерной безопасности.

Большое разнообразие вирусов классифицируется по различным признакам и категориям. К распространенным способам относятся следующие классификации.

По среде обитания вирусы делятся на:

- сетевые – распространяются по компьютерной сети путем внедрения в файлы, которые пересылаются электронной почтой;
- файловые – внедряются в исполняемые файлы, после запуска соответствующей программы размещаются в оперативной памяти компьютера и, как правило, сохраняют активность до перезагрузки операционной системы;
- загрузочные – внедряются в загрузочный сектор диска и активируются при запуске операционной системы.

По функциональным возможностям различают:

- безвредные вирусы, которые уменьшают свободную память на диске и на работу компьютера не оказывают существенного влияния;
- неопасные вирусы, кроме уменьшения памяти, могут производить графические, звуковые и прочие эффекты;
- опасные вирусы вызывают серьезные сбои, которые приводят к потере или блокировке файлов;
- очень опасные вирусы разрушают программное обеспечение, удаляют или декодируют файлы на дисках, перепрограммируют постоянную память, блокируют доступ к файлам или к компьютеру.

По особенностям алгоритма выделяют:

- вирусы-спутники, которые присоединяются и выполняют все свои действия при запуске прикладной программы;
- вирусы-черви (репликаторы) проникают по сети, внедряясь во вложения к электронным письмам, вычисляют адреса других компьютеров и рассылаются по этим адресам; искажают данные или полностью парализуют работу сети;
- вирусы-невидимки (стелс-вирусы) скрывают свое присутствие, перехватывают обращение операционной системы к пораженным файлам и подставляют вместо себя незараженные участки;
- вирусы-мутанты относятся к самомодифицирующимся вирусам, шифруют свой код, чтобы затруднить работу антивирусных программ;
- программы-боты автоматически выполняют действия по заданному алгоритму, имеют широкую область применения, в том числе в части вредоносного использования: сбор адресов электронной почты, загрузка и рассылка излишней рекламной информации, проведение вирусных атак.

Широко распространенным вирусам принято давать названия или собственные имена. Печальную известность получили группы вирусов под названием *логическая бомба* и *троянский конь*.

Логическая бомба представляет собой набор команд, который тайно встраивается в программу и проявляет свое действие при выполнении определенного логического условия, например при достижении обозначенной даты или реализации определенных информационных условий. Если в программе заложено срабатывание в заранее определенное время, то такую программу называют также логической бомбой с часовым механизмом. Код, который закладывается в логическую бомбу, приводит к неизвестным заранее последствиям для пользователя.

Одним из широко известных вирусов с часовым механизмом является вирус «Чернобыль», который 26 апреля 1999 г. в годовщину Чернобыльской аварии, активизировался и уничтожил файлы на жестких дисках инфицированных компьютеров. Вирус получил свое название в результате совпадения даты активации и даты аварии, этот же вирус известен под именем «СИН». По разным оценкам, от вируса «Чернобыль» пострадало примерно полмиллиона компьютеров по всему миру.

Троянский конь выражается в наборе команд, которые тайно вводятся в программу, маскируются под полезные фрагменты программы и осуществляют

незапланированные функции, сохраняя при этом прежнюю работоспособность программы. Программа троянский конь не предназначена для размножения, она скрытно выполняет вредоносные действия, о которых пользователь не всегда может догадаться. Результат действия троянского коня может выражаться в получении конфиденциальной информации, в частности:

- паролей для входа в информационную систему или в электронную почту;
- содержания файлов, которые хранятся в системе;
- сведений с кредитной карты.

Установка троянского коня может выполняться под видом загрузки полезной программы, которая скачивается из Интернета. В скачанной программе может оказаться небольшая программа-закладка, которая позже скачает из Интернета полный троянский файл.

Результатом работы троянского файла может быть передача доступа удаленному пользователю, который способен выполнять деструктивные действия:

- скачивать или удалять файлы;
- получать списки и пароли подключенных пользователей;
- отключать разделы меню и отдельные аппаратные средства;
- отслеживать порядок нажатия клавиш;
- перегружать компьютер и повреждать файлы операционной системы.

Задачи, выполняемые троянскими программами, могут быть весьма разнообразны. Один из троянских коней известен как семейство вредоносных программ Винлокер (вирус-вымогатель). Винлокер блокирует или затрудняет работу с операционной системой и требует перечисления денег злоумышленникам за восстановление работоспособности компьютера. Вредоносная программа впервые появилась в 2007 г. и за три года получила широкое распространение в тысячах версий. По разным данным, вирусом-вымогателем периодически заражаются миллионы компьютеров, преимущественная зона заражения находится в русскоязычном Интернете.

Для защиты данных от вирусов используются различные методы [23, 24]:

- профилактические меры, которые позволяют уменьшить вероятность заражения вирусом, например запреты на использование подозрительных файлов или предупреждения при открытии некоторых сайтов;
- общие средства защиты данных, например построение системы разграниченного доступа к информационным ресурсам или использование системы копирования данных;
- специализированные программы, которые широко применяются в компьютерах и являются наиболее эффективными средствами обнаружения и уничтожения вирусов.

В настоящее время существует множество специализированных программ, которые осуществляют антивирусную защиту. Большинство антивирусных программ предназначено для просмотра данных компьютера в целях обнаружения характерных признаков вируса (сигнатуры вируса). Антивирусные программы классифицируются по разным признакам.

По *принципу работы* выделяют:

- программы-мониторы, которые входят в состав операционной системы, осуществляют автоматический мониторинг и лечат все поступающие файлы; открытие файла программа разрешает только в том случае, если вирусы в файле не обнаружены; описание известных вирусов хранится в вирусной базе;

- программы-сканеры, которые размещаются в оперативной памяти, осуществляют поиск вирусных сигнатур путем сравнения с вирусной базой, выдают сообщения о подозрительных файлах и выполняют лечение;

- брандмауэры – программы, которые по заданному алгоритму фильтруют данные при обмене информацией между локальной и внешней сетью.

По *принципу обнаружения* вирусов используют:

- программы-полифаги, работа которых основана на проверке файлов оперативной памяти и поиске известных сигнатур вируса путем сравнения с имеющейся вирусной базой; вирусная база таких программ должна постоянно обновляться;

- программы-ревизоры ведут подсчет контрольных сумм файлов (сумма единиц двоичного кода) и их сравнение с результатами проверки сумм после возможной модификации данных; при изменении контрольных сумм ревизор выдает предупреждение о возможном заражении;

- программы-блокировщики следят за возникновением опасных ситуаций, таких, например, как появление новых записей в загрузочном секторе диска, и предупреждают пользователя о возможном заражении.

В нашей стране наряду с зарубежными программами уже не первое десятилетие используются отечественные антивирусные программные продукты.

Предприятие «ДиалогНаука» в стадии постоянной разработки поддерживает антивирусную программу «Доктор Веб» и является поставщиком многих антивирусных программных решений. Программа предназначена для работы с операционной системой Windows, набор излечиваемых вирусов ориентирован на отечественный рынок. Работа программы основана на сверке программных кодов, имеющихся на компьютере, с вирусной базой, содержащей коллекцию сигнатур. Имеющаяся коллекция сигнатур постоянно обновляется.

Компания «Лаборатория Касперского», которая специализируется на системах защиты от вирусов и от других кибернетических угроз, поддерживает продукт «Антивирус Касперского», распространяемый по всему миру среди частных пользователей и корпоративных клиентов. Антивирус Касперского является одной из мощных программ в мире, используется для обнаружения и удаления большинства вирусов, вирусная база обновляется несколько раз в месяц.

Все антивирусные программы могут работать только при условии постоянного обновления антивирусной базы, чтобы противодействовать атакам новых вирусов. Для снижения вероятности заражения вирусом рекомендуется использовать не только антивирусные программы, но и весь комплекс профилактических мер, что существенно снижает вероятность тяжелых последствий, но при этом не может обеспечить абсолютную защиту.

Противодействие несанкционированному доступу и спаму

Несанкционированный доступ к информации – это выход на компьютерные данные путем нарушения правил доступа с целью незаконного использования полученной информации. Несанкционированный доступ к информации относится к компьютерным преступлениям. Злоумышленники используют различные несанкционированные способы доступа к информации, в том числе:

- проникновение в сеть путем изменения уровня доступа с целью получения конфиденциальной информации;
- перехват данных в кабельных или проводных системах с помощью подслушивающих устройств или электромагнитных датчиков;
- хищение носителей данных или документальных отходов;
- копирование данных на носителях с преодолением мер защиты;
- маскировку под зарегистрированного пользователя с целью неуплаты за время нахождения в сети;
- внедрение компьютерных вирусов и др.

Все способы несанкционированного доступа за исключением вирусов поддаются блокировке путем подбора и использования *специальных мер* защиты информации.

К специальным мерам защиты от несанкционированного доступа относятся:

1. Оптимальное сочетание программных и аппаратных средств, а также организационных мер защиты.
2. Предоставление пользователям минимального уровня полномочий, обеспечивающих выполнение служебных обязанностей.
3. Идентификация каждого пользователя и протоколирование его действий.
4. Использование систем шифрования данных и цифровой подписи.
5. Обеспечение полного контроля над функционированием системы защиты.
6. Поддержание стоимости системы безопасности на уровне, не превышающем стоимость возможного экономического ущерба.

Для каждой компьютерной сети выбираются конкретные методы защиты от несанкционированного доступа, которые включаются в единую систему безопасности информационного ресурса. Надежным барьером от несанкционированного доступа является оптимальное соотношение технических, организационных и программных мер защиты.

Специфической разновидностью несанкционированного доступа является спам – множество сообщений, полученных в результате массовой несанкционированной рассылки рекламы по электронным почтовым ящикам.

Основным источником спама являются фирмы, рассылающие в автоматическом режиме электронные рекламные сообщения. Нежелательные последствия получения спама выражаются в захламлении электронного адреса множеством входящих сообщений, в потере времени на поиск важного сообщения, в необходимости периодической чистки электронного ящика от информационного мусора.

Способы противодействия спаму выражаются в ограничении на раздачу своих электронных реквизитов и ограничении подписки на рассылку электронной

информации. Одним из эффективных способов противодействия является установка программных фильтров, которые производят автоматическое выделение спама в отдельную папку или удаляют его при получении.

Криптографическая защита данных

Для надежной защиты от несанкционированного доступа можно использовать *шифрование данных* или, другими словами, *криптографическое кодирование* [25]. Зашифрованные данные представляют собой некоторый набор символов, который не имеет никакого смысла для злоумышленника, но позволяет обмениваться информацией между отправителем и получателем.

Шифрование – процесс кодирования сообщения с целью сокрытия его содержания от посторонних лиц.

Криптография – наука о методах кодирования данных, которые необходимо скрыть от всех посторонних лиц за исключением отправителя и получателя.

На практике для получения зашифрованных данных могут использоваться различные *алгоритмы шифрования*. Степень защиты информации при этом зависит от выбранного алгоритма и ключа шифрования.

Алгоритм шифрования – это порядок преобразования исходного сообщения в кодированное.

Ключ шифрования – это некоторый набор символов или параметр, который определяет правила преобразования данных; статические ключи остаются действительными для серии кодируемых сообщений, динамические – меняются для каждого сообщения.

В зависимости от выбранного типа ключа используют *симметричные* и *асимметричные* методы шифрования.

В симметричном шифровании *один криптографический ключ* используется для шифрования и дешифрования данных. Стороны, использующие симметричное шифрование, должны выбрать ключ до начала обмена сообщениями и в дальнейшем сохранять его в тайне. Симметричное шифрование удобно в том случае, когда зашифрованные данные хранятся без передачи. Передача данных, например по электронной почте, должна сопровождаться передачей ключа, что значительно снижает степень защиты. Основным недостатком симметричной системы заключается в необходимости обмена ключами и вытекающей отсюда невозможности некоторых применений, например, в формировании электронной цифровой подписи.

В асимметричном шифровании используются специальные *математические методы*, которые позволяют создать пару различных, но математически связанных ключей. Один из ключей называют *открытым*, получить открытый ключ довольно просто, поскольку он не является секретным. Если с помощью открытого ключа можно распаковать файл, то это означает, что отправителем файла является только владелец другого – *закрытого* ключа. Второй ключ называют закрытым, или секретным. Закрытый ключ остается у разработчика ключа и сохраняется в тайне. Только с помощью закрытого ключа можно открыть файл, зашифрованный открытым ключом.

Связь между ключами построена на математической основе таким образом, что невозможно вычислить один ключ на основе преобразования другого ключа. Для математической поддержки асимметричного шифрования используют криптографические *хэш-функции*.

Хэш-функция – это специфическая формула вида $f(x)$, которая в результате математической обработки позволяет заменить одну последовательность чисел или символов на другую последовательность символов фиксированной длины, которая однозначно соответствует содержанию исходной последовательности. При этом обратное преобразование, т. е. получение исходной последовательности с помощью этой же хэш-функции, невозможно.

Процесс обработки данных с помощью хэш-функции называется *хэшированием*.

Хэширование – преобразование входных данных произвольной длины в выходную строку установленной длины с помощью определенного алгоритма, который называется хэш-функцией, или функцией свертки. Результат хэширования представляется в виде *хэш-кода* – компактного набора случайных чисел, который однозначно повторяется при повторном хэшировании.

В криптографии хэширование используется, в частности, для выработки электронной подписи документа. Процесс шифрования и дешифрования состоит при этом из следующих этапов (рис. 40).

1. Документ, который требуется передать по сети, дополняют сведениями об авторе.

2. Сведения об авторе обрабатывают хэш-функцией и получают *подпись* в зашифрованном виде, которую называют *электронной цифровой подписью* (ЭЦП).

3. Передаваемый документ объединяют с ЭЦП в один файл, шифруют файл с использованием асимметричного метода и пересылают его принимающей стороне.

4. Принимающая сторона дешифрует полученный файл с помощью своего открытого ключа. Выделяет из файла ЭЦП и обрабатывает сведения об авторе той же хэш-функцией, которая использовалась при передаче.

5. Если полученный в результате обработки хэш-код совпадает с пришедшим в сообщении кодом, то подлинность подписи подтверждена.

Из перечисленных операций по шифрованию и дешифрованию следует определение ЭЦП.

Электронная цифровая подпись – реквизит электронного документа, полученный в результате криптографической обработки данных с использованием закрытого ключа подписи, который позволяет:

- проверить отсутствие искажения данных в документе с момента формирования подписи;
- проверить принадлежность подписи владельцу ключа подписи;
- подтвердить факт подписания электронного документа.

Если обе стороны используют один и тот же метод шифрования, то принято считать, что между ними организован защищенный канал связи. В случае компьютерной сети это защищенный информационный канал, используемый для

передачи кодированных пакетов данных, которые невозможно скрыть, изменить или просмотреть в процессе передачи.

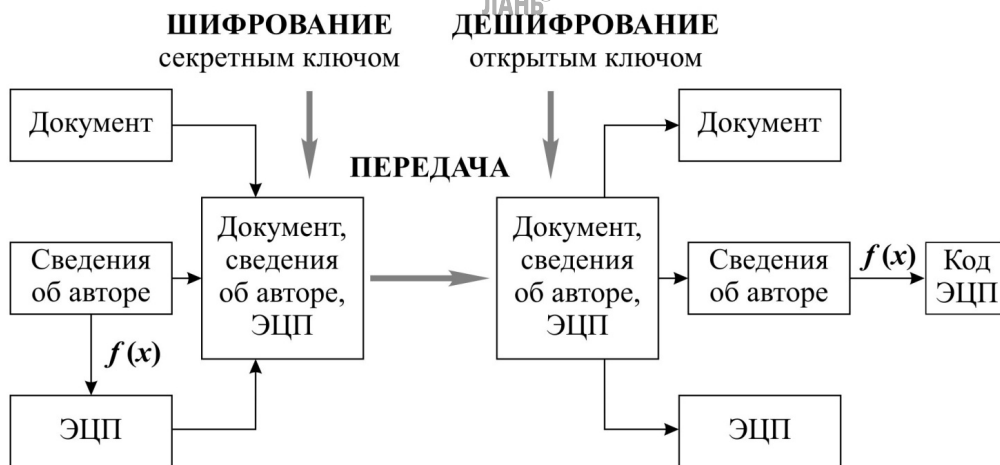


Рис. 40. Передача документа с электронной цифровой подписью

Общие меры обеспечения компьютерной безопасности

Предотвращение угроз компьютерной безопасности основано на создании базовой системы защиты информации с широким кругом мер, которые подразделяются на *правовые, административные, технические и программные* меры.

К правовым мерам относятся:

- разработка законодательных норм, устанавливающих ответственность за совершение компьютерных преступлений;
- совершенствование уголовного и гражданского законодательства в области информационных технологий.

На законодательном уровне правовые меры поддерживаются Федеральными законами «О связи», «Об информации, информатизации и защите информации», «О государственной тайне» и др. В Уголовном кодексе РФ, в котором в гл. 28 «Преступления в сфере компьютерной информации» определены составы преступлений в области компьютерной деятельности, содержатся три статьи:

статья 272 – Неправомерный доступ к компьютерной информации;

статья 273 – Создание, использование и распространение вредоносных программ для ЭВМ;

статья 274 – Нарушение правил эксплуатации ЭВМ.

К административным или организационным мерам относятся действия общего характера, предпринимаемые руководством организации в соответствии с реализуемой политикой безопасности. К мерам административной безопасности, в частности, относятся:

- выработка плана мероприятий по мерам безопасности локальной сети и каналов связи;

-
- утверждение правил обработки информации в автоматизированной информационной системе;
 - сертификация используемых технических и программных средств;
 - отладка системы подбора и подготовки персонала, организация системы допуска персонала;
 - организация охраны компьютерных систем и надежного пропускного режима;
 - организация системы контроля над работой пользователей и персонала;
 - организация системы хранения и использования носителей с конфиденциальной информацией;
 - исключение случаев ведения дела одним специалистом;
 - формирование правил учета и разграничения доступа для пользователей;
 - оформление документов, регулирующих ответственность лиц по обеспечению безопасности и т. п.

К техническим мерам обеспечения безопасности относятся профилактические мероприятия, предотвращающие потерю аппаратных средств и хранимых данных:

- резервирование компьютерных систем и резервное копирование данных;
- обеспечение компьютерных систем резервным электропитанием;
- создание препятствий на путях проникновения в помещения (кодовые замки, сигнализация, турникеты);
- организация охранных видеосистем и утверждения порядка их функционирования.

К программным мерам относятся средства, обеспечивающие информационную безопасность на уровне программного обеспечения. К числу программных мер относятся:

- идентификация пользователей путем установки учетных записей (защита паролем);
- использование систем управления доступом (разделение пользователей на уровни);
- постоянное использование и обновление антивирусных программ;
- использование программных фильтров, например для выделения спама;
- настройка правил безопасности при работе с браузером;
- введение систем шифрования и криптографии.

Базовая система защиты информации создается на основе использования правовых, административных, технических и программных мер. Выбор соотношения между перечисленными мерами с учетом целей функционирования сети и наличия возможных угроз лежит в основе формирования стратегии безопасности для конкретной компьютерной сети. Стратегия безопасности компьютерной сети, подключенной к сети Интернет, может быть выражена в жестком и разумном вариантах.

Жесткий вариант стратегии безопасности предполагает создание перечня разрешенных операций и запрет всех действий, которые не числятся разрешенными.

Этот вариант означает, что можно выходить только на те сайты, которые необходимы для выполнения служебных обязанностей и числятся разрешенными, остальные сайты закрыты для посещения. Аналогично с электронной почтой – личная переписка недопустима, служебную переписку можно вести только с адресами, которые разрешены в организации. Вся ценная информация в этом случае размещается на компьютерах, которые не могут иметь выход в Интернет.

Разумный вариант стратегии безопасности строится на основе запрета отдельных действий, при этом все остальные действия не оговариваются и считаются допустимыми. В части работы в Интернете это выражается в запрещении выхода на сайты, которые могут представлять угрозу безопасности организации. При посещении остальных сайтов должны соблюдаться общие меры предосторожности, предусмотренные установленными в организации правилами.

Выбор содержания стратегии безопасности зависит от рода деятельности организации, формы собственности, стоимости системы безопасности и других факторов.



Интернет и информационное общество

Возникновение и стремительное распространение сети Интернет послужило мощным катализатором развития цивилизации и формирования нового жизненного уклада, связанного с переходом в информационную фазу развития. Этот переход характеризуется крупными преобразованиями технологической и социальной основы общества. В истории цивилизации выделяются три социально-технологические фазы:

- аграрное общество было целиком ориентировано на производство продуктов питания и поддержание своей жизнедеятельности;
- индустриальное общество целенаправленно занималось производством промышленных товаров и средств производства этих товаров;
- информационное общество занято производством информации, информационных продуктов и знаний, основанных на получении новой информации.

Информационное общество – новая историческая фаза развития цивилизации, в которой главными продуктами производства являются *информация* и *знания* [7].

Начало развитию информационного общества было положено в конце XX в., когда были созданы аппаратные и программные основы глобальной компьютерной сети. Появление глобальной сети сопровождалось созданием связанной с ней инфраструктуры, содержащей серверы-накопители информации; коммуникации, необходимые для передачи и получения данных; систему управления составляющими новой структуры. Новая инфраструктура, наполненная содержанием, внесла кардинальные изменения в социальную жизнь общества, в том числе:

- изменила отношение общества к понятию «информация»;
- обеспечила доступ к информационным ресурсам мирового уровня;
- создала условия для безграничного информационного обмена на уровне социальной, производственной и управленческой сферы;
- значительно увеличила роль и долю информации в сфере традиционных знаний общества;
- обеспечила удовлетворение в информационных продуктах и услугах;
- перестроила характер занятости с индустриальной тематики на информационную сферу и многое другое.

Интернет существенно отличается от традиционных средств получения информации (газеты, журналы, радио, телевидение), поскольку объединяет все возможные информационные ресурсы и организует общее информационное пространство. Создание информационного пространства подтверждается рядом факторов.

1. *Информационные ресурсы* Интернета включают различные источники информации. Необходимым условием подключения к общим информационным ресурсам является оцифровка данных, которая переводит прессу, радио, телевидение и другие источники в единое информационное пространство. Помещение

в единое информационное пространство означает всеобщую доступность к информации по любой сфере человеческой деятельности в любое время суток.

2. *Средства связи* Интернета функционируют на основе общих принципов, которые распространяются на протоколы взаимодействия и системы адресации. В конечном итоге это обеспечивает глобальную связь, которая включает качественную письменную, голосовую, видео- и другие виды связи, существенно превосходящие традиционные системы. Средства связи глобальной системы доступны пользователям, которые практически не ограничены в доступе к необходимой информации. Наличие систем поиска позволяет выделить в огромном потоке данных именно то, что необходимо пользователю.

3. *Информационная инфраструктура* настроена таким образом, что позволяет компаниям организовать оперативный обмен информацией в режиме реального времени независимо от места нахождения компании и ее филиалов. Вся поступающая из сети информация отличается высоким качеством при сравнительно низкой цене трафика.

4. *Централизованный пункт управления* в глобальной сети отсутствует. Порядок поиска и передачи информации определяется программными системами, которые управляют процессом в автоматическом режиме. Способы управления сетью ограничены на уровне решения технологических проблем, присвоения доменных имен, внедрения новых стандартов и т. п. В отсутствие централизованного пункта управления не используются территориальные ограничения, в глобальной сети не выделяются государственные границы. Пользователи, как правило, не знают место нахождения источника информации, которую они получают.

5. *Вмешательство государства* в функционирование глобальной системы весьма затруднено, поскольку публикуемая в сети информация далеко не всегда привязана к конкретному государству и теоретически может располагаться на сервере любой страны. Регулирование и ограничение информации в рамках отдельной страны возможно, работу в этом направлении каждое государство организует на основе собственного законодательства.

Одним их характерных признаков информационного общества является рост доли информационной составляющей в совокупном общественном продукте. Основная часть информационной составляющей представлена на рынке программными продуктами, которые либо создаются заново, либо совершенствуются в соответствии с запросами общества. В развитии программного обеспечения можно отметить некоторые характерные тенденции нашего времени:

- программное обеспечение является товаром, который постоянно совершенствуется и пользуется широким спросом на рынках;
- крупные информационные компании, занятые разработкой программного обеспечения, становятся лидерами мировой экономики;
- с развитием аппаратной части увеличивается мощность и наукоемкость программ, а также расширяются области их применения и функциональные возможности;
- наукоемкое программное обеспечение лежит в основе повышения производительности труда.

Переход информационного общества к наукоемким технологическим процессам кардинально меняет распределение ресурсов, необходимых для развития общества. На смену производственным системам и привязанным к ним трудовым ресурсам приходят *автоматизированные системы*, развитие которых определяется уровнем интеллекта и финансовыми вложениями. Информационные и финансовые ресурсы по мобильности и эффективности значительно превосходят производительные силы индустриального общества, поэтому экономика информационного общества развивается ускоренными темпами. Ускоренное развитие обеспечивается возможностью быстрого перемещения финансовых средств в развиваемые регионы, подключением интеллектуальных ресурсов к развитию новых производств, созданием новых технологий на информационной основе, привлечением информационных систем на стадиях производства, продвижения и сбыта товаров и услуг.

Сеть Интернет обеспечивает дополнительные эффективные каналы для обмена информацией в науке, образовании, коммерческой деятельности, административном управлении и др. Наличие дополнительных каналов существенно увеличивает объемы данных, число форматов, а также скорости получения и передачи информации.

Информация превращается в значимый ресурс общества, которой выступает наравне с другими стратегическими ресурсами, в частности, с энергетическими. Более того, энергетические ресурсы начинают активно использоваться для поддержки информационных структур, а мощные информационные центры, занятые сбором и обработкой больших объемов данных, становятся основными потребителями электроэнергии. Деятельность общества в этих условиях плавно переходит из сферы создания промышленных продуктов в новые рыночные отношения, основу которых составляет сбор, обработка и использование информации. Процесс перехода сопровождается накоплением нематериального актива, а основную ценность приобретает интеллектуальный товар, который лишь в малой степени зависит от сырья и в большей степени – от числа и квалификации специалистов, от степени их образованности, от умения работать с информационными ресурсами. Человеческий потенциал в этих условиях выступает *главным ресурсом общества*, который поддерживает и развивает информационные и финансовые ресурсы.

Если посмотреть на развитие Интернета в историческом аспекте и сравнить его с информационными революциями, которые в свое время внесли значительный вклад в информатизацию общества, то следует отметить некоторые закономерности этого процесса.

Информационные революции – это процесс преобразования общественных отношений по причине кардинальных изменений в сфере обработки информации.

До наступления XX в. в истории развития человечества отмечено три информационных революции. Первая из них связана с изобретением письменности 4–6 тыс. лет назад и последующим появлением рукописей, которые лежали в основе системы письменной передачи знаний. Вторая информационная революция

относится к середине XVI в., когда был изобретен процесс книгопечатания, который позволил внедрить систему копирования и массового обучения. Третья революция произошла в конце XIX в., когда были изобретены телеграф, телефон и радио, которые обеспечили возможность передачи информации на большие расстояния. На развитие письменности человечеству потребовался значительный исторический отрезок времени, продолжительность которого измеряется веками. Процесс освоения и внедрения книгопечатания уложился в один век. Электрические технологии передачи информации распространились по миру в течение десятилетий.

Можно отметить, таким образом, что продолжительность каждой последующей информационной революции и промежутков между двумя революциями со временем сокращаются. Интернет в этой зависимости не стал исключением. Промежуток времени, в течение которого Интернет получил интенсивное развитие во всем мире, относится к последнему десятилетию XX в. и первому десятилетию XXI в. При этом существенное влияние на развитие общества Интернет оказывал уже в первое десятилетие своего активного существования. Историческое сравнение показывает, что Интернет значительно опередил все предыдущие информационные революции по скорости внедрения и тем более по степени воздействия на развитие общества. По этой причине можно говорить о *лавинообразном росте* объемов и потоков информации, который создается и поддерживается Интернетом.

Если следовать общей закономерности развития информационных революций, то этап, связанный с развитием Интернета, следует отнести к четвертой революции, которая произошла в конце XX в. и связана с внедрением цифровых технологий и способов передачи цифровых данных. Однако значительное превосходство Интернета перед другими революциями по всем показателям позволяет назвать этот этап *информационным взрывом*.

Информационный взрыв характеризуется ускоряющимся и лавинообразным темпом накопления и обмена информацией. По данным, приведенным в [7], объем знаний, накопленный человечеством с 90-х гг. прошлого века, удваивается каждые пять лет, а к 2020 г. удвоение объема будет происходить каждые 10 недель.

По мере развития информационного общества человечество может подвергаться вполне реальным угрозам. В условиях лавинообразного роста объемов информации в информационном обществе может возникать стремление к замене *реальной жизни на виртуальную*. Интернет устроен таким образом, что в нем идет постоянное накопление информации о разных сторонах окружающей жизни. При этом накопленная и представленная в сети информация может существенно отличаться от реальной жизни, поскольку она трансформируется по мере накопления и преобразования. Сетевая информация отличается от реальной жизни в ту или иную сторону, поскольку может либо отражать, либо исказить реальные данные.

Человек, проживающий в условиях информационного общества и регулярно получающий информацию из Интернета, формирует свое мировоззрение на

основе полученной информации. Дозирование или фильтрация информации может приводить к определенному смещению мировоззрения в ту или иную сторону, другими словами, сетевая информация может использоваться для *манипуляции сознанием* человека.

Опасность манипуляции вполне реальна, особенно для молодого поколения, которое использует Интернет как основной источник знаний об окружающем мире. Низкокачественная информация, оказывающая неуправляемое воздействие, может серьезно влиять на молодых людей. Для отдельных личностей это влияние может выражаться в переносе сетевых игр в реальную жизнь.

Возможные негативные последствия информатизации общества не могут оказать существенного влияния на колоссальные возможности, которые открываются перед народами разных стран, принимающих участие в развитии информационного общества. Приоритетные направления развития информационного общества, сформулированные международным сообществом государств в качестве основы всестороннего развития цивилизации, включают:

1. Доступность в сети информации о национальных и международных проектах и исследованиях, связанных с развитием информационного общества.
2. Организацию международных связей и взаимодействие с помощью высокоскоростных сетей.
3. Межкультурное обучение и образование (инновационные подходы к изучению языков, особенно для студентов и представителей малого бизнеса).
4. Электронные библиотеки как распределенную коллекцию знаний человечества, доступную большинству членов общества через сети.
5. Электронные музеи и галереи – мультимедийные коллекции.
6. Электронные информационные ресурсы по окружающей среде и природным ресурсам.
7. Глобальную информационную сеть управления ситуациями, связанными с чрезвычайными мерами и рисками.
8. Глобальное применение информационно-коммуникационных, телеметрических технологий в здравоохранении и медицине.
9. Глобальный рынок для бизнеса – развитие среды открытого и свободного обмена информацией и предоставления глобальных торговых услуг.

В нашей стране развитие информационного общества осуществляется в рамках государственной программы «Информационное общество» (2011–2020 гг.) [26]. Основная цель государственной программы – получение гражданами и организациями преимуществ от применения информационных и телекоммуникационных технологий за счет обеспечения равного доступа к информационным ресурсам, развития цифрового контента, применения инновационных технологий, радикального повышения эффективности государственного управления при обеспечении безопасности в информационном обществе. Для реализации поставленной цели в обществе решаются задачи:

- повышение качества жизни граждан и улучшение условий развития бизнеса в информационном обществе;

-
- построение электронного правительства и повышение эффективности государственного управления;
 - развитие российского рынка информационных и телекоммуникационных технологий, обеспечение перехода к экономике, осуществляемой с помощью информационных технологий;
 - преодоление высокого уровня различия в использовании информационных технологий регионами, слоями общества и создание базовой инфраструктуры информационного общества;
 - обеспечение безопасности в информационном обществе;
 - развитие цифрового контента и сохранение культурного наследия.

Одним из основных показателей развития государственной программы является постепенный рост удельного веса сектора информационных и телекоммуникационных технологий в валовом внутреннем продукте Российской Федерации. Для численной оценки развития информационного общества используется комплекс показателей, который в числе прочих показателей включает:

- уровень образования и профессиональной подготовки населения;
- результаты научно-исследовательских и опытно-конструкторских работ;
- результаты инновационной деятельности;
- уровень развития информационно-коммуникационных технологий связи и систем доступа;
- удельный вес специалистов, занятых в информационной индустрии;
- долю организаций, использующих информационную защиту и др.

С мониторингом развития сектора информационных технологий и динамикой основных показателей можно ознакомиться на сайте Федеральной службы государственной статистики [27] в разделе «Информационное общество».





Информатика — основа цифровой экономики

В июле 2017 г. распоряжением Правительства Российской Федерации утверждена государственная программа «Цифровая экономика Российской Федерации» [28]. Программа ориентирована на развитие цифровой формы представления данных, которая должна выступать ключевым фактором производства во всех сферах социально-экономической деятельности, в государственном управлении и бизнесе. Основная цель Программы заключается в создании экосистемы цифровой экономики, которая обеспечивает:

- эффективное взаимодействие государства и граждан, развитие научно-образовательного сообщества, трансграничное взаимодействие бизнеса;
- повышение конкурентоспособности отраслей экономики, в том числе на глобальных рынках;
- создание условий для развития высокотехнологичного бизнеса.

Программа сфокусирована на формировании двух базовых направлений цифровой экономики:

1) ключевых институтов, обеспечивающих цифровую экономику в части нормативного регулирования, кадров и образования, формирования технологических заделов и исследовательских компетенций;

2) основных инфраструктурных элементов цифровой экономики, включая информационную инфраструктуру и информационную безопасность.

Одной из основных сфер деятельности, связанной с развитием базовых направлений, является система образования.

В Программе отмечено, что в системе образования постепенно расширяется применение цифровых технологий. Большинство образовательных организаций имеет выход в Интернет и представлено в глобальной сети на своих сайтах, которые поддерживаются в актуальном состоянии. В программах общего образования преподается курс информатики, который обеспечен нормативно, технологически и содержательно. В рамках среднего и высшего образования ведется подготовка кадров цифровой экономики. Однако численность подготовки кадров и соответствие образовательных программ нуждам цифровой экономики недостаточны. Имеется серьезный дефицит кадров в образовательном процессе всех уровней образования. В процедурах итоговой аттестации недостаточно применяются цифровые инструменты учебной деятельности, процесс не включен целостно в цифровую информационную среду [28].

Для устранения отмеченных недостатков определены базовые направления развития цифровой экономики, одним из основных направлений является развитие образования. В Программе сформированы основные цели в отношении кадров и образования:

- создание ключевых условий для подготовки кадров цифровой экономики;
- совершенствование системы образования, которая должна обеспечивать цифровую экономику компетентными кадрами;

– формирование рынка труда, который должен опираться на требования цифровой экономики;

– создание системы мотивации по освоению необходимых компетенций и участию кадров в развитии цифровой экономики России.

Запланированные цели обозначены в Программе в виде конкретных показателей 2024 г. и сроков их достижения:

– количество выпускников образовательных организаций высшего образования по направлениям подготовки, связанным с информационно-телекоммуникационными технологиями, – 120 тыс. человек в год;

– количество выпускников высшего и среднего профессионального образования, обладающих компетенциями в области информационных технологий на среднемировом уровне, – 800 тыс. человек в год;

– доля населения, обладающего цифровыми навыками, – 40%.

Обучение указанного числа специалистов неразрывно связано с преподаванием и получением знаний в области информатики. Дисциплина «Информатика» в рамках реализуемой Программы приобретает особое значение и выступает в качестве базовой системы знаний. Владение основами информатики является обязательным для выпускников высшего и среднего профессионального образования, которые планируют свою профессиональную деятельность в условиях развитой цифровой экономики.

Особое значение информатика имеет для специалистов технического профиля, которые наряду с этой дисциплиной получают знания в области технологических, точных и естественных наук и планируют профессиональную деятельность в качестве инженера. Специалисты инженерного профиля представляют собой базовый кадровый потенциал Программы, который ориентирован на развитие перспективных сквозных цифровых платформ и технологий. Большинство перспективных технологий, обозначенных в Программе, относятся к технической области деятельности и включают такие наукоемкие направления, как искусственный интеллект, производственные технологии и промышленный интернет, компоненты робототехники и другие.

В заключение необходимо отметить, что бурное развитие информационных технологий в различных областях современной жизни неминуемо приведет к тому, что каждый специалист, в какой бы сфере он ни трудился, будет вынужден в совершенстве овладеть навыками работы с информационными объектами подобно тому, как современный человек владеет навыками чтения и письма.



Литература

1. *Симонович, С. В.* Информатика. Базовый курс : учебник. – 3-е изд. – СПб. : Питер, 2018. – 640 с.
2. *Паламарчук, Л. Н.* Информатика и программирование. Ч. 1. Основы теории информации : учеб. пособие / Л. Н. Паламарчук, А. С. Волосников. – Челябинск : Издат. центр ЮУрГУ, 2016. – 64 с.
3. *Коноплева, И. А.* Информационные технологии : учеб. пособие / И. А. Коноплева, О. А. Хохлова, А. В. Денисов ; под ред. И. А. Коноплевой. – М. : Проспект, 2008. – 304 с.
4. *Голицина, О. Л.* Информационные системы и технологии : учеб. пособие / О. Л. Голицина, Н. В. Максимов, И. И. Попов. – М. : ИНФРА-М, 2016. – 400 с. – (Высшее образование).
5. Советский энциклопедический словарь / гл. ред. А. М. Прохоров. – 2-е изд. – М. : Сов. энциклопедия, 1982. – 1600 с.
6. *Акулов, О. А.* Информатика: базовый курс : учеб. пособие для студентов вуза, бакалавров, магистров, обучающихся по направлению «Информатика и вычислительная техника» / О. А. Акулов, Н. В. Медведев. – 5-е изд., испр. и доп. – М. : Омега-Л, 2008. – 574 с.
7. *Гаврилов, М. В.* Информатика и информационные технологии : учебник. – М. : Гардарики, 2007. – 655 с.
8. *Лозовский, В. Н.* Информация, информатика, реальность : учеб. пособие. – СПб. : Лань, 2008. – 96 с.
9. *Гарднер, М.* Шифр Бэкона // Научно-популярный физико-математический журнал «Квант». – 1992. – № 8. – С. 21–26.
10. *Бройдо, В. Л.* Архитектура ЭВМ и систем : учебник / В. Л. Бройдо, О. П. Ильина. – СПб. : Питер, 2006. – 718 с.
11. *Гуров, В. В.* Микропроцессорные системы : учебник [Электронный ресурс]. – М. : ИНФРА-М, 2016. – 336 с. – Режим доступа: <http://www.znanium.com>.
12. *Гагарина, Л. В.* Технология разработки программного обеспечения : учеб. пособие / Л. В. Гагарина, Е. В. Кокорева, Б. Д. Виснадул ; под ред. Л. Г. Гагариной. – М. : ИД «ФОРУМ» ; ИНФРА-М, 2012. – 400 с. – (Высшее образование).
13. *Строгалева, В. П.* Имитационное моделирование : учеб. пособие / В. П. Строгалева, И. О. Толкачева. – 3-е изд. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2017. – 295 с.
14. *Свердлов, С. З.* Языки программирования и методы трансляции : учеб. пособие. – СПб. : Питер, 2007. – 638 с.

16. *Хомоненко, А. Д.* Базы данных : учебник / А. Д. Хомоненко, В. П. Цыганков, М. Г. Мальцев ; под ред. А. Д. Хомоненко. – 6-е изд. – М. : Бинوم-Пресс ; СПб. : КОРОНА–Век, 2007. – 736 с.

17. *Михеева, В. Д.* MicrosoftAccess 2002 / В. Д. Михеева, И. А. Харитоновна. – СПб. : БХВ-Петербург, 2003. – 1040 с.

18. *Олифер, В. Г.* Компьютерные сети. Принципы, технологии, протоколы : учебник / В. Г. Олифер, Н. А. Олифер. – 3-е изд. – СПб. : Питер, 2008. – 958 с.

19. *Волокитин, А. В.* Интернет для служащих государственных организаций и коммерческих фирм : учеб. пособие / А. В. Волокитин, А. П. Маношкин, И. Н. Курносов [и др.] ; под общ. ред. Л. Д. Реймана. – М. : ФИОРД-ИНФО, 2001. – 272 с.

20. *Дунаев, В. В.* HTML, скрипты и стили. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2008. – 1024 с.

21. *Соколов, В. А.* HTML и CSS в примерах, типовых решениях и задачах. Профессиональная работа. – М. : Издат. дом «Вильямс», 2007. – 416 с.

22. *Мельников, В. П.* Информационная безопасность и защита информации : учеб. пособие / В. П. Мельников, С. А. Клейменов, А. М. Петраков ; под ред. С. А. Клейменова. – М. : Издат. центр «Академия», 2006. – 336 с.

23. *Малюк, А. А.* Защита информации в информационном обществе : учебное пособие. – М. : ГЛТ, 2015. – 230 с.

24. *Шаньгин, В. Ф.* Информационная безопасность и защита информации. – М. : ДМК, 2014. – 702 с.

25. *Аверченков, В. И.* Криптографические методы защиты информации : учебное пособие [Электронный ресурс] / В. И. Аверченков, М. Ю. Рытов, С. А. Шпичак. – М. : ФЛИНТА, 2017. – 215 с. – Режим доступа: <https://e.lanbook.com/book/92914>.

26. Государственная программа «Информационное общество» (2011–2020 гг.), утв. Постановлением Правительства РФ от 15.04.2014 № 313 [Электронный ресурс]. – Режим доступа: minsvyaz.ru/ru/activity/programs/1/.

27. Официальный сайт Федеральной службы государственной статистики РФ [Электронный ресурс]. – Режим доступа: <http://www.gks.ru>.

28. Программа «Цифровая экономика Российской Федерации», утв. распоряжением Правительства Российской Федерации от 28 июля 2017 г. № 1632-р.



Оглавление

Введение.....	3
Информатика – наука об автоматической обработке информации	4
Информация и ее представление в информационно-вычислительной технике ...	7
Определения и основные свойства информации.....	7
Единицы измерения и хранения данных	13
Преобразование и обработка данных.....	17
Представление числовых данных.....	20
Системы счисления.....	20
Перевод чисел из одной системы счисления в другую.....	22
Двоичное кодирование целых чисел.....	25
Двоичное кодирование действительных чисел.....	26
Представление и обработка текстовых символов.....	27
Способы кодирования текста.....	28
Кодирование компьютерных символов	30
Компьютерные шрифты	31
Электронные документы.....	33
Формирование графических данных.....	34
Классификация цифровых изображений.....	34
Моделирование цветовых оттенков, законы Грассмана.....	36
Цветовая модель RGB.....	36
Цветовая модель CMYK.....	38
Формирование цветных изображений на экране и бумаге.....	39
Операции с логическими данными.....	40
Основные понятия алгебры логики.....	41
Базовые логические операции	42
Логические элементы и схемы.....	45
Логические операции с множествами.....	48
История развития вычислительной техники	50
Вычислительные средства домеханического и механического этапов.....	50
Предпосылки создания электронных счетных устройств	52

Поколения вычислительных машин	53
Принципы устройства и структура ЭВМ.....	55
Архитектура вычислительных машин	56
Основные характеристики вычислительных машин	58
Персональный компьютер.....	60
Базовая конфигурация	60
Внутренняя и внешняя память компьютера	62
Мониторы.....	65
Клавиатура и манипулятор мышью	66
Классификация компьютеров	67
Программное обеспечение компьютера	70
Системное программное обеспечение	70
Операционные системы.....	71
Операционные системы Windows и Linux.....	74
Системы управления файлами.....	75
Служебные программы и стандартные приложения ОС	77
Системы и технологии программирования	79
История развития технологий программирования	79
Этапы решения задач на программной основе	81
Моделирование как процесс упрощения задачи.....	82
Алгоритм и его основные свойства.....	85
Представление алгоритма на языке программирования	90
Разновидности языков программирования.....	93
Прикладное программное обеспечение	96
Редакторы текстов.....	98
Создание и вставка объектов в текстовом редакторе.....	99
Электронные таблицы	102
Обработка данных в электронных таблицах	105
Средства анализа данных	109
Базы данных и системы управления базами данных.....	113
Классификация и структура баз данных.....	114

Системы управления базами данных	118
Системы автоматизированного проектирования	119
Компьютерные сети	124
Аппаратное и программное обеспечение сетей	124
Порядок передачи данных в сетях	127
Классификация и топология компьютерных сетей	128
Модель компьютерной сети	129
Глобальная компьютерная сеть Интернет	131
История Всемирной паутины	131
Протоколы и система адресации	132
Службы Интернета	136
Публикация web-документов	138
Принципы построения сайтов в сети Интернет	140
Способы соединения с глобальной сетью	143
Защита компьютерных данных	144
Системы управления доступом	144
Компьютерные вирусы и методы защиты от вирусов	146
Противодействие несанкционированному доступу и спаму	150
Криптографическая защита данных	151
Общие меры обеспечения компьютерной безопасности	153
Интернет и информационное общество	156
Информатика – основа цифровой экономики	162
Литература	164