Н. Ю. Иванова, В. Г. Маняхина

Системное и прикладное программное обеспечение

Учебное пособие

Московский педагогический государственный университет

Издательство Прометей

Москва
2011

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное
образовательное учреждение
высшего профессионального образования
«Московский педагогический государственный университет»

Н. Ю. Иванова, В. Г. Маняхина

СИСТЕМНОЕ И ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Учебное пособие

МПГУ

ИЗДАТЕЛЬСТВО Прометей

Москва – 2011

УДК 004.45
ББК 32.973.2-018.2
И21

И21 Иванова Н. Ю., Маняхина В. Г. Системное и прикладное программное обеспечение: Учебное пособие. – М.: МПГУ, 2011. – 202 с.

Учебное пособие составлено в соответствии с требованиями ФГОС ВПО и предназначено для подготовки бакалавров по направлению 050100 «Педагогическое образование» по профилям «Информатика», «Информатика и математика», также может быть использовано для студентов, обучающихся по другим профилям педагогического образования.

Учебное пособие содержит теоретический и практический материал по следующим темам: аппаратное и программное обеспечение компьютера, системное программное обеспечение – операционные системы, обзор ОС Linux, сервисные программы, антивирусные программы; прикладное программное обеспечение – текстовые процессоры (на примере OOo Writer, LyX), электронные таблицы (на примере OOo Calc), системы управления базами данных (на примере OOo Base), системы компьютерной математики (основы работы в Maxima), графические редакторы (основные приемы работы в Gimp). Теоретический материал снабжен вопросами для самопроверки, практикум по программному обеспечению содержит большое количество примеров и заданий.

ISBN 978-5-4263-0078-1

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Учебное пособие составлено в соответствии с требованиями Федерального государственного образовательного стандарта высшего профессионального образования (ФГОС ВПО) и предназначено для подготовки бакалавров по направлению 050100 «Педагогическое образование» по профилям «Информатика», «Информатика и математика», также может быть использовано и для студентов, обучающихся по другим профилям педагогического образования.

Данное учебное пособие систематизирует знания о программном обеспечении на основе современных принципов его построения и использования.

В главе «Аппаратное и программное обеспечение компьютера» освещаются вопросы, связанные с устройством компьютера, дается классификация современного программного обеспечения.

Вторая глава «Системное программное обеспечение» содержит теоретический и практический материал по системному программному обеспечению. Раскрывается значение операционной системы (ОС) как средства распределения и управления ресурсами компьютера, ее основные функции. Дается классификация ОС. Подробно рассматривается операционная система Linux, основные характеристики, архитектура, пользовательский интерфейс, файловая система. Практикум ориентирован на работу с командным интерпретатором системы и создание командных файлов в оболочке shell. В этой же главе рассказывается о вспомогательных системных программах, производящих диагностику, тестирование и обслуживание компьютера. Рассматриваются вопросы безопасности и защиты от вирусов, а также средства и методы защиты информации от несанкционированного доступа.

Третья глава «Прикладное программное обеспечение» содержит как теоретические, так и практические аспекты, связанные с работой пользователя с различными приложениями, решающими те или иные прикладные задачи. Это, прежде всего, текстовый процессор, электронные таблицы, системы управления базами данных (на примере программ, входящих в пакет Open Office). С необходимостью проведения сложных математических расчетов сталкиваются студенты тех специальностей и направлений, для которых создан этот учебник. Поэтому большое внимание уделяется системам компьютерной математики, рассматривается пакет Maxima. Так же освещается вопрос создания научных текстов с математическими формулами в формате TeX с использованием редактора LyX.

Не забыта и компьютерная графика. Показаны основные приемы работы в растровом редакторе GIMP.

Учебное пособие ориентировано, главным образом, на изучение свободного программного обеспечения (ПО), так как в последнее время увеличивается количество вузов, перешедших на его использование. Предложенное для изучения прикладное ПО является большей частью кроссплатформенным, то есть его установка возможна как под Linux, так и под Windows. Использование этих приложений позволит студентам не зависеть от установленной на компьютере операционной системы.

Вопросы для самопроверки, большое количество примеров и упражнений позволит овладеть компетенциями в области программного обеспечения и компьютерной обработки информации.

ГЛАВА 1.
АППАРАТНОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

1.1. Основы архитектуры ЭВМ

Архитектура ЭВМ – это наиболее общие принципы построения ЭВМ, реализующие программное управление работой и взаимодействие основных ее функциональных узлов.

1.1.1. Принципы Джона фон Неймана

В 1946 г. ученые из США Д. Нейман, Г. Голдстайн и А. Бернс сформулировали основные принципы построения универсальных ЭВМ. Принципиальное описание устройства и работы компьютера принято называть архитектурой ЭВМ. Эти принципы получили название «*принципов Джона фон Неймана*» или «*архитектуры фон Неймана*».

В основе архитектуры многих современных компьютеров лежат именно эти принципы.

Согласно фон Нейману, ЭВМ должна состоять из следующих основных блоков (рис. 1.1): устройства управления (УУ) и арифметико-логического устройства (АЛУ) (в современных компьютерах эти устройства объединены в один блок – процессор), запоминающих устройств (внутренней и внешней памяти), устройств ввода и вывода.

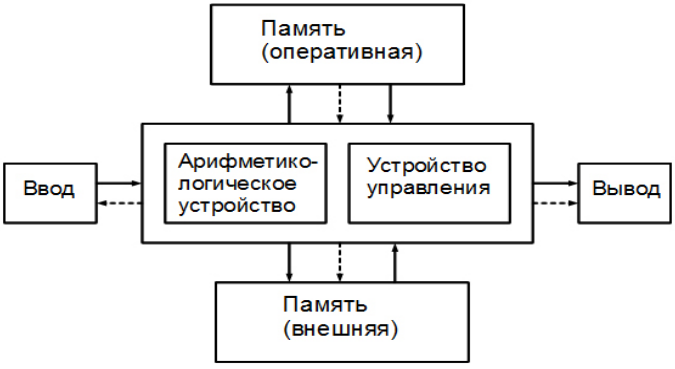


Рис. 1.1. Архитектура ЭВМ по Нейману.
Сплошными линиями показаны потоки информации,
пунктирными – управляющие сигналы (команды)

Принципы построения универсальных ЭВМ (Джона фон Неймана):

- принцип двоичного кодирования;
- принцип программного управления;
- принцип однородности памяти;
- принцип адресности.

Рассмотрим перечисленные принципы более подробно.

1. *Принцип двоичного кодирования* – вся информация, поступающая в ЭВМ, кодируется с помощью двоичных кодов.

Надо сказать, что в первых ЭВМ для представления информации не всегда использовался двоичный код. Например, в ENIAC был более привычный для человека десятичный код. Почему же был выбран именно двоичный способ кодирования? Ответ на вопрос довольно прост: два различных состояния, представляющих соответственно 0 или 1, технически реализовать значительно проще, чем все остальные случаи. Действительно, отсутствие напряжения можно закодировать как 0, наличие – 1; отсутствие намагниченности участка носителя информации – 0, намагниченность – 1 и т. д.

2. *Принцип программного управления* заключается в том, что работой ЭВМ управляет ***программа,*** представляющая собой последовательность команд (инструкций), которые выполняет процессор. ***Машинная команда*** – это двоичный код, определяющий выполняемую операцию, адреса используемых операндов (обрабатываемых данных) и адрес ячейки запоминающего устройства, по которому должен быть записан результат выполненной операции.

3. *Принцип однородности памяти* – программы и обрабатываемые данные хранятся вместе. ЭВМ не различает, что хранится в определенной ячейке памяти – команды или данные. Над командами можно выполнять такие же действия, что и над данными. Это позволяет получать команды одной программы в результате исполнения другой. Таким образом, ЭВМ может формировать для себя программу в соответствии с полученными результатами.

4. *Принцип адресности*: структурно основная память состоит из перенумерованных ячеек. Процессору в произвольный момент времени доступна любая ячейка.

Отсюда следует возможность давать имена областям памяти, так, чтобы к запомненным в них значениям можно было бы впоследствии обращаться или менять их в процессе выполнения программы с использованием присвоенных имен.

Выполнение программы происходит следующим образом.

Программа считывается (загружается) в оперативную память компьютера. Процессор последовательно считывает команды из оперативной памяти. Выборка команды осуществляется счетчиком команд – регистром устройства управления. Так как в памяти команды расположены последовательно друг за другом, то адрес очередной команды счетчик команд получает путем увеличения хранимого в нем адреса очередной команды на длину команды.

Несмотря на то, что команды выполняются последовательно, в программах можно реализовать возможность перехода к любому участку кода, так как существуют команды условного или безусловного перехода, которые позволяют занести в счетчик команд адрес внеочередной команды, и тем самым перейти не к следующей команде, а к той, адрес которой указывается.

Считанная в процессор команда расшифровывается, извлекаются необходимые данные и над ними выполняются требуемые действия.

Последовательное выполнение команд процессором может быть нарушено при поступлении сигнала прерывания.

Прерывание может быть вызвано внешним устройством (внешние прерывания) и самим процессором (внутрипроцессорные прерывания). Прерывание может быть и фатальным (неисправимым), например деление на 0, переполнение разрядной сетки, ведущие к прекращению выполнения программы.

После получения прерывания (не фатального) процессор запоминает текущее состояние прерванной программы, вызывает и выполняет специальную программу – обработчик прерываний, затем возвращается к исходной программе.

На основе принципов Джона фон Неймана производились первые два поколения ЭВМ. В более поздних поколениях происходили некоторые изменения, хотя принципы Неймана актуальны и сегодня.

1.1.2. Аппаратное обеспечение компьютера

Аппаратное обеспечение (hardware) – совокупность технических устройств, входящих в состав ЭВМ.

Рассмотрим аппаратное обеспечение современных компьютеров.

Современные компьютеры предназначены для работы с разными видами информации (числовой, текстовой, графической, звуковой и др.). Действия, выполняемые с информацией, называются информационными процессами. Выделяют несколько видов информационных процессов: получение

(ввод), хранение, обработка и передача (вывод) информации. За выполнение каждого действия отвечает одно или несколько устройств компьютера. Поэтому, как мы рассмотрели выше, ЭВМ имеет следующую структуру: устройство обработки информации – процессор (в современных компьютерах микропроцессор), устройства хранения информации – различные устройства внутренней и внешней памяти и устройства ввода и вывода информации.

Микропроцессор (МП), или ***центральный процессор*** (central processing unit – *CPU*), – программно-управляемое электронное устройство (интегральная микросхема), предназначенное для обработки информации. Нередко можно услышать, что микропроцессор сравним с «мозгом» машины.

В современных компьютерах процессоры выполнены в виде компактного модуля размерами около 5×5×0,3 см, вставленного в специальный разъем на материнской плате (рис. 1.2). Большая часть современных процессоров реализована в виде одного полупроводникового кристалла, содержащего миллионы и даже миллиарды транзисторов. Последние модели четырехъядерных процессоров Intel содержат 820 млн транзисторов. Для того чтобы уместить такое огромное количество элементов на площади, равной нескольким квадратным сантиметрам, нужно уменьшить их до микроскопических размеров. Размер транзистора передовых современных процессорах составляет всего 32 нм (для сравнения: толщина человеческого волоса равна 10 000 нм). К 2020 г. планируется, что производство процессоров перейдет уже на 6-нанометровую технологию.

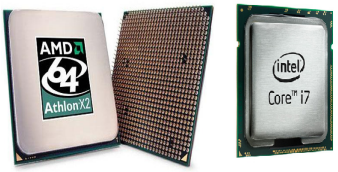


Рис. 1.2. Микропроцессоры

Микропроцессор состоит из следующих основных устройств:

- устройство управления (УУ), которое управляет работой всех компонентов компьютера при помощи команд;
- арифметико-логического устройство (АЛУ), которое выполняет арифметические и логические операции и таким образом производит обработку информации;
- регистры, которые используются для временного хранения информации в виде двоичного кода. Это счетчик ко-

манд (или регистр адреса команды), регистр кода команды, регистры обрабатываемых данных и др.).

Чтобы процессор смог обработать данные, любая информация (текст, графика, числа, звук и т. д.) должна быть представлена в виде двоичного кода. Поскольку информация представлена в виде двоичного кода, то процесс ее обработки осуществляется с использованием арифметических и логических операций, которые и производятся в арифметико-логическом устройстве процессора.

Процессор является одним из основных устройств компьютера. Характеристики процессора во многом определяют работоспособность всего компьютера.

Основные характеристики процессора:

- *тактовая частота* – частота синхронизирующих работу ЭВМ («тактовых») импульсов, задаваемых генератором тактовой частоты, которые регулируют выполнение циклов выборки и исполнения команд. Чем выше тактовая частота, тем выше быстродействие ЭВМ. Однако системы с одной и той же тактовой частотой могут иметь разную производительность (количество элементарных операций, выполняемых в одну секунду), так как на выполнение одной операции разным системам может требоваться разное количество тактов (от долей такта до десятков тактов). Измеряется тактовая частота в герцах (Гц). Современные процессоры имеют тактовую частоту порядка нескольких гигагерц (ГГц);
- *количество встроенных ядер (core)* – многоядерным процессором называется центральный процессор, содержащий два и более вычислительных ядра на одном процессорном кристалле или в одном корпусе. Это значит, что в одной микросхеме, по сути, находятся сразу несколько процессоров. В дальнейшем производительность процессора планируется увеличивать не за счет увеличения тактовой частоты, а путем увеличения количества встроенных в процессор ядер;
- *разрядность* – число одновременно обрабатываемых процессором битов (разрядность внутренних регистров). Процессор может быть 8-, 16-, 32- и 64-разрядным. Современные процессоры в основном имеют разрядность 64 бит. Но в процессор входят и другие важные устройства, основной характеристикой которых является разрядность: шина ввода и вывода данных, шина адреса памяти. Чем больше сигналов одновременно поступает на шину, тем больше данных передается по ней за определенный интервал времени и тем быстрее она работает. Количество разрядов – число переданных сигналов (двоичных кодов). Разрядность шин может отличаться

от разрядности регистров. Между шиной адреса и шиной данных есть эмпирическое соотношение: чем больше процессор должен адресовать памяти (то есть чем больше разрядность шины адреса), тем быстрее они должны поступать в процессор. Следовательно, тем шире шина данных. Разрядность этих шин является показателем возможностей процессора: количество разрядов в шине данных определяет способность процессора обмениваться информацией, а разрядность шины адреса – объем памяти, с которым он может работать. Также для определения производительности компьютера важна не только разрядность внутренних шин процессора, но и его интерфейс с системной шиной.

Рассмотрим теперь ***устройства хранения информации***. Память компьютера подразделяется на внутреннюю и внешнюю память.

В свое время это было связано с размещением устройства хранения информации внутри или вне процессорного шкафа. Однако с уменьшением размеров машин внешние запоминающие устройства удалось расположить внутри процессорного (системного) блока, и первоначальный смысл данного деления постепенно утратился, но терминология сохранилась.

Внутренняя память компьютера подразделяется оперативную, кэш-память, постоянную и полупостоянную.

Оперативная память – ОЗУ (оперативное запоминающее устройство), или RAM (random access memory – память с произвольным доступом). Это основная, рабочая память, она хранит данные и программы для решаемых в текущий момент задач. Все данные, которые обрабатываются процессором, считываются из оперативной памяти. Поэтому программа, с которой хочет работать пользователь, предварительно проходит процесс загрузки в оперативную память с внешнего носителя информации. Кроме того, это быстрая память. Однако она энергозависима, поэтому после выключения компьютера информация в оперативной памяти стирается.

Модули оперативной памяти устанавливается в специальные разъемы (слоты) на материнской плате. Современным стандартом для модулей оперативной памяти является третье поколение стандарта Double Data Rate, или DDR3 (рис. 1.3).

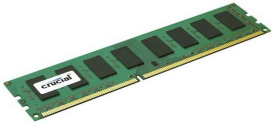


Рис. 1.3. Модуль оперативной памяти (SDRAM DDR3)

Кэш-память – это высокоскоростная память произвольного доступа, используемая процессором для временного хранения информации. Скорость обработки данных процессором намного больше, чем скорость обмена данными между процессором и оперативной памятью. Для того чтобы компенсировать эту разницу, массивы данных из оперативной памяти считываются в кэш-память и процессор непосредственно работает с быстродействующей кэш-памятью. Кэш-памятью управляет специальный контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память. Это энергозависимая память. Существует два вида кэш-памяти: внутренняя, размещаемая внутри процессора, и внешняя, устанавливаемая на системной плате.

Постоянная память – ПЗУ (постоянное запоминающее устройство), или ROM (read only memory – память только для чтения). Это энергонезависимая память. Поэтому в ней хранятся некоторые системные программы, в том числе программы, необходимые для первоначальной загрузки компьютера в момент включения питания (тестирование оборудования, управление работой процессора, инициирование загрузки операционной системы и др.). Большая часть программ связана с базовой системой ввода-вывода (BIOS). Это критически важная для компьютера информация, которая не зависит от выбора операционной системы. Из этой памяти можно только считывать информацию. Раньше содержимое ПЗУ формировалось на заводе, и невозможно было внести какие-либо изменения. Теперь современные технологии позволяют в случае необходимости обновлять содержимое ПЗУ – делать «перепрошивку» (рис. 1.4).

Рис. 1.4. Постоянное запоминающее устройство

Полупостоянная память – CMOS RAM, одна из разновидностей ПЗУ – микросхема памяти, на которой хранятся данные о запуске системы, которые используются BIOS во время запуска компьютера (рис. 1.5). Это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы. Содержимое CMOS изменяется специальной программой Setup, находящейся в BIOS (to set up – устанавливать).

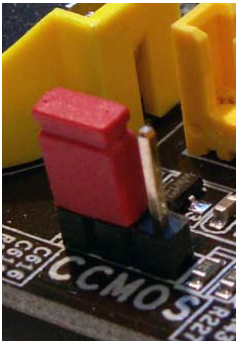


Рис. 1.5. Микросхема CMOS

Внешняя память компьютера – память, предназначенная для длительного хранения программ и данных.

Поскольку оперативная память энергозависима, а ПЗУ сильно ограничена по объему, для долговременного хранения программ и данных пользователя необходима энергонезависимая память, которая реализуется, в основном, при помощи магнитных и оптических накопителей, а также флеш-накопителей. Процессор не работает напрямую с этой памятью из-за низкой скорости обмена информацией.

Рассмотрим характеристики носителей внешней памяти:

- ***магнитные диски*** (носители на магнитных дисках – НМД):

гибкие (дискеты) – гибкие магнитные диски, покрытые ферромагнитным слоем, заключенные в пластиковый кожух. Имеют небольшой объем (1,44 МБайт), сейчас практически не используются.

жесткие (HDD – hard (magnetic) disk drive), или винчестеры, представляют собой несколько алюминиевых дисков с ферромагнитным покрытием, заключенных в единый корпус. Над поверхностью вращающегося диска скользит магнитная головка, которая считывает или записывает информацию на диск.

Окружность на магнитной пластине, которую описывает головка при вращении пластин, называется дорожкой, а совокупность таких дорожек, расположенных одна под другой (для каждого фиксированного положения головок), называется цилиндром. Каждая дорожка разбита на одинаковые блоки – сектора размером по 512 байт. На внешних дорожках секторов больше, чем на внутренних.

Таким образом, диски характеризуются совокупностью трех чисел: числом цилиндров / числом дорожек в цилиндре / числом секторов на дорожке или C/H/S (от английского Cylinder/Head/Sector, то есть цилиндр/головка/сектор). Эти три числа называют «геометрией диска». Диск с геометрией C/H/S имеет объем C*H*S*512 байт (рис. 1.6).

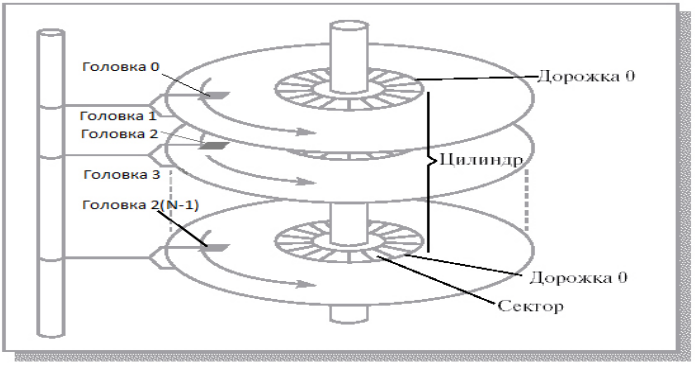


Рис. 1.6. Геометрия жесткого диска

Емкость современных жестких магнитных дисков исчисляется гигабайтами (1 Гбайт = 10^9 байт) и терабайтами (1 Тбайт = 10^{12} байт).

- ***магнитные ленты*** (носители на магнитных лентах – НМЛ), используемые до распространения жестких дисков. Неудобство заключалось в том, что доступ к данным был только последовательным, то есть чтобы считать информацию, записанную в конце ленты, нужно потратить время и промотать ее с начала до конца.
- ***оптические носители*** – диски из прочного пластика, требующие использования лазерного луча для выполнения операций чтения и записи информации.

CD – Compact Disc, объем – 650 или 700 Мбайт;

DVD Digital Versatile Disc (цифровой многоцелевой диск), объем – 4,7 Гбайт (однослойный), 8,5 Гбайт (двухслойный);

Blu Ray Disc, BD (blue ray – синий луч), объем – 23,3/25/27/33 Гбайт (однослойный), 46,6/50/54/66 Гбайт (двухслойный).

На оптических дисках указывается информация о возможности записи данных на диск:

ROM – только для чтения, то есть на этот диск невозможно записать данные;

R – однократная запись, то есть при помощи пишущего оптического привода можно один раз записать данные, потом только считывать их;

RW (RE) – многократная запись, при помощи пишущего оптического привода можно многократно записывать данные на диск.

- ***флеш-память*** – энергонезависимое полупроводниковое запоминающее устройство, выпускаемое в виде плат и карт памяти (memory stick, memory drive и др.). Данные во флеш-памяти перезаписываются целыми блоками. Благодаря своей компактности, дешевизне и низкому энергопотреблению флеш-память широко используется в цифровых портативных устройствах. Объем – несколько гигабайт.

Устройства ввода, вывода – средства связи ЭВМ с внешним миром. Принято также называть эти устройства периферийными устройствами. Их многообразие определяется разнообразием видов информации, с которой работает человек (текстовая, звуковая, графическая, видео и т. д.).

Клавиатура – основное устройство ввода информации. В современных компьютерах с графическим интерфейсом необходимы манипуляторы, помогающие пользователю управлять указателем и таким образом работать с графическими объектами на экране, чаще всего для этого используется *мышь*. В ноутбуках вместо мыши может использоваться другой манипулятор – *тачпад* (сенсорная панель). Для ввода графической информации применяют *сканер*, *световое перо, графический планшет,* для ввода звуковой информации – *микрофон*.

Стандартным устройством вывода является *монитор*, хотя исторически раньше появился *принтер*, и первые компьютеры распечатывали результаты вычислений.

Есть устройства, которые одновременно являются устройством ввода и вывода, например, монитор с сенсорным экраном.

1.1.3. Принцип открытой архитектуры

Принцип открытой архитектуры – принцип построения персонального компьютера, который регламентирует и стан-

дартизирует только описание принципа действия компьютера и его конфигурации (это позволяет собирать компьютер из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями), а также предусматривает наличие в компьютере внутренних расширительных гнезд, в которые пользователь может вставлять различные устройства, удовлетворяющие заданному стандарту.

Рассмотренные нами устройства компьютера соединяются друг с другом. Разработаны специальные стандартные интерфейсы (средства сопряжения двух устройств, в котором все физические и логические параметры согласуются между собой), утвержденные на уровне международных соглашений.

Все устройства компьютера связаны между собой через общую шину (магистраль), которая передает данные между этими функциональными блоками компьютера (см. рис. 1.7). Сигналы по шине идут по трем каналам (шинам) – шине данных, шине адресов, шине управления. По шине управления передаются только управляющие сигналы (команды). По шине адресов – адреса ячеек памяти, из которых считываются данные или команды, а по шине данных – только данные.

В первых ЭВМ процессор управлял не только работой внутренних, но и внешних устройств. Так как скорость обмена данными между процессором и внешними устройствами очень низкая по сравнению со скоростью обработки информации процессором, то большую часть времени процессор простаивал, ожидая завершения операций обмена. В дальнейшем управление работой внешних устройств было передано специальным блокам (электронным схемам) – контроллерам внешних устройств (controller – управляющий). Контроллеры часто называют адаптерами, так как они преобразуют информацию, поступающую от процессора, в соответствующие сигналы, управляющие работой устройств, тем самым обеспечивая совместимость интерфейсов этих устройств. Например, когда контроллер монитора (видеоадаптер или видеокарта) получает код буквы «А» – 01000001, то при помощи управляющих сигналов он организует работу монитора так, чтобы на экране появилась буква «А». Видеокарта, звуковая карта также являются контроллерами. Таким образом, обмен информацией между процессором, памятью и периферийными устройствами осуществляется по системной шине через контроллеры устройств.

Аппаратный порт – специализированный разъем в компьютере, предназначенный для подключения оборудования определенного типа (например, LPT – параллельный порт, COM – последовательный порт, USB –порт и др.).

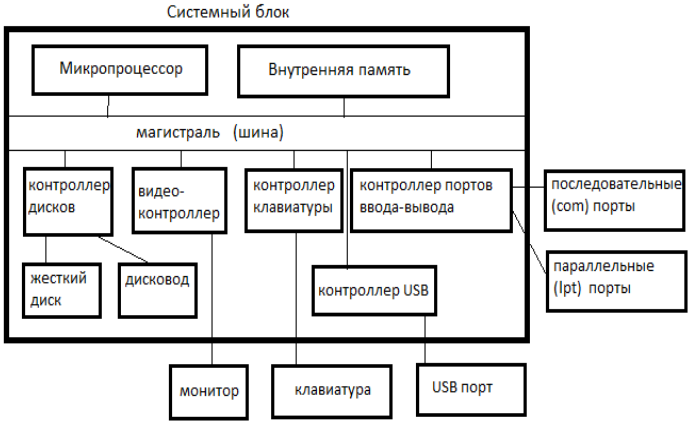


Рис. 1.7. Схема взаимодействия устройств компьютера

Применение данной схемы построения компьютера позволяет легко изменять конфигурацию компьютера путем добавления новых или замены старых устройств. Применение этого принципа позволило фирме IBM, выпустившей в 1981 г. свой первый персональный компьютер IBM PC, выйти в лидеры на рынке персональных компьютеров. Этот подход к построению компьютера был подхвачен другими фирмами, и таким образом появились компьютеры, совместимые с IBM PC (то есть выпущенные другими фирмами (не IBM), но по стандарту IBM PC), основанному на принципе открытой архитектуры.

1.2. Программное обеспечение ЭВМ

Компьютер – это единство двух составляющих: аппаратного и программного обеспечения.

***Программное обеспечение* (*software*)** – совокупность программных средств для создания и эксплуатации систем обработки данных средствами ЭВМ.

Разработано огромное количество программ, не только управляющих работой компьютера, но и решающих многообразные задачи пользователей.

Программное обеспечение (ПО) принято разделять на три класса:

- системное ПО;
- прикладное ПО;
- инструментальное ПО (системы программирования).

Системное ПО – совокупность программных средств, предназначенных для поддержания функционирования и эффективного выполнения основных задач ЭВМ, то есть программы, предназначенные для обслуживания компьютера, управления работой его устройств.

К этому классу ПО относятся операционные системы, которые обеспечивают управление процессом обработки информации и взаимодействие аппаратных средств и сервисные программы, расширяющие возможности ОС.

Прикладное ПО – совокупность программ, предназначенных для решения конкретных задач, стоящих перед пользователем, то есть программы, которые обеспечивают выполнение различных пользовательских задач (не прибегая к программированию). Это текстовые и графические редакторы, системы управления базами данных, табличные процессоры, бухгалтерские пакеты, математические пакеты, компьютерные игры и т. п.

Инструментальное ПО (системы программирования) – программы, которые обеспечивают создание новых программ.

Российское законодательство к объектам интеллектуальной деятельности относит и программные продукты. В связи с этим все программное обеспечение можно разделить на три группы:

- коммерческое (проприетарное от *proprietar* – собственник);
- условно-бесплатное (shareware);
- свободно распространяемое (freeware).

При приобретении программы как коммерческого продукта, способ использования программы определяется лицензионным соглашением. Лицензионное соглашение – это договор на передачу права использования ПО. При покупке ПО приобретается дистрибутив – программный продукт на каком-либо носителе информации (в основном, на DVD). Лицензионное ПО всегда имеет инструкцию по установке и руководство пользователя.

К условно-бесплатному ПО относятся программы, которые распространяются за деньги, но существует возможность бесплатно использовать их некоторое время.

Условно-бесплатными обычно бывают следующие программы:

- программы с ограниченным сроком действия (после истечения указанного срока программа перестает работать, если за нее не произведена оплата);
- программы с ограниченными функциональными возможностями (в случае оплаты пользователю сообщается код, включающий все функции);

- нормально работающие программы, которые размещают вновь образованные фирмы или отдельные программисты в целях рекламы и с предложением произвести добровольную оплату (обычно в небольшом размере).

Свободно распространяемое ПО не предусматривает никаких ограничений относительно срока работы или функционирования программы. «Свобода ПО» означает право пользователя свободно запускать, копировать, распространять, изучать, изменять и улучшать его. Многие программы этого класса имеют открытые исходные коды (Open sourse).

Программист Ричард Столлман создал некоммерческую организацию «Фонд свободного программного обеспечения». Своей основной целью Фонд ставит сохранение программного обеспечения, процесс разработки которого всегда будет гарантированно открытым, а исходные тексты всегда доступны. Для этого разработана специальная лицензия по аналогии с лицензиями на несвободное программное обеспечение: типовой договор автора программы (обладателя авторских прав) с пользователем, в котором автор оговаривает права пользователя по отношению к программе. В отличие от типовой собственнической лицензии, лицензия Столлмана предоставляет пользователю права, являющиеся критериями свободной программы: получать исходные тексты программ, изменять их, распространять измененные и неизмененные версии. Впоследствии лицензия Столлмана получила название GNU General Public License («Основная общественная лицензия GNU»), сокращенно – GNU GPL, или просто GPL. Согласно этой лицензии ни один пользователь, сделавший модифицированную версию свободной программы, не имеет права распространять ее, не соблюдая всех принципов свободного ПО, то есть делать модификацию свободной программы несвободной. Свободное ПО в основном бесплатное, хотя лицензия GNU GPL не запрещает коммерциализации.

Нужно отличать свободное бесплатное ПО от бесплатного проприетарного ПО. Многие производители программного обеспечения и компьютерного оборудования заинтересованы в широком бесплатном распространении своего программного обеспечения, которое не является свободным. К таким программным продуктам относятся следующие:

- новые недоработанные (бета) версии программных продуктов (это позволяет провести их широкое тестирование);
- программные продукты, являющиеся частью принципиально новых технологий (это позволяет завоевать рынок);

- дополнения к ранее выпущенным программам, исправляющие найденные ошибки или расширяющие возможности;
- устаревшие версии программ;
- драйверы к новым устройствам или улучшенные драйверы к уже существующим.

Вопросы для самоконтроля

1. Что такое архитектура ЭВМ? Перечислите основные устройства ЭВМ. В чем состоит принцип открытой архитектуры?
2. Кем были сформулированы основные принципы организации ЭВМ и в чем они состоят?
3. Опишите архитектуру современного компьютера.
4. Как можно классифицировать программное обеспечение по его назначению?
5. На какие классы можно разделить программное обеспечение по способу его распространения?
6. Что общего и в чем отличие условно-бесплатного и свободного программного обеспечения?

ГЛАВА 2.
ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

2.1. Функции операционной системы

Операционная система по праву считается самой главной программой компьютера. Это объясняется многообразием функций, которые она выполняет.

Операционная система (ОС) – совокупность программных средств, осуществляющих управление аппаратной частью ЭВМ, обеспечивающих эффективное использование ресурсов, запуск программ, их взаимодействие с внешними устройствами и другими программами, диалог пользователя с компьютером.

Есть и другие определения ОС, но большинство из них сводится к перечислению функций операционной системы.

По сути, операционная система обеспечивает три вида интерфейса (взаимодействия, взаимосвязи):

- ***Аппаратный интерфейс*** – совместное функционирование всех устройств компьютера, распределение ресурсов.
- ***Программный интерфейс*** – обеспечение эффективной работы и взаимодействия программ между собой и с аппаратурой.
- ***Пользовательский интерфейс*** – совокупность средств, при помощи которых пользователь взаимодействует с различными программами и устройствами.

Исходя из этого, можно перечислить **основные функции операционной системы**:

1. Управление устройствами ввода-вывода, обеспечение взаимодействия с устройствами компьютера.
2. Управление внутренней и внешней памятью компьютера, обслуживание файловой системы.
3. Управление работой приложений (программ).
4. Обеспечение взаимодействия пользователя с компьютером (пользовательского интерфейса).

Классифицируют ОС по нескольким признакам:

- однозадачные и многозадачные (по числу одновременно выполняющихся задач, или программ);
- однопользовательские и многопользовательские (по числу одновременно работающих пользователей);
- автономные и сетевые (по наличию сетевых служб и сервисов).

В многозадачных, многопользовательских и сетевых ОС функциональность системы расширяется.

Дополнительные функции многозадачной ОС:

- обеспечение одновременного выполнения нескольких программ, управление процессами;
- распределение ресурсов компьютера между выполняющимися программами;
- обеспечение обмена данными между программами.

Дополнительные функции многопользовательской ОС:

- обеспечение многопользовательского режима работы;
- разграничение прав доступа;
- защита данных то несанкционированного доступа.

Дополнительные функции сетевой ОС:

- обеспечение сетевого интерфейса.

Рассмотрим подробнее наиболее важные функции ОС.

2.1.1. Управление устройствами ввода-вывода и другим аппаратным обеспечением компьютера

Важная задача ОС – управлять работой различных устройств компьютера и обеспечивать взаимодействие программ компьютера с этими устройствами. Существуют сотни различных моделей разнообразных устройств компьютера. Ни один разработчик программного обеспечения не может учесть все варианты взаимодействия программы с тем или иным устройством. ОС выполняет роль посредника между программой и устройством, к которому обращается программа. В ОС содержатся специальные программы, называемые ***драйверами***, при помощи которых и осуществляется управление тем или иным устройством. Драйверы имеют точки входа для взаимодействия с программами, а ОС распределяет обращения приложений к драйверам. Обычно с ОС поставляются драйверы для основных устройств компьютера. Однако для некоторых устройств (таких, как графическая плата или принтер) могут потребоваться специальные драйверы, которые обычно предоставляются производителем устройства.

2.1.2. Управление памятью

Память, как внутренняя, так и внешняя, является важнейшим ресурсом, управление которым осуществляется ОС. Для увеличения объема оперативной памяти используется технология виртуальной памяти, когда оперативная память и часть внешней памяти на диске рассматриваются как единое пространство виртуальной памяти с единой адресацией, которая

называется логической адресацией, так как не совпадает с реальными физическими адресами запоминающих устройств.

ОС отслеживает свободную и занятую память, выделяет оперативную память процессам и освобождает ее при завершении процессов, выполняет преобразование логических адресов в физические адреса запоминающих устройств. Когда размеры основной памяти не достаточны для размещения в ней всех процессов, ОС вытесняет неактивные процессы из оперативной памяти на диск и возвращает их в оперативную память, когда в ней освобождается место.

Кроме того, технология виртуальной памяти позволяет разделять физическую память на отдельные блоки и распределять их между различными задачами (процессами). При этом предусматривается некоторая схема защиты, которая ограничивает задачу теми блоками памяти, которые ей принадлежат, и не разрешает записывать данные в другие блоки памяти.

Можно выделить три подхода к организации виртуальной памяти:

- **Страничная организация памяти** (*страница* – блок памяти фиксированного размера). При страничной организации памяти любому процессу доступна лишь та физическая память в пределах выделенного блока, которая ему соответствует, и не доступна память другого блока, выделенная другому процессу.
- **Сегментная организация памяти (***сегмент* – блок памяти с переменным размером, одновременно доступный нескольким процессам). Сегменты могут иметь различные размеры, причем размер сегмента может меняться динамически.
- **Сегментно-страничная организация памяти**, комбинирующая два предыдущих подхода.

В современных ОС используется сегментно-страничная организация памяти, при которой происходит двухуровневая трансляция виртуального адреса в физический. Виртуальный адрес состоит из трех полей: номера сегмента виртуальной памяти, номера страницы внутри сегмента и смещения внутри страницы. Соответственно, для преобразования виртуальных адресов в физические применяются две таблицы: таблица сегментов, которая связывает номер сегмента с таблицей страниц и называется также таблицей каталогов страниц, и таблица страниц для каждого сегмента. При преобразовании виртуального адреса элемента в физический адрес первоначально по номеру каталога страниц устанавливается требуемая таблица страниц. Далее с использованием таблицы страниц и

номера страницы определяется физический адрес страницы, а, зная смещение внутри страницы, можно определить физический адрес искомого элемента (рис. 2.1).

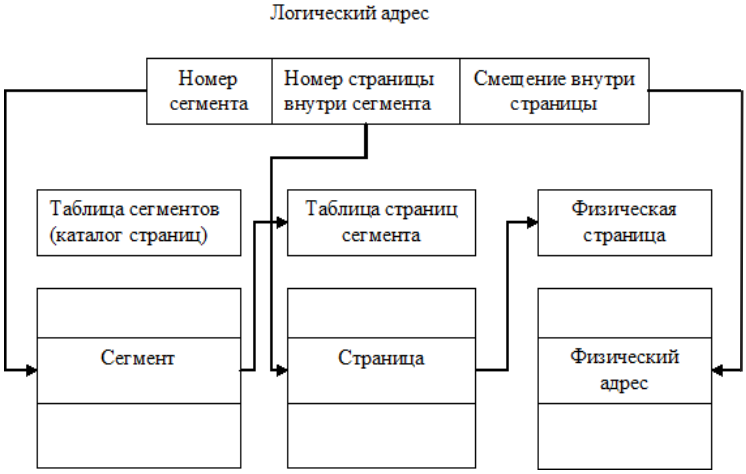


Рис. 2.1. Сегментно-страничная организация памяти

2.1.3. Организация файловой системы

ОС должна обеспечивать доступ к данным и организацию их долговременного хранения во внешней памяти. Эта функция реализуется с помощью *файловой системы* – системы хранения данных на внешних устройствах.

Логическая организация файловой системы

Файл – это имеющая имя совокупность данных на внешнем носителе информации. (Под данными понимаются и сами данные и программы для их обработки.)

Пользователи могут давать файлам символьные имена. Каждая ОС имеет свои правила составления имен файлов (допустимые символы имени, длина имени). В некоторых ОС, например в Windows, имя файла включает и расширение, или тип, файла. Обычно расширения txt, doc используются для текстовых файлов, exe – для исполняемых файлов и т. д. Однако в ОС семейства Unix не принято дописывать расширение к имени файла, хотя это и не является ошибкой.

Файлы для удобства использования объединяют в группы под общим названием. Так образуются каталоги (директории, папки).

Каталог (директория, папка) – совокупность файлов и каталогов, объединенных под одним именем. Каталог содержит

список файлов и каталогов, находящихся в нем. Для ОС каталог – это файл, в котором записана информация о файлах и каталогах: имя, расширение, местонахождение, размер, дата и время создания и др.

В логической организации файловой системы обязательно выделяется хотя бы один каталог – корневой (главный, основной), который содержит все остальные каталоги.

Каталог может содержать другой каталог, который называется *подкаталогом*. По отношению к подкаталогу каталог, содержащий его, называется *надкаталогом*, или *родительским каталогом*. Таким образом, формируется древовидная иерархическая структура – дерево каталогов (рис. 2.2).

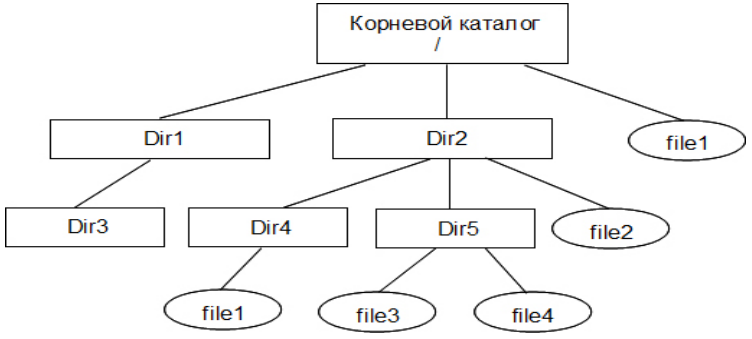


Рис. 2.2. Файловая система. Дерево файлов и каталогов

В разных каталогах могут содержаться файлы с одним и тем же именем. Поэтому, чтобы однозначно определить файл, необходимо указать *полное имя файла* – маршрут (*путь*) к этому файлу, то есть перечислить все каталоги, ведущие к этому файлу. При указании пути к файлу каталоги отделяются друг от друга специальным символом (в Windows это «\», в Linux – «/»). Так, file1 корневого каталога (рис. 2.2) имеет полное имя: /file1, а file1 каталога Dir4: /Dir2/Dir4/file1.

Физическая организация файловой системы

Каким образом логическая организация файловой структуры будет воплощена на физическом носителе (жестком или оптическом диске, флешке и др.), зависит от конкретной файловой системы. В каждой операционной системе этот вопрос решается по-своему. Поэтому существует большое количество файловых систем. Некоторые из них поддерживаются различными ОС, например, *FAT* (сейчас, в основном, используется для хранения данных на флешках небольшой емкости), *UDF*

(для оптических дисков), поэтому данные с этих носителей могут быть прочитаны любой ОС. Как правило, для организации хранения данных на жестком диске каждая ОС использует свою файловую систему, которая не обязательно поддерживается другими ОС. Например, в Linux поддерживаются такие файловые системы, как *XFS*, *RaiserFS*, *JFS*, *EXT3*, но эти файловые системы не поддерживаются в Windows, поэтому данные (файлы) перечисленных файловых систем не могут быть прочитаны в Windows.

Выбор того, какая файловая система будет использоваться, осуществляется при *форматировании* диска. Перед форматированием операционная система перечисляет поддерживаемые файловые системы и предлагает пользователю выбрать одну из них. После этого диск форматируется, то есть размечается на блоки (кластеры) в соответствии с правилами выбранной файловой системы и на нем размещаются служебные данные файловой системы.

Блок, или кластер – единица хранения информации, минимальный размер которого равен размеру сектора диска (512 байт). Чем больше размер диска, тем больше размер кластера. Диск разделяется на небольшие блоки (кластеры), они нумеруются и таким образом создается адресное пространство диска. Для того чтобы записать информацию на диск, надо «позиционировать головку», то есть указать контроллеру, в какой сектор эту информацию записать. Адрес сектора состоит из номера цилиндра (или дорожки), номера считывающей головки и порядкового номера сектора на дорожке.

При записи файла на диск файловая система определяет, какие блоки свободны и начинает запись данных в эти блоки (они не обязательно расположены последовательно на диске). То есть файл, записанный на диск, представляет совокупность блоков, а файловая система хранит служебную информацию о том, в какие блоки записан файл (адреса блоков). В служебных областях файловой системы хранятся сведения о самой файловой системе, о том, какой кластер к какому файлу относится, какие кластеры свободны, какие испорчены, размеры, названия и другие атрибуты элементов файловой системы.

Разделы диска и таблица разбиения диска

Первые версии MS-DOS не могли обеспечить доступ к большим дискам (а объемы дисков росли быстрее, чем возможности DOS). Поэтому физические диски в Intel-системах стали разбивать на разделы. Каждый раздел жесткого диска воспринимается ОС как отдельный физический диск. Это дает возможность установить на компьютер разные опе-

рационные системы, нужно только каждую систему установить в отдельный раздел.

Чтобы ОС знала адрес начала каждого раздела, в начало жесткого диска размещается *таблица разбиения диска на разделы* (partition table). Эта таблица находится в MBR (Master Boot Record – главная загрузочная запись), расположенной в самом первом секторе жесткого диска.

Таблица разделов содержит 4 записи по 16 байт для 4 разделов, которые называют *первичными*. Именно поэтому диск можно разделить не более чем на четыре первичных раздела.

Когда выяснилось, что четырех разделов для больших дисков мало, были изобретены *логические* разделы. Для организации логических разделов один из первичных разделов объявляется «*расширенным*», и в нем создаются логические разделы. Расширенные разделы сами по себе не используются, они могут лишь хранить логические разделы. Каждый расширенный раздел имеет свою таблицу разбиения на разделы. Число логических разделов в принципе не ограничено. Однако реально ограничения все же существуют и зависят от операционной системы и типа жесткого диска.

Процедура разбиения диска на разделы осуществляется специальной программой fdisk или ее усовершенствованными аналогами. При установке ОС обычно пользователю дается возможность определить, на какие разделы необходимо разделить диск, а после этого выбрать, какие файловые системы устанавливать в каждый раздел. Разделы форматируются, и только после этого производится установка операционной системы.

2.1.4. Управление работой приложений

Прикладную компьютерную программу часто называют приложением. Приложения не взаимодействуют напрямую с устройствами компьютера, а только через операционную систему. ОС позволяет программистам абстрагироваться от деталей реализации и функционирования устройств, предоставляя необходимый для работы приложений набор функций. Этот набор функций (несколько тысяч функций) называется ***интерфейс прикладного программирования API*** (*application programming interface).* Программисты, разрабатывая приложения, в программах используют необходимые функции API, тем самым дают команды операционной системе сделать ту или иную операцию. Поэтому без ОС работа приложения не возможна. Нельзя запустить на компьютере

текстовый редактор, игру или какое-либо другое приложение, если на нем не установлена ОС.

Операционная система осуществляет запуск, выполнение и завершение приложений. Прежде, чем запустить приложение, ОС считывает соответствующий файл из внешней памяти, как правило, с жесткого диска и загружает его в оперативную память компьютера. При выполнении приложения ОС обеспечивает выделение ресурсов компьютера, необходимых для работы данного приложения. Если запущено несколько приложений, то ОС управляет правильным распределением ресурсов компьютера между этими приложениями.

Для правильной работы приложение должно пройти операцию установки (инсталляции). Необходимость установки связана с тем, что разработчики ПО не могут заранее предвидеть аппаратную и программную конфигурацию конкретного компьютера. Таким образом, дистрибутивный пакет (установочный комплект) – это полуфабрикат ПО, из которого в процессе установки формируется полноценное рабочее приложение, осуществляется привязка к существующей аппаратно-программной среде.

Современные ОС берут на себя управление процессом установки приложений. Они управляют распределением ресурсов компьютера между приложениями, обеспечивают доступ устанавливаемого приложения к драйверам устройств, формируют общие ресурсы, регистрируют приложение.

Если приложение не будет больше использоваться, его необходимо деинсталлировать. В ОС, реализующей принцип совместного использования ресурсов, процесс удаления имеет свои особенности и происходит под контролем ОС (нельзя допустить удаления ресурсов, используемых другими приложениями).

2.1.5. Интерфейс пользователя

Посредством интерфейса пользователя осуществляется диалог пользователя с компьютером. Пользовательский интерфейс дает возможность при помощи команд, меню, графической оболочки управлять работой программ и аппаратуры и, таким образом, скрывает от пользователя сложные и ненужные подробности взаимодействия с аппаратурой.

Изначально диалог пользователя с компьютером осуществлялся на машинном языке (в двоичных кодах). Вести такой диалог могли специалисты высокой квалификации. Неудобство ведения диалога на машинном языке очевидно, поэтому впоследствии был разработан более дружелюбный для человека ин-

терфейс – ***командный,*** или ***интерфейс командной строки*** (Command line interface, CLI), в котором диалог ведется при помощи команд, записываемых пользователем в командной строке. Строка, в которой записываются команды, называется строкой-приглашением, так как появление этой строки на экране показывает готовность операционной системы к диалогу с пользователем, приглашает его к диалогу. Вид строки-приглашения может отличаться в разных операционных системах, обычно в ней указывается имя текущего каталога. До сих пор в современных операционных системах остается возможность использования командного интерфейса. Например, в Windows можно запустить командный интерпретатор (*Пуск – Программы – Стандартные – Командная строка* или *Пуск – Выполнить* – набрать команду *cmd*). Введите в строку-приглашение команду *help*, чтобы вывести справку о командах Windows (рис. 2.3).

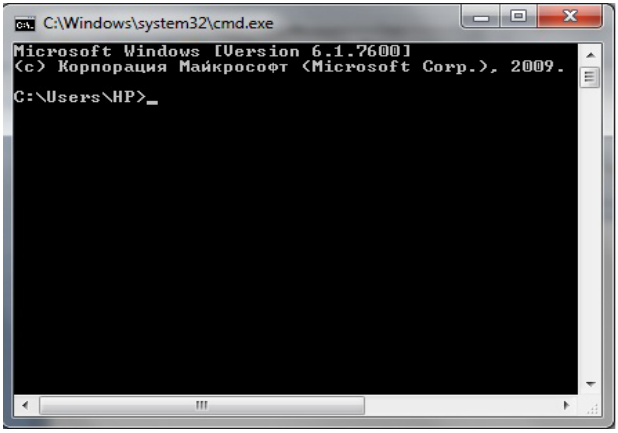


Рис. 2.3. Командная строка в Windows

В Linux в консольном режиме диалог пользователя с компьютером осуществляется при помощи команд. Но и в графическом режиме можно запустить приложение *Терминал*, которое эмулирует консоль.

Командный интерпретатор операционной системы расшифровывает введенную пользователем команду и выполняет ее. Конечно, операционные системы отличаются и набором команд, и синтаксисом записи этих команд. Можно выделить общий формат команды:

имя_команды [параметр_1][параметр_2][…]

Когда получили распространение персональные компьютеры и его широко стали использовать не только програм-

мисты, диалог с операционной системой при помощи команд многим пользователям казался сложным, требовалось разработать для них более дружелюбный интерфейс. В 1973 г. в исследовательском центре компании Xerox была разработана концепция графического интерфейса WIMP (Windows – окна, Icons – пиктограммы, Menus – меню, Point-n-Click – указатель, указывать и щелчок), которой до сих пор следуют разработчики современного графического интерфейса. ***Графический интерфейс*** (*graphical user interface, GUI)* – интерфейс, обеспечивающий диалог пользователя с компьютером при помощи графических объектов (пиктограмм, окон, меню и т. д.), то есть команда дается посредством манипуляций с графическими объектами.

Эволюция пользовательского интерфейса связана с появлением более дружелюбного интерфейса, способного выполнять голосовые команды человека, реагировать на жесты и т. п.

2.1.6. Поддержка многозадачности

Задачей (или процессом) *называют программу в стадии выполнения.* Конечно, один процессор не может выполнять одновременно несколько действий, эффект многозадачности достигается за счет быстрого переключения с одной задачи на другую. В оперативной памяти компьютера одновременно находятся несколько программ. При этом каждая программа загружается в свой участок оперативной памяти, называемый разделом, и не влияет на выполнение другой программы. В каждый момент времени процессор выполняет только одну из них.

Многозадачными ОС являются UNIX, OS/2, Windows 95 и выше, Linux, Mac OS и др. Самые известные однозадачные ОС: MS DOS, первая Mac OS, которая называлась «System 1.0».

Многозадачная ОС должна обеспечивать оптимальное разделение ресурсов компьютера между работающими процессами. Ресурсы – это то, в чем нуждается процесс для своей работы (оперативная память, процессор, внешний носитель, программа и др.). Выделение ресурсов процессам и их освобождение осуществляет менеджер ресурсов. Самый главный ресурс – процессорное время между выполняющимися задачами. ОС делит процессорное время на промежутки, *кванты*, и распределяет их между всеми задачами. Задача распределения квантов процессорного времени в системе называется диспетчеризацией или *планированием задач* и занимается этим диспетчер задач. Диспетчер задач распределяет про-

цессорное время между процессами, руководствуясь определенными правилами управления очередью процессов (эти правила иногда называют дисциплиной обслуживания). В зависимости от того, какие правила используются, различают несколько видов многозадачности.

Невытесняющая многозадачность – тип многозадачности, при котором ОС одновременно загружает в память два или более приложений, но процессорное время предоставляется только основному приложению (foreground). Другие приложения должны быть активизированы в фоновом режиме (background).

Добровольная, или кооперативная, совместная многозадачность – процесс занимает столько времени, сколько ему необходимо. Освободить процессор процесс может только добровольно. Планировщик может переключать процессы только после того, как текущая задача освободит процессор. Плюсом такой организации является простота реализации. Однако добровольная многозадачность часто приводит к зависанию всей системы, если процесс, занявший процессор, не освобождает его.

Вытесняющая многозадачность реализуется с использованием следующих методов:

- *Равномерное квантование* – выделение процессам времени по принципу обычной очереди. Каждому процессу выделяется одинаковый квант процессорного времени. Если за отведенное время процесс не завершился, то он переходит в конец очереди и ждет нового кванта процессорного времени.
- *Планирование согласно приоритетам* – планировщик присваивает каждому из работающих процессов *приоритет* и решает, в каком порядке, как долго и какие будут выполняться процессы. Планировщик в любой момент времени может отнять процессор у одной задачи и передать другой. Обычно такой акт происходит в результате реакции на *событие*. В случае, если происходит событие, относящееся к процессу с большим приоритетом, планировщик вытесняет текущий процесс и начинает выполнять тот, у которого приоритет больше. Например, щелчок мыши – это событие, которое может привести к передаче процессора другому процессу, которому принадлежит окно, где щелкнули мышью. Но планировщик может решить, что текущий процесс более важен. При такой организации планировщик получает систему в свое распоряжение, когда процесс освобождает процессор или истекает отведенное данному процессу время.

Таким образом, каждый процесс в ОС имеет момент своего начала и завершения и в течение своего существования может

находиться в различных состояниях и переходить и одного состояния в другое. Модели состояний процесса различаются в разных ОС. Например, в Windows NT различают 7 состояний процесса, в Unix – 9 состояний. Основными состояниями процесса являются: запуск, готовность, выполнение, ожидание и завершение работы. При ***запуске*** процессу выделяются все необходимые ресурсы и он загружается в оперативную память. ***Готовность*** – процесс встает в очередь и может начать выполнение сразу, как только будет предоставлено процессорное время. ***Выполнение*** (активное состояние) – процесс занимает процессорное время, выполняется процессором. ***Ожидание*** – процесс блокируется, так как ожидает выполнения некоторого события, например, ввода с клавиатуры, освобождения какого-либо необходимого ему ресурса и др., и, как только ожидаемое событие выполняется, процесс переходит в состояние готовности. При ***завершении*** работы все ресурсы, которые использовались процессом, освобождаются, в том числе и оперативная память.

2.1.7. Поддержка многопользовательского режима

Операционные системы, которые могут поддерживать работу нескольких пользователей, называются *многопользовательскими*. Многопользовательская система должна обеспечить разграничение прав доступа и защиту информации каждого пользователя от несанкционированного доступа других пользователей. Наличие средств защиты от несанкционированного доступа – главное отличие от однопользовательских ОС.

В многопользовательских ОС каждый пользователь может настраивать для себя графический интерфейс пользователя, то есть может создать собственные наборы ярлыков, группы программ, задать индивидуальную цветовую схему, переместить в удобное место панель задач.

Многопользовательские ОС: UNIX, Windows NT и выше, Linux, Mac OS X и др.; однопользовательские ОС: MS DOS, OS/2 ранние версии, операционные системы для мобильных устройств, например, Windows Mobile и др.

2.1.8. Поддержка сети

Если операционная система отдельного компьютера позволяет ему работать в сети, то есть предоставлять свои ресурсы в общее пользование и использовать ресурсы других компьютеров сети, то такая операционная система называется сетевой ОС.

В сетевых ОС пользователи получают информацию о наличии других компьютеров в сети и могут осуществить логический вход в другой компьютер, чтобы воспользоваться его ресурсами, преимущественно файлами. Сетевая ОС обязательно содержит программную поддержку для сетевых интерфейсных устройств (драйвер сетевого адаптера), а также средства для удаленного входа в другие компьютеры сети и средства доступа к удаленным файлам.

Вопросы для самоконтроля

1. Расскажите о назначении операционных систем.
2. Перечислите функции ОС.
3. По каким признакам классифицируются ОС?
4. Какие дополнительные функции должна выполнять многозадачная ОС?
5. Какие дополнительные функции накладываются на многопользовательскую ОС?
6. Расскажите об организации файловой системы.
7. Многозадачность, классификация, способы реализации многозадачности.

2.2. Архитектура операционных систем

Большинство современных ОС представляют собой хорошо структурированные модульные системы, способные к развитию, расширению и переносу на другие аппаратные платформы. Какой-либо единой архитектуры ОС не существует, но есть универсальные подходы к структурированию ОС.

2.2.1. Кольца защиты

ОС должна иметь исключительные права для того, чтобы управлять работой приложений, выделять для них ресурсы компьютера в многозадачном режиме, обеспечивать защиту данных, контролировать доступ приложений к памяти и др. Таким образом, должны быть обеспечены как минимум два уровня привилегий – высокий для ОС (привилегированный режим, или ***режим ядра***) и низкий для остальных приложений (непривилегированный, или ***пользовательский режим*** работы).

В современных системах чаще всего используется аппаратно реализованное разделение уровней привилегий. В схемах защиты современных процессоров предусмотрены четыре кольца защиты. То есть архитектура современных про-

цессоров позволяет организовать четыре уровня привилегий, однако большинство ОС использует только два. Внутреннее кольцо имеет номер 0, в нем работает сама ОС (режим ядра). Внешнее кольцо – номер 3, в нем работают приложения (режим пользователя). ОС использует кольца защиты для гарантии, что только компоненты ОС могут обращаться к ее внутренним механизмам. Другими словами, приложение не может изменить какие-либо параметры, способные привести к отказу всей системы.

Программа, исполняющаяся в привилегированном режиме, имеет все права, поэтому процессор выполняет любую команду этой программы. Напротив, в пользовательском режиме процессор может исполнять только обычные команды обработки данных и не имеет доступа к системному адресному пространству. Поэтому, если приложение, работающее в пользовательском режиме, обратится к системной области памяти, процессор немедленно прервет работу этого приложения и оно будет закрыто. Пользовательские программы имеют доступ только к своим внутренним объектам и тем внешним объектам, доступ к которым им разрешен ОС.

Такие модули ОС, как планировщик, менеджер памяти, драйверы внешних устройств, должны работать только в системном режиме. Обычно в системном режиме исполняется все ядро ОС, в том числе и те модули, для которых это необязательно – файловые системы, различные сервисные модули и т. д. Поэтому привилегированный режим часто называют режимом ядра.

Переключение из пользовательского режима в режим ядра осуществляется специальной командой. Обычно такая команда сразу же передает управление одному из модулей ядра системы.

2.2.2. Операционные системы с монолитным ядром

Монолитное ядро (monolithic kernel) – старейший способ организации операционных систем. В этом случае компоненты операционной системы являются не самостоятельными модулями, а составными частями одной большой программы. Монолитное ядро представляет собой набор процедур (подпрограмм), каждая из которых может вызвать каждую. Все процедуры работают в привилегированном режиме. Таким образом, монолитное ядро – это такая схема операционной системы, при которой все ее компоненты являются составными частями одной программы, используют общие структуры данных и взаимодействуют друг с другом путем непосредственного вызова процедур.

Примером систем с монолитным ядром является большинство Unix-систем, в том числе Linux, MS-DOS и другие ОС.

Достоинствами этой архитектуры являются высокая скорость работы ОС.

Есть и недостатки у этой архитектуры. Поскольку все ядро работает в одном адресном пространстве, сбой в одном из компонентов может нарушить работоспособность всей системы.

2.2.3. Операционные системы с микроядром

Современная тенденция в разработке операционных систем состоит в перенесении значительной части системного кода на уровень пользователя и одновременной минимизации ядра.

Суть микроядерной архитектуры в том, что в привилегированном режиме остается работать только очень небольшая часть ОС, называемая микроядром. Микроядро защищено от остальных частей ОС и приложений. В состав микроядра обычно входят машинно-зависимые модули, а также модули, выполняющие базовые функции ядра по управлению процессами, обработке прерываний, управлению виртуальной памятью, пересылке сообщений и управлению устройствами ввода-вывода. Данные функции очень трудно выполнить в пользовательском пространстве.

Все остальные функции ядра оформляются в виде приложений, работающих в пользовательском режиме.

Примеры ОС с микроядерной архитектурой: Symbian OS; микроядро Mach, используемое в GNU/Hurd и Mac OS X; Minix и другие ОС.

Формально Windows NT и выше часто называют микроядерной операционной системой. Однако микроядро этой ОС слишком велико (более 1 Мбайт), чтобы носить приставку «микро». Кроме того, все компоненты ядра работают в одном адресном пространстве и активно используют общие структуры данных, что свойственно операционным системам с монолитным ядром. Таким образом, в Windows NT и выше сочетаются два подхода, поэтому часто говорят, что эта ОС имеет гибридное ядро.

Главное достоинство микроядерной архитектуры – устойчивость к сбоям оборудования, ошибкам в компонентах системы. Микроядерная архитектура более надежна, чем с монолитным ядром, поскольку ошибка на уровне непривилегированной программы менее опасна, чем отказ на уровне режима ядра.

Кроме этого, высокая степень модульности ядра операционной системы упрощает добавление в него новых компо-

нентов. В микроядерной ОС можно, не прерывая ее работы, загружать и выгружать новые драйверы, файловые системы и т. д. Компоненты ядра операционной системы ничем принципиально не отличаются от пользовательских программ, поэтому для их отладки можно применять обычные средства.

Недостатком этой архитектуры является некоторое снижение производительности, так как передача данных между процессами требует накладных расходов.

2.3. Операционная система Linux

Популярная сейчас операционная система Linux является представителем операционных систем семейства UNIX. Поэтому для начала сделаем обзор UNIX и причин популярности операционных систем, входящих в это семейство.

2.3.1. Семейство UNIX

Операционная система UNIX была разработана группой сотрудников Bell Labs под руководством Д. Ричи, К. Томпсона и Б. Кернигана в 1969 г. Но в наши дни, когда говорят об этой операционной системе, чаще всего имеют в виду не конкретную ОС, а скорее целое семейство UNIX-подобных операционных систем.

Вот некоторые представители семейства UNIX: System V UNIX, BSD UNIX, OSF/1, Solaris, Linux, freeBSD и др.

Рассмотрим причины 40-летней популярности ОС семейства UNIX.

1. UNIX – многозадачная многопользовательская система с широким спектром услуг. Один мощный сервер может обслуживать запросы большого количества пользователей. Система может выполнять различные функции – работать как вычислительный сервер, обслуживающий сотни пользователей, как сервер базы данных, как сетевой сервер, поддерживающий важнейшие сервисы сети (telnet, ftp, электронную почту, службу имен DNS и т. д.), как сетевой маршрутизатор.
2. UNIX относится к классу наиболее надежных и безопасных ОС.
3. Переносимость. Код системы написан на языке высокого уровня Си, что сделало ее простой для понимания, изменений и переноса на другие аппаратные платформы. Все части системы, не считая ядра, являются полностью машинно-независимыми. Эти компоненты написаны на языке Си, и для их переноса на новую платформу требуется только перекомпиляция исходных текстов в коды целевого компьютера.

4. Стандартизация. Несмотря на многообразие версий UNIX, основой всего семейства являются принципиально одинаковая архитектура и ряд стандартных интерфейсов. Опытный администратор без большого труда сможет обслуживать другую версию системы.

IEEE и POSIX – стандартизация интерфейса прикладного программирования API UNIX, которая обеспечивает переносимость приложений в разные версии ОС семейства UNIX. То есть приложения, написанные, например, для Solaris должны выполняться и под Linux, и под freeBSD.

5. Наличие большого количества приложений, в том числе свободно распространяемых, начиная от простейших текстовых редакторов и заканчивая мощными системами управления базами данных.

2.3.2. Краткая история Linux

Операционные системы типа UNIX изначально разрабатывались для работы на больших многопользовательских компьютерах – мейнфреймах. В начале 90-х гг. студент хельсинкского университета Линус Торвальдс приступил к разработке UNIX-подобной ОС для IBM-совместимых персональных компьютеров. Файлы первого варианта Linux (исходные коды) были опубликованы в Интернете в 1991 г. Л. Торвальдс не стал патентовать или иным образом ограничивать распространение новой ОС. С самого начала Linux распространяется на условиях, определяемых лицензией General Public License (GPL), принятой для программного обеспечения, разрабатываемого в рамках движения Open Source и проекта GNU. Разработка Линуса Торвальдса представляла собой только ядро операционной системы.

Ядро – это основная, определяющая часть ОС, которая управляет аппаратными средствами и выполнением программ. У*тилиты* выполняют служебные функции.

К 1991 г. в рамках проекта GNU уже было разработано большое количество разного рода утилит. Но для превращения GNU в полноценную ОС не хватало ядра. Разработка ядра также велась, но по разным причинам задерживалась. Поэтому появление разработки Л. Торвальдса было очень своевременным. Таким образом, более правильным было бы называть операционную систему Linux – GNU/Linux.

2.3.3. Основные характеристики ОС Linux

Linux – самая современная, устойчивая и быстроразвивающаяся система, почти мгновенно вбирающая в себя самые последние технологические новшества. Она обладает всеми возможностями, которые присущи современным полнофункциональным операционным системам.

1. Надежная многозадачная многопользовательская ОС для персональных компьютеров.
2. Осуществляет эффективное управление памятью.
3. Поддерживает различные файловые системы.
4. Предоставляет сетевые возможности.
5. Работает на разных аппаратных платформах (на всех версиях микропроцессоров Intel, на процессорах AMD, разработаны версии ОС и для других типов процессоров.

2.3.4. Дистрибутивы Linux

Самые первые версии Linux помещались на двух дискетах. Первая дискета была загрузочной и содержала ядро, а вторая – корневую файловую систему и основные утилиты, разработанные в рамках проекта GNU. Процесс конфигурирования и настройки системы производился вручную и требовал обширных знаний. Чтобы установка Linux стала доступна не только экспертам, стали разрабатываться дистрибутивы Linux.

Дистрибутив Linux – это набор пакетов программного обеспечения, включающий базовые компоненты операционной системы, набор программных приложений, программу инсталляции, которая позволяет установить на компьютер пользователя операционную систему GNU/Linux и набор прикладных программ, необходимых для конкретного применения системы.

Поскольку разработкой дистрибутивов занимается большое количество независимых групп программистов, то сейчас в мире существует уже сотни различных дистрибутивов Linux (см. http://distrowatch.com/) и все время появляются новые. Новые дистрибутивы создаются, в основном, не на пустом месте, а на основе одного из уже существующих дистрибутивов. Отличаются дистрибутивы, прежде всего:

– программой инсталляции;
– используемым средством установки программных пакетов (системой управления пакетами);

– составом утилит и прикладных программ, включенных в дистрибутив;
– сценарием начальной загрузки;
– требованиями к аппаратуре.

Можно выделить три основные группы дистрибутивов:

1. На основе дистрибутива Red Hat, переименованного позднее в Fedora Core. Наиболее известные дистрибутивы этой группы – Mandrake (или Mandriva), в том числе русифицированные – ASPLinux, Linux Ink, AltLinux (на основе Mandrake) и др.
2. На основе дистрибутива Debian. К этой группе относятся наиболее популярный сейчас во всем мире дистрибутив Ubuntu, также Knoppix, Storm и др.
3. На основе дистрибутива Slackware. К этой группе относится openSuSe.

В России сложилось три команды разработчиков, создающих и поддерживающих русифицированные дистрибутивы.

Одна из команд – «ALTLinux» (http://www.altlinux.ru), которая выпускает собственный дистрибутив ALTLinux. На протяжении последних лет эта команда активно работает в направлении внедрения свободного программного обеспечения в образовательные учреждения России. Они разработали специальный «Пакет свободного программного обеспечения для образования».

Вторая команда представлена фирмой «ASPLinux» (http://www.asplinux.ru), которая тоже выпустила собственный дистрибутив ASPLinux.

Третья команда – Санкт-Петербургская фирма «Linux Ink» (http://www.linux-ink.ru), которая выпускает дистрибутив «НауЛинукс», основанный на всемирно известном дистрибутиве Scientific Linux, а также –версии дистрибутивов, специально ориентированных для использования в образовательных учреждениях.

2.3.5. Интерфейс пользователя

Linux – многопользовательская система, поэтому чтобы начать работать, пользователь должен «представиться» системе, введя свой ***логин*** и ***пароль***. Регистрацию новых пользователей обычно выполняет администратор системы. Пользователь не может изменить свое учетное имя, но может установить и изменить свой пароль.

Ядро ОС UNIX идентифицирует каждого пользователя по его ***идентификатору*** (UID – User Identifier), уникальному целому значению, присваиваемому пользователю при реги-

страции в системе. Кроме того, каждый пользователь относится к некоторой группе пользователей, которая также идентифицируется некоторым целым значением (GID – Group IDentifier).

Администратор системы обладает большими возможностями, чем обычные пользователи, он имеет нулевой UID и называется ***суперпользователем*** или ***root***. Он имеет неограниченные права на доступ к любому файлу и на выполнение любой программы, возможность полного контроля над системой.

Изначально в системах семейства UNIX использовался **командный интерфейс**. Интересной особенностью является то, что было разработано несколько командных интерпретаторов с похожими, но различающимися своими возможностями командными языками. Общее название для любого командного интерпретатора ОС семейства UNIX – shell (оболочка), так как интерпретатор представляет внешнее окружение ядра системы. Наиболее известными из них являются: sh (Bourne Shell), более мощный bash (Bourne Again Shell), самый мощный zsh (The Z Shell). Пользователь может выбрать любой интерпретатор.

Вызванный командный интерпретатор выдает приглашение на ввод пользователем командной строки (символ $ для обычных пользователей, # – для суперпользователя root.)

Команды

Команды в shell обычно имеют следующий формат:

<имя команды> <флаги> <аргументы>

После нажатия на клавишу Enter начинается выполнение команды.

Командный интерпретатор является удобным средством программирования. Программы на языке shell часто называются скриптами или сценариями (script). Интерпретатор считывает строки из файла-скрипта (командного файла) и выполняет их, как если бы они были введены в командной строке.

Графический интерфейс

Современные ОС семейства UNIX, в том числе и Linux, обеспечивают и **графический пользовательский интерфейс**. Для вывода графики используется система XWindow, известная еще как X11 (или Xfree86). XWindow использует архитектуру клиент–сервер. X-сервер управляет оборудованием ввода (клавиатура, мышь) и вывода (монитор). Программы, осуществляющие ввод и вывод графических данных, являются клиентами (X-клиентами), то есть для операций ввода и вывода обращаются к X-серверу. Таким образом, X-сервер стыкует аппаратную часть с программной.

При запуске одного X-сервера экран становится черным и появляется курсор в виде крестика. Чтобы появился привыч-

ный «оконный» интерфейс, необходимо запустить программу X-клиент, которая будет прорисовывать окна, следить за изменением размеров окон, их перемещением и т. д. Такая программа называется *менеджером окон*, она обеспечивает любые манипуляции с окнами.

Программы, которые используют графический интерфейс, только выводят информацию в окна, созданные менеджером окон. Оконных менеджеров в мире Unix очень много: fvwn, IceWM, Windows Maker, Motif, LessTif и др.

Оконный менеджер не обеспечивает связи между программами, как это делается в Windows. Для обеспечения такой связи используются более сложные интегрированные графические среды, в которых оконный менеджер является одной из многих подпрограмм. Примерами интегрированных графических оболочек являются системы KDE и Gnome. Таким образом, в Linux пользователь может выбрать наиболее понравившуюся по дизайну и функциональности графическую оболочку, учитывая, конечно, и аппаратные требования.

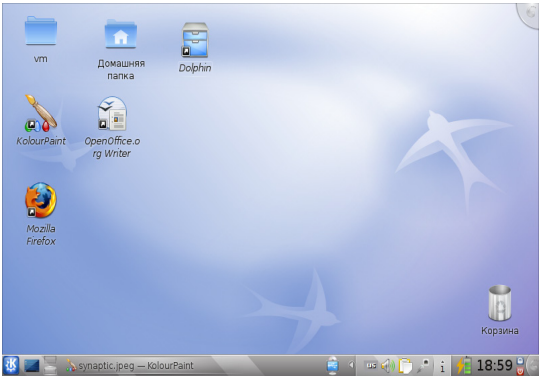


Рис. 2.4. Графическая оболочка KDE (дистрибутив AltLinux 5.0.0)

Интерфейс KDE (Kool Desktop Environment) построен по тем же принципам, что и графический интерфейс Windows, поэтому у пользователей обычно не возникает особых проблем, связанных с работой в KDE. В состав KDE входит набор тесно интегрированных между собой программ для выполнения повседневной работы (набор программ может отличаться в разных дистрибутивах):

- Dolphin – файловый менеджер.
- K3b – программа для записи CD-, DVD- и BluRay-дисков.

- Konsole – эмулятор терминала.
- Kontact – электронный секретарь, персональный информационный менеджер, включающий клиент электронной почты, адресную книгу, планирование задач, календарь и многое другое.
- Kopete – клиент мгновенных сообщений.
- Konqueror – веб-браузер.
- Gwenview – для просмотра изображений.
- Okular – для просмотра документов различных типов, в частности, PDF, DjVu, FB2, CHM.
- KOffice – офисный пакет и другие программы.

Почти все параметры внешнего вида и поведения KDE можно настроить, используя менеджер настройки CompizConfig или KCC – Центр управления KDE.

Вызов приложений осуществляется из стартового K-меню (см. рис. 2.5).

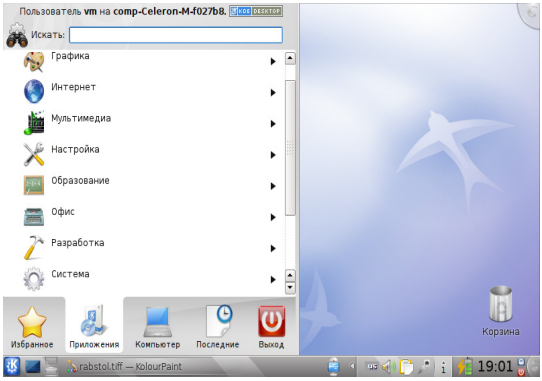


Рис. 2.5. Меню запуска приложений Kickoff в KDE

2.3.6. Файловая система

ОС Linux поддерживает различные файловые системы, которые различаются своими возможностями, производительностью, надежностью.

- *EXT2FS*, *MINIX-1* и *XENIX* – нежурналируемые файловые системы.
- *XFS*, *RaiserFS*, *JFS*, *EXT3* – более надежные журналируемые файловые системы.
- *FAT16, FAT32 (имеют тип VFAT), NTFS* – совместимые с Windows файловые системы. Если один из разделов диска отформатировать под *FAT*, то данные этого раздела бу-

дут доступны и в Windows. Данные разделов, отформатированных под другие файловые системы, не доступны в Windows).

Более надежными и современными являются *журналируемые* файловые системы. В журнал или лог сохраняется список изменений, которые будут произведены с файловой системой перед фактическим их осуществлением. Если вдруг произойдет сбой, например, отключится электропитание компьютера в момент записи информации на диск, то благодаря журналу файловая система будет приведена в непротиворечивое состояние, то есть данные недописанного файла будут удалены с диска. В нежурналируемой файловой системе в этом случае недописанный файл остался бы на диске, но доступ к нему был бы невозможен.

В UNIX-подобных ОС логическая структура файловой системы одинакова независимо от ее типа. Все файлы организованы в виде древовидной иерархической структуры (дерева). Корнем этого дерева является ***корневой каталог*** (root directory), имеющий имя «/». В отличие от файловых систем Windows, где на каждом носителе создается свой корневой каталог, в UNIX-подобных ОС все файловое пространство объединено в единое дерево каталогов, корнем которого является каталог «/». Полное имя файла содержит ***путь*** – список каталогов, которые необходимо пройти, чтобы достичь файла, и, в отличие от Windows, не содержит идентификатора устройства (дискового накопителя, CD-ROM и др.), на котором он фактически хранится.

Можно сказать, что в файловой структуре UNIX-подобных ОС не диск содержит каталоги, а каталоги могут содержать диски. Существуют процедуры монтирования и размонтирования устройств, которые позволяют к единому дереву каталогов подсоединять и отсоединять разные устройства (разделы жесткого диска или съемные устройства). Точками монтирования (то есть местами, куда подключаются устройства) служат каталоги. Данные, содержащиеся на подключаемом устройстве, становятся доступными внутри этого каталога.

В Linux ***имена файлов*** и каталогов могут быть длиной не более 256 символов, и могут содержать любые символы, кроме «/». Нужно помнить, что Linux различает прописные и строчные буквы в именах файлов и каталогов. В файловой системе Unix-подобных ОС не обязательно указывать ***расширение*** (часть имени файла, отделенная точкой и характеризующая тип содержащихся в файле данных). Однако многие пользователи часто указывают его, это не является ошибкой.

В Linux, как во всех UNIX-подобных ОС, используются общепринятые имена основных каталогов. Это существенно

облегчает работу в операционной системе, ее администрирование и переносимость. Эта структура используется в работе системы, например при ее инициализации и конфигурировании, при работе почтовой системы и системы печати. Нарушение этой структуры может привести к неработоспособности системы или отдельных ее компонентов.

Описание основных каталогов файловой системы

Корневой каталог «/» является основой любой файловой системы UNIX.

/bin – в этом каталоге находятся наиболее часто используемые команды и утилиты системы.

/boot – этом каталоге находятся файлы, необходимые для загрузки ОС.

/dev – содержит специальные файлы устройств, являющиеся интерфейсом доступа к периферийным устройствам. Файлы используются не только для хранения данных. Файлы определяют привилегии пользователей, обеспечивают доступ к периферийным устройствам компьютера, включая диски, накопители на магнитной ленте, CD-ROM, принтеры, терминалы, сетевые адаптеры и даже память. Для приложений UNIX-подобных ОС доступ к дисковому файлу «неотличим» от доступа к принтеру. ***Специальный файл устройства*** обеспечивает доступ к физическому устройству. Доступ к устройствам осуществляется путем открытия, чтения и записи в специальный файл устройства.

/proc – в этом каталоге все «виртуальные» файлы, которые располагаются не на диске, а в оперативной памяти. В этих файлах содержится информация о программах (процессах), выполняемых в данный момент в системе.

/root – домашний каталог администратора системы – пользователя root.

/sbin – каталог для важнейших системных утилит.

/etc – в этом каталоге находятся системные конфигурационные файлы и многие утилиты администрирования. Среди наиболее важных файлов – скрипты инициализации системы.

/lib – в каталоге находятся библиотечные файлы языка С и других языков программирования.

/lost+found – каталог «потерянных» файлов. Ошибки целостности файловой системы, возникающие при неправильном останове ОС или аппаратных сбоях, могут привести к появлению «безымянных» файлов – структура и содержимое файла являются правильными, однако для него отсутствует имя в каком-либо из каталогов. Программы проверки и вос-

становления файловой системы помещают такие файлы в каталог /lost+found под системными числовыми именами.

/mnt – стандартный каталог для монтирования внешних файловых систем к корневой файловой системе для получения единого дерева каталогов. Чтобы работать с какой-либо существующей на внешнем носителе файловой системой, пользователь должен ее смонтировать («привить» в виде ветви к общему дереву каталогов). Завершив работу с файловой системой, необходимо ее размонтировать. Монтирование и размонтирование файловой системы выполняются командами (утилитами) *mount* и *umount*.

/media – в этот каталог монтируются съемные носители: компакт-диски, флешки, внешние жесткие диски и т. п. В современных версиях Linux пользователь вручную не монтирует файловую систему подключаемого накопителя к коренной файловой системе, это делается автоматически. В каталоге /media создается каталог, имя которого совпадает с именем носителя. Этот каталог – точка монтирования внешней файловой системы подключенного носителя к коренной файловой системе.

/home – каталог для размещения домашних каталогов пользователей.

/usr – в этом каталоге находятся подкаталоги различных сервисных подсистем – системы печати, электронной почты и т. д. (/usr/spool), исполняемые файлы утилит и прикладные программы (/usr/bin), электронные справочники (/usr/man) и т. д.

/tmp – этот каталог предназначен для временных файлов, в которых программы хранят промежуточные данные, необходимые для работы. После завершения работы программы временные файлы удаляются.

/var – здесь размещаются те данные, которые создаются в процессе работы разными программами и предназначены для передачи другим программам и системам) или для сведения системного администратора (системные журналы). В отличие от каталога /tmp сюда попадают те данные, которые могут понадобиться и после того, как создавшая их программа завершила работу.

Особенности имени файла в UNIX-подобных ОС

Каждый файл имеет связанные с ним *метаданные* (хранящиеся в *индексных дескрипторах* – inode), содержащие все характеристики файла и позволяющие ОС выполнять операции, запрошенные прикладной задачей: открыть файл, прочитать или записать данные, создать или удалить файл. В частности, метаданные содержат указатели на дисковые блоки хранения данных файла. Имя файла в файловой си-

стеме является указателем (жесткой ссылкой) на его метаданные, в то время как метаданные не содержат указателя на имя файла.

Фактически каталог – это файл, содержащий имена находящихся в нем файлов, а также указатели на метаданные. Каталоги определяют положение файла в дереве файловой системы, поскольку сам файл не содержит информации о своем местонахождении. По существу каталог представляет собой таблицу, каждая запись которой соответствует некоторому файлу. Первое поле каждой записи содержит указатель на метаданные (номер inode), а второе определяет имя файла.

Такая организация файловой системы позволяет одному файлу иметь несколько имен. Имена файлов жестко связаны с метаданными и, соответственно, с данными файла, в то время как сам файл (данные файла) существует независимо от того, как его называют в файловой системе. Такая связь имени файла с его данными называется ***жесткой ссылкой*** (hard link). С помощью команды ***ln*** можно создать жесткую ссылку – еще одно имя для файла. Например, для файла *name1* создать жесткую ссылку *name2*. С точки зрения пользователя – это два разных файла. Атрибуты файлов *name1* и *name2* абсолютно одинаковые, отличие только в имени. Фактически эти имена ссылаются на одни и те же метаданные и блоки данных. Поэтому изменения, внесенные в любой из этих файлов, затронут и другой. Удаление одного из файлов не приведет к удалению самого файла, то есть его метаданных и данных. Фактически файл будут удален тогда, когда будет удалена последняя жесткая ссылка на метаданные.

Жесткие ссылки используются, например, для того, чтобы хранить под разными именами одну и ту же команду (выполняемый файл) командного интерпретатора.

Существуют также и *символьные ссылки* (symbolic link) – аналог ярлыков в Windows. Например, для каталога нельзя создать жесткую ссылку, поэтому для того, чтобы обращаться к нему с другим именем, необходимо создать символьную ссылку. Символьная ссылка – это файл, в котором содержится имя другого файла. Символьная ссылка – это отдельный объект, поэтому удаление ссылки не оказывает никакого влияния на сам файл с данными, более того, файл с данными может быть удален, а ссылка на него будет существовать, и лишь при попытке вызвать по этой ссылке сам файл система выдаст сообщение об ошибке.

2.3.7. Установка программного обеспечения в ОС Linux. Пакеты

В Linux каждый компонент системы или прикладной программы представлен в виде пакета. Любая, даже самая простая программа при работе использует дополнительные файлы, содержащие различные ресурсы (библиотеки, конфигурационные файлы, файлы-дырки и другие программы). Поэтому для полноценной работы программы, необходимо помимо главного исполняемого файла обеспечить наличие в системе всех нужных файлов с ресурсами, которых может быть очень много.

Все файлы, необходимые для работы программы, объединяются в архивы – пакеты. Специальная программа – менеджер пакетов – занимается установкой, удалением, обновлением и проверкой пакетов. Менеджер пакетов определяет, какие пакеты нужны для установки программы, проверяет, какие пакеты уже были установлены в системе другими программами, отслеживает, чтобы в разных пакетах не оказалось файлов с одинаковым именем и путем, то есть чтобы файл одного пакета не был заменен файлом другого пакета при установке. Менеджер пакетов скачивает их из специальных хранилищ – репозиториев.

Наиболее известный и популярный менеджер пакетов называется APT (Advanced Package Tool). В файле /etc/apt/sources.list хранится список доступных APT репозиториев. Для каждого дистрибутива Linux доступны тысячи пакетов, и APT дает возможность поиска нужного пакета среди доступных в репозитории пакетов. Менеджеры пакетов позволяют выполнять и комплексные обновления всей системы.

В графических оболочках управление пакетами осуществляется еще проще. Например, во многих дистрибутивах используется менеджер пакетов Synaptic, который в KDE можно запустить следующим образом:

KDE (Пуск) – Система (или Приложения – Настройка) – Менеджер пакетов Synaptic. С помощью Synaptic (рис. 2.6) можно устанавливать, удалять, настраивать и обновлять пакеты в системе, просматривать списки доступных и установленных пакетов, управлять репозиториями и обновлять систему до новой версии.

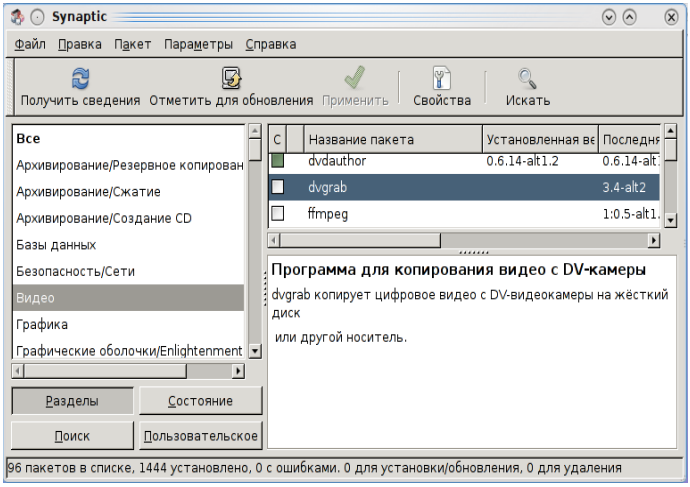


Рис.2.6. Менеджер пакетов Synaptic

В левом окошке перечислены разделы (должна быть нажата кнопка «Разделы» под этим окном), то есть можно выбрать, по какой теме отображать доступные пакеты, если выбрана категория «Все», то отображаются все пакеты.

В правом верхнем окошке – список доступных пакетов с указанием их версии, который загружается с репозитория. Если пакет установлен, то вы увидите версию установленного пакета. Если в репозитории доступна более новая версия установленного пакета, то вы можете сделать его обновление.

В правом нижнем окне дается краткая информация о том пакете, на котором установлен курсор в правом верхнем окне.

Для **установки пакета:** щелкните по кнопке «Получить сведения» для того, чтобы скачать список самых последних версий доступных в репозитории пакетов. Затем в правом верхнем окне выберите нужный пакет и в контекстном меню (щелчок правой кнопкой мыши) выберите «Отметить для установки». Если пакет требует установки другого пакета, то появится диалоговое окно с изменениями, которые будут сделаны, если продолжить установку. Чтобы запустить процесс установки, щелкните кнопку «Применить».

Для **удаления пакета:** в правом верхнем окне выберите нужный пакет и в контекстном меню (щелчок правой кнопкой мыши) выберите «Отметить для удаления». Если выбрать «Отметить для полного удаления», то удалится не только выбранный пакет, но и все зависимости, то есть все связанные с

ним пакеты. Далее появится диалоговое окно с перечнем изменений, которые будут произведены. Чтобы запустить процесс удаления, щелкните кнопку «Применить».

Вопросы для самоконтроля

1. Перечислите основные характеристики Linux.
2. Что представляет собой дистрибутив Linux, приведите примеры дистрибутивов.
3. Какой интерфейс пользователя в Linux?
4. Опишите файловую систему Linux.

2.3.8. Работа в командном интерпретаторе shell

Командный интерпретатор ОС Linux. Основные команды

Работа в этой ОС начинается с регистрации пользователя. Поэтому в строке «Имя пользователя» (Login) нужно ввести свое учетное имя, а в строке «Пароль» (Password) – пароль (эту информацию можно узнать у администратора).

Автоматически запускается командный интерпретатор. Команды записываются в строке-приглашении, в которой указывается учетное имя пользователя, имя машины и имя текущего каталога. Вид строки-приглашения можно настраивать, поэтому вы можете увидеть и другую информацию в этой строке. Заканчивается стока знаком $, если зарегистрировался обычный пользователь, и # – при регистрации суперпользователя root.

[student@h203 ~]$

Значит, вошел в систему пользователь student, работает на машине h203, текущим является домашний каталог пользователя, который обозначается знаком «~». Домашний каталог всегда находится в /home и совпадает с именем пользователя, при входе в систему всегда устанавливается текущим.

Виртуальные консоли

Работая в Linux, можно переключаться с одной консоли на другую (переключать экраны). Это позволяет одному пользователю одновременно регистрироваться в системе под разными именами, работать одновременно с разными программами и т. д.

Для переключения используются сочетания клавиш Alt+F1 (1-ая консоль), Alt+F2 (2-ая консоль и т. д.)

Примечание: Если запущен графический режим, то переключение осуществляется сочетанием клавиш: Ctrl + Alt + FN (N –номер консоли от 1 до 12). Обычно графические консоли имеют номер от 7 до 9.

В графическом режиме можно использовать приложение Терминал (Konsole) для работы с командной строкой.

Общий формат команд

Команда записывается в строке-приглашении сразу после знака $ (или #).

Общий формат команд:

имя_команды -f1…-fn A1 A2 … An

-f1…-fn – флаги (ключи, ощии – настройки команды), допускается и объединение нескольких флагов с одним «минусом». Флаги в некоторых командах многобуквенные (полнобуквенные), тогда перед ними указывается два знака «минус», например, --help.

A1 A2 … An – аргументы (имя каталога, файла и т. д.)

Обратите внимание: Linux различает строчные и прописные буквы!

Получение справочной информации о командах

man имя_команды

Например: man cd

Man от manual – руководство, подробное описание команды, возможных флагов, примеры. Чтобы выйти из справки и перейти в командный режим, нажмите клавишу «Q».

В основном, все команды имеют и короткое описание, которое вызывается следующим образом:

имя_команды -- help

Команды для работы с каталогами

Напомним, что в UNIX/Linux файлы организованы в виде древовидной структуры (дерева).

Файлы объединяются в группы – каталоги или папки. Каталог может помимо файлов содержать и вложенные каталоги (подкаталоги).

Каждый файл имеет имя, определяющее его расположение в дереве файловой системы. Корнем этого дерева является корневой каталог (root directory), имеющий имя «/», он содержит в себе все файлы и каталоги.

Полное имя файла содержит путь – список каталогов (ветвей), которые необходимо пройти, чтобы достичь файла. Полное имя любого файла в UNIX/Linux начинается с «/» и не содержит идентификатора устройства (дискового накопителя, CD-ROM или удаленного компьютера в сети), на котором он фактически хранится.

Например, /home/stud/gr1-3/Ivanov/text1.txt – полное имя файла text1.txt

В UNIX/Linux расширения файлов не обязательны.

Примечание: во всех командах путь указываются в том случае, если действия совершаются не с текущим каталогом.

Просмотр каталога (list): ls -ключи путь/имя_файла
ключи:

-F – (full) вывод информации о принадлежности объекта
(Система использует следующие обозначения:
* – исполняемый файл; / – каталог; @ – символьная ссылка)

-*l* – (long) длинный формат, указываются свойства и атрибуты файла

-R – вывести оглавление каталога рекурсивно вместе с оглавлениями подкаталогов

-a – (all) вывести оглавление каталога, показывать и специальные (скрытые) файлы, имена которых начинаются с точки.

Примеры:

!!! Обратите внимание, что имена файлов могут быть длиннее 8 символов и содержать точку в любой позиции. Можно даже использовать несколько точек в одном имени.

ls – вывести оглавление текущего каталога

ls / – просмотреть корневой каталог

ls /bin /home – просмотреть оглавление каталогов / bin и /home

ls -F / – вывести оглавление корневого каталога, указывая принадлежность объектов.

ls -*l* /bin – вывести в длинном формате оглавление каталога /bin

ls -R /etc – вывести и дерево подкаталогов с их содержимым.

(Для пролистывания страниц используйте Shift+PgUp, Shift+PgDn).

ls –a –F /home – вывести оглавление домашнего каталога, показывать и специальные (скрытые) файлы, имена которых начинается с точки.

!! Обратите внимание на названия «.» и «..» , которые присутствуют в каждом каталоге.

«..» – обозначение родительского каталога

«.» – обозначение самого каталога

При написании команд эти обозначения часто используются.

Узнать текущий каталог: pwd

Сменить текущий каталог: cd имя_каталога
(change directory)

Примеры:

(выполните команды, перейдя к другому каталогу, посмотрите его оглавление, используя команду ls).

cd или cd ~ – переход в домашний каталог пользователя

cd.. – переход в надкаталог

cd / – переход в корневой каталог

cd /mnt – переход в каталог /mnt

Создание нового каталога: mkdir путь/имя_каталога (make directory)

Примеры:

cd ~ – переход в домашний каталог пользователя

mkdir dir1 – создать каталог dir1 в домашнем каталоге (текущем)

mkdir dir1/dir2 – создать каталог dir2 в каталоге dir1

mkdir dir3 – создать dir3 в текущем каталоге (домашнем)

С помощью команды ls -R проверьте дерево подкаталогов домашнего каталога.

Удаление пустого каталога: rmdir путь/имя_каталога (remove directory)

!!! Удаляемый каталог должен быть пуст (т.е. из него должны быть удалены все файлы – команда rm и подкаталоги – команда rmdir).

Примеры:

rmdir dir3 – удалить каталог dir3 из текущего каталога (домашнего).

rmdir dir1/dir2 – удалить каталог dir2 из каталога dir1.

С помощью команды ls -R проверьте, что каталоги действительно удалены.

Команды для работы с файлами

Использование шаблонов имен файлов

Для обозначения группы файлов (при копировании, удалении и др.) используются шаблоны имен файлов.

В шаблонах используют символы «*» и «?».

«*» – заменяет любое количество любых символов.

Примеры:

* – файлы с любым именем (то есть все файлы)

s – все файлы, в имени которых содержится s

x* – все файлы с именем, начинающимся на x

«?» – заменяет один любой символ.

Примеры:

????? – все файлы, длина имени которых 5 символов.

??d* – все файлы, в имени которых третий символ d.

Можно в шаблонах использовать *диапазоны*.

[набор] – любой один символ из заданного набора

[^набор] – любой один символ, не заданный в наборе

Примеры:

*[a-c s] – все файлы, имя которых заканчивается на одну из букв: a, b, c, s.

[^a-d]* – все файлы, имя которых не начинается на одну из букв: a, b, c, d.

6. Узнать тип файла: file путь/имя_файла

Примеры:

file ~/* – узнать тип файлов, хранящихся в домашнем каталоге.

file /bin/* – определить тип файлов, хранящихся в каталоге bin.

7. Запуск исполняемых файлов: указать полное имя файла

или sh путь/имя_файла

Примеры:

Просмотрите каталог /usr/bin (с ключом –F) или используйте команду file, чтобы определить тип файлов в этом каталоге. В этом каталоге содержатся исполняемые файлы-утилиты. Запустите, например, файл cal, который отобразит на экране календарь.

/usr/bin/cal

Многие команды, которые мы используем, не являются встроенными в командный процессор, а представляют собой исполняемые файлы. Большинство таких полезных файлов-утилит находятся в каталогах /bin и /usr/bin.

Примечание: исполняемые файлы нельзя запускать просто по имени, если они находятся в текущем каталоге и этот каталог не входит в переменную окружения PATH, в которой содержится список каталогов, в которых командный интерпретатор будет искать указанный файл.

Создание текстового файла: cat > путь/имя_файла

Значение символа > подробно будет рассмотрено ниже (см. 17. Ввод и вывод).

Примеры:

cat > ~/dir1/mytext.txt – создание текстового файла в каталоге dir1, находящегося в домашнем каталоге.

После ввода команды наберите текст файла.

Окончание набора и закрытие файла: ctrl + D , enter.
Создайте еще несколько текстовых файлов в каталоге dir1.

Копирование файлов (copy):

cp путь/имя_файла(ов) путь/имя_каталога
(откуда и что копировать) (куда копировать)

Можно при копировании одновременно переименовывать файлы. Для этого нужно указать новое имя файла в качестве второго аргумента (куда копировать).

Примеры:

cp /bin/??? ~/dir1 — копировать файлы, длина имени которых 3 символа из каталога bin, расположенного в корневом каталоге, в dir1, находящегося в домашнем каталоге.

cp ~/dir1/* ~ – копировать все файлы из каталога dir1 в домашний каталог.

Используя команду ls, просмотрите содержимое домашнего каталога и проверьте правильность копирования.

cd /sbin – сделать текущим каталог sbin.

cp mk* ~/dir1 – копировать все файлы, начинающиеся на mk из текущего каталога (sbin) в каталог dir1 .

cd ~/dir1 – сделать текущим каталог dir1.

cp /bin/vi . – копировать файл vi из каталога bin в текущий каталог (dir1). Для обращения к текущему каталогу используется «.» (точка).

Просмотр текстового файла:

cat путь/имя_файла
less путь/имя_файла (постраничный просмотр)

Просмотрите созданный файл cat ~/dir1/mytext.txt
Чем отличается просмотр файла с помощью разных команд.
cat /usr/share/doc/HTML/index.html
less /usr/share/doc/HTML/index.html
Выход из режима просмотра less: Ctrl+z

Переименование файла (каталога) (move):

mv стар_имя_файла нов_имя_файла

Перемещение файла(ов): mv путь/имя_файла путь
(откуда) (куда)

Переименуйте созданный текстовый файл в studdoc.txt:
mv ~/dir1/mytext.txt ~/dir1/studdoc.txt
С помощью команды ls проверьте переименование файла.
Переместите studdoc.txt в свой домашний каталог:
mv ~/dir1/studdoc.txt ~

Удаление файлов: rm путь/имя_файла(ов)

rm -r имя_каталога – позволяет удалять каталог вместе со всем содержимым.

Для подтверждения удаления файла нажмите клавишу Y, для отмены N.

rm ~/dir1/* – удалить все файлы из каталога dir1.

Просмотрите каталог dir1 и проверьте удаление файлов.

Поиск файла: find каталог_поиска -ключи

Каталог_поиска – каталог, включая дерево всех подкаталогов, в которых будет производиться поиск.

-name – позволяет задать имя искомого файла или каталога

-type – определяет тип файла: f – файл, d- каталог, *l* – символьная ссылка.

-xdev – ограничить поиск одной файловой системой, не выходить за границы устройства хранения.

find /home –name dir1 – искать dir1 во всех подкаталогах каталога /home.

find ~ –type d – искать все каталоги и подкаталоги, расположенные в домашнем каталоге.

find /usr/bin –name k* – искать файлы, имена которых начинаются на k во всех подкаталогах /usr/bin.

Изменение прав доступа к файлу

При создании объектов файловой системы (файлов, каталогов и т. д.) атрибуты объекта несут информацию о том, кто и что имеет право делать с этим объектом файловой системы.

Посмотреть атрибуты файлов и каталогов поможет команда ls –*l* (ключ *l* определяет длинный формат). Список атрибутов прав доступа состоит из 10 позиций (рис. 2.7).

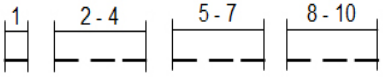


Рис. 2.7. Список атрибутов прав доступа файла

1. Тип файла (d – каталог). 2-4. Права хозяина (создателя) файла (rwx). 5-7. Права группы создателя файла (rwx). 8-10. Права посторонних (всех остальных) (rwx).

R – чтение, w – запись, x – выполнение

Например, -rwxr-xr-x означает

1. дефис значит, что это обыкновенный файл.

2-4. rwx – на данный файл владелец имеет полные права (чтение, запись, выполнение).

5-7. r-x – пользователи, входящие в группу владельца, имеют права на чтение и выполнение.

8-10. r-x – все остальные – права на чтение и выполнение.

Изменять права доступа к файлам и каталогам можно, используя следующую команду:

chmod права имена_файлов

Право изменять права есть лишь только у владельца файла и суперпользователя. Режим доступа можно указывать как в виде восьмеричного числа, так и виде мнемообозначения.

Обозначения: u (user) – для пользователя (владельца), g (group) – для группы, o (other) – для остальных пользователей, a (all) – для всех.

Примеры:

chmod go–w myfile – для файла запретить доступ на изменение/запись для группы и остальных пользователей.

chmod a+x file1 – всем разрешить выполнение файла file1, то есть сделать его выполняемым.

Восьмеричное представление атрибутов:

В рассмотренном выше примере -rwxr-xr-x вместо букв подставим 1, в противном случае, если атрибут не установлен – 0. Получим:

0 111 101 101 переведем каждую тройку в десятичную систему счисления: 0 7 5 5.

Поскольку значение полученного числа в каждой позиции может быть только от 0 до 7, то говорят, что атрибуты заданы в восьмеричном представлении.

Команда chmod в этом случае запишется следующим образом:

chmod 755 file2

Создание жесткой ссылки: ln имя_файла имя_ссылки

Жесткая ссылка фактически второе название физического файла на диске, так как указывает на тот же, что и у исходного файла индексный дескриптор.

Создайте текстовый файл text в домашнем каталоге.

Создадим жесткую ссылку hltext на этот файл:

```
ln ~/text ~/hltext
```

С помощью команды ls -i можно убедиться, что оба файла имеют один и тот же индексный дескриптор (см. число перед именем файла).

```
ls -i ~/text~/hltext
```

Создайте еще одну жесткую ссылку hltext2 на text.

Команда ls *-l* позволяет увидеть число жестких ссылок на файл (см. число во второй колонке, следующее за правами доступа к файлу). В нашем случае у всех созданных файлов

количество жестких ссылок будет 3. И все эти файлы будут иметь один и тот же индексный дескриптор.

Обращаясь к hltext или text, мы фактически обращаемся к одному и тому же файлу. Поэтому, если мы меняем что-то в одном файле, эти же самые изменения произойдут в другом файле.

Используя команду cat >> hltext допишите в hltext несколько строчек. Затем просмотрите содержимое файла text, произошли ли изменения в этом файле?

При удалении файла, на самом деле удаляется только одна ссылка на файл. Введете команду:

rm ~/text

Просмотрите домашний каталог. Удалилась только ссылка, имеющая имя text, а hltext и hltext2 по-прежнему существуют. Файл только тогда действительно удаляется, когда на него больше нет ссылок. Обычно файлы имеют только одну ссылку, так что команда rm действительно приведет к удалению файла. Однако, если файл имеет много ссылок, применение rm приведет только к удалению одной из них; для того, чтобы удалить файл, вы должны удалить все ссылки на этот файл.

Создание символьной ссылки: ln –s имя_файла имя_ссылки

Создадим символьную ссылку sltext на hltext.

ln –s ~/hltext ~/sltext

Используйте команду ls –i, чтобы убедиться, что эти два файла имеют различные файловые дескрипторы.

Обратите внимание на вывод команды ls *–l*, свойства файлов sltext и hltext отличаются, что еще раз доказывает, что это разные объекты файловой системы, кроме того при символьной ссылке всегда указывается файл, на который она ссылается.

Ввод и вывод. Перенаправление ввода и вывода

Каждый процесс в Linux получает при старте три потока данных (специальные файлы). Первый открыт на чтение и является стандартным вводом процесса stdin (процесс из него берет данные, читает), по умолчанию это клавиатура. Второй поток открыт на запись и называется стандартным выводом процесса stdout (процесс в него записывает данные), по умолчанию это экран. Третий поток данных предназначен для вывода сообщений об ошибках и называется стандартный вывод ошибок.

Рассмотрим команду cat, которая берет данные с потока ввода и передает их на поток вывода, то есть с stdin на stdout.

Введите команду cat. Поскольку не указаны потоки ввода и вывода, то они считаются стандартными, то есть ввод с кла-

виатуры, вывод на экран. Поэтому система будет ожидать ввода текста, а после сразу же выводить его на экран.

В команде: cat имя_файла – поток ввода указан из файла, поэтому данные будут читаться из файла и передаваться на стандартный поток вывода – экран.

Оболочка shell дает возможность перенаправлять стандартные потоки, например, вывод делать не на экран, а в файл, чтобы данные были записаны и сохранены. Для этого используются символы « >» и « <».

> – перенаправление стандартного вывода

< – перенаправление стандартного ввода

cat > textfile – поток ввода не указан, следовательно ввод с клавиатуры, а поток вывода перенаправлен в файл. Таким образом, данные с клавиатуры записываются в файл.

Просмотрите созданный файл.

Если файл уже существует, то командная оболочка запишет его заново. Чтобы сохранить информацию в этом файле и добавить новую с клавиатуры, используются символы >>.

Добавьте в созданный файл textfile еще данные:

cat >> textfile

Снова просмотрите его.

cat < textfile > textfile2 – эта команда читает данные из файла textfile и записывает их в файл textfile2.

Поиск строк в файле, соответствующих регулярному выражению:

grep –ключи шаблон файлы

Работа с текстом – одна из сильных сторон Linux, так как система содержит большое количество команд, выполняющих различные преобразования с текстом. Рассмотрим наиболее популярные из них.

Регулярные выражения – система синтаксического разбора текстовых фрагментов по формализованному шаблону, основанная на системе записи образцов для поиска. Общая задача механизма регулярных выражений – находить или не находить совпадения строки или ее части с заданным шаблоном.

Ключи:

–v – выводить только те строки, которые не соответствуют регулярному выражению (шаблону поиска);

–o – выводить только ту часть строки, которая совпала с регулярным выражением;

–l – выводить только имена файлов, содержащих шаблон поиска, без вывода самих строк;

–c – выводить только количество найденных строк, соответствующих регулярному выражению;

–n – указывать номера найденных строк;

–i – поиск, нечувствительный к регистру;

–r – рекурсивный поиск во всех файлах и подкаталогах.

Примеры:

grep –*l* “rm –r” cmd create delcreate

В перечисленных файлах (cmd, create, delcreate) ищет подстроку «rm –r», то есть мы хотим определить, в каком из сценариев (командных файлов) содержится эта команда.

grep –c “rm –r” cmd create delcreate

Конвейеры

Нередко возникает ситуация, когда нужно обработать вывод одной команды какой-нибудь другой командой. То есть перенаправляется вывод одной команды не в файл, а на вход другой команды. Такой способ передачи называется конвейером и обозначается символом |.

С помощью конвейера можно объединять в цепочку много команд:

Команда1 | команда2 | … | командаN

Например, нужно отсортировать оглавление каталога в алфавитном порядке, то есть результат команды ls передать команде sort.

ls | sort

Чтобы просматривать постранично длинный список файлов, выдаваемой командой ls, нужно использовать конвейер и обратиться к команде less, которая постранично выводит текст.

ls /usr/bin | sort | less

Создание командных файлов (скриптов)

Командный файл (скрипт или сценарий) – это текстовый файл, состоящий из команд интерпретатора. При запуске этого файла последовательно выполняются все команды, содержащиеся в нем.

Как файл его можно создать командой cat или использовать текстовый редактор, например, vi.

Если вы работаете в графической оболочке, то воспользуйтесь, например, текстовым редактором Leafpad или KWrite.

Создадим файл с именем cmd (в домашнем каталоге), и запишем в него следующие команды:

```
#! /bin/sh                          (выполнение файла – в shell)
echo  “How do you do!” (вывод строки на экран, приветствие)
date                                (вывод текущей даты)
```

pwd (вывод текущего каталога)
ls (вывод оглавления текущего каталога)

Примечание: в скобках содержится комментарий команды, включать его в командный файл не нужно.

Сделайте этот файл исполняемым командой: chmod +x ~/cmd.

То есть в правах доступа к файлу для всех пользователей разрешите выполнение этого файла (+x).

Теперь чтобы запустить этот файл используйте команду: ~/cmd

Итак, для создания командного файла:

1. Запустите текстовый редактор, создайте текстовый файл.
2. Последовательно запишите команды в этом файле, располагая каждую команду на отдельной строке.
3. Сохраните созданный файл, сделайте его исполняемым, применив команду:

chmod +x имя_файла.

4. Запустите созданный файл и проверьте правильность выполнения команд. В случае нахождения ошибки, в текстовом редакторе внесите изменения в командный файл, сохраните его и проверьте еще раз.

Задания.

А) Создайте в домашнем каталоге командный файл create, который выполняет следующие операции:

1. В домашнем каталоге создает каталоги katalog1 и katalog2.
2. В каталоге katalog2 создает katalog3 и katalog4.
3. В каталог katalog1 копирует все файлы, длина имени которых 5 символов из каталога /bin.
4. Копирует файлы, содержащие в имени одну из букв a-c, из каталога /bin в каталог katalog4.
5. В каталоге katalog2 создает текстовый файл user.txt, в котором будет храниться имя пользователя (свое имя пользователь введет с клавиатуры после запуска командного файла).
6. В каталоге katalog1 создает файл filelist, в который записывается список файлов, находящихся в домашнем каталоге пользователя и его подкаталогах, отсортированный по алфавиту.
7. В каталоге katalog1 создается жесткая ссылка на файл user.txt.
8. В каталоге katalog2 создается ссылка на каталог /usr/bin.
9. Организует поиск файла, результат поиска помещает в файл file_find.
10. Перемещает user.txt в katalog3.

Б) Создайте в домашнем каталоге командный файл delcreate, который удаляет все файлы и каталоги, созданные файлом create.

2.4. Сервисные программы

Сервисная программа, или утилита (utility или tool) – программный продукт, предназначенный для расширения возможностей операционных систем и предоставления набора дополнительных услуг.

Компьютерные утилиты можно разделить на три группы: утилиты сервисного обслуживания компьютера, утилиты расширения функциональности и информационные утилиты.

Самые необходимые для работы утилиты входят в состав операционной системы. Это программы, которые позволяют обслуживать диски (проверять, сжимать, дефрагментировать и т. д.), выполнять мониторинг работы ОС и основных устройств компьютера, делать оптимизацию системы, резервирование данных и их восстановление и др.

В Windows основной набор утилит находится в папке Программы – Стандартные – Служебные.

В Linux утилиты также входят в состав системы, но их набор зависит от конкретного дистрибутива. Не все утилиты имеют графический интерфейс, так как традиционно входили в состав системы еще появления графического интерфейса и запускались в консольном режиме.

2.4.1. Обслуживание дисков

Деление на разделы и форматирование

Для того чтобы компьютер мог хранить, читать и записывать информацию, жесткий диск предварительно должен быть размечен. На нем с помощью соответствующих программ создаются разделы – это и называется «разбить жесткий диск». Без этой разметки на жесткий диск не удастся установить операционную систему (обычно предлагается произвести разметку диска во время установки ОС, в этом случае программа разметки диска запускается автоматически). Существует много программ для разделения диска на разделы. При установке Windows используется **fdisk**, однако эта программа не имеет графического интерфейса. После установки Windows для создания и изменения существующих разделов можно воспользоваться оснасткой «Управление дисками».

В Linux используются **sfdisk** или **fdisk**, не имеющие графического интерфейса или **GParted c** графическим интерфейсом.

После разметки диска созданные разделы необходимо отформатировать и разместить какую-либо подходящую файловую систему.

В Windows это делается командой **format** или при помощи команды ***Форматировать*** из контекстного меню диска в файловом менеджере. Оснастка «Управление дисками» также позволяет отформатировать разделы диска.

В Linux для форматирования используется **mkfs** или **GParted** с графическим интерфейсом.

Тестирование диска

При сбоях в работе компьютера, зависаниях ОС и по другим причинам системные области на диске могут быть изменены. Для выявления и исправления этих нарушений производится проверка нарушений файловой системы, проверка физической поверхности диска на наличие «сбойных секторов» и перенос данных из этих участков в безопасные.

В Windows: ***Проверка диска*** (Checkdisk) в свойствах диска в контекстном меню, вкладка Сервис – Проверка диска.

В Linux: **fsck** в консольном режиме, утилита запускается автоматически при загрузке ОС.

Дефрагментация диска

Для записи файлов на диск ОС выделяет участки памяти (кластеры), которые не обязательно расположены друг за другом. При удалении и перезаписи файлов на диске образуется много пустых мест, а файлы оказываются фрагментированными, это замедляет работу.

Приведем пример. Представим, что у нас данные находятся в полном порядке и идут друг за другом. Затем мы удалили какой-то файл и записываем на его место больший по размеру файл. Запись пойдет с первого пустого фрагмента, и системе придется записать этот файл, разбив его на две части. А если файл находится в результате такой записи в конце диска, то это приведет к потере быстродействия. Если же фрагментирован win386.swp (область подкачки, свопинга) – файл, с помощью которого Windows устраняет недостаток ОЗУ, то потеря быстродействия будет еще большей из-за очень частых операций записи/чтения этого файла.

Поэтому периодически нужно устранять фрагментацию диска – дефрагментировать его. В Windows: ***Дефрагментация диска*** (Defrag), в свойствах диска, вкладка Сервис – Дефрагментация диска. В Linux: используемые файловые системы не фрагментируют диск, поэтому нет программ дефрагментации диска.

2.4.2. Сводная информация о компьютере и системе

Производительность компьютера зависит от технических характеристик составляющих его устройств. В случаях возникновения сбоев в работе компьютера или перед его модернизацией полезно осуществить тестирование различных устройств компьютера.

В Windows: программа ***Сведения о системе*** дает сводную информацию об устройствах компьютера. ***Системный монитор*** позволяет в фоновом режиме отслеживать параметры функционирования компьютера, получать информацию о возникших проблемах.

В Linux входят традиционные для Unix-подобных ОС утилиты, связанные с мониторингом системы, которые не имеют графического интерфейса: uname – сведения о системе, uptime – загрузка системы, free – объем свободной памяти, xload – статистика загрузки системы (в графическом виде) и др.

Также в графических средах KDE и Gnome имеются служебные программы, которые позволяют пользователю в графическом формате получать сведения о системе и процессах.

2.4.3. Оптимизация системы

Периодически необходимо оптимизировать ОС, так как после продолжительной работы Windows в реестре накапливается ненужная информация, которая тормозит работу системы. Проблема решается следующими способами.

- Удаление программ: например, Revo Uninstaller бесследно удаляет самые «живучие» приложения.
- Можно использовать программы для съемных носителей, которые позволяют избежать засорения реестра (например, свободное ПО www.portablefreeware.com).
- Удаление ненужных программ и служб из автозагрузки.

В строке «Выполнить» наберите: msconfig для «настройки системы», возможность отредактировать boot.ini, win.ini, автозагрузку, службы и т. д.

Проверка процессов автозагрузки часто позволяет выявить программы, которые не особо нужны, а иногда и вредные программы, которые могут вредить компьютеру. Если опытный пользователь может справиться с такой задачей, как мониторинг процессов автозагрузки, то не очень опытный может испытывать затруднения. Тогда можно использовать свободную утилиту OSAM – Autorun Manager.

• Удаление временных файлов с жесткого диска: чистка диска или утилита CCleaner (свободное ПО).

• Чистка реестра, например, бесплатная утилита Wise Registry Cleaner – оптимизирует системный реестр Windows, удаляя потерянные ключи и ненужные поля, тем самым сокращая его длину. Создает резервную копию реестра, сканирует реестр и выводит перечень всех записей в виде таблицы, если установлена опция «безопасно», то будут выведены лишь те ключи, удаление которых не вызовет проблем в системе. Позволяет устанавливать режим автоматической чистки.

2.4.4. Сервисные программы, обеспечивающие резервирование информации и восстановление данных

Данные – самое ценное из того, что есть на компьютере. Если компьютер используется для создания документов, ценность труда, вложенного в их создание, обычно превышает стоимость самого компьютера.

В аварийной ситуации рабочие программы можно переустановить, если сохранился дистрибутив (а он должен храниться), восстановление программного обеспечения займет максимум несколько дней, а восстановление данных, которые, быть может, создавались годами, может оказаться уже невозможным.

Искажения или разрушение данных, хранящихся на внешних носителях, может происходить по следующим причинам:

• несанкционированные действия пользователей или вирусных программ;

• сбои и отказы в работе аппаратно-программного обеспечения;

• физическое разрушения носителя.

Устранить последствия потери данных можно только в том случае, если утерянная информация была зарезервирована.

Резервирование данных бывает:

• фоновое – при каждой записи на диск данные параллельно записываются еще и на другой носитель. Эта функция поддерживается ОС. Обеспечивает автоматическое восстановление данных в случае отказа носителя или возникновения дефектов, но не в случае некорректной работы программы или пользователя.

• периодическое – новые данные с рабочих носителей дублируются на резервные носители через определенные интервалы времени.

Наибольшая эффективность достигается при сочетании этих способов резервирования.

Для предотвращения потери данных, хранящихся на компьютере, рекомендуется придерживаться ***принципа раздельного хранения данных и программ***: все, что создается, хранится и обрабатывается на компьютере не должно храниться в тех же папках, что и сами приложения (то есть нельзя хранить документы, созданные, например, редактором WORD в папке …\WORD, где находится сам текстовый редактор. При переустановки приложения или ОС можно потерять все, что хранилось в этих папках).

Если на одном компьютере работают несколько человек, то каждый пользователь создает для себя одну основную папку и хранит свои документы в этой папке.

Виды резервирования

1. Резервирование информации, хранящейся в файловых областях может быть выполнено двумя способами:
    - Обычное копирование. Имеет ряд неудобств: большое количество памяти, необходимой для хранения копий, трудности в работе с копиями.
    - Архивация. Используя специальные программы-архиваторы, позволяющие сохранять данные в более компактном (сжатом) виде.
2. Резервирование информации, хранящейся в нефайловых областях и энергонезависимой памяти компьютера (CMOS) выполняется с помощью специальных программ, которые сохраняют ее в виде файла.
3. Резервирование ПО и средств восстановления данных. Для восстановления ПО необходимо иметь дистрибутив установленных программ. «Ремонтный набор» должен включать в себя средства, которые позволят:
    - Загрузить ОС.
    - Восстановить системные данные.
    - Найти и устранить ошибки на дисках, выполнить разбиение диска на разделы.
    - Проверить на наличие вирусов.
    - Разархивировать данные.

Организационная схема резервирования данных

Прежде, чем приступить к работе со средствами архивации данных, необходимо разработать организационную схему резервного копирования и затем придерживаться ее постоянно.

Необходимо принять решение:

1) О числе резервных копий. Обычно используют две копии.

2) О порядке ротации (круговорота) копий. Пока одна копия хранится в удаленном месте, происходит работа с другой копией. Обычно применяют еженедельную ротацию.

3) Об обновлении копий. Архивация может быть полной или частичной при еженедельной ротации полную архивацию проводят раз в неделю, а частичную – каждый день.

Правила резервирования

1. Резервирование необходимо выполнять на внешнем носителе, удовлетворяющим требованиям:
   - носитель используется только для хранения копий;
   - носитель должен быть защищен от несанкционированного доступа;
   - носитель должен быть доступен в случае необходимости.
2. Перед резервированием и в случае восстановления необходим антивирусный контроль.
3. Зарезервирована должна быть вся ценная информация.
4. Необходимо подготовить программные средства восстановления работоспособности системы.

В Windows есть утилита **«Восстановление системы»**, которая создает на жестком диске точки восстановления – резервные копии конфигурационных данных системы. Работает в автоматическом режиме, создает точки отката при установке программ, после обновления системы. Можно сделать точку восстановления и вручную. При сбое операционная система использует эти данные.

В Vista и Windows 7 добавлено резервное копирование файлов, резервное копирование данных компьютера.

Свободная утилита **Erunt** – для резервного копирования реестра.

Свободная утилита **Cobian Backup** достаточно удобна и относительно проста. Имеет встроенный архиватор, возможность копирования данных на удаленном сервере и планировщик, с помощью которого можно автоматизировать процесс создания резервных копий.

Acronis True Image Home (коммерческая) – одна из лучших программ для резервного копирования. На диске создается зона безопасности, куда размещаются резервные копии, можно также копировать на CD, DVD, сетевые диски и т. д.

Полное копирование, только изменения с момента последней копии, а также изменения после последнего полного резервного копирования. Также позволяет восстанавливать систему даже в случаях, когда она перестает запускаться. Ути-

лита позволяет делать образы разделов диска, образы отдельных папок, образы параметров установленного ПО.

Вопросы для самоконтроля

1. Каково назначение сервисных программ?
2. В чем заключается обслуживание диска сервисными программами? Какие повреждения диска могут быть устранены, а какие нет?
3. Что представляет собой дефрагментация диска? Для чего и в каких случаях ее необходимо выполнять?
4. Для чего применяется архивация данных?
5. Что представляет собой архивный файл? Какие основные операции с архивным файлом?
6. Приведите примеры программ- архиваторов.

2.5. Вредоносные программы. Средства защиты компьютера

Вредоносные программы (*malware от „malicious software“*) – это программное обеспечение, специально созданное для того, чтобы причинять ущерб отдельному компьютеру, серверу или компьютерной сети.

2.5.1. Типы вредоносных программ

Выделяют несколько типов программ, выполняющих какие-либо деструктивные действия и причиняющих ущерб пользователям:

- троянские программы (троянцы, троянские кони);
- сетевые черви;
- вирусы;
- прочие вредоносные программы.

Троянцы – программы, которые при запуске выполняют скрытые деструктивные действия (например, кодирование информации на диске, удаление файлов, использование ресурсов машины в злонамеренных целях, кража информации и другие несанкционированные операции). Термин «троянец» произошел от троянского коня – деревянной фигуры, с помощью которой, согласно легенде, греки обманным путем проникли в город Трою и захватили его. От вирусов и червей троянские программы отличает неспособность к самостоятельному распространению. Такие программы не подлежат лечению, их нужно удалять. Как правило, троянцы устанавливаются на компьютере скрытно и доставляют свой вредоно-

сный «груз» без ведома пользователя. Существует множество видов троянских программ, и каждая из них предназначена для выполнения конкретных вредоносных функций.

Троянские утилиты удаленного администрирования (backdoors) очень распространены. Они похожи на «легальные» утилиты администрирования, однако позволяют без ведома пользователя удаленно управлять компьютером (от выключения компьютера до манипуляций с файлами). Троянские программы данного класса являются одними из наиболее опасных, так как превращают компьютер в зомби-машину. Злоумышленники, удаленно управляя зараженными компьютерами, создают ботнеты. *Ботнет* – это сеть компьютеров, зараженных вредоносной программой, которая позволяет киберпреступникам удаленно управлять зараженными машинами. Ботнеты могут использоваться злоумышленниками для решения криминальных задач разного масштаба: от рассылки спама до DDoS-атак (Distributed Denial of Service – распределенная атака типа «отказ в обслуживании»). В ходе такой атаки с зараженных машин создается поток ложных запросов на атакуемый сервер в сети. В результате сервер из-за перегрузки становится недоступным для пользователей. За остановку атаки злоумышленники, как правило, требуют выкуп. DDoS атаки могут использоваться и как средство политического воздействия. В этих случаях атакуются, как правило, серверы государственных учреждений или правительственных организаций.

Шпионские программы (trojan-spy) осуществляют шпионаж за пользователем: записывание информации, набранной с клавиатуры, кража паролей, информации о системе, файлов, номера счетов, снимков экрана и т. д.

Троянские загрузчики (trojan-downloader) и установщики (trojan-dropper) занимаются несанкционированной загрузкой и установкой вредоносного программного обеспечения на компьютер-жертву.

Существуют и другие не менее опасные виды троянских программ.

Сетевые черви – это вредоносные программы, распространяющие свои копии по локальным и глобальным сетям с целью проникновения на компьютер-жертву, запуска своей копии на этом компьютере и дальнейшего распространения. Червей часто рассматривают как разновидность вирусов. Однако между червями и вирусами есть существенные различия. Червь – это компьютерная программа, самостоятельно распространяющая свой код, но не способная к заражению

других файлов. В отличие от вирусов, черви создают единственную копию своего кода на каждой машине. Червь устанавливается на компьютер-жертву и ищет возможность распространения на другие компьютеры.

Для распространения черви используют электронную почту (почтовые черви – email-worm), irc-каналы (irc-worm), файлообменные сети (p2p-worm) и другие сети, в том числе сети обмена данными между мобильными устройствами. Большинство червей распространяются в файлах (вложение в письмо, ссылка на файл и т. д.). Но существуют и черви, которые распространяются в виде сетевых пакетов.

Вирусы – это вредоносные программы, способные к самовоспроизведению путем заражения файлов, размещенных на жестком диске. Соответственно, чем дольше вирус остается незамеченным, тем больше зараженных файлов будет на компьютере. Вирусы могут распространяться как в пределах одной машины, так и передавая себя на другие компьютеры.

Основные разновидности вирусов (по среде обитания):

Файловые вирусы – наиболее известны, распространялись еще в DOS. Для своего распространения используют стандартные функции ОС. Первоначальное заражение происходит при запуске зараженной программы.

Загрузочные вирусы (бутовые) – местом их обитания является загрузочный сектор диска. Подменяя собой стандартный загрузчик ОС, такой вирус попадает в память раньше ОС. Заражение происходит в момент загрузки с зараженного диска.

Макрокомандные вирусы живут в среде макрокоманд и заражают документы MS Office. Ряд таких вирусов жизнеспособен и в среде обычных файлов.

Скриптовые вирусы – это вирусы, написанные на скриптовых языках (vbs, js, bat, php и т. д.).

2.5.2. Антивирусные программы

Наиболее эффективны в борьбе с компьютерными вирусами, троянскими программами антивирусные программы. Однако не существует антивирусов, гарантирующих стопроцентную защиту.

Типы антивирусных программ

Сканеры (детекторы)

Принцип работы антивирусных сканеров основан на проверке файлов, секторов и системной памяти и поиске в них известных и новых (неизвестных сканеру) вирусов. Для поиска известных вирусов используется сигнатура вируса – некото-

рая постоянная последовательность программного кода, специфичная для каждого конкретного вируса. Каждый сканер имеет обширную базу данных, содержащую сигнатуры известных вирусов. Однако, если вирус новый и его сигнатуры нет в вирусной базе, то сканер не сможет обнаружить этот вирус.

Во многих сканерах используются также алгоритмы «эвристического анализа» (анализа последовательности команд в проверяемом объекте), которые могут определить возможность заражения проверяемого объекта, высказать предположение о заражении.

Бывают:

- *резидентные сканеры* (мониторы), производящие сканирование «на лету», то есть постоянно проверяющие на вирусы объекты, к которым происходит обращение (запуск, открытие, создание и т. п.). В этом режиме антивирус постоянно активен, он присутствует в памяти «резидентно» и проверяет объекты без запроса пользователя;
- *нерезидентные сканеры*, обеспечивающие проверку системы только по запросу пользователя. В этом режиме антивирусная программа неактивна до тех пор, пока не будет вызвана пользователем из командной строки, командного файла или расписания работы антивирусной программы.

Как правило, резидентные сканеры обеспечивают более надежную защиту системы, поскольку они немедленно реагируют на появление вируса, в то время как нерезидентный сканер способен опознать вирус только во время своего очередного запуска.

Достоинство сканеров: универсальность.

Недостатки сканеров: большие размеры антивирусных баз и относительно небольшая скорость поиска вирусов.

CRC-сканеры (ревизоры)

Принцип работы CRC-сканеров основан на подсчете CRC-сумм (контрольных сумм) для присутствующих на диске файлов и системных секторов. Эти CRC-суммы и некоторая другая информация (длина файлов, дата их последней модификации и т. д.) сохраняются в базе данных антивируса. При последующем запуске CRC-сканеры сверяют данные, содержащиеся в базе данных, с реально подсчитанными значениями. Если информация о файле, записанная в базе данных, не совпадает с реальными значениями, то CRC-сканеры сигнализируют о том, что файл был изменен или заражен вирусом.

Достоинства ревизоров: CRC-сканеры, использующие алгоритмы, позволяют обнаружить практически 100% вирусов почти сразу после их появления на компьютере.

Недостатки ревизоров: CRC-сканеры не способны поймать вирус в момент его появления в системе, а делают это лишь через некоторое время, уже после того, как вирус разошелся по компьютеру. CRC-сканеры не могут определить вирус в новых файлах, поскольку в их базах данных отсутствует информация об этих файлах.

Блокировщики

Антивирусные блокировщики – это резидентные программы, делающие поведенческий анализ работы программного обеспечения, то есть перехватывающие «вирусоопасные» ситуации и сообщающие об этом пользователю. К «вирусоопасным» относятся вызовы, которые характерны для вирусов в моменты их размножения (вызовы на открытие для записи в выполняемые файлы, запись в boot-сектора дисков или MBR винчестера, попытки программ остаться резидентно и т. д.).

Достоинство блокировщиков: способность обнаруживать и останавливать вирус на самой ранней стадии его размножения.

Недостатки блокировщиков: большое количество ложных срабатываний, замедление работы компьютера, существование путей обхода защиты блокировщиков.

Сетевой экран (брандмауэр, firewall)

Сейчас подавляющее большинство пользователей работают в глобальных сетях. Для безопасной работы в сети разработано специальное программное обеспечение.

Файервол, или сетевой экран, или брандмауэр – это программно-аппаратный комплекс, который позволяет обеспечивать безопасность при работе в локальной сети и сети Интернет (хакерские атаки, вирусы, спам и т. д). Сетевой экран контролирует входящий и исходящий трафик на компьютере или в локальной сети и разрешает или блокирует соединения в соответствии с заданными политиками безопасности. Он обеспечивает фильтрацию на уровне приложений, то есть позволяет задавать правила для популярных приложений (браузеров, программ мгновенного обмена сообщениями и др.). Кроме того, обеспечивается пакетная фильтрация, анализируются передаваемые пакеты данных (заголовки, используемые протоколы, порты, IP-адреса и пр.) и осуществляется фильтрация пакетов в соответствии с заданными политиками безопасности.

Перечисленные выше средства защиты обязательно должны комбинироваться. Современные антивирусные пакеты обеспечивают комплексную защиту компьютера, используя сканеры, CRC-сканеры, блокировщики и сетевые экраны.

Существуют и бесплатные версии антивирусного ПО как для Windows, так и для Linux, которые можно скачать с сайтов разработчиков:

1. Антивирус Avast! Free.
2. Avira AntiVir Personal – Free Antivirus.
3. AVG Anti-Virus Free.

Есть также бесплатные версии Kaspersky и Dr.Web.

2.5.3. Правила безопасности

Вредоносное ПО попадает на компьютер через следующие каналы:

- Флеш-накопители, карты памяти, внешние жесткие диски.
- Электронная почта (вложения, ссылки на зараженные веб-сайты).
- Системы обмена мгновенными сообщениями (ссылки на зараженные файлы и веб-сайты).
- Веб-страницы (скрипты).
- Интернет и локальные сети (сетевые черви).

Чтобы защитить свой компьютер, нужно выполнять следующие несложные правила:

1. Установить на своем компьютере антивирусное ПО, обеспечивающее комплексную защиту компьютера.
2. Регулярно обновлять антивирусное ПО.
3. Не реже раза в неделю производить антивирусную проверку всего компьютера.
4. Работать в учетной записи с ограниченными правами пользователя, учетную запись с правами администратора использовать только в тех случаях, когда нужно установить программы или изменить настройки системы.
5. Крайне осторожно относиться к программам и документам, которые приходят из глобальных сетей. Перед тем, как запустить файл на выполнение или открыть документ, обязательно проверить их на наличие вирусов. Не открывать вложение, если отправитель письма неизвестен.
6. Периодически сохранять на внешнем носителе файлы, с которыми ведется работа.

Вопросы для самоконтроля

1. Какие виды вредоносного ПО вы знаете?
2. Объясните, что такое компьютерный вирус.
3. Какие разновидности вирусов встречаются?
4. Какие средства борьбы с компьютерными вирусами существуют?

5. Выполнение каких правил позволит снизить вероятность заражения компьютера вирусом?

6. Каковы пути заражения компьютерным вирусом?

ГЛАВА 3.
ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

3.1. Программное обеспечение для обработки текстовой информации

Каждый пользователь сталкивается с проблемой ввода и редактирования текста. Программы, позволяющие это делать, установлены на каждом компьютере. Многие называют этот класс программ текстовыми редакторами, однако все не так просто.

3.1.1. Классификация программного обеспечения для создания и редактирования текста

Текстовые редакторы – программы для создания, редактирования, сохранения и печати документа.

Редактирование – внесение каких-либо изменений в набранный текст (добавление, удаление, перемещение и т. д.).

Примеры: vi (Linux); LeafPad (Linux); Блокнот (Windows); редакторы, содержащиеся в системах программирования.

Текстовые процессоры – текстовые редакторы, поддерживающие форматирование текста.

Форматирование – изменение формы представления документа (отступы и интервалы, выравнивание абзаца, размеры шрифта и т. д.).

Современные текстовые процессоры позволяют также вставлять таблицы, рисунки, объекты других приложений, производить проверку орфографии и многое другое.

Издательские системы – мощные текстовые процессоры, предназначенные для подготовки документов к публикации.

Примеры: Scribus (Linux); Adobe Design; QuarkXPress и др.

3.1.2. Редактор научных текстов TeX

Помимо текстовых процессоров общего назначения, к которым относится Word, существуют программы для работы с текстовыми документами, предназначенные для набора специальных текстов. Специальные тексты – тексты, содержащие большое количество специальных символов, не встречающихся в обычных текстах, например математические тексты.

Было время, когда при наборе математических текстов просто оставляли пустые места, в которые потом, после распечатки, формулы вписывались вручную.

Одним из первых редакторов, позволяющим набирать математические формулы, был ChiWriter. Способ набора формул там был достаточно громоздким.

Встроенные редакторы математических формул есть и в OpenOffice.org – Math, и в MS Office – Equation, сокращенная версия редактора MS MathType. Эти редакторы широко используются, когда нужно в текстовый документ вставить некоторое количество формул. Однако формулы внедряются в текстовый документ как объекты других приложений (редакторов формул). Поэтому если необходимо набрать документ, в котором очень большое количество математических формул, то документ становится слишком громоздким, медленно идет его загрузка и т. п. В этом случае, как правило, используются программы другого класса – издательские системы, специально разработанные для подготовки к изданию математических и других научных текстов, содержащих много формул и специальных знаков.

На сегодняшний день лучшими системами для набора математических текстов являются издательские системы, в основе которых TeX. Первую систему для верстки текстов с формулами создал американский математик и программист Дональд Кнут. Сам по себе TeX представляет собой специализированный язык программирования, на котором пишутся издательские системы, используемые на практике. LaTeX, MikTex, LyX и другие – это издательские системы, созданные на базе TeX.

Многие издательские системы на базе TeX не поддерживают режим WYSIWYG. Формулы, математические символы, форматирование и разметка осуществляется только при помощи специальных команд языка TeX. Но неудобства, которые влечет за собой отсутствие возможности видеть текст при наборе таким, каким он будет напечатан, не являются препятствием в работе. Во-первых, к такому способу набора быстро привыкаешь, а, во-вторых, TeX, создававшийся именно как редактор математических текстов, предоставляет гораздо большие возможности для набора различных формул, даже достаточно сложной структуры, нежели встроенные офисные редакторы математических формул.

Достоинства TeX:

- текст на печати имеет высокое полиграфическое качество;
- гибкость верстки абзацев и математических формул;
- невысокие системные требования;
- машинонезависимость.

Недостатки:

- текст при создании не видим так, как он будет выглядеть при печати (математические знаки – в виде специальных команд);
- медленная работа.

Последовательность работы:

1. Подготовка исходного текстового файла с расширением .tex в любом текстовом редакторе.
2. Обработка этого файла с помощью программ-трансляторов, получение dvi-файла.
3. Просмотр или печать полученного файла.

Современные системы, основывающиеся на языке TeX, например LyX, позволяют вводить большинство математических символов при помощи графического интерфейса – специальных палитр с математическими значками, система автоматически генерирует соответствующий TeX код. Таким образом, верстка документа в TeX существенно упростилась.

3.1.3. Кодировки текста

Для кодирования символа требуется 1 байт (8 бит) информации. Это позволяет закодировать 256 символов, что вполне достаточно для представления текстовой информации, включая строчные и прописные буквы алфавита, цифры, знаки, графические символы и т. д. Для сопоставления символов и кодов используется *таблица кодировки* – стандарт, ставящий в соответствие каждому символу уникальный порядковый номер от 0 до 255 (или соответствующий ему двоичный код от 00000000 до 11111111).

Международным стандартом стала таблица ASCII, в которой первые 33 кода соответствуют управляющим символам (пробел, перевод строки и т. д.), коды с 33 по 127 соответствуют символам латинского алфавита, цифрам, знакам препинания, знакам арифметических действий, а коды с 128 по 255 являются национальными и отличаются в разных странах. Таблицу кодировки символов 128–255 принято называть *кодовой страницей*. Существует несколько кирилличных кодовых страниц. CP866 используется для кодирования кирилличных символов в MS DOS, CP1251 – в Windows.

Существуют и другие кодовые таблицы, широко используемые на практике. Например, КОИ-8 (Код Обмена Информацией), применяемая в глобальных компьютерных сетях, на ЭВМ, работающих под управлением ОС Unix. Очень часто этот стандарт используется в электронной почте.

Сейчас разработан новый международный стандарт Unicode, который отводит на каждый символ два байта или 16 бит и позволяет закодировать 65 536 символов. Такой широкий диапазон позволяет представить в численном виде символы любого языка, в том числе и китайского. Юникод имеет несколько форм представления UTF-8, UTF-16, UTF-32 соответственно восьми, шестнадцати и тридцати двух байтные.

Так как существует несколько широко используемых стандартных кодировок, то часто возникают случаи, когда пользователь не может прочитать текст, поскольку кодировка текста отличается от кодировки, установленной в приложении, с которым работает пользователь. Специальные программы-конверторы, встроенные в приложения, производят перекодирование текста.

3.1.4. Универсальные форматы для представления текста и документов

Формат файла определяет способ хранения текста в файле.

Простейший формат текстового файла содержит только символы (числовые коды символов). Другие форматы содержат дополнительные управляющие числовые коды, которые обеспечивают форматирование текста.

Существует несколько универсальных текстовых форматов, которые могут быть прочитаны любым редактором:

- **txt** – только текст (Text only). Наиболее универсальный формат. Сохраняет текст без форматирования, за исключением только управляющих символов конца абзаца. Применяется для документов, которые должны быть прочитаны приложениями, работающими в различных операционных системах;
- **rtf** (Rich Text Format). Сохраняет все форматирование. Преобразовывает управляющие коды в команды, которые могут быть прочитаны и интерпретированы многими приложениями;
- **pdf** (Portable Document Format) используется для обмена отформатированными документами. Разработан фирмой Adobe. Этот формат хорошо использовать в том случае, когда необходимо сохранить точное форматирование документа. Документ PDF, так же, как и текст факса, изменить очень непросто.

Adobe Acrobat – это целая технология, включающая несколько программ для создания документов в формате PDF и работы с ними.

Создаются такие документы с помощью программы Acrobat Distiller, которая устанавливает на компьютер одноименный принтер. Документ, напечатанный из любого приложения на этот «принтер», на самом деле преобразуется в PDF-файл. Open

Office Writer позволяет преобразовать текстовый документ в формат PDF (нажатием кнопки PDF на панели инструментов).

PDF-файл может быть открыт для просмотра и напечатан без всяких изменений на любом компьютере, независимо от типа процессора, операционной системы, установленных шрифтов и т. п., лишь бы на нем была установлена программа для просмотра PDF-файлов (Acrobat Reader, Foxit Reader и др.).

Внести же изменения в PDF-файл можно, воспользовавшись собственно программой Adobe Acrobat.

Технология Acrobat очень удобна для передачи макета в электронном виде заказчику (который при этом не сможет его присвоить);

- **html** (HyperText Markup Language). Файл имеет расширения .html, .htm . Используется для электронных гипертекстовых документов (web-страниц), в которых содержатся ссылки на другие документы (страницы), доступные через глобальную сеть Интернет или расположенные на локальном компьютере. Это специальный язык – язык разметки гипертекста. Файлы HTML – это текстовые файлы с элементами разметки в виде тегов, которые заключаются в угловые скобки. Иными словами, тег – это указание, как оформлять текст. HTML поддерживает графику, именно это обстоятельство и сделало популярным World Wide Web.

3.1.5. Сканирование текста. Системы оптического распознавания текста (OCR)

Современные программно-аппаратные системы позволяют автоматизировать ввод больших объемов печатной информации в компьютер, используя сканер и распознавание текстов.

Сначала печатная страница сканируется, в результате получается растровое изображение (картинка). Растровое изображение страницы может быть получено и через факс-модем, сканер, цифровую фотокамеру или другое устройство. Работать с изображением как с текстом, то есть редактировать, форматировать и т. д., естественно, нельзя. Поэтому необходимо использовать программу оптического распознавания текста (OCR – Optical Character Recognition) для получения полноценного текстового документа.

На первом этапе OCR разбивает страницу на блоки текста, основываясь на особенностях правого и левого выравнивания и наличия нескольких колонок. Затем распознанный блок разбивается на строки. Потом строки разбиваются на непрерывные области изображения (отдельные буквы). Алгоритм

распознавания каждую область изображения соотносит с наиболее близким по начертанию символом. В результате растровое изображение текстовой страницы восстанавливается в символах текста.

OCR-системы могут достигать наилучшей точности распознавания – свыше 99,9% для чистых изображений, составленных из обычных шрифтов. Но полностью избежать ошибок не удается. Процент ошибок распознавания для «нечистых» текстов намного выше.

Основное назначение OCR-систем состоит в анализе растровой информации (отсканированного символа) и присвоении фрагменту изображения соответствующего символа. После завершения процесса распознавания OCR-системы должны уметь сохранять форматирование исходных документов, присваивать в нужном месте атрибут абзаца, сохранять таблицы, графику и т. д. Современные программы распознавания поддерживают все известные текстовые и графические форматы и форматы электронных таблиц, а некоторые поддерживают такие форматы, как HTML и PDF.

При распознавании текстов, в которых использовано несколько языков, эффективность распознавания зависит от умения OCR-системы формировать группы языков. В то же время в некоторых системах уже имеются комбинации для наиболее часто используемых языков, например русский + английский.

На данный момент существует огромное количество программ, поддерживающих распознавание текста как одну из возможностей.

Лидер в этой области — FineReader. Это программный продукт фирмы ABBYY Software. FineReader поддерживает большое количество форматов для сохранения, включая PDF, имеет возможность прямого распознавания из PDF-файлов. FineReader точно воспроизводит документы сложной верстки, поддерживает большое количество языков.

OCR CuneiForm (свободное ПО) – один из главных конкурентов FineReader. Производителем является российский разработчик программного обеспечения Cognitive Technologies. OCR CuneiForm имеет высокий уровень распознавания, в том числе текстов низкого качества. Отличается удобным интерфейсом. Распознает любые полиграфические и машинописные гарнитуры всех начертаний и шрифтов, получаемые с принтеров, за исключением декоративных и рукописных.

Вопросы для самоконтроля

1. Как можно классифицировать программы для обработки текстовой информации?
2. Что происходит в процессе редактирования текста?
3. В чем заключается форматирование текста?
4. Как происходит процесс создания документа в TeX-ориентированных издательских системах?
5. Перечислите универсальные текстовые форматы.
6. Для чего предназначены системы оптического распознавания текста?

3.2. Текстовый процессор OpenOffice.Org Writer

Мы будем знакомиться с пакетом OpenOffice.org[1]. Это свободное ПО, дистрибутив пакета можно скачать с сайта http://ru.openoffice.org.

Поскольку это кроссплатформенное приложение, его можно установить и под Windows, и под Linux.

3.2.1. Интерфейс

Запуск приложения осуществляется из группы OpenOffice.Org, которая обычно находится в меню «Пуск».

Открывается окно приложения, в котором сразу создается новый документ. Каждый документ открывается в отдельном окне.

Строка меню обеспечивает доступ ко всем функциям Writer.

Если справа от названия пункта имеется стрелка, то при выборе этого пункта будет выведено подменю.

Если название пункта меню заканчивается многоточием, то будет выведено дополнительное диалоговое окно.

Обычно по умолчанию открываются две панели инструментов: «Стандартная» и «Форматирование». Набор панелей, выведенных на экран, можно изменять (Вид – Панели инструментов).

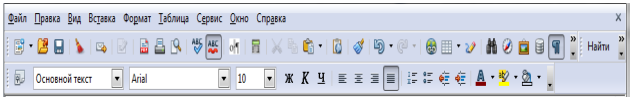


Рис. 3.1. OpenOffice.org Writer. Панели инструментов

Набор кнопок на панелях тоже можно настроить под свои потребности (Вид – Панели инструментов – Настройка).

[1] С 2010 г. параллельно развивается и другой свободный офисный пакет Libre Office как ответвление от разработки OpenOffice.org.

3.2.2. Ввод и редактирование текста

Текст вводят с помощью алфавитно-цифровых клавиш. Для ввода прописных букв используют одновременное нажатие клавиши с буквой и клавиши Shift или включают режим прописных букв нажатием на клавишу Caps Lock.

Когда текст доходит до конца строки, то он автоматически переходит на новую строку.

Чтобы начать новый абзац необходимо нажать Enter, это приводит к тому, что в текст вставляется непечатаемый символ конца абзаца ¶ . Увидеть текст со всеми непечатаемыми символами можно, нажав на кнопку ¶ на панели инструментов «Стандартная» или выполнив команду Вид – Непечатаемые символы.

Абзац – любая часть документа, содержащая текст, графику, объекты, (например формулы) за которой следует маркер конца абзаца (маркер конца абзаца содержит информацию о форматировании, которое применяется к этому абзацу).

Переход на новую страницу осуществляется автоматически при заполнении очередной страницы, принудительно перейти на новую страницу можно нажав Ctrl+Enter или вставив разрыв страницы (Вставка – Разрыв – Разрыв страницы).

Место, куда вводится текст в данный момент, отмечается вертикальной чертой, которая называется курсором. Щелчок мышью в нужном месте текста перемещает туда курсор.

Отмена и возврат действия

Все операции ввода, редактирования и форматирования текста протоколируются текстовым редактором, и поэтому можно отменить некоторые последние действия (Правка – Отменить или кнопка в панели инструментов «Стандартная»).

Отмененное действие можно вернуть обратно (Правка – Вернуть или кнопка).

Использование автозамены при вводе

Использование автозамены позволяет заменить ввод часто повторяющихся длинных последовательностей символов произвольным (желательно коротким) сочетанием других символов.

Настройку автозамены выполняют следующим образом: Формат – Автозамена – Параметры автозамены, введя в поле «Заменить» сокращенное слово, а в поле «Заменить на» полностью слово или словосочетание.

Ввод специальных символов

В текст можно вставлять такие символы, как ¼ и ©. Типы доступных для вставки символов определяются имеющимися шрифтами. Например, стандартный шрифт символов (обыч-

ный текст) включает символы дробей (¼), буквы национальных алфавитов (Ç, ë), а также международные символы денежных единиц (£, ¥).

Вставка символов в текст документа: Вставка – Специальные символы. Открывается таблица символов, в которой устанавливается необходимый шрифт, выбирается нужный символ и вставляется в текст.

Для ввода символов форматирования воспользуйтесь Вставка – Символ форматирования (неразрывный пробел или дефис, мягкий перенос).

Проверка правописания

Writer может проверить документ или текущее выделение на наличие орфографических ошибок.

Для этого нужно выбрать команду Сервис – Проверка орфографии или нажать кнопку  на панели инструментов «Стандартная».

Если при проверке обнаруживается орфографическая ошибка, то открывается диалоговое окно «Орфография», в котором даются рекомендации по исправлению ошибки. Если рекомендации не точны или неприемлемы, от них можно отказаться (Пропустить). Если слово отмечено как ошибочное только потому, что его нет в словаре, его можно добавить во встроенный словарь.

Можно выбрать режим автоматической проверки орфографии  . Вводимый текст будет автоматически проверяться и слова, содержащие ошибки, подчеркиваться волнистой красной чертой. Если навести курсор на отмеченное таким образом слово, можно открыть контекстное меню со списком предлагаемых исправлений. Если выбрать исправление для замены слова, то в случае повторения такой же ошибки при редактировании этого документа она будет исправлена автоматически.

Задание 3.2.1

Наберите с клавиатуры текст. Организуйте автозамену для слова «математика».

«Математика (от др.-греч. μάθημα – изучение, наука) – наука о структурах, порядке и отношениях, которая исторически сложилась на основе операций подсчета, измерения и описания форм реальных объектов.

До начала XVII в. математика – преимущественно наука о числах, скалярных величинах и сравнительно простых геометрических фигурах, изучаемые математикой величины рассматриваются как постоянные. Областью применения математики являлись: счет, торговля, землемерные работы, астрономия и отчасти архитектура.

В XVII-XVIII вв. потребности бурно развивавшихся естествознания и техники привели к введению в математику идей движения и изменения, прежде всего в форме переменных величин и функциональных зависимостей между ними. Это повлекло к созданию и развитию дифференциального и интегрального исчисления.

В XVIII в. возникают и развиваются теория дифференциальных уравнений (например, вида dy=2xdx), дифференциальная геометрия и т. д.

В XIX-XX вв. математика поднимается на новые ступени абстракции, развиваются новые дисциплины: теория функций комплексного переменного, теория групп, проективная геометрия, теория множеств, математическая логика, функциональный анализ…»

Выполните проверку правописания.

Перейдите принудительно на новую страницу после второго абзаца. Включите отображение непечатаемых символов. Обратите внимание на маркеры конца абзаца и страницы.

Отмените переход на новую страницу.

3.2.3. Форматирование текста

Форматирование – изменение внешнего вида документа.

Осуществляется форматирование командами меню «Формат» и кнопками панели инструментов «Форматирование» (наиболее употребляемые команды меню дублируются кнопками). Многие команды форматирования дублируются также в контекстном меню (вызывается нажатием правой кнопки мыши).

К форматированию текста относят:

- Форматирование символов (выбор и изменение шрифта (гарнитуры), размера, начертания и цвета шрифта).
- Форматирование абзаца (в том числе списка).
- Форматирование страницы.
- Создание таблиц.
- Многоколоночная верстка.

3.2.4. Параметры страницы

Параметры страницы задаются с помощью диалогового окна «Стиль страницы», которое вызывается: Формат – Страница.

Диалоговое окно «Стиль страницы» имеет несколько вкладок.

Вкладка «Страница» позволяет задавать формат и размер бумаги, выбирать ориентацию листа бумаги (альбомная или

книжная), устанавливать размеры полей (отступы от краев страницы до текста) и настройки разметки.

Размеры полей также можно задать и с помощью линеек:

– левое и правое поля с помощью горизонтальной линейки  , перетаскивая мышью разделитель (границу между белым и серым фоном линейки);

– верхнее и нижнее поле устанавливается аналогично с помощью вертикальной линейки.

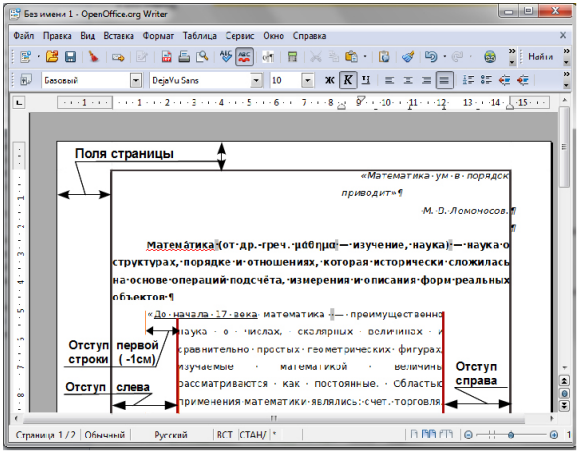


Рис. 3.2. Поля и отступы

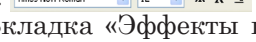
Задание 3.2.2

Установите для текста следующие поля: левое – 2 см, правое – 1,5 см, нижнее и верхнее – 1 см.

Установите размер бумаги – 18 х 15 см и альбомное положение листа.

3.2.5. Настройки шрифта

Настройку шрифта выполняют в диалоговом окне «Символы» (Формат – Символы), имеющем несколько вкладок:

Вкладка «Шрифт» позволяет изменять шрифт, кегль (размер), начертание символа. Поскольку форматирование символов достаточно частая операция, основные шрифтовые настройки вынесены на панель форматирования:  .

Вкладка «Эффекты шрифта» позволяет задавать некоторые другие эффекты оформления (зачеркивание, подчеркивание, цвет шрифта и др.).

Вкладка «Положение» позволяет задавать верхний (a^n) и нижний (a_n) индексы, смещение символов относительно основной строки, масштаб и вращение символов, интервал между символами в строке.

Задание 3.2.3

Установите для всего текста шрифт DejaVu Sans (или Times New Roman), размер – 10.

Выделите определение математики жирным шрифтом.

Подчеркните даты, встречающиеся в тексте.

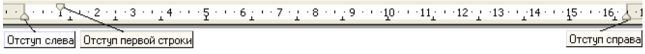
Название разделов математики выделите курсивом.

3.2.6. Форматирование абзаца

Форматирование абзаца осуществляется в диалоговом окне «Абзац» (Формат – Абзац), которое также состоит из нескольких вкладок.

Вкладка «Отступы и интервалы» настраивает величины абзацных отступов слева (от левого поля), справа (от правого поля), отступ первой строки («красная строка»).

Более простой и наглядный способ определения абзацных отступов связан с использованием маркеров линейки. Если линейка отсутствует, то отобразите ее (Вид – Линейка).



Также в этой вкладке можно задать отбивку до и после абзаца (отступ перед абзацем и после него) и междустрочный интервал (расстояние между строками текста). Междустрочный интервал задается и с помощью соответствующей команды контекстного меню.

Вкладка «Выравнивание» позволяет выбрать метод выравнивания текста (по левому краю, по центру, по правому краю, по ширине).

Наиболее быстрый способ выполнить выравнивание абзаца – использовать кнопки  панели инструментов «Форматирование».

Задание 3.2.4

В набранном тексте (задание 1) сделайте выравнивание абзацев по ширине.

Задайте отступ первой строки первого абзаца в 1,5 см.

Задайте полуторный междустрочный интервал.

Перед первым абзацем наберите эпиграф: «Математика ум в порядок приводит» М. В. Ломоносов (см. рис. 3.2).

Для эпиграфа установите отступ слева (левую границу абзаца) – 8,2 см, отступ первой строки – 1 см (то есть примерно на 9,2 см, обратите внимание, как установлены маркеры линейки на рис. 3.2).

Аналогично для третьего абзаца: отступ слева – 2,3 см, отступ справа – 2 см, абзацный заступ – 1 см (см. рис. 3.2).

После первого абзаца сделайте интервал (отбивку) в 0,2 см.

3.2.7. Многоколоночная верстка

Часто при верстке журнала или газеты необходимо отформатировать текст в две колонки или более.

Для того чтобы расположить текст в несколько колонок, надо выделить необходимый фрагмент текста и выполнить команду Формат – Колонки (в некоторых версиях Writer – Столбцы). В появившемся окне диалога определить количество колонок, интервал между колонками и при необходимости ширину каждой колонки.

Задание 3.2.5

Отформатируйте текст про математику в две колонки.

3.2.8. Создание маркированных и нумерованных списков

Самый быстрый способ создания одноуровневых списков – использовать кнопки панели «Форматирование», для нумерованных списков кнопку  , для маркированных  . Автоматически открывается панель «Маркеры и нумерация», которая облегчает редактирование списка. С помощью кнопок этой панели можно повысить или понизить уровень элемента списка, переместить его выше или ниже по списку и т. д.

Настройка параметров списка осуществляется в диалоговом окне «Маркеры и нумерация» (Формат – Маркеры и нумерация) или кнопкой  панели «Маркеры и нумерация». Данное окно имеет несколько вкладок.

Вкладки «Маркеры», «Изображения» позволяют выбрать вид символьного или графического маркера для создания маркированного списка.

С помощью вкладки «Тип нумерации» можно создать различные нумерованные списки.

Для создания многоуровневого списка следует использовать вкладку «Структура». Отрегулировать интервалы и отступы в многоуровневом списке можно с помощью вкладки «Положение», для этого в диалоговом окне выбирается номер уровня и для этого уровня определяются отступ, расстояние до текста и способ выравнивания нумерации.

Если пользователя не удовлетворяют предложенные программой варианты оформления маркированного или нуме-

рованного списка, то вкладка «Настройка» позволит создать уникальные способы оформления маркированных, нумерованных и многоуровневых списков.

Создание списка происходит сразу после настройки его оформления или при нажатии кнопки на панели инструментов.

При нажатии на Enter в конце пункта списка автоматически будет проставлен маркер или номер для следующего абзаца.

При вводе многоуровневого списка для перехода на более низкий уровень используется клавиша Tab и кнопки , или , если требуется понизить уровень вместе с подпунктами. Аналогично, для повышения уровня используют Shift+Tab и кнопки , .

Двойное нажатие на Enter в конце списка, завершает его.

Задание 3.2.6

Создайте многоуровневый список, в котором для нумерации первого уровня используются римские числа, для нумерации второго – арабские, а третий уровень обозначается маркером §. Используйте вкладку «Настройка» для задания параметров в каждом уровне, для третьего уровня в выпадающем списке «Нумерация» выберите «Рисунок», «Стиль символа» определите как маркеры списка, а символ маркера найдите в таблице символов, нажав на кнопку «Символ» (см рис. 3.3).

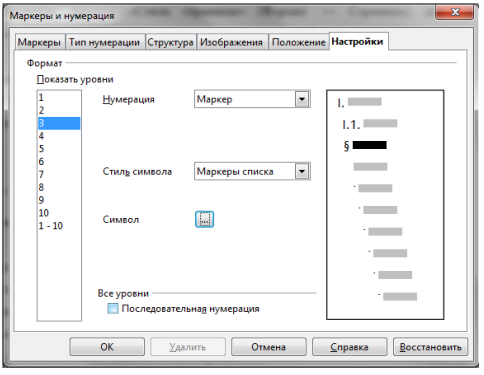


Рис. 3.3. Настройки многоуровневого списка

3.2.9. Обрамление и фон

Обрамление и фон используются для выделения фрагмента текста или всей страницы.

Применить обрамление можно как ко всей странице целиком, так и только к одному или нескольким абзацам. Обрамле-

ние может быть как с четырех сторон, так и с одной, двух или трех сторон. При этом могут быть применены различные стили и цвета линий, а также графические границы. Аналогично и фоновую заливку можно применить целиком к странице, к нескольким выделенным абзацам или только к одному абзацу.

Чтобы задать обрамление страниц, следует использовать вкладку Обрамление диалога «Стиль страницы» (Формат – Страница), для определения фоновой заливки нужно открыть вкладку «Фон».

Если стоит задача заключить в рамку абзацы, то их необходимо выделить, вызвать окно диалога «Абзац» (Формат – Абзац) и открыть вкладку «Обрамление», в которой задать все нужные параметры. Фоновая заливка абзацев задается во вкладке «Фон» диалога Абзац.

Задание 3.2.7

Выделите первый абзац текста про математику (задание 1) в рамку с тенью и залейте его серым цветом.

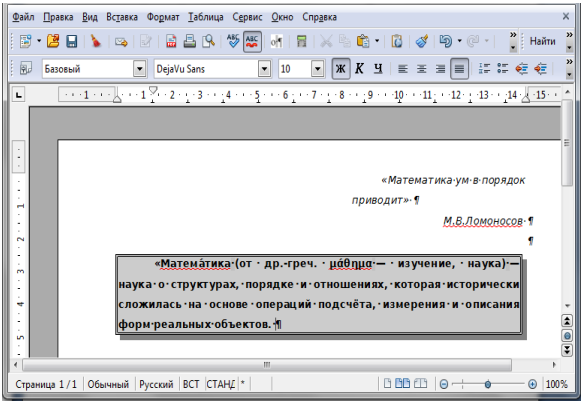


Рис. 3.4. Обрамление и фоновая заливка абзаца

3.2.10. Колонтитулы

Колонтитул – это текст и/или рисунок, который печатается внизу или вверху каждой страницы документа. В зависимости от места расположения (на верхнем или на нижнем поле страницы) колонтитулы бывают верхними и нижними. Например, нумерация страниц выставляется в нижнем или верхнем колонтитуле.

По умолчанию один и тот же верхний колонтитул и нижний колонтитул создается для всего документа. Если нуж-

но отличать колонтитулы некоторых разделов, колонтитулы для четных и нечетных страниц, а также создать уникальный колонтитул для первой страницы документа, то необходимо использовать различные стили страниц для четных и нечетных страниц в каждом разделе и отдельный стиль для первой страницы (см. «Стили»).

Работа с верхним и нижним колонтитулом осуществляется с помощью диалога «Стиль страницы» (Формат – Страница), вкладок «Верхний колонтитул», «Нижний колонтитул» (см. рис. 3.5).

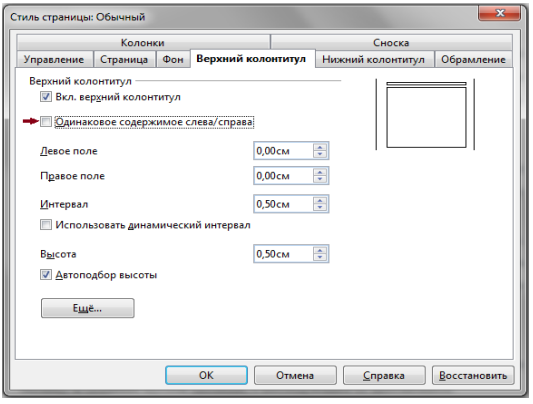


Рис. 3.5. Формат страницы. Колонтитулы

Чтобы отличать колонтитулы четных и нечетных страниц, в соответствующей вкладке настройки колонтитулов (верхний или нижний колонтитул) снять галочку «Одинаковое содержимое слева/справа». Это позволит создать разные колонтитулы на четных и нечетных страницах. Можно отличать, например, только верхние колонтитулы, а нижние оставить одинаковыми (то есть оставить галочку).

Например, создадим несколько страниц (Вставка – Разрыв – Новая страница). Вызовем диалог «Стиль страницы». Во вкладке «Страница» установим различную разметку для четных и нечетных страниц (слева и справа). В верхнем колонтитуле будем различать содержимое справа/слева, а в нижнем оставим одинаковым (пометим галочкой). Вверху и внизу страницы появятся небольшие прямоугольные текстовые блоки.

В текстовом блоке верхнего колонтитула первой страницы напишите свою фамилию, выравнивание по правому краю. В нижнем колонтитуле вставьте поле для нумерации

страниц (Вставка – Поля – Номер страницы), сделайте выравнивание по центру.

Перейдите на следующую (четную) страницу. Поскольку в нижнем колонтитуле мы не отличали содержание для четных и нечетных страниц, там будет поле с номером страницы. В верхнем колонтитуле вставим поле с текущей датой (Вставка – Поля – Дата), выравнивание по левому краю.

Просмотрите другие страницы. У всех страниц нижний колонтитул будет содержать номер страницы, а верхний колонтитул: у нечетных – фамилию, у четных – дату.

Если в разных частях документа должны быть разные колонтитулы, нужно вставить создать новый раздел документа (Вставка – Раздел) и назначить страницам этого раздела другие стили четных и нечетных страниц (Подробнее см. «Стили страниц»).

3.2.11. Создание таблиц

Существует два способа создания таблиц: преобразование текста в таблицу и создание пустой таблицы, с последующим ее заполнением.

Первый способ – преобразование текста в таблицу – может быть применен для создания простых таблиц.

Построчно ввести данные, находящиеся в ячейках таблицы, при этом содержимое одной ячейки отделяется от другой нажатием Tab, в конце строки нажимается Enter.

Выделить текст, который надо преобразовать в таблицу, и выбрать Таблица – Преобразовать в таблицу. Проверить правильность указанных там параметров.

Второй способ. Создание пустой таблицы.

С помощью команды Таблица – Вставить – Таблица или используя кнопку . Необходимо указать количество строк и столбцов в табли Затем заполнить ячейки таблицы.

Задание 3.2.8

Одним из перечисленных способов создайте таблицу:

	1	2	3	4	5
Большой театр	«Жизель»	«Щелкунчик»	«Аида»	«Пиковая дама»	«Снегурочка»
Малый театр	«Борис Годунов»	«Корсиканка»	«Чайка»	«Гроза»	«Царь Федор Иоанович»

3.2.12. Редактирование таблиц

Все действия с таблицами выполняются с помощью пункта верхнего меню «Таблица». Сюда относятся:

- Вставка и удаление строк и столбцов. Если надо вставить несколько строк (столбцов), то необходимо установить курсор в ту строку (столбец), куда предполагается сделать вставку и выполнить: Таблица – Добавить строки (столбцы), указать количество строк (столбцов) и куда вставлять (перед текущим или после).
- Добавьте в созданную таблицу сверху одну строку и слева один столбец.
- Задание ширины столбцов и высоты строк. Эти параметры устанавливаются командами контекстного меню: Строка – Высота и Столбец – Ширина или перемещением ограничителей строк и столбцов на горизонтальной и вертикальной линейках.

Задание 3.2.9

Отрегулируйте длину и ширину ячеек предложенной таблицы (задание 8).

Объединение ячеек и разбиение ячеек осуществляется с помощью команд «Объединить ячейки», «Разбить ячейки» в меню Таблица или кнопок  в панели инструментов «Таблица». Предварительно эти ячейки нужно выделить.

Объедините ячейки в добавленной строке и столбце по образцу.

Театры	месяц	сентябрь				
	число	1	2	3	4	5
	Большой театр	«Жизель»	«Щелкунчик»	«Аида»	«Пиковая дама»	«Снегурочка»
	Малый театр	«Борис Годунов»	«Корсиканка»	«Чайка»	«Гроза»	«Царь Федор Иоанович»

Измените направление текста в первом столбце (Формат – Символы, вкладка «Положение»).

3.2.13. Форматирование таблицы

Обрамление, толщина границы, стиль и цвет линий, заливка фона ячеек задаются при помощи окна диалога «Таблица» (Таблица – Свойства таблицы) или с использованием соответствующих кнопок панели инструментов «Таблица».

Задание 3.2.10

Определите в созданной таблице (задания 8-9) обрамление и заливку ячеек.

Театры	месяц	сентябрь				
	число	1	2	3	4	5
	Большой театр	«Жизель»	«Щелкунчик»	«Аида»	«Пиковая дама»	«Снегурочка»
	Малый театр	«Борис Годунов»	«Корсиканка»	«Чайка»	«Гроза»	«Царь Федор Иоанович»

Текст в таблице оформляется при помощи форматирования символов и абзацев. Выравнивание текста можно производить не только по ширине, но и по высоте ячейки. Для выравнивания удобно пользоваться кнопками  панели инструментов «Таблица».

Для форматирования таблицы может быть применен автоформат (Таблица – Автоформат), то есть оформление таблицы можно выбрать из имеющихся образцов.

Отформатируйте текст в ячейках таблицы по образцу.

Чтобы заголовки таблицы автоматически дублировались в начале каждой страницы, если таблица занимает несколько страниц, нужно в диалоге «Оформление таблицы», на вкладке «На странице» поставить галочку «Повторить заголовок».

3.2.14. Сноски

Сноска состоит из двух частей: собственно символа сноски, расположенного в тексте, и текста этой сноски. В зависимости от того, где располагается текст сноски, различают обычные сноски (располагаются в конце страницы) и концевые сноски (располагаются в конце документа). Сноски могут быть нумерованные или помеченные любым символом.

Чтобы вставить сноску, необходимо поставить курсор в нужное место текста и выбрать команду Вставка – Сноска. Указать тип сноски (обычная – внизу страницы или концевая – в конце документа), а также вид нумерации (автоматическая – числовая или символ, например, *).

Задание 3.2.11

Создайте в своем документе про математику (задания 1-4) обычную автоматическую сноску к эпиграфу: «М. В. Ломоносов – первый русский ученый-естествоиспытатель мирового значения, энциклопедист, химик и физик».

3.2.15. Вставка объектов в документ (формул, рисунков и т. д.)

В документ могут быть помещены объекты различных приложений. Объекты внедряются в документ с помощью коман-

ды Вставка – Объект – Объект OLE, затем в списке выбирается приложение, объект которого вставляется в документ.

Для редактирования объекта нужно щелкнуть по нему два раза, после чего элементы интерфейса приложения, которое создало этот объект, появляются в окне Writer, что позволяет изменять и редактировать этот объект.

Для вставленного объекта можно задать его взаимное расположение с остальным текстом (обтекание). Для этого необходимо выделить объект и выбрать команду Формат – Обтекание (или в контекстном меню «Обтекание»).

Вставка формул

Формулы создаются с помощью приложения OpenOffice Math. Вставка формулы в текст документа производится через внедрение объекта OpenOffice Math (Вставка – Объект – Формула).

Окно делится на две части. Нижняя часть – окно команд, в которой формула записывается на командном языке. В верхней части окна отображается вид формулы. С помощью панели инструментов «Выбор» (Вид – Выбор), содержащей шаблоны основных математических выражений, можно вводить математические формулы. Греческие буквы вставляются с помощью окна «Символы», которое вызывается нажатием на кнопку Σ панели инструментов.

По умолчанию курсор находится в окне команд. Чтобы сделать доступными объекты формулы, нужно сделать активным курсор формулы нажатием на кнопку . При выделении объекта формулы в верхней части окна одновременно выделяется соответствующая запись в команде, записанной в нижней части окна.

По окончании ввода формулы нужно щелкнуть в любом месте документа.

Задание 3.2.12

Внимательно просмотрите математические шаблоны панели «Выбор». Используя подходящие шаблоны, наберите следующую формулу:

$$\sqrt[3]{\frac{\sqrt{\delta^n + \gamma^n}}{\lambda^{\frac{1}{n}}}}$$

Вставка рисунков

А) Через буфер обмена.

Откройте графический редактор OpenOffice Draw или другой графический редактор, нарисуйте треугольник. Используя буфер обмена, скопируйте этот рисунок в текстовый документ.

Б) Из файла.

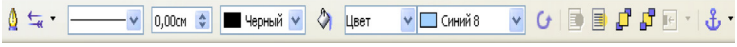
Сохраните созданный в редакторе OpenOffice Draw рисунок (экспортируйте (Файл – Экспорт) в универсальный формат, например в emf). Вставьте его в текстовый документ (Вставка – Изображение – Из файла).

В) Используя панель инструментов «Рисование».

Некоторые простые инструменты рисования встроены в текстовый редактор. Эти инструменты доступны через панель инструментов «Рисование» (Вид – Панели инструментов – Рисование). С помощью этих инструментов можно рисовать линии, прямоугольники, эллипсы, кривые, блок-схемы, автофигуры и др. Полученные рисунки будут векторными графическими объектами, поэтому их можно свободно масштабировать без потери качества.

Внимательно изучите инструменты рисования, наведя мышкой на инструмент можно получить подсказку о его назначении.

Выберите любой инструмент (щелкните по нему), обратите внимание, что при этом панель инструментов Форматирование закрывается, а вместо нее появляются инструменты панели «Свойства рисунка», которые необходимы для редактирования изображений.



С помощью этой панели можно менять свойства линии (стиль, толщину, цвет, стиль стрелок), стиль и цвет заливки, вращать фигуры, перемещать их на передний/задний план, привязывать к тексту, группировать/разгруппировывать и др.

Если необходимо линию объекта сделать прозрачной, например, для надписи, то в контекстном меню графического объекта (щелчок правой кнопкой мыши по объекту) выберите «Линия» и установите «Стиль» – «Невидимая».

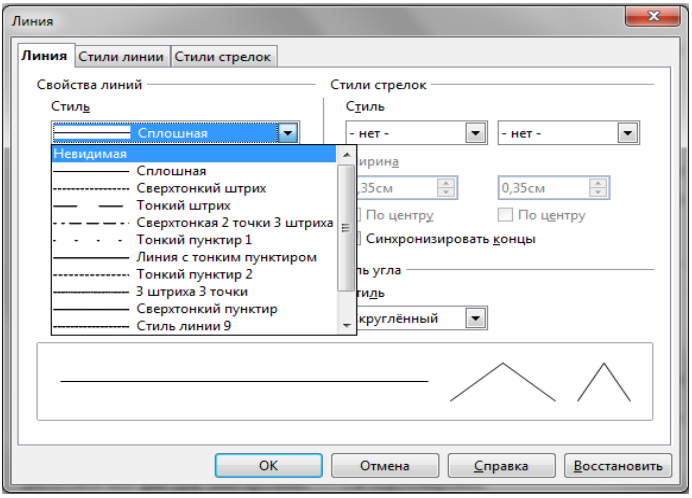


Рис. 3.6. Линия. Настройки

Аналогично, чтобы заливку объекта сделать прозрачной, в контекстном меню выберите «Область» и установите в выпадающем списке «Цвет» настройку «Нет».

Все свойства изображения настраиваются в контекстном меню объекта. Можно устанавливать тип обтекания и отступы от текста, располагать на переднем или заднем плане, группировать с другими графическими объектами, обрамлять изображение и т. д.

Задание 3.2.13

1. Вставьте автофигуру из группы звезды, установите соответствующий цвет и заливку. В эту фигуру вставьте надпись (используйте кнопку T). Сделайте прозрачными линии и заливку. Добавьте звезду из автофигур. Выделите все фигуры (инструмент) и сгруппируйте.

Рис. 3.7. Автофигуры и надпись (к заданию 13.1)

2. Используя инструменты панели рисования, создайте следующую блок-схему (см. рис. 3.8). Для сложных фигур, состоящих из нескольких объектов, используйте группировку.

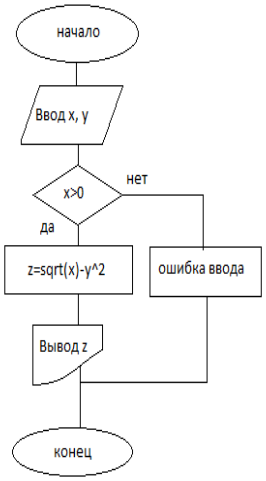


Рис. 3.8. Блок-схема (к заданию 13.2)

3.2.16. Стили

Стили – это набор характеристик, определяющих внешний вид и форматирование текста, к которому они применяются.

Стиль – это имеющая имя совокупность таких параметров, как шрифт, размер шрифта, цвет, выравнивание абзаца, межстрочный интервал и т. д.

Например, присвоив абзацу некоторый стиль, можно автоматически задать все элементы оформления, которые заданы для данного стиля. Если для нескольких абзацев задан один стиль, то они будут оформлены одинаково. Изменение стиля автоматически изменит оформление абзацев, которым назначен этот стиль.

Writer позволяет создавать и использовать стили символов, абзацев, страниц, врезок, списков. Команда Формат – Стили вызывает окно «Стили и форматирование». В этом окне перечисляются названия всех существующих стилей, которые сгруппированы по нескольким категориям: стили абзаца , стили символа , стили врезок , стили страницы и стили списка . Переключение между этими категориями осуществляется нажатием на соответствующую кнопку.

Чаще всего приходится пользоваться стилями абзацев.

Назначение стиля абзацу

Поставить курсор в любое место абзаца и выбрать нужный стиль из списка «Стиль», который открывается на панели инструментов «Форматирование» или выполнить команду: Формат – Стили, в результате которой открывается окно «Стили и форматирование» с установленными стилями.

Очень удобно пользоваться инструментом «Копировать форматирование»  . Абзац, форматирование которого мы хотим взять за образец, нужно выделить. Затем скопировать форматирование  . Далее следует выделить часть текста, которой мы хотим присвоить это форматирование. Выделенные абзацы автоматически переформатируются по образцу.

Изменение параметров существующего стиля

Чтобы внести изменения в установленные стили, необходимо вызвать окно «Стили и форматирование» (Формат – Стили), с помощью кнопок-переключателей определить категорию стиля (абзаца, символа и т. д.), выбрать в списке стилей название стиля, форматы которого нужно изменить, далее в контекстном меню выбрать «Изменить».

В открывшемся окне указать следующие параметры шрифта, абзаца, отступы и интервалы, обрамление и т. д.

Создание нового стиля

Формат – Стили, кнопкой-переключателем определить категорию стиля, в контекстном меню выбрать «Создать стиль», далее указать форматы шрифта, абзаца и другие параметры.

Задание 3.2.14

1. Измените настройки стилей:

Заголовок1 (DejaVu Sans, 14, п/ж, подчеркнутый, интервал после абзаца 6 пт).

Заголовок 2 (DejaVu Sans, 12, п/ж, курсив, интервал после абзаца 4 пт).

Заголовок 3 (Times New Roman, 12, п/ж, интервал после абзаца 3 пт).

2. Создайте новый стиль абзаца «Рабочий» (Times New Roman, 12, отступ первой строки 1 см, междустрочный интервал полуторный, выравнивание по ширине).

Создайте новый стиль страницы «Моя страница», в котором включите нижний колонтитул.

3. Измените настройки стиля «Первая страница»: во вкладке «Управление» следующий стиль определите «Моя страница». (Обычно на титульном листе не используются колонтитулы. Для того, чтобы отличать форматирование первой страницы от остальных, и применяются отличные стили для первой

страницы и для всех других. Аналогично поступают и в тех случаях, когда необходимо использовать отличающийся текст в колонтитулах разных разделов документа).

4. Наберите текст:

«Глава 1. Название 1 главы.
§1. Название первого параграфа.
1.1. Название пункта 1.
Текст первого пункта…. (Вставьте разрыв страницы).
1.2. Название пункта 2.
Текст второго пункта… (Вставьте разрыв страницы).
§2. Название второго параграфа.
Текст второго параграфа (Вставьте разрыв страницы).
Глава 2. Название второй главы.
§1. Название первого параграфа.
Текст первого параграфа (Вставьте разрыв страницы).
§2. Название второго параграфа.
1.1. Название пункта 1.
Текст первого пункта…. (Вставьте разрыв страницы).
1.2. Название пункта 2.
Текст второго пункта…»

5. Страницам присвойте стиль «Моя страница». Используйте стили заголовков 1, 2, 3 соответственно для названий глав, параграфов и пунктов, а также стиль «Рабочий» для основного текста. В нижний колонтитул вставьте нумерацию страниц.

6. Вставьте титульный лист в начало документа, присвойте ему стиль «Первая страница» (на первой странице колонтитулы должны исчезнуть).

3.2.17. Создание оглавления

1. С помощью встроенных стилей заголовков

Сначала нужно назначить абзацам, которые являются заголовками, стили заголовков (Заголовок 1, Заголовок 2 и т. д.). Например, названию главы присвоить стиль «Заголовок 1» (заголовок первого уровня), названию параграфа – «Заголовок 2», названию пункта – «Заголовок 3» и т. д.

Затем необходимо курсором указать место в документе, куда будет вставлено оглавление. Обычно оглавление вставляют в конце или в начале документа.

Выполнить команду: Вставка – Оглавление и указатели – Оглавление и указатели.

В появившемся окне «Вставить оглавление/указатель» несколько вкладок. Вкладка «Вид» позволяет определить за-

головок для оглавления. Выберите нужный вид оглавления, укажите уровни заголовков, которые войдут в оглавление.

Существует возможность разработки пользовательских стилей оглавления.

Задание 3.2.15

Сделайте оглавление документа (задание 14), разместите его в начале документа после титульного листа.

2. С помощью указателей

Выделите заголовок и выполните команду: Вставить – Оглавление и указатели – Элемент. В появившемся окне «Вставить элемент указателя» необходимо тип указателя в поле «Указатель» установить как «Оглавление» и определить уровень данного заголовка. Нажатие на кнопку «Вставить» завершает процедуру.

Вставка оглавления в документ осуществляется так же, как и в случае использования встроенных стилей заголовков (см. выше).

3.2.18. Создание библиографии

В курсовых и дипломных работах обязательно нужно приводить список использованной литературы, а в тексте делать ссылки на источники, из которых были взяты цитаты. Многие делают это «вручную»: составляют список литературы, а затем в текст вставляют «ссылки» – указывают номер источника в списке. При этом, если в середину списка добавляется новый источник, нумерация сбивается и ее нужно переделывать. Чем длиннее список использованной литературы, тем сложнее становится эта задача.

В некоторых текстовых редакторах для составления библиографии используются закладки и перекрестные ссылки.

В Writer используется специальная база данных «Библиография», в которую можно заносить сведения об использованных книгах. Команда Сервис – База данных библиографии открывает таблицу этой базы. Данные можно заносить как непосредственно в таблицу, так и в форму, расположенную под таблицей.

Задание 3.2.16

Занесите в базу данных «Библиография» следующие книги:

Хахаев И., Машков В. и др. OpenOffice.org. Теория и практика. – М.: Издательство «Бином», 2008.

Такет Дж., Барнет С. Использование Linux. – М.: Вильямс, 2000.

Примечание: Обязательно следует присвоить книге сокращенное название в поле «Сокращенно» (или Identifier). Без сокращенного названия невозможно будет сделать ссылку на этот источник.

Создание ссылок в документе на источники, содержащиеся в библиографии

Если в документе цитируется отрывок из произведения другого автора, необходимо дать ссылку на источник, то есть указать автора и книгу, по которой цитируется фрагмент. Обычно в конце документа приводится список литературы (библиография), а в самом документе после приведенной цитаты в квадратных или круглых скобках дается ссылка на библиографию, как правило, указывается номер источника в списке литературы.

Writer позволяет автоматизировать этот процесс. Пользователю необходимо используемые литературные источники занести в базу данных библиографии, а в тексте документа после каждой приведенной цитаты вставить ссылку на источник в библиографии. Список литературы в конце документа будет сгенерирован автоматически, в него будут включены источники из базы данных библиографии, ссылки на которые встретились в документе.

Чтобы вставить ссылку на источник в библиографии в документ, нужно установить курсор в то место документа, куда требуется вставить ссылку на источник. Выполняется команда: Вставка – Оглавление и указатели – Элемент списка литературы. В появившемся окне «Вставить библиографическую ссылку» отметьте, что данные находятся в базе данных библиографии и в выпадающем списке «Сокращенное название» выберите то сокращенное название, которое соответствует нужному литературному источнику. До генерации в документе списка литературы ссылка будет содержать сокращенное название, которое после генерации заменится соответствующим номером из списка литературы.

На разных страницах создайте ссылки на книги, занесенные в библиографическую базу.

Генерация списка литературы в документе

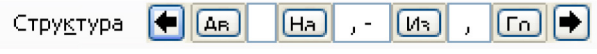
Чтобы вставить список литературы в документ, нужно установить курсор в то место документа, куда предполагается вставить библиографию. Выполнить команду: Вставка – Оглавление и указатели. Появившееся окно «Вставить оглавление/указатель» имеет несколько вкладок.

Откройте вкладку «Вид» и выберите в выпадающем списке «Вид» значение «Библиография». По умолчанию

в поле «Заголовок» написано «Библиография». Этот заголовок можно исправить на «Список литературы:». Обязательно нужно поставить галочку в опции «Пронумеровать элементы».

Откройте вкладку «Записи», в которой указывается, какой вид будет иметь каждая запись в списке литературы и какие данные нужно взять из базы данных библиографии. В поле «Структура» задается шаблон записи. Элементы шаблона – это поля записи в базе данных библиографии, разделенные пробелами, знаками препинания и т д.

Изменим шаблон «Структура». Удалим первый элемент «Со» (сокращенное название) – выделить элемент и нажать кнопку «Удалить» и удалим двоеточие, следующее за ним. Оставим элемент «Ав» (автор) после ставим пробел и элемент «За» (заголовок). Если в базе данных название источника вы заносили в поле «Название книги», то следует удалить элемент «Заголовок», а вместо него вставить элемент «Название книги» (в выпадающем списке под словом «Структура» выбрать элемент «Название книги» и нажать кнопку «Вставить»). Далее нужно поставить точку, пробел, тире и снова пробел, затем вставить элемент «Издатель», после которого поставить запятую, пробел и элемент «Год». Примерный вид шаблона будет следующим:



Укажите, что необходимо сортировать записи по содержимому и укажите в ключе сортировки «Авторы», нажмите кнопку «По возрастанию»  .

Нажатие на кнопку «ОК» запускает генерацию списка литературы в документе.

3.3. Редактирование научных текстов в LYX

3.3.1. Специализированный язык разметки документа TeX

TEX представляет собой специализированный язык, который используется для создания документов с большим количеством математических или других научных символов.

Эти издательские системы не поддерживают режим WYSIWYG (What You See Is What You Get), то есть увидеть, как будет выглядеть отформатированный документ, можно только после его компиляции.

Последовательность работы при создании TeX-документа:

1. Подготовка в редакторе исходного текстового файла и сохранение его с расширением .tex.
2. Обработка этого файла с помощью программ-трансляторов, получение .dvi или .pdf файла.
3. Просмотр полученного .dvi или .pdf файла в специальной программе для просмотра или печать этого файла.
4. При необходимости внесения каких-либо изменений, файл .tex открывается в редакторе, в него вносятся поправки, он сохраняется и заново компилируется.

Исходный файл TeX

Он представляет собой собственно текст документа вместе со спецсимволами и командами. Этот файл может быть создан в любом тестовом редакторе (ASCII). Как правило, для подготовки исходного текста, можно использовать обычный текстовый редактор, который не вставляет свои специальные символы форматирования, например, Блокнот. Некоторые системы TeX имеют свой редактор для набора исходного текста.

Текст не должен содержать переносов. Слова отделяются друг от друга любым количеством пробелов (лишние пробелы игнорируются). Абзацы отделяются пустой строкой.

В различных системах, созданных на базе TeX'а, правила организации исходного документа различны.

Структура документа:

\documentclass{article}

(определяем стиль документа -- оформление)

\usepackage[cp866]{inputenc}

(cp866 - в кодировке MS-DOS. Можно указать и cp1251, и koi8-r).

\usepackage[russian]{babel} \begin{document}

... текст документа

\end{document}

В первой строке (\documentclass) определяется стиль документа. Существует 4 стандартных стиля: article (статья), book (книга), report (отчет), letter (письмо). Книга и отчет могут оформляться с колонтитулами, другие отличия сводятся к размеру шрифта в заголовках, правилам нумерации разделов, таблиц, формул.

Текст документа:

С помощью команд задаются отсутствующие на клавиатуре специальные знаки и буквы, устанавливаются элементы форматирования.

Команда начинается символом «\» .

Пробелы:

TeX различает только один пробел в тексте, остальные игнорируются.

Конец строки воспринимается тоже как пробел.

\QUAD – двойной матем. пробел
\quad – матем. пробел
\Quad; – большой пробел

Специальные символы:

Особую роль играют следующие символы:

{,} – группировка
$ – выделение формулы
& – табуляция
% – начало комментария
// – разрыв строки
_ – указатель нижнего индекса
^ – указатель верхнего индекса
~ – неразрывный пробел
\ – признак начала команды

Шрифт:

размер:

\tiny – крошечный
\scriptsize – индексный
\footnotesize – для сносок
\small – мелкий
\normalsize – нормальный
\large – увеличенный
\Large – большой
\LARGE – очень большой
\huge – громадный
\Huge – колоссальный

начертание:

\rm – прямой
\it – курсив
\sl – наклонный
\bf – полужирный
\sc – малые прописные (капитель)
\tt – машинописный

Группы и окружения:

группы: определяются {}.

Внутри группы задается команда, которая будет применяться только к символам внутри скобок.

Н-р, {\bf Теорема} – все символы слова теорема будут выделены полужирным шрифтом.

окружения (для выделения больших фрагментов текста):

\begin{имя окружения}

...

\end {имя окружения}

Формулы:

внутритекстовые заключаются в $...$

выключные (как отдельный абзац) заключаются в $$...$$

степень: ^	x^{1987} соответствует x^{1987}
нижний индекс: _	a_{i,j} соответствует $a_{i,j}$
дробь:	\frac{числитель}{знаменатель}
корень:	\sqrt[степень]{подкоренное выражение}
интеграл:	\int_a^b подинтеграл.выр-е (a,b-пределы интегрирования)
предел:	\lim
сумма:	\sum
скобки: фигурные	\{,\}
круглые	\left(,\right)

Математические символы:

Группа	Набор	Операция	Группа	Набор	Операция
Акценты	\hat z	$\hat{z}$	Геометрия	\triangle	$\triangle$
	\tilde z	$\tilde{z}$		\angle	$\angle$
	\dot z	$\dot{z}$		\perp	$\perp$
	\bar z	$\bar{z}$		\top	$\top$
	\vec z	$\vec{z}$		\circ	$\circ$
Арифмет.	\div	$\div$	Множества	\cup	$\cup$
	\times	$\times$		\cap	$\cap$
	\cdot	$\cdot$		\in	$\in$
	\pm	$\pm$		\notin	$\notin$
	\ast	$\ast$		\subset	$\subset$
	\vert	$\vert$		\backslash	$\backslash$
	\colon	$\colon$		\oslash	$\oslash$

Отношен.	\le \ge \ll \gg \ne \sim \approx	$\leq$ $\geq$ $\ll$ $\gg$ $\neq$ $\sim$ $\approx$	Следования	\leftarrow \rightarrow \Rightarrow \uparrow \downarrow \prec \succ	$\leftarrow$ $\rightarrow$ $\Rightarrow$ $\uparrow$ $\downarrow$ $\prec$ $\succ$
Логика	\land \lor \not \iff \exists \forall	$\wedge$ $\vee$ $/$ $\Leftrightarrow$ $\exists$ $\forall$	Анализ	\Re \Im \to \partial \infty \nabla	$\Re$ $\Im$ $\to$ ∂ ∞ ∇

3.3.2. Работа в LyX

LyX – это система подготовки документов, надстройка LaTeX, использует парадигму стилевой разметки. Благодаря LyX можно набирать математические формулы, не зная языка TeX, так как в программе предусмотрены различные шаблоны с математическими символами для набора разнообразных формул. Несмотря на то, что в окне редактора вы видите текст и формулы, LyX только примерно показывает, как выглядит ваш текст, показывая и выделяя его структуру.

Для того чтобы увидеть, как будет выглядеть текст при печати, необходимо экспортировать созданный файл в один из форматов: dvi, ps или pdf (Файл – Экспортировать в или используя кнопки ).

Затем полученный файл вы можете просмотреть, используя меню Просмотр или соответствующие кнопки панели инструментов  .

Примечание:

Последние версии LyX автоматически создают файл соответствующего формата при обращении пользователя к кнопкам просмотра.

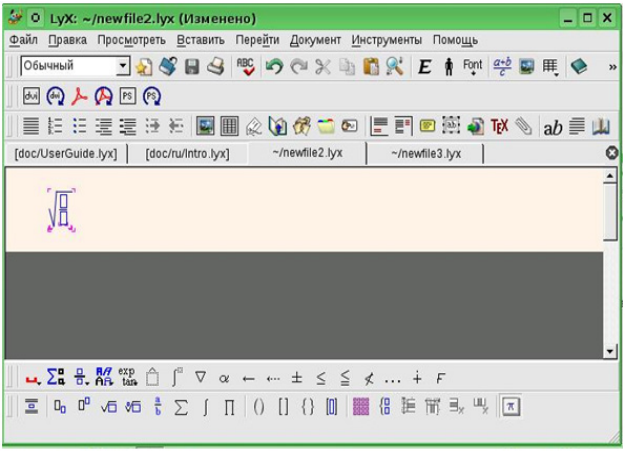


Рис. 3.9. Окно LyX

Создание нового документа

Для создания нового документа и указания его класса необходимо произвести следующие действия:

- Файл – Создать... (File – New) создать новый документ.
- Изменить настройки документа, открыв окно настроек документа: Документ – Настройки (Document – Settings).

– установить класс документа (Document – Class) (по умолчанию article – статья), выбрать из выпадающего списка требуемый класс документа;

– выбрать необходимые шрифты (Fonts);

– определить макет текста (Text Layout);

– формат страницы (Page Layout);

– язык (Language); установить русский, снять галочку «Use languag's default encoding) и установить кодовую страницу (Encoding) либо «cp1251», либо «cp866», либо «koi8-r». Без этой настройки при использовании кириллицы будут генерироваться ошибки.

Структура документа

Документ LyX состоит из различных частей, имеющих разное назначение. Эти логически обособленные части документа называются окружениями (environments), по сути это различные стили оформления абзаца. Каждое окружение включает в себя один или несколько абзацев.

Окружение абзаца определяет такие свойства абзаца, как стиль шрифта, отступы, схема нумерации и т. д. Название окружения абзаца, в котором находится курсор, показывается

слева в панели инструментов Форматирование. Здесь же его можно изменить на другое.

Используя возможности LyX, будем набирать текст, содержащий как внутритекстовые (заключенные в $), так и выключные (заключенные в $$) математические формулы и одновременно анализировать автоматически создаваемый программой код TeX. Для этого вызовите окно просмотра исходного текста (Просмотреть – Просмотреть исходный текст), подтвердите автоматическое обновление и показ всего файла целиком (поставьте галочки).

Для работы нам понадобятся панели инструментов: «Мат. Панели» и «Формула» (Просмотреть – Панели инструментов).

Внимательно ознакомьтесь с инструментами этих панелей.

Эти инструменты понадобятся вам для набора следующих формул.

Код LaTeX	Вид при печати
{\Large Дроби}\\ Простой пример: ${\large \frac{a}{b}}$ $$\frac{a+b}{c \cdot d}= \frac{a}{c \cdot d}+\frac{b}{cd}$$	Дроби Простой пример: $\frac{a}{b}$ $\frac{a+b}{c \cdot d}= \frac{a}{c \cdot d}+\frac{b}{cd}$
{\Large Корни}\\ Простой пример: $\sqrt{x}$, со степенью - $\sqrt[n]{x}$ $$\sqrt[m]{\frac{x-a}{x+a}}$$	Корни Простой пример: $\sqrt{x}$, со степенью – $\sqrt[n]{x}$ $\sqrt[m]{\frac{x-a}{x+a}}$
{\Large Индексы}\\ Простой пример: A_n $$C_{\psi+1}^{\tau-a}$$	Индексы Простой пример: A_n $C_{\psi+1}^{\tau-a}$
{\Large Степени}\\ Простой пример: a^b $$(a-b)^{n+m};$$ $$a^{\frac{a}{b}}$$	Степени Простой пример: a^b $(a-b)^{n+m}$ $a^{\frac{a}{b}}$

{\Large Большие скобки}\\ $$\left(\frac{a}{b}\right)^n; \left|\frac{a}{b}\right|$$ $$ \left\{ \frac{a}{a+\frac{1}{a}} \right\}; $$	Большие скобки $$\left(\frac{a}{b}\right)^n;\ \left|\frac{a}{b}\right|$$ $$\left\{\frac{a}{a+\frac{1}{a}}\right\}$$
{\Large Интегралы}\\ Простой пример:$\int_0^1 x dx $ $$ \int\limits_{-\infty}^{+\infty} x^2 \, dx $$	Интегралы Простой пример: $\int_0^1 x dx$ $$\int\limits_{-\infty}^{+\infty} x^2\,dx$$
{\Large Сумма}\\ Простой пример:$\sum_{i=1}^n 2n $ $$\sum_{i=1}^\infty (a_k^2+b_k^2) $$	Сумма Простой пример: $\sum_{i=1}^{n} 2n$ $$\sum_{i=1}^{\infty}(a_k^2 + b_k^2)$$
{\Large Пределы}\\ $$\lim_{x\to\infty}f(x) $$	Пределы $$\lim_{x\to\infty} f(x)$$
{\Large Матрица}\\ % в матрице строки разделяются \\, а элементы отделяются &. $$\begin{array}{ccc} % в последних скобках задается кол-во столбцов (кол-во букв) и выравнивание (c-центр, l-лев.край,r-пр.край) a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\\ \end{array} $$	Матрица $$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$
Система ур-ний, записанная с помощью матрицы: $$\left\{ \begin{array}{ccl} x^2+y^2 & = & 7\\ x+y & = & 3\\ \end{array} \right . $$	Система ур-ний, записанная с помощью матрицы: $$\begin{cases} x^2+y^2 &= 7 \\ x+y &= 3 \end{cases}$$ *Примечание:* LyX проверяет парные операторы, т.е. для каждого \left должен быть \right, но т.к. в данном случае правая фигурная скобка не нужна, после оператора \right мы вместо скобки ставим точку.

{\Large Таблица}\\ $$ \begin{tabular}{| |c|c| |} \hline a & b\\ \hline \hline c & d\\ \hline \end{tabular} $$	Таблица a b c d
{\Large Специальные символы}\\ $$ \vec{v}$$\\ $$ \forall a \in N \exists c \ Leftrightarrow \leftrightarrow \rightarrow $$ $ a_1+a_2+\ldots+a_n ;\qquad a_1+a_2+\cdots+a_n ;$\\	Специальные символы $\vec{v}$ $\forall a \in N \exists c \Leftrightarrow \leftrightarrow \rightarrow$ $a_1 \mid a_2 \mid \ldots \mid a_n; \quad a_1 \mid a_2 \mid \cdots \mid a_n$

Задание 3.3.1

В основном окне набирайте текст, который расположен в правой колонке таблицы и отслеживайте сгенерированный код в окне «Исходный текст LaTeX», в основном, он будет совпадать с кодом в левом столбце таблицы.

Во всех примерах формулы в тексте – внутритекстовые (этот режим в LyX установлен по умолчанию), а формула вне текста, выравненная по центру строки – выключная (выключный режим устанавливается нажатием на кнопку панели инструментов «Формула»).

Размер шрифта устанавливается при помощи Edit – Text Style – Customized, Size.

Задание 3.3.2

В режиме TeX (Вставить – TeX код) наберите следующий текст:

Выражение тригонометрических функций через тангенс половинного аргумента

$$\sin\alpha - \frac{2tg\frac{\alpha}{2}}{1+tg^2\frac{\alpha}{2}}$$

Корень n-й степени

$$\sqrt[n]{\sqrt[k]{a}} = \sqrt[n \cdot k]{a}$$

Длина дуги (натуральный периметр) линии

$$\vec{r} = \vec{r}(t) \rightarrow s = \int_{t_1}^{t_2} \left|\vec{r}'(t)\right| dt$$

$$\begin{cases} x = x(t) \\ y = y(t) \\ z = z(t) \end{cases} \rightarrow s = \int_{t_1}^{t_2} \sqrt{x'^2 + y'^2 + z'^2}\, dt$$

3.4. Табличные процессоры

Появление электронных таблиц (ЭТ) исторически совпадает с началом распространения персональных компьютеров. Первая программа для работы с ЭТ – ***табличный процессор***, была создана в 1979 г., предназначалась для компьютеров типа Apple II и называлась VisiCalc. В 1982 г. появляется знаменитый табличный процессор Lotus 1-2-3, предназначенный для IBM PC. Lotus объединял в себе вычислительные возможности ЭТ, деловую графику и функции реляционной СУБД. Популярность табличных процессоров росла очень быстро. Появлялись новые программные продукты этого класса: Multiplan, Quattro Pro, SuperCalc и др. Одним из самых популярных табличных процессоров сегодня является MS Excel, входящий в состав пакета Microsoft Office, в пакете OpenOffice.org – Calc.

Области применения табличных процессоров разнообразны.

1. ***Выполнение вычислений.*** Издавна многие расчеты выполняются в табличной форме, особенно в области делопроизводства: многочисленные расчетные ведомости, сметы расходов и т. п. Кроме того, решение численными методами целого ряда математических задач удобно выполнять в табличной форме. Электронные таблицы представляют собой удобный инструмент для автоматизации таких вычислений. Решения многих вычислительных задач на ЭВМ, которые раньше можно было осуществить только путем программирования, стало возможно реализовать на электронных таблицах.

2. ***Математическое моделирование***. Использование математических формул в ЭТ позволяет представить взаимосвязь между различными параметрами некоторой реальной системы. Основное свойство ЭТ – мгновенный пересчет формул при изменении значений входящих в них операндов. Благодаря этому свойству таблица представляет собой удобный инструмент для организации численного эксперимента: подбор параметров, прогноз поведения моделируемой системы, анализ зависимостей, планирование. Дополнительные удобства для моделирования дает возможность графического представления данных.

3. ***Использование электронной таблицы в качестве базы данных***. Конечно, по сравнению с СУБД электронные таблицы имеют меньшие возможности в этой области. Однако некоторые операции манипулирования данными, свойственные реляционным СУБД, в них реализованы. Это поиск информации по заданным условиям и сортировка информации.

3.4.1. Структура электронной таблицы

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								

Рис. 3.10. Вид электронной таблицы

Электронная таблица – матрица, состоящая из столбцов и строк.

Заголовки строк обозначаются числами 1, 2, 3 …

Заголовки столбцов обозначаются латинскими буквами A, B, C, … AA, AB, AC, …

Пересечение столбца и строки образует ***ячейку.***

Каждая ячейка имеет ***адрес*** (имя) – например, A5, B6 и т. д.

Текущая, активная ячейка, с которой производятся какие-либо действия, выделяется рамкой (курсором).

Ячейка может содержать:

- текст (признаком текстовых данных являются кавычки);

- число;
- формулу (начинается со знака =).

Ячейку можно рассматривать как *переменную*, так как ячейка имеет имя (адрес) и хранит значение какого-либо типа.

3.4.2. Формулы

В электронных таблицах используются два вида выражений:

- арифметические;
- логические.

Арифметические – определяют способ вычисления некоторого числового значения.

Правила записи арифметических формул аналогичны тем, что используются в языках программирования.

Формулы состоят из констант, переменных, знаков операций, функций.

Например, = 5*SQRT (C5^2-4*B3)

Логические выражения строятся с помощью операций отношения (<, >, =, <=, >=, < >) и логических операций («И», «ИЛИ», «НЕ»). Результатом вычисления логического выражения являются логические величины «истина» или «ложь».

Часто в формулах и функциях операции нужно произвести с группой (диапазоном) ячеек. Например, найти максимум, сумму диапазона ячеек.

Диапазон обозначается именами (адресами) верхней левой и нижней правой ячеек.

Например, A3:C7.

3.4.3. Адресация

В формулах используются имена переменных – ссылки на адреса ячеек.

В электронных таблицах используются ***два вида адресации***:

- относительная;
- абсолютная.

Различия проявляются при копировании формулы из активной ячейки в другие.

Относительная адресация используется для указания адреса ячейки, вычисляемого относительно ячейки, в которой находится формула.

Например, в E2 написана формула: = B2*D2.

E2 – текущая ячейка, B2 – ячейка, расположенная на 3 ячейки левее текущей, D2 – ячейка, расположенная на 1 левее текущей.

Копируем формулу в E3. Теперь текущей будет ячейка E3, поэтому адреса ячеек, использующихся в формуле, будут определяться относительно E3. Ячейка, расположенная на 3 ячейки левее – B3, на 1 ячейку левее – D3. Формула примет вид = B3* D3 (см. рис. 3.11).

	A	B	C	D	E	F
1						
2		12	2	5	=B2*D2	
3					=B3*D3	
4						
5						
6						

Рис. 3.11. Относительная адресация. Копирование формулы

Относительная адресация в электронных таблицах используется по умолчанию.

Абсолютная адресация используется для указания фиксированного адреса ячейки.

При перемещении и копировании абсолютные адреса не изменяются. Для указания абсолютного адреса используется знак $. Зафиксировать можно весь адрес, например, A7 или отдельные его части (только строку или столбец), например, $A7 или A$7 (смешанные ссылки).

3.4.4. Графическая обработка данных

Представление табличных данных в графической форме часто используется на практике. Графическая обработка делает наглядными результаты расчетов. Табличные процессоры предоставляют пользователю на выбор множество типов диаграмм (гистограмм, графиков).

Вопросы для самопроверки

1. В каких областях деятельности применяются табличные процессоры?
2. Какова структура электронной таблицы?
3. Каковы особенности абсолютной и относительной адресации?

3.5. Работа в табличном процессоре OpenOffice.org Calc

Электронные таблицы – это программа для создания и использования документов с автоматическим расчетом вносимых данных. Данные организованы в виде таблиц.

В электронную таблицу можно вводить данные, обычно числовые, и затем манипулировать этими данными для получения определенных результатов.

3.5.1. Вид окна Calc

Главное окно табличного процессора Calc имеет следующий вид (см. рис. 3.12):

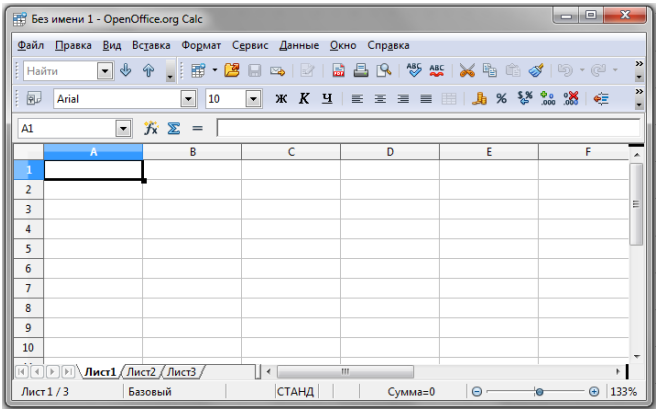


Рис. 3.12. Окно Calc

Вверху – Заголовок окна. Ниже расположена строка меню. Еще ниже идут панели инструментов, обычно установлены панели «Стандартная» и «Форматирование». Многие инструменты этих панелей вам знакомы по программе OpenOffice.org Writer.

Под панелями инструментов находится «Панель формул»:

Эту панель можно разделить на три зоны: слева расположены координаты активных ячеек; в центре – кнопки мастеров и справа – строка ввода.

Рабочее поле окна представляет собой лист, разделенный на отдельные ячейки (таблицу). Столбцы озаглавлены буквами, строки – цифрами.

Каждый лист Calc может иметь максимум 65 536 строк и максимум 245 столбцов (от A до IV). Каждый лист содержит 16 056 320 индивидуальных ячеек.

Каждая электронная таблица может иметь много листов. Для переключения между ними используют ярлыки рабочих листов, расположенные в самом низу таблицы: Лист1, Лист2 и т. д.

3.5.2. Создание таблиц

Таблица состоит из столбцов, обозначенных латинскими буквами (A, B, C…) и строк, обозначенных числами 1, 2, 3 … таким образом, каждая ячейка имеет адрес, например A2, B5, который используется для ссылки на ячейку (например, в формулах).

Одна из ячеек является активной (выделена рамкой). Сделать ячейку активной можно, щелкнув по ней мышью или переместив рамку с помощью стрелок на клавиатуре.

В активную ячейку можно вводить данные. Если данные не помещаются в ячейке, то они будут отображаться в соседних справа ячейках (если они не заполнены) или частично не будут видны (если ячейка справа заполнена), тогда в ячейке справа отображается маленький красный треугольник.

В ячейке могут быть размещены данные различных форматов. Формат данных определяет правила их обработки, в частности, допустимые операции, которые можно производить с данными ячейки.

Указать формат данных в ячейке можно с помощью команды: Формат – Ячейки (вкладка «Числа»).

Некоторые форматы данных:

Текстовый – состоит из букв, цифр, знаков препинания. По умолчанию выравнивается по левому краю.

Числовой – состоит из цифр и десятичной запятой (или точки). По умолчанию выравнивается по правому краю.

Денежный – для отображения денежных величин.

Процентный – содержимое ячейки умножается на 100 и дописывается знак %.

Дробный – для отображения обыкновенных дробей.

Дата – используется для отображения дат. По умолчанию выравнивается по левому краю.

Время – используется для отображения времени.

Научный – представление в экспоненциальной форме.

Задание 3.5.1

Создайте следующую таблицу (см. рис. 3.13).

	A	B	C	D
1	**Отчет по продажам за первый квартал**			
2		Январь	Февраль	Март
3	Отдел 1	200	180	220
4	Отдел 2	300	320	260
5	Отдел 3	260	160	120
6	Отдел 4	330	450	160

Рис. 3.13. Данные таблицы (к заданию 3.5.1)

Обратите внимание, что текст заголовка таблицы, находящийся в ячейке A1, не поместился и занял соседние ячейки (но относится он все равно только к ячейке A1).

Сохраните созданную таблицу (Файл – Сохранить как).

Выделение ячеек

Для того чтобы произвести некоторые действия с группой ячеек, их надо выделить.

	Первый способ	**Второй способ**
Строка	Щелкнуть на заголовке строки	Shift + пробел
Столбец	Щелкнуть на заголовке столбца	Ctrl + пробел
Блок ячеек	Удерживая левую кнопку мыши, «протащить» курсор из правого верхнего угла в левый нижний	Shift + стрелки
Всего листа	Щелкнуть на кнопке в правом верхнем углу листа	Ctrl + Shift + пробел Ctrl + A

Задание 3.5.2

Попробуйте разные способы выделения ячеек. Выделите в исходной таблице ячейки, содержащие числовые данные, и присвойте им денежный формат (Формат – Ячейки, вкладка «Число»).

3.5.3. Ввод формул

Формула начинается со знака равенства «=». Формула может содержать числа, адреса ячеек, функции, знаки математических операций, но не может содержать текст.

Формула набирается в строке ввода. При нажатии клавиши Enter или кнопки ✔ формула считается введенной и в ячейке отображается результат, сама формула после этого отображается только в строке ввода. При изменении исходных значений результат пересчитывается автоматически.

В OpenOffice.org Calc можно использовать следующие операторы.

Арифметические операторы

Эти операторы возвращают числовые значения.

Оператор	Название	Пример
+ (плюс)	Сложение	1+1
- (минус)	Вычитание	2-1
- (минус)	Унарный минус	-5
* (звездочка)	Умножение	2*2
/ (косая черта)	Деление	9/3
% (процент)	Процент	15%
^ (крышка)	Возведение в степень	3^2

Операции сравнения

Эти операторы возвращают значение TRUE (Истинно) или FALSE (Ложь).

Оператор	Название	Пример
=	Равно	A1=B1
>	Больше	A1 >B1
<	Меньше	A1 <B1
>=	Больше или равно	A1 >=B1
<=	Меньше или равно	A1 <=B1
< >	Не равно	A1 < >B1

Текстовые операторы

Этот оператор объединяет несколько текстовых строк в одну.

Оператор	Название	Пример
& (И)	объединение строк И	Выражение «Воскре» & «сенье» эквивалентно строке «Воскресенье»

Операторы ссылки

Эти операторы используются для обозначения диапазонов ячеек.

Оператор	Название	Пример
: (Двоеточие)	Диапазон	A1:C108 Блок ячеек, в левом верхнем углу которого расположена ячейка A1, а в правом нижнем – C108.

! (Восклицательный знак)	Пересечение диапазонов	SUM(A1:B6!B5:C12) Вычисляет сумму всех ячеек в пересечении двух диапазонов; в данном примере результат равен сумме ячеек B5 и B6.

Задание 3.5.3

В созданной таблице (задание 1) добавим столбец «Итого по отделам» и вычислим суммы продаж по отделам (то есть в ячейке E3 сумму ячеек B3, C3, D3, в ячейке E4 сумму ячеек B4, C4, D4 и т. д.).

Для этого в ячейке E3 надо записать формулу: = B3+C3+D3 или =SUM(B3:D3). Вторая формула удобнее при большом количестве ячеек.

Самостоятельно запишите формулы в остальных ячейках столбца E.

Копирование формул

Формулу можно копировать из одной ячейки в другую, при этом адреса ячеек, которые использованы в формуле, меняются автоматически.

Для копирования формулы:

- Сделайте ячейку, содержащую нужную формулу, активной.
- Подведите мышку к маркеру заполнения – маленькому черному квадратику в левом нижнем углу рамки выделения, при этом указатель мыши примет вид черного крестика.
- Нажмите левую кнопку мыши и, не отпуская ее, протащите выделение на нужное количество ячеек. Формула будет скопирована во все выделенные ячейки.

Примечание. Копировать формулы можно и с помощью команд меню Правка – Копировать, затем Правка – Вставить.

Задание 3.5.4

Добавьте в таблицу еще одну строку (в ячейку A7): «Итого по месяцам» и для каждого месяца вычислите сумму продаж.

В ячейку B7 запишите формулу =SUM(B3:B6).

Скопируйте эту формулу в ячейки C7 и D7.

Обратите внимание на изменение адресов в формуле при переходе от ячейки к ячейке.

Примечание: Calc позволяет делать автосуммирование. Для этого необходимо выделить ячейки, данные которых нужно просуммировать и нажать кнопку **Σ** в панели формул.

Абсолютные и относительные ссылки

При копировании формулы адреса ячеек, которые использованы в этой формуле, изменяются, так как обычный адрес

ячейки (например, B6) является относительной ссылкой: запоминается расположение этой ячейки относительно той, где содержится формула. Пример: пусть в ячейке A2 записана формула =A1+B2, она фактически означает «сумма содержимого ячейки, находящейся над данной и ячейки справа от нее», если эту формулу скопировать в ячейку E6, то она примет вид =E5+F6.

Для того чтобы адреса ячеек не изменялись при копировании формулы, используются абсолютные ссылки или имена ячеек.

Для задания абсолютной ссылки на ячейку в ее адресе надо поставить знак $ перед именем столбца и/или строки, например A5 или $A5 или A$5 «координата», перед которой стоит знак $ изменяться при копировании не будет.

Задание 3.5.5

На листе 2 заполните таблицу (см. рис. 3.14).

Примечание. Заполнение строк 4 и 5 можно упростить, если заметить, что числа в них представляют прогрессию. Один из способов *автоматического заполнения ячеек* таков:

Заполните ячейки B4 и C4.

Выделите эти ячейки. Удерживая левую кнопку мыши, захватите за маркер заполнения (черный квадратик в правом нижнем углу рамки) и «растащите» выделение до нужной ячейки (аналогично процессу копирования формул).

В B6 введите формулу =B4+B5. Скопируйте вправо. Как меняется формула в соседних ячейках?

Введите формулы в ячейки B7, B8. Скопируйте вправо.

В ячейку B9 запишите формулу умножения x на константу k. Скопируйте вправо. Получился ли правильный результат в соседних ячейках?

В формуле B9 зафиксируйте ячейку B3 (абсолютная ссылка). Скопируйте вправо. Проверьте результат.

	A	B	C	D	E	F
1						
2						
3		*k*= 45				
4	*x*	1	3	5	7	9
5	*y*	0	-3,5	-7	-10,5	-14
6	*x*+*y*					
7	*x**y*					
8	*x**k*					
9						

Рис. 3. 14. Таблица для задания функций (к заданию 3.5.5)

Присвоение имен (названий) ячейкам

Для возможности использования ссылок на ячейки и диапазоны ячеек в формулах имеет смысл именовать ячейки и диапазоны ячеек. Например, диапазону ячеек A1:B2 можно присвоить имя «Начало». Формула будет иметь вид: =SUM(Начало).

Гораздо проще понять формулу для вычисления налога с продаж, если вместо = A5 * B12 написать фомулу: = Сумма * Ставка_налога. В этом случае имя ячейки A5 – «Сумма», а ячейки B12 – «Ставка_налога».

Имена в Calc могут содержать буквы, цифры и некоторые специальные символы. Имена должны начинаться с буквы или символа подчеркивания.

Допустимые специальные символы:

подчеркивание (_)

точка (.) может стоять внутри имени, но не может использоваться в качестве первого или последнего символа.

пробел ( ) может стоять внутри имени, но не может использоваться в качестве первого или последнего символа и для диапазона ячеек.

Имена не должны совпадать со ссылками на ячейки. Например, имя A1 недопустимо, поскольку A1 является ссылкой на левую верхнюю ячейку.

Имена диапазонов ячеек не должны содержать пробелы. Пробелы допустимы внутри имен для отдельных ячеек, листов и документов.

Определение имени ячейки или имени диапазона ячеек:

Выделите диапазон ячеек, затем выберите в меню команды: Вставка –Названия – Определить. Появится диалоговое окно «Определение имен».

Введите имя выбранной области в поле «Имя». Нажмите кнопку «Добавить». Новое имя появится в списке ниже. Нажмите кнопку «ОК», чтобы закрыть диалоговое окно.

Посредством ввода имени в поле и последующего выбора соответствующих ячеек в этом диалоговом окне можно присвоить имена различным диапазонам ячеек.

Если начать вводить имя в формулу, то после ввода первых символов появится все имя в виде подсказки.

Нажмите клавишу Enter, чтобы принять имя из подсказки.

Если с одних и тех же символов начинается несколько имен, можно прокрутить все имена с помощью клавиши TAB.

Задание 3.5.6

Дополните таблицу, созданную в заданиях 1–4, столбцом «Доля» (столбец F), в котором вычислите долю каждого отдела в общей сумме продаж (отношение продаж по каждому отделу к общей сумме продаж). Запишите формулу в ячейку F3 и скопируйте вниз в остальные ячейки.

В соседнем столбце (столбце G) выполните те же вычисления, только с использованием именования ячейки. Присвойте ячейке E7 имя «итого» (без кавычек). В ячейке G3 запишите формулу: =E3/итого и скопируйте ее в остальные ячейки.

Сравните результаты, полученные в столбцах F и G.

!!! Использование в формуле именованной ячейки все равно, что использование абсолютной ссылки на эту ячейку.

Удалите данные столбца F, а данные столбца G сдвиньте влево (в столбец F).

3.5.4. Форматирование таблицы

Форматирование таблицы в целом и отдельных ячеек производится с помощью вкладок окна «Формат ячеек» (пункт меню Формат – Ячейки или «Формат ячеек» в контекстном меню) или соответствующих кнопок панели инструментов «Форматирование».

Форматирование включает в себя: изменение шрифта и размера, расположение данных в ячейке (выравнивание по левому или правому краю, центрирование), оформление границ, цветовое оформление и др.

Форматирование данных. Вкладка «Числа» позволяет устанавливать категорию данных ячейки (числовой, процентный, дата, и др.), формат записи, определить параметры дробной части (количество знаков после запятой), разделение разрядов для лучшего зрительного восприятия больших чисел и др.

Форматирование шрифта. Вкладки «Шрифт» и «Эффекты шрифта» позволяют настроить шрифтовые форматы. Выбрать шрифт, установить его размер, начертание можно и при помощи инструментов панели «Форматирование». Цвет шрифта, рельеф устанавливается во вкладке «Эффекты шрифта».

Выравнивание текста и данных ячейки. Во вкладке «Выравнивание» можно настроить параметры горизонтального и вертикального выравнивания текста относительно границ ячейки. Изменить направление текста можно с помощью круговой диаграммы или указав угол наклона текста в градусах, при этом каждый символ текста поворачивается на заданное число градусов (по часовой стрелке). Если необходимо распо-

ложить текст вертикально, при этом не поворачивая символы на 90 градусов, нужно поставить галочку в настройке «Накопление по вертикали».

Также во вкладке «Выравнивание» есть настройки, позволяющие управлять размещением текста в ячейке. По умолчанию, если текст не вмещается в одной ячейке, он переносится в соседние справа ячейки, при этом, если в соседнюю ячейку вводятся данные, то не вместившийся в первоначальную ячейку текст прячется и может быть виден полностью только в строке ввода. Избежать этого можно, если включить настройку (поставить галочку) «Переносить по словам» (можно переносить и по слогам). При этом текст автоматически будет переноситься на следующую строку, а высота ячейки увеличиваться, вмещая нужное количество строк. Если не желательно увеличивать размер ячейки, то нужно использовать настройку «Уменьшить по размеру ячейки», тогда размер символов текста будет уменьшен так, что текст полностью поместится в ячейке.

Примечание: для красивого оформления длинных заголовков обычно используют *объединение ячеек*. Для этого нужно выделить ячейки и выполнить команду: Формат – Объединить ячейки или нажать кнопку  панели инструментов «Форматирование». Тогда текст можно центрировать относительно границ объединенных ячеек.

Обрамление и фоновая заливка. Вкладка «Обрамление» позволяет определить параметры рамки как для целой таблицы, выделенного блока ячеек, так и для отдельной ячейки. По умолчанию при печати граница между ячейками таблицы невидимы. Чтобы на распечатке были обозначены границы таблицы, отдельных ячеек необходимо применить обрамление к этим ячейкам. Рамка может быть полной, с четырех сторон, и не полной, с двух сторон, устанавливается при помощи кнопок «Предопределение». Можно настроить вид и цвет рамки, а также отступы от содержимого ячейки.

С помощью вкладки «Фон» можно определить фон отдельной ячейки или блока выделенных ячеек.

Изменение размеров строк и столбцов. Calc позволяет изменять высоту строк и ширину столбцов: Формат – Строка – Высота и Формат – Столбец – Ширина. Следует иметь в виду, что изменение распространяется на все строку или на весь столбец текущего листа.

Изменение размеров можно сделать и при помощи мыши, установив курсор в серой зоне на границе столбцов (строк), курсор меняет форму на двунаправленную стрелку, можно, не отпуская левую кнопку мыши, перетащить границу столбца (строки).

Задание 3.5.7

Оформите созданную на 1 листе таблицу по образцу.

	A	B	C	D	E	F
1	Отчет по продажам за первый квартал					
2		Январь	Февраль	Март	Итого по отделам	Доля
3	Отдел 1	200 000 руб.	180 000 руб.	220 000 руб.	600 000 руб.	20,27%
4	Отдел 2	300 000 руб.	320 000 руб.	260 000 руб.	880 000 руб.	29,73%
5	Отдел 3	260 000 руб.	160 000 руб.	120 000 руб.	540 000 руб.	18,24%
6	Отдел 4	330 000 руб.	450 000 руб.	160 000 руб.	940 000 руб.	31,76%
7	Итого по месяцам	1 090 000 руб.	1 110 000 руб.	760 000 руб.	2 960 000 руб.	100,00%

Рис. 3.15. Форматирование таблицы (к заданию 3.5.7)

Объедините ячейки с A1 до F1 и расположите текст заголовка таблицы по центру.

В столбцах B, C, D, E используйте для данных денежный формат, а для данных столбца F – процентный.

Шрифтовое оформление данных в ячейках и выравнивание сделайте по образцу. Используйте фоновую заливку.

Сделайте обрамление границ ячеек таблицы.

Задание 3.5.8

На листе 2 создайте таблицу умножения (рис. 3.16).

Таблица умножения									
	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

Рис. 3.16. Таблица умножения (к заданию 3.5.8)

Введите формулу в B24. Скопируйте вниз и вправо. Правилен ли результат?

Какие адреса нужно сделать абсолютными? В полученной формуле в каком сомножителе фиксируется только строка, в каком только столбец?

Задание 3.5.9

Вычислите цепную дробь

$$\cfrac{1}{1+\cfrac{1}{3+\cfrac{1}{5+\cfrac{1}{\cdots}}}}\Bigg|_{101+\frac{1}{103}}$$

Задание 3.5.10

Вычислите значение выражения:

$$\sqrt{3+\sqrt{6+\ldots+\sqrt{96+\sqrt{99}}}}$$

3.5.5. Автозаполнение данных

Команды «Автозаполнение» и «Ряд» позволяют автоматически заполнить ячейки данными.

Функция автозаполнения автоматически создает ряды данных по заданному образцу.

Например, нужно заполнить столбец А значениями целых чисел от 1 до 11 с шагом 2.

Введите число 1 в ячейку А1. В ячейку А3 введите следующее число 3.

(Если нужно с помощью автоматически заполнить строку, то соответственно следующее число нужно ввести в чейку В1).

Выделите две ячейки с данными и с помощью мыши протащите маркер заполнения (черный квадратик в правом нижнем углу рамки), по ячейкам, которые нужно заполнить, и отпустите кнопку мыши.

Ячейки заполняются числами по возрастанию.

Примечание. Удерживая Ctrl при выделении ячеек, можно заполнить ячейки одинаковыми значениями.

Использование заданных рядов данных

Выберите на листе диапазон ячеек, подлежащих заполнению.

Откройте диалоговое окно «Заполнить ряды» (Правка – Заполнить – Ряды).

Выберите параметры для ряда.

Если выбран параметр «Линейный», то введенное приращение последовательно прибавляется к каждому числу ряда, давая следующее значение.

Если выбран параметр «Рост», то введенное приращение последовательно умножается на каждое число ряда, давая следующее значение.

Если выбран параметр «Дата», то введенное приращение последовательно прибавляется к указанной единице времени.

3.5.6. Функции

Calc имеет большое число встроенных функций: математических, статистических, логических, финансовых и т. д., которые позволяют производить сложные расчеты в электронной таблице. С некоторыми из них мы с вами познакомимся.

Встроенные функции

Функции задают вычисления по заданным величинам, называемыми аргументами, и в указанном порядке, называемом синтаксисом.

Аргументами функции могут быть:

- Числа
- Текст
- Логические величины (истина, ложь)
- Ссылки на ячейки
- Константы
- Функции

Необходимо следить за соответствием типов аргументов.

Написание функции начинается с указания имени функции, затем вводится открывающая скобка, указываются аргументы, отделяющиеся точкой с запятой, а затем – закрывающая скобка.

Чтобы вставить функцию можно воспользоваться «Мастером функций», который вызывается командой Вставка – Функция или кнопкой на панели инструментов (или CTRL+F2).

Например, построим таблицу значений функции y=sin(x) на отрезке [-3;3] с шагом h (h=0,5; 0,1…).

Этап 1. Заполним столбец с аргументами (можно заполнить и строку).

В столбце A будем располагать значения аргумента (x), в столбце B – значения функции.

В ячейках A1 и B1 запишем заголовки столбцов – «X» и «У», соответственно.

Заполните ячейки столбца А значениями аргумента функции: -3 -2,8 -2,6 ...3 (используя автозаполнение).

Этап 2. Заполним ячейки столбца В значениями функции sin(x).

Сделайте ячейку В2 активной и вызовите «Мастер функций»:

Укажите категорию функции (математические) и выделите нужную функцию в списке (двойным щелчком).

Укажите аргумент функции (в нашем случае аргументом будет ссылка на ячейку, в которой хранится значение x, то есть A2). Можно просто щелкнуть на нужную ячейку таблицы.

Нажмите ОК.

Скопируйте формулу из ячейки В2 на остальные ячейки.

Примечание. В случае, если правила написания функции известны, можно не обращаться к «Мастеру функций», а ввести функцию непосредственно с клавиатуры в строку ввода.

При заполнении ячеек значениями аргумента можно было использовать формулу =A2+0,5 (для ячейки A3) и скопировать ее в остальные ячейки до достижения нужного значения (3) или (более общий случай) в отдельной ячейке (например C2) записать значение h, тогда в ячейке A3 будет формула =A2+C2 (объясните, почему использована абсолютная ссылка).

Вложенные функции

Аргументом функции может быть функция, то есть допускаются вложенные функции. Для задания вложенной функции в Мастере функций в качестве аргумента надо указать функцию (нажать кнопку).

Задание 3.5.11

В столбце С вычислите значения функции $g(x) = \sqrt{\tan(x) - 1}$.

Логические функции

Функция IF задает логическую проверку условия и выполняет различные действия в том случае, если условие выполнено и если не выполнено.

Синтаксис

IF (Условие; Тогда_значение; Иначе_значение)

Условие: любое значение или выражение, которое может иметь значение TRUE или FALSE.

Тогда_значение (необязательно): значение, которое возвращается, если условие выполняется (то есть возвращает значение TRUE).

Иначе_значение (необязательно): значение, которое возвращается, если условие не выполняется (то есть возвращает значение FALSE).

Рассмотрим пример. Пусть в ячейке A1 пользователь запишет число, а в ячейке B1 должен появиться текст «больше 10» или «меньше 10» в зависимости от заданного числа.

Решение: так как мы не знаем какое число будет записано в A1, используем функцию IF.

В B1 вызовем «Мастер функций» и выберем категорию – «Логические».

В списке логических функций дважды щелкнем на функции IF.

Заполним аргументы:

Тест (условие) – A1 > 10

тогда значение – «больше 10»

иначе значение – «меньше 10», нажмите «ОК», чтобы закончить ввод функции.

Заполните ячейку A1 числом. Посмотрите результат в ячейке B1.

Кроме функции IF есть еще логические функции, изучите их самостоятельно.

Рассмотрим пример построения кусочно-заданной функции. На Листе 2 построим таблицу кусочно-заданной функции на отрезке [-2, 4] с шагом 0,1:

$$y = \begin{cases} \cos(x) & 1, \text{ при } x \le 0 \\ \sin(x), & \text{при } x > 0 \end{cases}$$

Заполните столбец A аргументами функции.

В ячейке B2 задайте функцию:

Вызовите «Мастер функций» и выберите логическую функцию IF.

«Тест» в нашем случае имеет вид A2 <= 0.

«Тогда значение» – вложенная функция cos (A2)-1.

«Иначе значение» – вложенная функция sin(A2).

Скопируйте формулу из ячейки B2 в остальные ячейки столбца B.

Задание 3.5.12

Постройте таблицу значений функции:

$$f(x) = \begin{cases} -1, \text{при } x < 0 \\ 0, \text{при } x = 0 \\ 1, \text{при } x > 0 \end{cases}$$

Задание 3.5.13

Создайте таблицу для вычисления стипендии студентам по следующему правилу: если средний балл выше 4,5 – стипендия 1 000 рублей, если средний балл от 4 до 4,5 – 800 рублей. В случае, если средний балл меньше 4, стипендия не выплачивается.

Таблица должна содержать список студентов (не менее 5 фамилий) и оценки по 4 предметам для каждого студента. Средний балл и размер стипендии должны быть вычислены по формулам.

3.5.7. Диаграммы и графики

Для визуального представления большого количества числовых данных используются графики и диаграммы. В электронных таблицах всегда имеются инструменты для построения различных графиков, диаграмм, гистограмм и пр.

Автоматическое заполнение данных на основе содержимого смежных ячеек

Calc предоставляет широкие возможности для построения графиков и диаграмм на основе табличных данных. Для построения диаграммы (график рассматривается как частный случай диаграммы) используется «Мастер диаграмм», который вызывается Вставка – Диаграмма или соответствующей кнопкой на панели инструментов.

Построим график функции $y = sin\ x$. Для этого создайте в столбцах A и B таблицу значений функции $y = sin\ x$ на отрезке [-3;3] с шагом 0,1.

После этого:

Выделите данные в столбцах A и B.

Вызовите «Мастер диаграмм».

Шаг 1. Выбор типа диаграммы. В нашем случае – Диаграмма XY (Далее).

Шаг 2. Указание ячеек – источника данных для диаграммы.

Диапазон: проверить правильность указания диапазона данных или указать необходимый диапазон.

Укажите, как располагаются данные в строках или столбцах.

Шаг 3. Ряд данных. Настройка параметров для каждого ряда. Пока оставляем как есть.

Шаг 4. Элементы диаграммы. Настройка параметров диаграммы – заголовка, легенды и т. д.

Примечание. После того, как процесс создания диаграммы завершен, можно внести изменения не только во внешний вид

диаграммы, но и изменить источник данных для нее. Выделив необходимый элемент диаграммы (оси, сетку, точки графика или столбцы диаграммы, подписи под осями, легенду и т. д.), открыть контекстное меню и внести необходимые изменения.

Задание 3.5.14

Постройте график функции $f(x)=1/x$, где $x \in [-5;5]$ шаг 0,5.

Обратите внимание на поведение графика в точке $x=0$. Для исправления ситуации удалите значение функции в точке $x=0$.

Задание 3.5.15

Постройте семейство косинусоид вида $y=a\cdot \cos(b\cdot x)+c$ на отрезке [0;4] с шагом табуляции 0,2.

Значения констант *a, b, c* занесены в отдельные ячейки.

На листе 3. Начиная с ячейки А6 создайте таблицу значений коэффициентов.

a	***b***	***c***
2	-2	2
-4	0,7	-1
0,4	3	0,3

А ниже, начиная с ячейки А11 и до А14, создайте таблицу значений функции при разных коэффициентах (значения располагаются по строкам).

x
y1
y2
y3

1. В ячейку В11 введите 0, в ячейку С11 введите 0,2. С помощью маркера заполнения заполните 11-ую строку таблицы вправо.

2. В ячейку В12 введите формулу косинусоиды, используя коэффициенты a, b, c из строки 7. Относительными или абсолютными должны быть адреса? Скопируйте формулу вправо.

3. Аналогично введите формулы в В13 и В14, используя последующие значения из таблицы коэффициентов.

4. Выделите полученную таблицу. Постройте диаграмму. Используйте тип «Точечная».

5. Измените некоторые значения коэффициентов в таблице коэффициентов. Как меняются графики функций?

Задание 3.5.16

Постройте параметрическую функцию:

$x=2cos^3t$

$y=2sin^3t$ где $t \in [0 ; 6]$, шаг табуляции 0,2.

Отведем для таблицы строки с 40 по 42. В ячейках A40-A42 запишите:

t
$x=2cos^3t$
$y=2sin^3t$

1. С помощью автозаполнения заполните строку 40 значениями от 0 до 6 с шагом 0,2.
2. В ячейку B41 введите формулу для вычисления значений x. Скопируйте вправо.
3. Введите формулу для вычислений значений y в ячейку B42. Скопируйте вправо.
4. Выделите 41 и 42 строки таблицы и постройте точечную диаграмму.

Задание 3.5.17

Построение поверхности гиперболического параболоида $z = x^2-y^2$

$x \in$ [-3;3], $y \in$ [-3,3], шаг табуляции 0,5.

1. С помощью автозаполнения введите значения x. Заполните столбец A (например строки от 71 до 83) значениями от -3 до 3 с шагом 0,5.
2. Аналогично введите значения y (ячейки B70: N70).
3. В ячейку B71 введите формулу для вычисления значений z.

Подумайте, как использовать здесь абсолютную и относительную адресацию. Вспомните пример с таблицей умножения.

Заполните формулу вправо и вниз на все ячейки таблицы.

4. Выделите блок B71:N83. Постройте диаграмму, тип – линии.

3.5.8. Поиск оптимального решения

Есть целый класс задач, связанных с поиском оптимального решения. Рассмотрим одну из них.

Некоторое предприятие производит продукцию двух видов П1 и П2 из сырья трех видов С1, С2 и С3. Запасы сырья на складе С1 – 200 единиц, С2 – 400 единиц и С3 – 300 единиц. Для производства продукции П1 требуется 2 единицы

сырья С1, 8 единиц сырья С2, 5 единиц сырья С3. Для производства продукции П2 требуется 5 единиц сырья С1, 5 единиц сырья С2, 6 единиц сырья С3. Завод не может выпускать более 100 единиц каждой продукции в день. Доход от продажи единицы продукции П1 – 50 у.е., П2 – 40 у.е. Сколько единиц каждой продукции надо произвести, чтобы прибыль была максимальна?

Для решения подобных задач можно использовать Решатель, который запускается Сервис – Поиск решения (если этого пункта меню нет, то необходимо установить расширение Solver).

Составьте следующую таблицу (рис. 3.17).

	A	B	C	D	E	F
1	Продукт	Кол-во	Затраты сырья			Доход
2			C1	C2	C3	
3	П1	1	2	8	5	=50*B3+40*B4
4	П2	1	5	5	6	
5			=C3*B3+C4*B4	=D3*B3+D4*B4	=E3*B3+E4*B4	
6			200	400	300	
7						
8						

Рис. 3.17. Составление таблицы для поиска оптимального решения

В ячейках B3 и B4 указано произвольное количество вырабатываемой продукции, ячейки C5, D5, E5 содержат формулы для расчета затрат сырья каждого вида, ячейки C6, D6, E6 – запасы сырья на складе, в ячейке F3 вычисляется доход от производства продукции.

1. Выделите ячейку F3.
2. В пункте меню Сервис найдите «Поиск решения» (или Solver). В зависимости от того, какое расширение установлено, откроется окно «Решатель» или окно «Оптимальное решение».
3. В открывшемся окне «Решатель»:
   - Установить целевую ячейку F3;
   - Результат – максимальное значение;
   - Изменяя ячейки – B3:B4
   - Добавить ограничения:
     - Ссылка на ячейку: C5
     - Условие: < =
     - Ограничение: C6

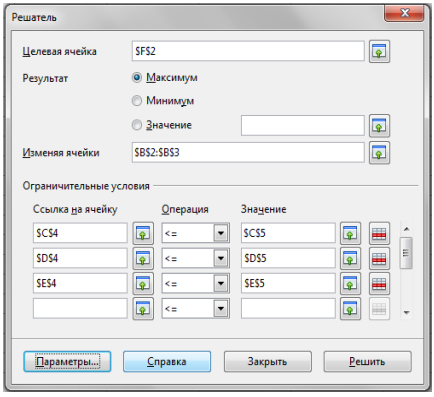


Рис. 3.18. Решатель

- Аналогично для D5, D6 и E5, E6.
- Далее следует установить еще некоторые ограничения, открыв окно «Параметры»: переменные принять как неотрицательные и целочисленные.
- После указания всех ограничений нажать кнопку «Решить».

4. Автоматически должно быть найдено решение: П1 = 40, П2 = 16. Решение необходимо сохранить.

Задание 3.5.18

Фирма занимается составлением диеты, содержащей по крайней мере 20 единиц белков, 30 единиц углеводов, 10 единиц жиров и 40 единиц витаминов. Как дешевле всего достичь этого при указанных в таблице параметрах продуктов?

	Хлеб	Соя	Сушеная рыба	Фрукты	Молоко
Белки	2	12	10	1	2
Углеводы	12	0	0	4	3
Жиры	1	8	3	0	4
Витамины	2	2	4	6	2
Цена	12	36	32	18	10

Задание 3.5.19

Фирма выпускает два набора удобрений для газонов: обычный и улучшенный. В обычный набор входят 3 кг азотных удобрений, 4 кг фосфорных и 1 кг калийных, а в улучшенный

– 2 кг азотных, 6 кг фосфорных и 2 кг калийных удобрений. Известно, что для некоторого газона требуется по меньшей мере 10 кг азотных, 20 кг фосфорных и 7 кг калийных удобрений. Обычный набор стоит 3$, улучшенный – 4$. Сколько и каких наборов удобрений надо купить, чтобы обеспечить потребности и минимизировать стоимость?

3.6. Средства компьютерной графики

Компьютер как устройство для обработки данных применялся изначально для обработки числовой информации, а вовсе не графической. Однако понимание того, что графическое представление данных значительно удобнее для восприятия, чем числовое, привело к возникновению и развитию компьютерной графики.

На первых порах развитие и применение графических изображений, построенных компьютером, сдерживалось несовершенством механических устройств вывода графической информации (принтер, графопостроитель). Первые рисунки были очень примитивны, они выводились в символьном режиме и состояли из различных символов – звездочек, крестиков, букв (псевдографика). Таким образом печатались графики функций, изображения результатов научных исследований и даже художественные изображения (репродукции Джоконды, Эйнштейна и т. д.).

Появление графических дисплеев произвело настоящую революцию в компьютерной графике, так как позволяло получать на экране электронного дисплея не только статические, но и движущиеся и даже взаимодействующие изображения. С тех пор область применения компьютерной графики все время расширяется.

3.6.1. Области применения компьютерной графики

Научная графика

Это направление появилось самым первым. Используется для визуализации (наглядного изображения) объектов научных исследований, графической обработки результатов.

Например, широко применяется компьютерная графика для изучения природных и географических явлений. Графические системы, предназначенные для такого использования, должны обеспечить создание и обработку географических и рельефных карт, океанографических карт, карты погоды и изолиний.

Кроме того, компьютерная графика используется при исследовании астрономических, физических, химических и других явлений.

Инженерная графика (конструкторская)

Основное направление развития систем инженерной графики связано с автоматизацией чертежных и конструкторских работ.

Инженерная графика является обязательным элементом систем автоматизации проектирования (САПР). Эти системы применяются при проектировании компонентов и систем механических, электрических, электромеханических и электронных устройств. Сочетание расчетов с графикой позволяет проводить в наглядной форме поиск оптимальной конструкции, прогнозировать последствия, к которым могут привести изменения в конструкции. Инженерная графика позволяет получать двухмерные изображения (проекции и сечения) трехмерных объектов.

Деловая графика

Системы деловой графики предназначены для графического отображения данных, хранимых в электронных таблицах и БД. Таким образом оформляют плановые показатели, отчетную документацию, статистические сводки и т. д. Обычно исходные данные отображают с помощью графиков, круговых и столбчатых диаграмм.

Художественная и рекламная графика

Позволяет создавать рекламные ролики, мультфильмы, компьютерные игры, видеопрезентации и т. д.

Графические пакеты, предназначенные для решения подобных задач, требуют больших ресурсов компьютера по быстродействию и памяти. Эти программы позволяют создавать реалистические изображения и компьютерную анимацию. Для создания реалистических изображений используется сложный математический аппарат, при помощи которого рассчитываются приближения, удаления, повороты, деформации пространственных объектов, передача освещения, расположения теней с учетом законов оптики. Движущиеся изображения также получаются на основании расчетов. Художник создает начальную и конечную фазу движения объекта, а все промежуточные состояния рассчитываются и изображаются компьютером.

Иллюстративная графика

Программные средства иллюстративной графики позволяют использовать компьютер для произвольного рисования и черчения. Простейшие программные средства этой направленности называются графическими редакторами.

3.6.2. Способы формирования графического изображения

Существует два основных способа представления графических изображений – при помощи растра и векторный.

Растровая графика

Растровое изображение строится из отдельных точек (пикселей), каждая из которых имеет определенное положение и цвет. Изображение описывается конкретным расположением и цветом каждой точки в растровом изображении не существует объектов как таковых (линий, фигур и т. д.).

В памяти компьютера растровое изображение представлено набором кодов: координаты точки (или номер) и код цвета. Если изображение состоит только из черных и белых точек, то для хранения цвета одной точки требуется 1 бит. Чем больше цветов, тем больше бит требуется для хранения цветовой характеристики каждой точки, тем больше будет объем файла с таким изображением.

Растровое изображение характеризуется разрешением (количеством пикселей в изображении) и глубиной цвета (количеством цветов). На качество растрового изображения оказывает существенное влияние то устройство, с помощью которого его просматривают (монитор или принтер). Вывод растровой графики на устройства с более низким разрешением, чем разрешение самого изображения, понизит его качество.

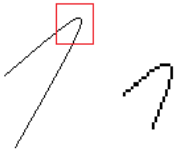


Рис. 3.19. Масштабирование растрового изображения

При изменении размеров изображения (масштабировании), качество изображения ухудшается (см. рис. 3.19). При уменьшении несколько соседних точек преобразуются в одну, поэтому исчезают мелкие детали. При увеличении – увеличивается размер каждой точки, картинка превращается в набор цветных квадратиков.

Растровые изображения могут быть получены при сканировании, фотографировании и видеосъемки при помощи цифровых аппаратов и камер.

Достоинства:

- Растровые изображения очень хорошо передают реальные образы. Они замечательно подходят для фотографий, картин и в других случаях, когда требуется максимальная «естественность».
- Легко выводятся на монитор или принтер, поскольку эти устройства тоже основаны на растровом принципе формирования изображения.

Недостатки:

- Большой объем, необходимый для хранения изображения. Растровое изображение высокого качества может занимать сотни мегабайт памяти. Для обработки их нужны мощные компьютеры.
- Плохая масштабируемость. Любое изменение размеров неизбежно приводит к ухудшению качества: при увеличении пиксели не могут появиться «из ничего», при уменьшении часть пикселов просто выбрасывается.

Векторная графика

Векторное изображение формируется из объектов (точка, линия, окружность, прямоугольник, кривая и т. д.), называемых примитивами. При этом сохраняется не сам рисунок, а правила его построения. Например, для построения круга сохраняются не все пиксели круга, а команда «построить круг радиусом 30 с центром в точке (50,135) и закрасить его красным цветом». Точка задается своими координатами, кривая – координатами начала и конца и уравнением. Поэтому файлы, хранящие векторные изображения, имеют сравнительно небольшой объем.

При редактировании элементов векторного изображения изменяются параметры линий, задающих форму этих элементов. Можно переносить элементы, менять их форму и цвет, но это не отразится на качестве их визуального представления. В векторном изображении все части (примитивы) могут быть изменены независимо друг от друга. Любой из них можно увеличить, повернуть, деформировать, перекрасить и даже стереть – остальных объектов это не изменит.

Векторная графика не зависит от разрешения, то есть может быть показана в разнообразных устройствах вывода с различным расширением без потери качества, так как изображение пересчитывается в растровое представление для данного устройства непосредственно перед выводом на устройство.

Достоинства:

- Небольшой объем файла, хранящего векторное изображение.

- Объекты векторной графики легко трансформируются, ими легко манипулировать, что не влияет на качество изображения.
- Качественное представление изображения на любом устройстве вывода.

Недостаток: не предоставляет возможность для создания фотореалистических изображений.

3.6.3. Фрактальная графика

Фрактальная графика возникла и начала развиваться как средство изучения сложных динамических систем. Результаты компьютерных экспериментов открыли перед учеными удивительный мир графических объектов, которые до этого не попадали в поле изучения науки или были отброшены ею, как «бесформенные».

Термин «фрактал» ввел в употребление в 1975 г. американский математик Бенуа Мандельброт. Фракталами он назвал структуры, обладающие двумя важными признаками: изломанностью и самоподобием (любая часть структуры подобна всему целому). Слово «подобный» не всегда имеет классический смысл «линейно увеличенный или уменьшенный», но всегда находится в соответствии с широким толкованием слова «похожий».

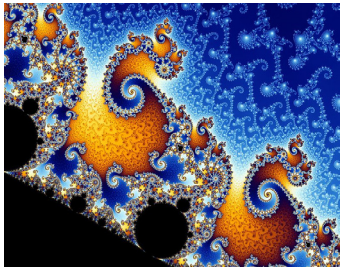


Рис. 3.20. Множество Мандельброта (фрагмент границы)

Понятие фрактала во всей его красоте математических свойств и глубине физических следствий вошло в сознание математиков и физиков после опубликования в 1983 г. книги Мандельброта «Фрактальная геометрия природы». В 1984 г. состоялась выставка «Границы хаоса», на которой были представлены компьютерные изображения фракталов, полученные в Бременском университете. В 1986 г. вышла книга Х.-О. Пайтгена и П. Рихтера «Красота фракталов». Фрактальный бум охватил планету.

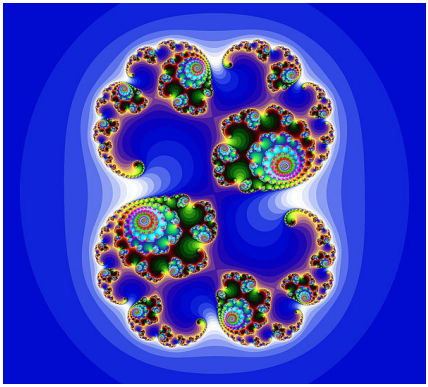


Рис. 3.21. Множество Жюлиа

Фрактальная графика относится к вычисляемой графике, то есть изображение не строится из некоторого набора примитивов, а является результатом выполнения некоторого алгоритма (программы). Свойство самоподобия позволяет получить изображения похожие на реальные природные объекты: деревья, горы, ландшафты, что широко используется в компьютерных играх для создания виртуальной реальности.

3.6.4. Цветовые модели

Для описания цветовых оттенков, которые могут быть воспроизведены на экране или при печати, разработаны специальные средства – *цветовые модели* или *системы цветов*. Наиболее часто используются модели RGB и CMYK.

RGB

В этой цветовой модели цвет каждого пикселя определяется тремя основными цветовыми компонентами: Red (красным), Green (зеленым), Blue (синим), сочетания которых могут дать 16 миллионов различных оттенков. Изображения в данном режиме содержат три канала – по одному на каждый цвет. Пиксель в каждом из каналов описывается 8-битовым кодом и может принимать значения от 0 до 255. Если пиксель имеет значения яркости 0 для каждого цвета, то он имеет абсолютно черный цвет. Если 255, то абсолютно белый.

В режиме RGB работают мониторы, сканеры и телевизоры.

CMYK

Режим используется при подготовке цветных изображений к печати. Изображения в данном режиме содержат четыре кана-

ла, по одному для каждой основной краски: Cyan (голубой), Magenta (пурпурной), Yellow (желтой), Black (черной). Сочетание точек, напечатанных этими четырьмя красками, создает все богатство цветов, получаемых при печати. В режиме CMYK черный пиксель получается смешением всех трех основных цветов, при отсутствии краски получается белый цвет – цвет бумаги.

Две модели RGB и CMYK «противоположны». Изображения в режиме CMYK занимают на 25% больше места, чем в кодировке RGB, поэтому рекомендуется выполнять редактирование в режиме RGB, а затем переводить его в режим CMYK для печати.

3.6.5. Обзор программных средств для создания и обработки графических изображений

CorelDraw, Adobe Illustrator – наиболее популярные векторные редакторы, среди свободного ПО наиболее популярен редактор **Inkscape**. Эти векторные графические редакторы обладают мощным инструментарием для создания векторных изображений. Область применения – деловая графика, макеты обложек для печатных изданий, Интернет-графика.

Adobe Photoshop – самый известный среди коммерческого ПО растровый графический редактор, аналог среди свободного ПО – редактор **GIMP**. Эти растровые редакторы предназначены для обработки фотографических изображений: ретуши, коррекции цветовых и тоновых отношений, добавления оптических эффектов. Основное назначение – допечатная подготовка изображений для последующей публикации в печатных изданиях.

Autodesk 3D Studio, свободный редактор **Blender** – трехмерные графические редакторы. Позволяют создавать трехмерные сцены с имитацией текстур, освещенности, пространственного окружения объектов, сочетая их с анимацией объектов.

AutoCad, КОМПАС-3D (есть бесплатная версия) – системы автоматизированного проектирования (САПР), получившие широкое распространение среди инженеров-проектировщиков. Позволяет проектировать объекты самой различной природы от технических устройств до архитектурных сооружений, имеет возможности для расширения применения системы в различных направлениях благодаря дополнительным пакетам программ.

ArchiCad, 3D Dream House Designer, ArCon – системы для разработки архитектурно-строительных проектов, для разработки интерьеров и ландшафтного дизайна с использованием трехмерной графики. Позволяют создавать фотореалистиче-

ские изображения архитектурных объектов окружающего ландшафта, имеет встроенную библиотеку предметов интерьера.

3.6.6. Универсальные форматы графических файлов

Форматы определяют способ хранения информации в файле (растровый или векторный), а также используемый алгоритм сжатия.

BMP (Bit MaP images) – универсальный формат растрового изображения, используемый в ОС Windows. Поддерживается многими графическими редакторами, рекомендуется для обмена данными с другими приложениями.

TIFF (Tagged Image File Format) – растровый, поддерживается всеми основными графическими редакторами и компьютерными платформами. Использует алгоритм сжатия без потери информации. Рекомендуется для использования при работе с издательскими системами.

GIF (Graphics Interchange Format) – растровый. Использует эффективный алгоритм сжатия без потери информации (поиск повторяющихся в рисунке узоров). Рекомендуется для хранения контурных (с четким контуром) графических изображений с ограниченным количеством цветов.

PNG (Portable network graphics) – растровый формат хранения графической информации, использующий сжатие без потерь. Формат PNG спроектирован для замены устаревшего и более простого формата GIF. Формат PNG рекомендуется для использования в Интернете.

JPG – растровый. Использует эффективный алгоритм сжатия с потерями (данные, которые не могут быть восприняты человеческим глазом). Рекомендуется для многоцветных фотографических изображений, вызывает заметные искажения четко очерченных контурных рисунков.

EPS (Encapsulated PostScript) – формат векторных графических файлов, поддерживается компьютерами разных платформ. Рекомендуется для печати и создания иллюстраций в настольных издательских системах.

WMF (Windows MetaFile) – векторный универсальный формат, используется в Windows-приложениях.

Вопросы для самопроверки

1. Назовите области применения компьютерной графики.
2. Как можно классифицировать компьютерную графику по способу формирования изображения?

3. Какие цветовые модели используются в компьютерной графике?

4. Приведите примеры программ, предназначенных для работы с компьютерной графикой, в том числе и свободное ПО.

5. Перечислите основные форматы графических файлов.

3.7. Растровый графический редактор GIMP

GIMP – GNU Image Manipulation Program. Этот редактор пригоден для решения множества задач по созданию изображений, их изменению, включая ретушь фотографий и множество других возможностей. Программа GIMP многофункциональна.

GIMP – свободное программное обеспечение, выпускаемое под лицензией GPL. GIMP входит в состав большинства дистрибутивов GNU/Linux.

Это кроссплатформенная программа, может быть использована и в таких операционных системах, как Linux, Microsoft Windows и Mac OS X.

GIMP поддерживает форматы файлов: GIF, JPEG, PNG, XPM, TIFF, TGA, MPEG, PS, PDF, PCX, BMP и многие другие.

Официальный сайт проекта GIMP: www.gimp.org.

3.7.1. Окна, панели инструментов, настроек и диалогов

Работа в редакторе организуется с помощью нескольких окон.

Как такового главного окна не существует. Каждое изображение открывается в отдельном окне.

В отдельном окне располагается и Панель инструментов.

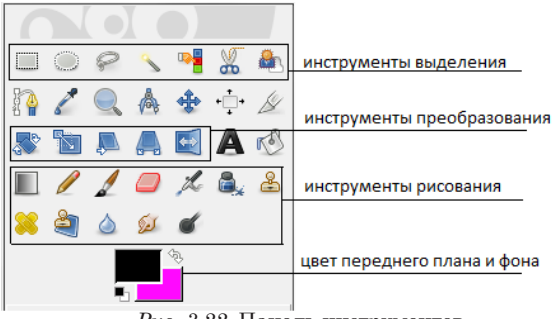


Рис. 3.22. Панель инструментов

Если навести курсор мыши на какой-либо элемент панели инструментов, то появится подсказка о назначении этого элемента.

В панели представлены только основные инструменты. Перечень всех инструментов доступен через меню «Инструменты» окна изображения.

При выборе инструмента параметры его настройки отображаются в окне диалога, по умолчанию прикрепленном к панели инструментов. Если по каким-то причинам это окно отсутствует, то вернуть его можно через меню «Окна» – «Прикрепляющиеся диалоги», перенеся этот диалог под панель инструментов.

Кроме панели инструментов на экране могут располагаться другие панели – вспомогательные окна, предназначенные для выполнения различных операций над изображениями.

По умолчанию вспомогательная панель включает следующие диалоги: «Слои», «Каналы», «Контуры», «История действий», «Цвет», «Кисти», «Текстуры», «Градиенты» и др. Каждый диалог оформлен в отдельной вкладке. Можно нажатием на кнопку войти в меню настройки вкладки и добавить новый диалог или удалить ненужный.

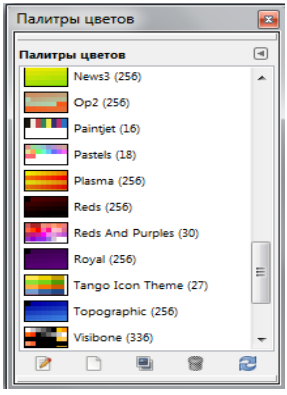


Рис. 3.23. Диалог «Палитры»

3.7.2. Открытие изображения

Команда Файл – Открыть вызывает диалог открытия файла. Если изображение было создано с помощью GIMP, то можно открыть его через меню Файл – Открыть последние. Появится список недавних изображений, над которыми вы работали. При выборе нужной пиктограммы, GIMP откроет изображение.

Находясь в файловом менеджере, можно нажать на пиктограмму нужного файла и перетащить на панель инструмен-

тов GIMP. Изображение откроется в новом окне. Если пиктограмму перетащить на существующее изображение в GIMP, то файл добавится как новый слой или слои этого изображения.

Во многих приложениях возможно нажать на изображение (а не только на пиктограмму) и перетащить его на панель инструментов GIMP.

3.7.3. Отмена действий или операций с изображением

Отменить последнее действие можно используя команду Правка – Отменить.

Если необходимо отменить большее количество операций, то используется вкладка «История действий». Активной является последняя команда списка. Она подсвечена синим цветом. Щелчок по любой команде в списке истории действий возвращает изображение в соответствующее состояние, отменяя все следующие за ней команды. Отменить действия можно и удалив соответствующие команды из списка, нажатием на Del.

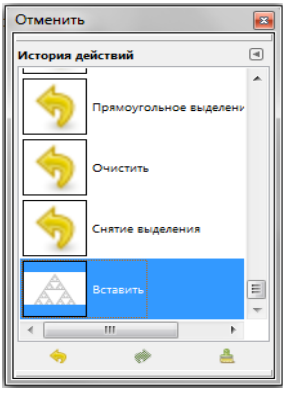


Рис. 3.24. Диалог «История действий»

3.7.4. Масштаб изображения

В GIMP изображения можно просматривать в разном масштабе.

При этом изменяется только экранное представление изображения, а не его фактический размер. В строке состояния окна изображения (внизу слева) выводится масштаб изображения в процентах.

Откройте произвольный графический файл. Увеличьте, а затем уменьшите масштаб этого изображения.

Это можно сделать несколькими способами:

1. Выполнить команду Вид – Масштаб, установить удобный для работы масштаб.

2. Выбрать инструмент «Лупа», установить в параметрах инструмента приблизить или отдалить, и в дальнейшем для быстрого переключения удерживать клавишу CTRL.

3. Открыть окно «Навигации», используя команду Вид – Окно навигации. Панель Navigator (Навигатор) состоит из окна просмотра, строки управления масштабом и кнопками управления (в нижней части). В окне просмотра видно все изображение целиком. Рамка отмечает часть изображения, видимую в окне документа. Перемещая рамку, можно управлять видимой частью изображения в окне изображения.

Для изменения масштаба можно переместить регулятор по шкале вправо для увеличения, влево для уменьшения или использовать кнопки управления масштабом.

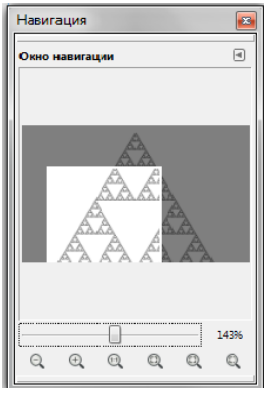


Рис. 3.25. Панель «Навигатор»

3.7.5. Выделение фрагмента изображения

Умение выделять фрагменты изображений имеет большое значение для редактирования и монтажа фотографий. Дело в том, что только выделенную область можно переместить с одного изображения на другое. При ретушировании и цветокоррекции все изменения производятся только в выделенной области и не могут повлиять на остальную часть изображения.

В растровой графике не существует объектов. То, что глаз воспринимает как дерево, цветок, лицо человека – всего лишь набор разноцветных пикселей. Прежде, чем переместить цветок на фотографии или изменить его яркость, необходимо сообщить растровой программе, какая группа пикселей составляет этот цветок, то есть выделить его. Вокруг выделенной области появится мерцающая пунктирная линия.

Изображение, расположенное за пределами выделения, называется маскированной областью. Эта область недоступна для редактирования и, следовательно, защищена от случайных изменений.

В растровой графике не всегда просто создать выделение. Этим объясняется разнообразие средств выделения: инструменты прямоугольное выделение, эллиптическое выделение, лассо (произвольное выделение), волшебная палочка (выделение связанной области), выделение по цвету, умные ножницы (распознавание краев), режим «Быстрая маска» и др.:

- выделение прямоугольником;
- эллиптическое выделение.

Выделение области в виде круга или квадрата: начните выделение, нажмите и удерживайте клавишу Shift.

Выделение области равномерно во все стороны от центральной точки области: начните выделение области, нажмите и удерживайте клавишу Ctrl.

Снять выделение: щелкнуть вне области выделения или выполнить команду Выделение – Снять выделение.

Задание 3.7.1

Найдите графический файл[2*], содержащий групповой портрет, как, например, на рисунке 3.26 (http://www.sxc.hu/photo/1251759). Требуется получить индивидуальные портреты в отдельных файлах (см. рис. 3.27).

Рис. 3.26. Групповой портрет

[2] Использованы свободные фотографии с сайта http://www.sxc.hu.

Рис. 3.27. Индивидуальные портреты

Для этого выделенный объект через буфер обмена скопируйте в новый созданный файл. При создании файла необходимо указать размер холста, на который будет помещено изображение. Размер выделенной области можно посмотреть в параметрах настройки инструмента выделения, расположенных под панелью инструментов.

При работе с фотографией может возникнуть желание отрезать от изображения все лишнее (поля, фрагменты окружающей обстановки и т. д.). Этот процесс называется *кадрированием*.

Задание 3.7.2

Выполните кадрирование изображения на рис. 3.28 (http://www.sxc.hu/photo/1152142) так, чтобы на изображении осталась одна собака (рис. 3.29).

Рис. 3.28. Изображение для кадрирования

Рис. 3.29. Изображение, полученное в результате кадрирования

С помощью прямоугольного выделения обведите рамкой изображение, которое необходимо оставить. Затем выполните команду Изображение – Откадрировать в выделение.

Виньетка, как средство художественного оформления фотографии (особенно портрета) создается просто, но выглядит эффектно.

Задание 3.7.3

Создайте виньетку для оформления портрета (рис. 3.30 http://www.sxc.hu/photo/1282098).

Рис. 3.30. Портрет

Рис. 3.31. Виньетка и портрет

С помощью инструмента «Эллиптическое выделение» выделите эллиптическую область на фотографии.

Растушевка создает плавный переход между пикселями выделенной области и пикселями, окружающими выделенную область. Растушевка определяется в параметрах инструмента выделения, нужно поставить галочку «Растушевать края» и задать радиус растушевки, например, 20. Или можно выполнить команду Выделение – Растушевать и в появившемся окне задать значение растушевки. Пока вы не заметите никаких изменений.

Выполните команду Выделение – Инверсия. Выделенная и маскированная области поменялись местами. Удалите выделенную область. Снимите выделение.

Так как размер изображения больше размера виньетки, остаются белые поля. Уберите их, изменив размер холста. Прикиньте, каков размер виньетки с помощью инструмента «Измеритель». Затем выполните команду Изображение – Размер холста. В открывшемся окне введите новые размеры длины и ширины (10 см х 13 см) и укажите при необходимости смещение.

Однако быстрее выполнить команду Изображение – Автокадрировать изображение.

Для **создания выделений неправильной формы** служат инструменты «Лассо» и «Умные ножницы».


Использование инструмента «Лассо» напоминает процесс рисования карандашом, с помощью которого вы обводите нужный фрагмент изображения.


Инструмент «Умные ножницы» очень удобен при выборе объектов неправильной формы с четким контуром. Граница выделения «прилипает» к линии, разделяющей светлые и темные участки изображения. Щелчком мыши ставятся контрольные точки. Если инструмент неточно определяет нужный контур, можно удалить неправильные контрольные точки и поставить новые контрольные точки щелчком мыши.

3.7.6. Трансформация выделенного фрагмента изображения

Выделенный фрагмент изображения можно трансформировать (менять масштаб, вращать, отображать по вертикали и по горизонтали и др.) при помощи инструментов преобразования  .

Задание 3.7.4

Возьмите фотографию, подобную изображенной на рис. 3.32 (http://www.sxc.hu/photo/815412). Увеличьте количество птиц в исходной фотографии (рис. 3.33).

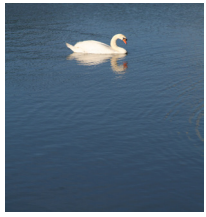


Рис. 3.32. Исходное изображение

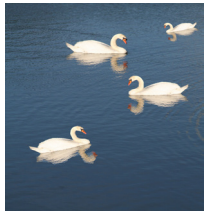


Рис. 3.33. Полученное изображение

Выделите изображение лебедя с его отражением в воде, используя Умные ножницы. Каждый щелчок левой клавишей мыши создает новую контрольную точку, контрольные точки соединяются кривой, идущей по границе в изображении. Для завершения щелкните первую контрольную точку. Кривую можно исправить перемещением контрольных точек или созданием новых точек. Когда процесс выделения окончен, щелкните внутри кривой, чтобы преобразовать ее в выделение.

Выделенную область скопируйте в буфер обмена (Правка – Копировать), а затем вставьте в изображение (Правка – Вставить).

Вставленный фрагмент изображения можно переместить в нужное место фотографии .

Выберите инструмент . Перемещение и перетащите выделенный фрагмент вправо.

Используя механизм перемещения копии и инструменты Масштаб и Зеркало , одну уменьшенную копию лебедя разместите на заднем плане фотографии, а другую, отраженную по горизонтали, справа.

Инструмент «Волшебная палочка» используется для выделения близких по цвету пикселей. При этом степень близости цвета пользователь может задать сам.

Задание 5

Поместите изображение рыбки в аквариум (рис. 3.35). Исходные изображения – рис. 3.34 (http://www.sxc.hu/photo/256923, http://www.sxc.hu/photo/8126).

Рис. 3.34. Исходные изображения

Рис. 3.35. Полученное изображение

Выделите рыбку в исходном файле инструментом «Волшебная палочка».

В панели настройки инструмента «Волшебная палочка», установите порог чувствительности цвета, который определяет, какое количество более светлых и более темных цветовых оттенков, близких к указанному, будет включено в выделение. Чтобы увеличить диапазон оттенков, включаемых в область выделения, нужно ввести большее значение порога. Щелкните Волшебной палочкой по рыбке. Подберите значение порога так, чтобы была выделена вся рыбка. Поскольку цвет зрачка сильно отличается от основного цвета рыбки, нужно выделить его отдельно, но при этом удерживать клавишу Shift (удерживание этой клавиши добавляет новую область в текущее выделение).

3.7.7. Работа со слоями

В GIMP изображения могут содержать один или несколько слоев. Слой можно сравнить с листом прозрачной пленки, на которую нанесен рисунок. Если сложить такие листы стопкой, то получится изображение из нескольких рисунков. GIMP позволяет удалять, перемещать, вращать, масштабировать отдельные слои, изменять порядок слоев, переносить фрагменты изображений с одного слоя на другой. Новые файлы создаются на слое заднего плана (Background). Это означает, что, если отсканировать картинку и открыть ее в программе GIMP, она будет размещаться на слое заднего плана. Слой Background является особым слоем, так как он – всегда самый дальний, не может быть перемещен, не может иметь прозрачных участков.

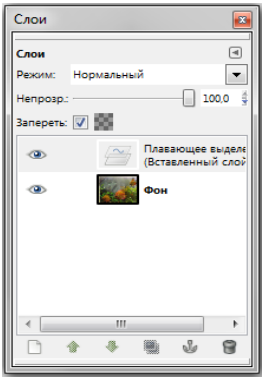


Рис. 3.36. Диалог «Слои»

Для работы со слоями используется вкладка вспомогательной панели «Слои» (рис. 3.36). На ней отображается информация о слоях активного документа. Эта панель позволяет выполнять различные операции над слоями (создавать, удалять, делать видимыми или невидимыми и пр.). Эти же операции можно сделать, используя команды меню «Слой». Активный слой в панели «Слои» выделен синим, видимые слои помечаются пиктограммой «глаз».

Используя операции над слоями, создадим коллаж, в котором соединим фрагменты различных изображений.

Задание 3.7.6

Возьмите несколько фотографий с пейзажем и животными и создайте из этих фотографий коллаж (рис. 3.38). Исходные фотографии на рис. 3.37 (http://www.sxc.hu/photo/1350249, http://www.sxc.hu/photo/1187342, http://www.sxc.hu/photo/535270).

Рис. 3.37. Исходные изображения

Рис. 3.38. Коллаж

В изображении пингвинов создайте удобным вам способом выделение двух пингвинов и скопируйте его в буфер обмена.

В изображении пейзажа создайте новый слой Layer 1, выполнив команду Слой – Создать Слой или нажав на кнопку  в панели «Слои». В диалоге «Новый слой» тип заливки сделайте прозрачным. Убедитесь, что созданный слой явля-

ется видимым и активным. Вставьте из буфера обмена фрагмент изображения животных, например пингвинов, в текущий слой. Используя инструмент «Перемещение», разместите пингвинов в нужном месте. При необходимости измените размер группы пингвинов, используя инструменты преобразования. Аналогично в тот же слой вставьте изображение двух оставшихся пингвинов из файла.

Используя буфер обмена, вставьте изображение лисы в другой новый слой. Отразите изображение лисы по горизонтали и уменьшите его.

Расположите лису и пингвинов так, как показано на рисунке.

Порядок расположения слоев можно менять (перемещать вперед, назад). Для этого используйте одну из команд в отрывающемся меню Слой – Стопка Слоев или кнопки со стрелками в панели «Слои».

Попробуйте изменить настройку непрозрачности слоя (применяется к активному слою). Непрозрачность определяется диапазоном от 0 до 100, где 0 означает полную прозрачность, и 100 означает полную непрозрачность. Непрозрачность настраивается в панели «Слои» с помощью регулятора непрозрачности.

Альфа-канал кодирует информацию о том, насколько прозрачен слой в каждой точке. При создании нового изображения оно состоит только из одного слоя. Если изображение было создано с непрозрачным типом заполнения, то у этого слоя нет альфа-канала. Если добавить новый слой, даже с непрозрачным типом заполнения, альфа-канал создается автоматически. Это относится ко всем слоям, кроме фонового. Чтобы получить фоновый слой с прозрачностью либо создайте новое изображение с прозрачным заполнением, либо используйте команду Слой – Прозрачность – Добавить альфа-канал.

Несколько слоев изображения можно связать и затем воздействовать на них, как на группу элементов. Для этого достаточно щелкнуть по пустой пиктограмме справа от пиктограммы «глаз». При этом появляется символ связи цепь. Связанные слои будут перемещаться вместе.

Каждый новый слой, добавляемый к документу, увеличивает размер файла. Чтобы избежать излишнего увеличения размера файла, рекомендуется объединить слои, работа с которыми закончена, используя команду Слой – Объединить с предыдущим. При этом теряется возможность работать с каждым слоем отдельно.

3.7.8. Тоновая коррекция

Средства регулировки цветовых оттенков позволяют превращать тусклые и плохо окрашенные изображения в яркие и красочные. Хотя результат коррекции некоторых фотографий и рисунков может оказаться великолепным, чудес не бывает. Если в оригинале отсутствуют некоторые детали, то программа редактирования не сможет создать их из «ничего». Не стоит ждать больших результатов, если исходное изображение имеет очень низкое качество.

Тона (оттенки цветов) изображения характеризуются яркостью пикселей. Эти яркости лежат в диапазоне от 0 до 255. Самому темному оттенку соответствует яркость 0, а самому светлому – 255. Диапазон яркостей пикселей изображения называется тоновым диапазоном изображения. Например, если тоновый диапазон фотографии лежит в пределах от 100 до 255, то такая фотография выглядит слишком светлой, так как в ней отсутствуют темные оттенки. При тоновом диапазоне от 0 до 70 изображение, наоборот, будет очень темным из-за отсутствия светлых тонов.

Для каждого изображения GIMP показывает распределение яркостей пикселей в виде гистограммы, которая вызывается Цвет – Инфо – Гистограмма.

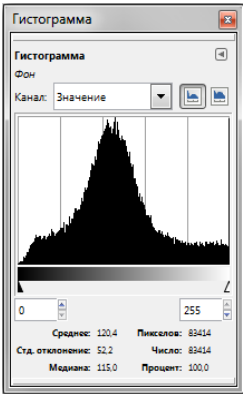


Рис. 3.39. Гистограмма

По горизонтальной оси располагаются значения яркостей: от 0 до 255, а по вертикальной – количество пикселей каждого уровня яркости (рис. 3.39). Тени – самая темная часть изображения с малым значением яркости. Светлые тона – самая

светлая часть изображения с большим значением яркости. Между тенями и светлыми тонами находятся средние тона.

Откройте в редакторе темное изображение (рис. 3.40), светлое изображение (рис. 3.41) и, например, тусклое изображение (рис. 3.42). Проанализируйте распределение яркостей пикселей различных изображений, выполнив команду Цвет – Инфо – Гистограмма для каждого изображения.

Основная задача тоновой коррекции – обеспечить правильное распределение яркостей пикселей в изображении.

Рис. 3.40. Затемненное изображение

Рис. 3.41. Светлое изображение

Рис. 3.42. Неконтрастное изображение

Необходимость тоновой коррекции объясняется рядом причин. Очень часто, когда изображение оцифровывается с помощью сканера, его тоновый диапазон сужается (из-за особенностей этого устройства). Поэтому отсканированная фотография на экране выглядит тускло по сравнению с оригиналом. С другой стороны, невысокое качество фотографии может быть связано с непрофессиональной съемкой. С помощью GIMP такой оригинал можно существенно улучшить.

Обычно для тоновой коррекции используются команды, находящиеся в меню «Цвет»: Яркость – Контраст, Уровни, Кривые, Авто (Автоматическая коррекция).

Команда «Яркость – Контраст» – самое простое, но и наименее гибкое средство. Команда «Уровни» более эффективна, однако наилучшие результаты обеспечивает команда «Кривые».

Задание 3.7.7

Выполните тоновую коррекцию какой-нибудь бледной тусклой фотографии с использованием команды Яркость – Контраст.

Для сравнения результата с оригиналом сначала создайте копию изображения Изображение – Создать копию и с ней работайте.

В диалоговом окне Яркость – Контраст перемещайте регулятор по шкале Яркость, что приведет к затемнению или осветлению изображения. Регулятор Контраст позволяет улучшить четкость изображения.

Задание 3.7.8

Выполните тоновую коррекцию этой же фотографии с использованием команды Цвет – Авто. Сравните результаты, полученные при помощи автоматической коррекции и с использованием команды Яркость – Контраст.

Эта команда выполняет перераспределение яркостей пикселей автоматически. Анализируя изображение, GIMP находит самый светлый и самый темный пиксели и определяет их как белый и черный, а все остальные тона распределяет между ними.

Иногда результаты выполнения этой команды бывают неудовлетворительными.

Задание 3.7.9

Выполните тоновую коррекцию изображения с использованием команды Кривые.

Настройка кривых позволяет настраивать цвета более тонко.

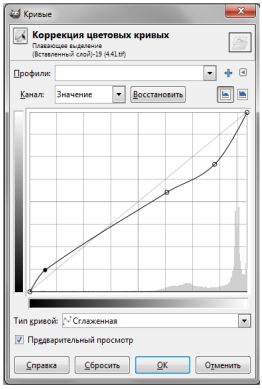


Рис. 3.43. Коррекция цветовых кривых

Когда вы открываете диалоговое окно «Кривые», самой кривой вы не увидите. Вместо этого вы увидите сетку, на которой расположена прямая линия, направленная по диагонали. Горизонтальная ось этого графика соответствует исходным значениям (входным уровням) изображения или выделенной области, а вертикальная ось соответствует новым значениям (выходящим уровням). При первом открытии этого диалогового окна график отображается в виде прямой, так как значения еще не изменялись. Все пиксели имеют одинаковые входящие и выходящие значения.

Можно поставить на кривой несколько контрольных точек и с их помощью отрегулировать вид кривой и, следовательно, яркость изображения. На изображении выбирайте точки, цвета которых необходимо откорректировать. В результате вы увидите эти точки на графике. Перетягивайте точки, соответствующие темным тонам, вверх, а точки, соответствующие светлым тонам, – вниз. Чтобы удалить точку, которую вы поставили, перетяните ее за пределы графика.

3.7.9. Рисование

В GIMP к инструментам рисования относятся:

«Кисть», «Карандаш», «Аэрограф», «Ластик», «Заливка», «Градиент», «Штампы», «Палец», «Осветление и Размывание».

Рис. 3.44. Панель выбора цвета

Цвет переднего плана (или основной) используется для рисования, заливки выделенных областей, а также в качестве начального цвета градиента.

Цвет заднего плана (или фоновый) можно сравнить с цветом холста, на который наносится краска. Именно этот цвет появляется при удалении пикселей изображения (после удаления краски виден холст). Кроме того, фоновый цвет используется для завершения градиента.

По умолчанию основной цвет – черный, а фоновый – белый (рис. 3.44). Цвета переднего и заднего плана указываются в цветовых полях на панели инструментов. Используя «Переключатель цветов», основной и фоновый цвета можно менять местами. Щелчок на пиктограмме «Основной и фоновый цвета по умолчанию» восстанавливает черный цвет в качестве основного, а белый – в качестве фонового.

Щелчок по пиктограмме «Фоновый цвет» или «Основной цвет» откроет окно «Палитра цветов», с помощью которой устанавливается нужный цвет.

Кончик инструментов рисования имеет определенный размер и форму – размер кисти и форму кисти. Кроме того, каждый инструмент имеет параметры, которые определяют характер их работы. Форма инструмента отображается во вкладке «Кисти» вспомогательной панели, а параметры – в панели «Параметры инструмента», настройка «Масштаб» определяет размер кончика инструмента рисования.

Задание 3.7.10

Установите размер и параметры карандаша и создайте простой рисунок.

В панели «Параметры карандаша» обратите внимание, что исходная степень непрозрачности карандаша равна 100%. Откройте вкладку «Кисти» вспомогательной панели, где установите форму кончика карандаша и ее размер.

Если вам не нравится то, что получается, можно ластиком стереть неудачные штрихи.

Чтобы закрасить фрагменты рисунка, воспользуйтесь инструментом «Заливка».

Выберите цвет заливки (основной цвет) и закрасьте замкнутую область.

3.8. Системы компьютерной математики (математические пакеты)

Еще недавно для решения математических задач пользователь должен был не только разбираться в математике, но и освоить язык программирования (и не один), знать сложные численные методы. Сейчас разработаны математические пакеты, которые позволяют решать различные математические задачи.

Ранние математические пакеты были ориентированы только на ***численные расчеты***, для чего использовались возможности развитых численных методов. Но применение численных методов имеет ряд недостатков. Это, прежде всего, нахождение только частных решений и некоторая погрешность вычислений.

Высочайшим достижением стали системы компьютерной математики, выполняющие ***аналитические (символьные) преобразования***. Еще недавно это было фантастикой. Аналитические методы решения задач дают общие (формульные) результаты и без погрешностей. Поэтому сейчас системы аналитических вычислений – наиболее интересное и развивающееся направление компьютерной математики.

Современные системы компьютерной математики обеспечивают и численные и символьные вычисления.

Спектр задач, решаемых подобными системами, очень широк:

- проведение математических исследований, требующих вычислений и аналитических выкладок;
- разработка и анализ алгоритмов;
- математическое моделирование и компьютерный эксперимент;
- анализ и обработка данных;
- визуализация, научная и инженерная графика;
- разработка графических и расчетных приложений.

3.8.1. Команды главного меню

Обычно главное меню систем компьютерной математики включает следующие команды.

Author – ввод пользователем исходной задачи.

Build – построение математических выражений из имеющихся фрагментов.

Calculus – вычисление.

Declare – объявление переменных, векторов, матриц и функций.

Expand – раскрытие выражения, перемножение его членов, упорядочение по степеням.

Factor – разложение на простые множителя чисел и выражений.

Solve – решение уравнений.

Simplify – упрощение выражений.

Approx – приближенное вычисление выражений.

3.8.2. Особенности работы с математическими пакетами

Современный уровень развития компьютерной алгебры не исключает отказа от выполнения символьных вычислений (система возвращает исходное выражение). Многие преобразования имеют ограниченную область применения. Поскольку при упрощении выражений система, должна выполнять только корректные преобразования, компьютеру может потребоваться квалифицированная помощь: подсказать направление дальнейших преобразовании, сузить класс значений параметра. Поэтому во многих системах имеются средства указания областей определения переменных.

Результаты некоторых преобразований выглядят неожиданными. Такие сюрпризы могут быть связаны с неоднозначностью функций, их взаимозаменяемостью и главным образом – с вынужденной стандартизацией методов решения. Именно поэтому системы компьютерной математики не могут справиться с отдельными задачами, решения которых вошли в широко известные справочники. Это объяснимо: красивые нестандартные решения копились сотни лет и иногда достигались весьма искусственными приемами. Поэтому, например, при сравнении результата интегрирования с табличным следует иметь в виду возможность эквивалентных преобразований и различного представления констант. В таких случаях нужно получить разность интегралов, упростить ее и убедиться в том, что эта разность не зависит от переменной интегрирования.

Результатом символьного решения дифференциальных уравнений в общем случае является уравнение, связывающее

зависимую и независимую переменные. Полезно выполнить его дифференцирование и затем контрольную подстановку.

При решении нелинейных уравнений и систем уравнений обычно требуется указание начальных приближений или области нахождения корней. В первом случае, как правило, находится единственное решение, в «области притяжения» которого оказалось исходное приближение, и для получения нового решения требуется сменить исходную точку. Если возможны комплексные корни, то, по крайней мере, одно приближение должно быть комплексным. Во втором за одно обращение возможно получение нескольких решений, но можно остаться ни с чем, поскольку процесс при выходе из указанной области завершается аварийно. «Свободный» поиск не имеет этих недостатков, но идет дольше и может завершиться отказом по исчерпанию предельного числа шагов. Во многих пакетах имеется возможность применить для решения системы несколько процедур, различающихся способами подготовки исходных данных.

В математических пакетах, конечно же, есть ошибки как на концептуальном уровне, так и на уровне программной реализации. Поэтому необходимо критически относится к полученным результатам и осуществлять проверку. Такие возможности может дать только знание математики.

3.8.3. Наиболее распространенные системы компьютерной математики

Самыми богатыми возможностями обладают профессиональные пакеты *Mathematica* и *Maple*. Они же предъявляют самые высокие требования к аппаратуре и наиболее сложны в освоении.

Mathematica

Система Mathematica разработана фирмой Wolfram Research и является мощным средством выполнения математических исследований как в символьной, так и в численной форме. Система справедливо считается мировым лидером среди компьютерных систем символьной математики. Она используется в ведущих университетах мира и получила широкое распространение в образовательных учреждениях всех континентов. Любая серьезная научная лаборатория или кафедра вуза должна иметь подобную программу, если там всерьез заинтересованы в автоматизации выполнения математических расчетов любой степени сложности.

Mathematica демонстрирует высокую скорость символьных преобразований и численных расчетов. Программа

Mathematica наиболее полна и универсальна. Ввод может осуществляться как с помощью палитр математических знаков, так и с использованием команд специального входного языка. Mathematica как система программирования имеет все возможности для разработки и создания практически любых управляющих структур, организации ввода-вывода, работы с системными функциями и обслуживания любых периферийных устройств, а с помощью пакетов расширения (Add-ons) появляется возможность подстраиваться под запросы любого пользователя. На базе Mathematica создано около 100 специализированных коммерческих пакетов.

Сильной стороной этой системы является развитая двух- и трехмерная графика. Набор функций графики и изменяющих их действие опций очень широк.

Разработчик поддерживает в сети Internet свободный доступ к большому числу научных, методических и учебных продуктов, созданных сотрудниками фирмы и пользователями, число которых превысило миллион.

Maple

Maple – это среда для выполнения символьных, численных и графических вычислений профессиональными математиками, разработанная фирмой Waterloo Maple Software (University of Waterloo, Канада) и Высшей технической школой в Цюрихе. Она воплотила колоссальный математический потенциал, включает широчайший арсенал средств («от элементарном арифметики до общей теории относительности») и активно используется в научной среде. Программа Maple является одним из лидеров среди универсальных систем символьных вычислений. Она предоставляет пользователю удобную интеллектуальную среду для математических исследований любого уровня и пользуется особой популярностью в научной среде. Символьный анализатор программы Maple является наиболее сильной частью этого ПО, поэтому именно он был позаимствован и включен в ряд других пакетов, таких как MathCad и MatLab, а также в состав пакетов для подготовки научных публикаций Scientific WorkPlace и MS Math.

Система имеет очень удобный пользовательский интерфейс. Набор осуществляется с помощью входного языка или с использованием палитры математических знаков. В систему встроен хороший текстовый редактор, позволяющий выполнять форматирование шрифта, абзаца. Maple позволяет преобразовать математический текст в формат TeX.

Набор графических возможностей уникален. При построении двумерных графиков Maple поддерживает 15 систем

координат, а в трехмерном случае – 31 (с возможностями преобразования из одной системы в другую). Графические средства Maple позволяют строить двумерные графики сразу нескольких функций, строить графики функций в логарифмической, двойной логарифмической, параметрической, фазовой, полярной и контурной форме. Можно графически представлять неравенства, неявно заданные функции, решения дифференциальных уравнений и т. д.

Maple может строить поверхности и кривые в трехмерном представлении, включая поверхности, заданные явной и параметрической функциями, а также решениями дифференциальных уравнений. При этом представлять можно не только в статическом виде, но и в виде двух- или трехмерной анимации. Эту особенность системы используют для отображения процессов, протекающих в режиме реального времени.

Maxima

Maxima относится к классу свободного ПО. В программе Maxima для математической работы используется язык, сходный с языком в пакете Mathematica, а графический интерфейс построен по тем же принципам. Изначально программа называлась Xmaxima и создавалась для UNIX-систем. Maxima имеет мощный, эффективный и дружественный графический интерфейс.

Mathcad

Пакет *Mathcad* (MathSoft Inc., 1986–1998 гг.) является одним из наиболее удобных для несложных расчетов на ПЭВМ. Пакет имеет *естественный* входной язык представления математических выражений и инструменты их набора (палитры математических знаков), именно очень дружелюбный пользовательский интерфейс сделал Mathcad очень популярным, несмотря на скромные математические возможности.

Это физико-математический пакет: он позволяет вводить размерности переменных задачи и автоматически контролирует соответствие размерностей операндов и результата. Mathcad имеет встроенный текстовый процессор, который позволяет оформить статью без помощи специализированных текстовых редакторов.

Вместе с пакетом могут использоваться прикладные дополнения (обработка сигналов и изображений, анализ электрических цепей, численный анализ, статистика, теория очередей), включающие в себя электронные гипертекстовые книги и программные компоненты.

MatLab

Своим названием (MATrix LABoratory) система MatLab обязана ориентации на матричные и векторные вычисления.

MatLab прошла многолетний путь развития от системы для больших ЭВМ до интегрированной среды, ориентированной на массовые ПК (с середины 80-х). MatLab – хорошо апробированная и надежная система, рассчитанная на широкий круг задач и представление данных в универсальной (матрично-векторной) форме.

MatLab называют «симфонией алгоритмов». Считается, что эта система фактически стала международным стандартом учебного программного обеспечения. Она используется более чем в 70 ведущих университетах мира, включая Стэнфордский, Кембриджский и Калифорнийский, и в таких отечественных вузах, как МАИ, МГТУ, МГУ, МИФИ, Балтийский университет.

Недостатки MatLab – отсутствие спецзнаков математических операторов (корня, интеграла и т. д.), невозможность редактирования ранее введенных команд, комбинирования текста и графиков и ряд других неудобств в работе с последними, ограниченные возможности символьных вычислений.

В целом MatLab можно характеризовать как мощную и хорошо сбалансированную математическую систему, ориентированную преимущественно на инженерные приложения теории управления, электро- и радиотехники.

Вопросы для самопроверки

1. Опишите структуру системы компьютерной математики.
2. Какие особенности следует учитывать при работе с системами компьютерной математики?
3. Приведите примеры современных систем компьютерной математики. Какие достоинства и недостатки имеет каждая из них?

3.9. Математический пакет MAXIMA

Скачайте дистрибутив пакета с сайта: http://sourceforge.net/projects/maxima/files/ и установите на своем компьютере.

После запуска Maxima появляется окно программы.

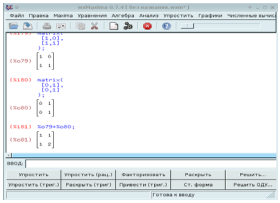


Рис. 3.45. Окно Maxima

Основные элементы окна:

- Меню.
- Панель инструментов.
- Рабочая область.
- Строка ввода.
- Кнопки наиболее часто употребляемых команд.

Рассмотрим некоторые возможности этого математического пакета.

3.9.1. Ввод выражений

Осуществляется в строке ввода (ВВОД:). Последовательность выполнения вычислений:

1. Набрать с клавиатуры в строке ввода выражение, содержащее символы, числа, строки. В конце команды рекомендуется ставить точку с запятой «;».
2. Запустить вычисление выражения, нажав Enter.

Если выражение не содержало ошибок, то система вычислит его. Одновременно с этим набранное выражение будет помечено %i (номер) – это входная ячейка, а появившийся ответ – %o (номер) – это выходная ячейка. При дальнейших вычислениях можно использовать и входные данные, и результаты, указывая номер строки.

Например, %o79 + %o80 (рис. 3.45).

3.9.2. Численные вычисления

Арифметические выражения составляются с использованием знаков арифметических действий и с соблюдением обычных соглашений о порядке действий.

+ сложение, - вычитание, * умножение, / деление, ^ степень.

Если необходимо вычислить приближенное значение, то надо применить оператор numer. Например, 3/8+2/3, numer.

Иначе результат будет выдан в виде обыкновенной дроби.

Функции численных выражений

Функция	**Описание**
Оператор numer	Приближенное числовое значение выражения, указанного перед оператором
mod(m,n)	Сравнение m по модулю n (остаток от деления m на n)
quotient(m,n)	Целая часть от деления m на n

gcd(n,m,…)	НОД(n,m,…)
lcm(n,m,…)	НОК(n,m,…)
factor(n)	Разложение на простые множители числа n

Задание 3.9.1

1) Выполните вычисления

45+98

34*(26-78)

1/5+3/11

$(1+3)^3$-5(2+4)

7^{25}

2) Определите, что больше – e^p или π^e?

Примечание. Для задания константы *e* надо написать %e. Аналогично, для π: %pi.

3) Найдите наибольший общий делитель чисел 2 476 и 5 634 и убедитесь в правильности ответа, разложив эти числа на простые множители.

Некоторые элементарные математические функции

Функция	**Описание**
sqrt(x)	Квадратный корень
exp(x)	e^x
log(x)	Натуральный логарифм
sin(x), cos(x), tan(x), cot(x), asin(x), acos(x), atan(x), acot(x)	Тригонометрические функции
N!	Факториал
abs(x)	Модуль
round(x)	Ближайшее целое число
Mod()	Остаток от деления
random(N)	Случайное число между 0 и N
max(x,y,…) min(x,y,…)	Наибольшее и наименьшее число из набора

3.9.3. Символьные вычисления

В выражениях можно использовать и переменные.

Maxima использует знак «:» для присвоения значений, например, a : 3; и «:=» для определения функций, например, f(x) := x^2;.

Пример:

Определите функцию:

t(x):=(x+1)^4-x^4-2*x-1.

Задайте значение переменной *x*:

x:4.

Теперь, если во входной строке написать t(x), в выходной получим значение функции t в точке 4.

Если есть необходимость отменить присвоение значений переменной, используется метод kill, например kill(x) удаляет значение переменной *x*. Можно удалить значения всех переменных kill(all).

3.9.4. Преобразование рациональных выражений

Название функций для работы с рациональными выражениями начинаются с символов rat.

1. Преобразование выражения к каноническому виду.

rat(выражение), например,

rat((x-1)^2/(x^2+x)+1/(x+1)+0.25)

даст ответ $\frac{5x^2-3x+4}{4x^2+4x}$

2. Раскрытие скобок в рациональном выражении:

ratexpand(выражение), expand(выражение)

Сравните:

ratexpand((a+b)^(x^2+(2-x)*(2+x)),

expand((a+b)^(x^2+(2-x)*(2+x)).

3. Разложить на множители многочлен:

– в строке ввода задать многочлен, например, 2*x^4+4*x^2+2*x;

– в нижней части окна нажать кнопку «Факторизовать» или выбрать в меню: Упростить – Факторизовать выражение.

4. Раскрытие скобок в многочлене:

– в строке ввода задать выражение;

– в нижней части окна нажать кнопку «Раскрыть» или выбрать в меню: Упростить – Раскрыть выражение.

Функция	Описание
Expand(f)	Раскрыть скобки в выражении f
Factor(f)	Разложить многочлен на множители

Также можно выполнить следующие операции над многочленами: найти частное от деления многочлена на многочлен (Анализ – Делить полиномы), НОД и НОК (в пункте меню «Анализ»).

Задание 3.9.2

1) Разложите на множители $-1-x^2+x^3+x^5$.

2) Раскройте скобки в выражении $(1+x)(2+2x+3x^2)(x^3+4x+3)$, проверьте правильность ответа с помощью обратной функции.

3.9.5. Решение уравнений

Для решения уравнений служит пункт меню «Уравнения».

Например, решим уравнение 2*x^3-3*x^2+6*x+4=0.

Для этого

– зададим в строке ввода левую часть уравнения: 2*x^3-3*x^2+6*x+4;

– выбираем в меню: Уравнение – Решить, проверяем правильность выражения и задаем переменную, по которой решаем уравнение;

– нажать «ОК».

Для решения уравнений может использоваться функция:

solve ([аргументы], [переменные]).

Аргументами являются левая часть уравнения или система уравнений, записанных через запятую, далее – переменная, относительно которой решается уравнение или список переменных.

Например, solve([2*x^3-3*x^2+6*x+4],[x])

Задание 3.9.3

Решите уравнение:

$$x^2 + \sqrt{20 + x^2} = 22$$

Решение систем уравнений

Для решения систем используется пункт меню: Уравнения – Решить линейную систему... или Уравнения – Решить алгебраическую систему...

– задается количество уравнений в системе;

– задаются левые части уравнений и список переменных;

– нажмите «ОК».

Для решения системы уравнений может быть использована функция Solve, при этом уравнения системы записываются в квадратных скобках через запятую, список аргументов – в квадратных скобках через запятую.

Например, solve([x^3-3*y^2, x+y-2],[x,y]).

Задание 3.9.4

Решите следующие системы уравнений:

1) $\begin{cases} x + y = 2 \\ x - y = 4 \end{cases}$

2) $\begin{cases} (x + 0{,}2)^2 + (y + 0{,}3)^2 = 1 \\ x + y = 0{,}9 \end{cases}$

3.9.6. Вычисление сумм

Для вычисления суммы ряда используется функция sum, которая может быть вызвана командой меню Анализ – Вычислить сумму и заполнить форму (рис. 3.46).

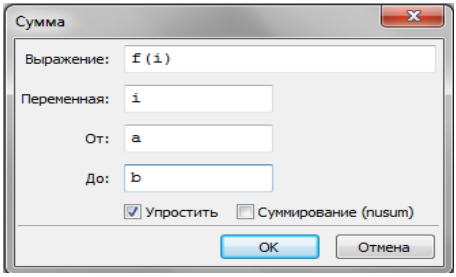


Рис. 3.46. Окно вычисления суммы

Вычисление суммы $\sum_a^b f(i)$: sum(f(i), i, a, b);

Задание 3.9.5

Вычислите следующие суммы:

1) $\sum_1^\infty \frac{1}{i^2}$;

2) $\sum_1^\infty \frac{1}{i!}$.

3.9.7. Вычисление пределов и дифференцирование

Для нахождения пределов нужно выбрать пункт меню Анализ – Найти предел и заполнить следующую форму (см. рис. 3.47).

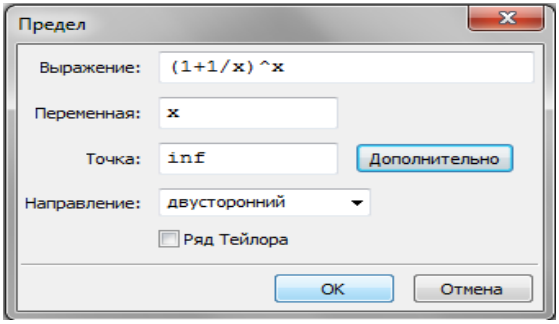


Рис. 3.47. Окно вычисления пределов

Или использовать функцию:
limit(выражение, переменная, предел)
Например, limit((1+1/x)^x, x, inf);

Для нахождения производной надо выбрать пункт меню Анализ – Дифференцировать. В окне «Дифференцировать» задается выражение, переменная, по которой надо искать производную и множитель – порядок производной (рис. 3.48).

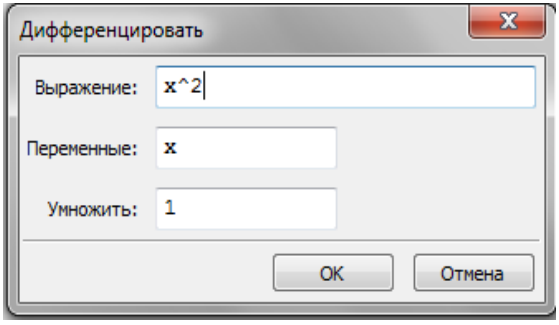


Рис. 3.48. Окно дифференцирования

Можно использовать функцию diff, например, diff(x^2, x).
Если нужно найти производную более высокого порядка, то в поле «Умножить» введите порядок производной.

Задание 6
Вычислите пределы:
1) $\frac{1}{\ln x} - \frac{1}{1-x}$ при $x \to 1$;

2) $\frac{\sin(nx)}{x}$ при $x \to 0$;

3) $\frac{x^2+x+2}{x^2-2x-3}$ при $x \to 3$ справа и слева и при $x \to \infty$;

4) Вычислите производные функции $e^{\frac{1}{x}}$ с 1 по 5 порядок двумя способами (последовательно и сразу).

3.9.8. Интегралы

Для нахождения интегралов используется пункт меню Анализ – Интегрировать.

Задание 7

Вычислите интегралы:

1) $\int_1^2 \frac{2^{\sqrt{x}}}{\sqrt{x}} dx$;

2) $\int \sqrt{10^{3x} dx}$;

3) $\int x^2 dx$;

4) $\iiint xdxdydz$.

3.9.9. Графики функций

1. Построим график функции $y=x^2$ на отрезке [-4;4].

Пункт меню: Графики – График 2D. Откроется окно параметров графика (рис. 3.49).

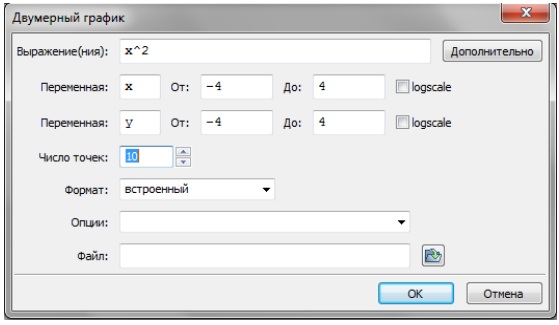


Рис. 3.49. Окно параметров графика 2D

– В поле «Выражение» задаем функцию: x^2.
– В поле переменная x из -4 к 4.
– Количество точек – 10 (как по умолчанию).

2. Построим поверхность $z=x^2-y^2$ на отрезке [-5;5] (рис. 3.50).

Пункт меню: Графики – График 3D.

– В поле «Выражение» задаем фунцию: x^2-y^2.
– В поле переменная x из -5 к 5.
– В поле переменная y из -5 к 5.
– Сетка 30*30 (как по умолчанию).

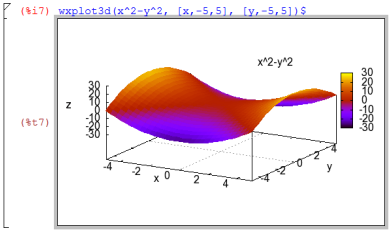


Рис. 3.50. Поверхность

Примечание. В поле выражение можно задавать несколько функций через запятую, чтобы построить несколько графиков на одном чертеже.

3. Для построения графиков функций, заданных параметрически:

- В окне «График 2D» нажать кнопку «Дополнительно» и выбрать «Параметрический график».
- Задать параметрические уравнения функции, диапазон изменения аргумента.
- Нажать «ОК».

Для построения графика кривой, заданной параметрически используют параметр parameric. В общем случае команда выглядит так:

Plot2D([paramentric, x-выражение, y-выражение, [параметр, начало, конец], [nticks, количество]]),

где x-выражение, y-выражение задают зависимость координат от параметра, nticks, количество – количество кусочков, на которые будет разбит интервал изменения параметра.

Пример. Построим параметрическую окружность

Plot2D([paramentric, cos(t), sin(t), [t, -%pi, %pi], [nticks, 300]])

Задание 3.9.8

Постройте следующие графики:

1) Функции, заданной параметрическими уравнениями: $x = t^2$, $y = t^* (3-t^2)$.

2) На одном графике постройте кривые x^3+3x^2+1 и x^2 установите равные длины единичных отрезков по осям.

3) Постройте на одном чертеже параметрические кривые x1 = 2cos(t)^3, y1 = 2sin(t)^3 и
x2 = sin (t/3)*cos(t), y2 =sin(t/3)*sin(t).

4) Постройте трехмерный график cos(x+y)+sin(x-y) при -3<x<3p и -3<y<3p.

3.9.10. Операции с матрицами

Чтобы задать матрицу выполните: Алгебра – Ввести матрицу и задайте элементы матрицы.

Поэлементные операции (сложение, вычитание, умножение, деление, возведение в степень) – производятся над каждым элементом матрицы, матрицы должны быть одинакового размера.

Например, x и y матрицы.

1) Определим эти матрицы:
x : matrix([5,2],[3,9]);
y : matrix([1,3],[2,5]).

2) Вычислим сумму и разность:
x+y;
x-y.

Матричные операции:

- Умножение (обозначается точкой).
- Возведение в степень матрицы (умножение саму на себя заданное число раз).
- Поиск обратной матрицы invert(x).
- Ранг матрицы – rank(x) и определитель – determinant(x), только для квадратных матриц.
- Транспонирование – transpose(x).

Задание 3.9.9

Решите систему линейных уравнений матричным способом.

$$\begin{cases} 5x_1 + 3x_2 - 2x_3 + x_4 = 9 \\ 2x_1 + 3x_2 + 4x_3 - 5x_4 = 0 \\ 3x_1 - 4x_2 + x_3 - x_4 = 6 \\ 2x_1 - 2x_2 + x_3 + x_4 = 5 \end{cases}$$

Задание 3.9.10

Исследуйте функции и постройте их график.

$y = \dfrac{x^3}{16-x^2}$;

$y = \dfrac{x^3}{(x-2)(x+4)}$.

3.10. Системы управления базами данных

В основе решения многих задач лежит хранение и обработка больших массивов информации об объектах и явлениях реального мира. Такие массивы данных вместе с программно-аппаратными средствами для их обработки называют автоматизированными информационными системами или просто информационными системами (ИС) учитывая, что современные ИС являются, как правило, автоматизированными.

Сама идея ИС и некоторые принципы их организации возникли задолго до появления ЭВМ. Библиотеки, архивы, адресные книги, телефонные справочники, словари – все это ИС.

3.10.1. Базы данных

Основа информационной системы, объект ее обработки – база данных.

База данных – в широком смысле – это совокупность сведений о конкретных объектах реального мира в какой-то предметной области.

В этом определении, вообще говоря, отсутствует упоминание о компьютере. Базой данных можно считать, например, картотеку в регистратуре поликлиники, которая ведется на картонных карточках и хранится на полках.

Если хранить эти же сведения в текстовых файлах, то некоторые действия можно выполнять быстрее и аккуратнее (например, распечатать рекомендации врача какому-либо пациенту), однако многие возможности компьютера при таком подходе останутся не задействованы. Дело в том, что в текстовых файлах обычно содержится неструктурированные данные, то есть у компьютера отсутствует формальная информация о том, где какие данные находятся, поэтому использовать возможности ЭВМ для поиска и отбора нужных данных (например, отобрать пациентов определенного года рождения или с определенным диагнозом) очень сложно.

Процесс приспособления форматов и значений данных к нуждам ЭВМ, то есть введение соглашений о способах представления данных, можно назвать ***структурированием информации***.

Теперь можно уточнить определение базы данных и информационной системы.

База данных (БД) – совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области.

Информационная система – это совокупность структурированных данных (базы данных) и комплекса аппаратно-программных средств для хранения данных и манипулирования ими.

3.10.2. Компоненты информационной системы:

1. База данных.
2. Система управления базами данных (СУБД) – пакет программ, позволяющий:
   - Обеспечивать пользователя языковыми средствами описания и манипулирования данными.
   - Обеспечить поддержку логических моделей данных (то есть взаимосвязи между данными).
   - Обеспечить операции создания и манипулирования логическими данными и одновременное отображение этих операций на физическом уровне.
   - Обеспечить защиту и целостность данных.
3. Администратор БД – это специалист или группа специалистов, занятых обслуживанием пользовательской БД. Администратор должен координировать процессы сбора информации и эксплуатации БД, а также обеспечивать целостность и защиту данных. При разработке ИС обязательно учитываются интересы конечного пользователя. Главный принцип состоит в том, что от конечных пользователей не должно требоваться каких-либо специальных знаний в области вычислительной техники и языковых средств. Поэтому пользовательский интерфейс должен быть предельно прост и понятен.
4. Вычислительная система (ВС) – совокупность взаимосвязанных и согласованно действующих ЭВМ и других устройств, обеспечивающих автоматизацию процессов приема, обработки и выдачи информации.
5. Словарь данных – подсистема БД, предназначенная для централизованного хранения информации о структуре данных, хранящихся в БД, типах данных и форматах их представления, взаимосвязях файлов БД, принадлежности пользователей, кодах защиты информации и проч. Присутствует во всех ИС, но не всегда имеет такое название, эти функции обычно поддерживает СУБД.

3.10.3. Классификация информационных систем

В зависимости от используемых технических средств можно выделить следующие типы БД:

- Локальные (ЛБД) – размещаются в памяти одной машины.
- Распределенные (РБД) – размещаются в памяти нескольких машин, соединенных в сеть.

Совокупность ЛБД, входящих в сеть ЭВМ, является РБД.

Информационные системы можно условно разделить на ***фактографические***, ***документальные*** и ***экспертные***.

В ***фактографических*** ИС регистрируются факты – конкретные значения данных об объектах реального мира. Основная задача таких ИС заключается в том, что все сведения об объектах (фамилии людей, названия предметов, даты и т. д.) сообщаются системе в каком-то заранее обусловленном формате (например, дата в виде ДД.ММ.ГГ). Информация, с которой работает фактографическая ИС, имеет четкую ***структуру***, позволяющую машине отличать одно данное от другого. Поэтому фактографическая система способна давать однозначные ответы на поставленные вопросы. Например, ИС магазина может ответить на вопрос «Сколько велосипедов марки Салют продал магазин за май месяц?», ИС вокзала – «Какие поезда проходят через город Нижний Новгород?» и т. д.

Документальные ИС работают с неструктурированными текстовыми документами (статьи, книги, рефераты, тексты законов …), графическими или мультимедийными объектами. Такие ИС имеют формализованный аппарат поиска нужного документа. Цель такой системы – выдать в ответ на запрос пользователя список документов или объектов, в какой-то мере удовлетворяющих сформулированным в запросе условиям. Например, выдать список всех статей, в которых встречается слово «рак». Принципиальной особенностью документальной ИС является ее способность, с одной стороны, выдавать ненужные пользователю документы (например, если нужно «рак» – животное, а система отберет «рак» – болезнь), а с другой стороны – не выдавать нужные (например, если в запросе употреблен синоним или произошла ошибка в написании).

Современные фактографические системы часто работают с неструктурированными блоками информации (текстом, графикой, видео), снабженными структурированными ***описателями.***

Рассмотрим пример.

Пусть объектом обработки фактографической ИС служит список эстрадных певцов, причем, для каждого певца имеются данные:

- Сценическое имя (не более 20 символов).
- Дата рождения в формате ДД.ММ.ГГГГ.
- Пол (М или Ж).
- Биография.
- Фонограмма.

Располагая структурированными описателями (имя, дата рождения, пол), система может выдать строгие ответы на вопросы:

- О каждом певце персонально.
- О распределении певцов по возрасту и полу (в любых сочетаниях).

Та же информация дублируется в тексте биографии, однако там она не структурирована. Если удалить описатель, то получится документальная ИС.

Важнейший этап обработки любого документа, поступающего на хранение в документальную ИС – ***индексирование***. Индексирование состоит из двух этапов:

1. Выявление основного содержания документа.
2. Описание содержания документа на информационно-поисковом языке и получение соответствующего ***поискового образа документа*** (ПОД).

При получении запроса на поиск документа составляется ***поисковый образ запроса***, который в процессе поиска сравнивается с поисковыми образами документов до тех пор, пока не будет найдено совпадение.

Информационно-поисковые языки (ИПЯ) можно разделить на три группы:

- Классификационные:
  - ИПЯ с иерархической структурой.
  - ИПЯ с фасетной структурой.
  - Эмпирические (неиерархические) языки.
- Дескрипторные.
- Комбинированные.

В ***иерархических классификационных системах*** термины находятся между собой в отношении включения. При записи они располагаются в порядке постепенного перехода от общих к частным. Примером такой системы является универсальная десятичная классификация (УДК), применяемая в библиотечном деле (создана в XIX – XX в.). Каждый класс (первая ступень) содержит группу более или менее близких наук, например 5 – математика и естественные науки, 6 – прикладные науки (в том числе и информатика)… Каждая следующая цифра, не меняя значения предыдущих, уточняет их, обозначая более частное понятие. УДК указывается на оборотной стороне титульного листа книги. Пример: 51.938 («Красота фракталов. Образы комплексных динамических систем»).

Примером ***эмпирической классификационной системы*** может быть алфавитно-предметная классификация.

В ***дескрипторных ИПЯ*** смысловое содержание документа выражается списком ***ключевых слов*** (дескрипторов). ***Де-***

скриптор – слово или словосочетание естественного языка, отражающее один из аспектов содержания документа. Для обеспечения поиска используется поисковый образ данных, который содержит фиксированный набор дескрипторов, быть может, связанных между собой логическими операциями.

Еще одним видом информационных систем являются экспертные системы.

Экспертные ИС имитируют поведение эксперта (специалиста) в той или иной области. Экспертная система может генерировать новую информацию на основе имеющейся у нее и давать разумные советы исследователям. В основе операций экспертной системы – обработка базы знаний, составленной специалистами в данной области. База знаний содержит факты (изменяющуюся информацию, характеризующую состояние объекта) и правила (информацию о том, как получать новые факты).

3.10.4. Объекты, атрибуты, связи

Целью ИС является обработка данных об объектах реального мира с учетом связей между ними. В теории фактографических ИС данные часто называются атрибутами, а объекты – сущностями.

Объект (сущность) – это нечто существующее и различимое, то есть объектом можно назвать то, для чего существует название и способ отличать один подобный объект от другого. Например, город, человек, класс в школе, фирма, химическое соединение могут быть объектами в той или иной БД.

Группа подобных объектов образует ***класс объектов***. Например, классами объектов могут быть города, люди, работающие на предприятии, товары на складе и т. д. Конкретный объект в такой группе является экземпляром класса.

Атрибут (или данное) – это некоторый показатель (параметр, признак, свойство), который характеризует некий объект и принимает для конкретного экземпляра конкретное значение. Например, класс объектов – города России. Название города – атрибут, принимающий текстовое значение (например, Москва, Красноярск, Тамбов), численность населения – атрибут, принимающий числовое значение и т. д.

ИС оперирует классами объектов, выделенных применительно к данной предметной области, используя при этом конкретные значения атрибутов тех или иных экземпляров класса.

Атрибут некоторого класса объектов сам может быть объектом, имеющим собственные атрибуты. Например,

атрибутом служащего является ВУЗ, который он окончил. С другой стороны, конкретный ВУЗ – это объект, характеризующийся собственным набором атрибутов: адрес, фамилия ректора, список факультетов. Таким образом, возникает возможность установления ***связи*** между экземплярами объектов разных классов.

Пример. Рассмотрим пример промышленной компании. В этом случае желательно хранить информацию о: проектах, деталях (для этих проектов), поставщиках (деталей), складах (где хранятся эти детали), служащих и т. д. Все перечисленное представляет собой сущности.

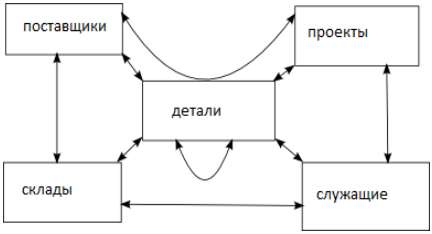


Рис. 3.51. Сущности и связи

Между этими объектами существуют связи или отношения (показаны стрелками). На приведенной схеме большинство связей объединяют два типа объекта, но это не всегда так. Например, стрелка может объединять три типа объекта (поставщики – детали – проекты). Это отражает тот факт, что определенные поставщики поставляют определенные детали для определенных проектов. Связь может охватывать и один тип объекта (детали). Она отражает то, что некоторые детали могут являться компонентами других деталей.

В общем случае одни и те же объекты могут быть соединены произвольным числом связей (проекты – служащие две связи: одна из них «работает над», другая «руководит проектом»).

Таким образом, подобная БД представляет собой сущности, атрибуты и связи, иначе говоря, содержит логически связанные данные.

3.10.5. Модели данных

Общее описание БД называется ***моделью БД.***

Модель данных – интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.

Таким образом, модель данных представляет собой структуру БД и включает в себя описание всех классов объектов, их атрибутов и связей между ними. Модель данных описывает правила, по которым построена БД, определяет допустимые операции с данными.

В фактографических ИС выделяют следующие типы моделей данных:

- Реляционная.
- Сетевая.
- Иерархическая.

Иерархическая модель

Иерархическая модель исторически появилась первой. Ее появление связано с тем, что в реальном мире очень многие связи соответствуют иерархии, когда один объект выступает как родительский, а с ним может быть связано множество подчиненных объектов. Иерархия проста и естественна в отображении взаимосвязи между классами объектов.

Основными информационными единицами в иерархической модели являются: БД, сегмент, поле.

Поле – минимальная, неделимая единица данных, описывает одну из характеристик объекта.

Сегмент (запись) – совокупность данных об одном объекте.

В иерархической модели сегменты объединяются в древовидный граф. При этом направленные ребра графа отражают иерархические связи между сегментами: каждому экземпляру сегмента, стоящему выше по иерархии (сегменты-предки) соответствует множество экземпляров подчиненного типа сегмента (сегменты-потомки).

Таким образом, БД представляет собой совокупность отдельных деревьев. При этом действуют следующие ограничения:

- В каждом дереве есть один корневой сегмент, то есть сегмент, у которого нет логически исходного.
- Каждый исходный сегмент может иметь произвольное число подчиненных сегментов.
- Подчиненный сегмент может иметь только одного родителя.

Сетевая модель

Сетевая модель данных – модель, состоящая из записей данных и связей, установленными между записями.

Сетевую модель можно представить как граф с записями в виде узлов и связями в виде ребер. На связи между объектами не накладывается ограничений.

Сетевой БД фактически является Всемирная паутина (WWW) сети Интернет. Гиперссылки связывают между собой сотни миллионов документов в единую сетевую БД.

При использовании сетевых и иерархических моделей от пользователя требуется знание физической организации БД, к которой он должен осуществлять доступ.

Реляционная модель

Реляционная БД представляет собой совокупность логически связанных таблиц. Основной структурой данных является отношение (таблица). Можно доказать, что любую структуру можно преобразовать в набор двумерных таблиц.

В таблицу помещаются данные об объектах одного типа. Заголовками столбцов служат атрибуты (свойства) описываемых объектов. Столбцы таблицы называются ***полями***, строки – ***записями или кортежами***.

Поле характеризуется своим именем и типом данных (текстовый, числовой, денежный и т. д.).

Ключевым называется поле, значение которого однозначно определяет запись таблицы.

Как правило, реляционные БД состоят из более, чем одной таблицы. Таблицы связываются между собой по общему полю. Связи, установленные между таблицами, позволяют осуществить поиск одних записей по значению других.

Отделы:

Номер отдела	Улица	Город	Индекс	Телефон	Факс
B5	…	Москва	…	1223454	8767652
B7	…	Новгород	…	1286653	5435435
B3	…	Москва	…	8766542	6544322
B4	…	…	…	4354376	9876552
B2	…	…	…	9876543	9876543

Сотрудники:

Номер сотрудника	Фамилия	Имя	Адрес	Телефон	Должность	Дата рождения	Доход	Номер отдела
СЛ-21	Иванов	Николай						B5
СГ-37	Петров	Иван						B3
СГ-14	…	…						B3
СА-9	…	…						B7
СГ-5	…	…						B3
СЛ-41	…	…						B5

Между таблицами «Отделы» и «Сотрудники» существует связь: сотрудник работает в отделении компании. Но существование этой связи можно заметить, только зная, что атрибут «номер отдела» в таблице «Сотрудники» эквивалентен атрибуту «Номер отдела» в таблице «Отделы».

Этот же пример в сетевой модели приведен на рис. 3.52.

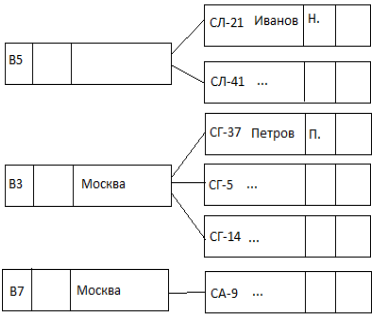


Рис. 3.52. Сетевая модель базы данных

Иерархическая модель (рис. 3.53).

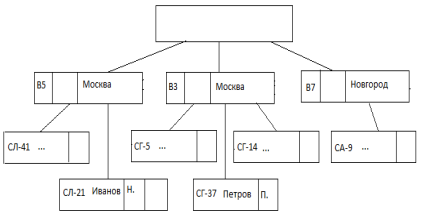


Рис. 3.53. Иерархическая модель базы данных

К СУБД реляционного типа относятся: dBASE, FoxPro, Paradox, Access и др. Интерфейсы различных СУБД менее унифицированы, чем интерфейсы текстовых или графических редакторов, поэтому трудно дать общее описание.

Но, как правило, реляционные БД состоят из следующих объектов:

Таблицы. Это базовый объект БД. В таблицах хранится вся информация.

Запросы. Главное предназначение запросов – отбор данных на основании заданных условий.

Формы. Отображает данные таблицы или запроса в более удобном виде.

Отчеты. Предназначен для печати данных из таблиц и запросов.

Вопросы для самопроверки

1. Что представляет собой современная информационная система (ИС)?
2. Перечислите виды ИС.
3. Назовите компоненты, входящие в состав ИС.
4. Проведите классификацию БД.
5. Опишите основные модели данных.

3.11. Система управления базами данных OpenOffice.org Base

OpenOffice Base – инструмент для работы с внешними источниками данных и со встроенной СУБД, который предоставляет мощный набор инструментов: редакторы таблиц, запросов, форм, отчетов БД.

Мы будем пошагово разрабатывать базу данных, иллюстрируя работу инструментов Base.

3.11.1. Создание базы данных

Реляционная БД представляет собой совокупность взаимосвязанных таблиц, поэтому таблица является одним из основных способов представления данных. Другим способом представления данных является форма.

Описание создаваемой базы данных

В качестве примера будем рассматривать магазин, который торгует посадочным материалом цветочных культур. Администрация магазина создает БД, в которой хотелось бы хранить следующую информацию: название культуры, является ли культура многолетней, цена, данные о заказчиках, данные о заказах, количество посадочного материала в заказе. Кроме того, по имеющимся данным хотелось бы иметь возможность получить информацию о стоимости заказа и имеющихся скидках.

Проектирование БД – достаточно сложный процесс, он выходит за рамки данного курса, поэтому опишем лишь конечный результат (рис. 3.54).

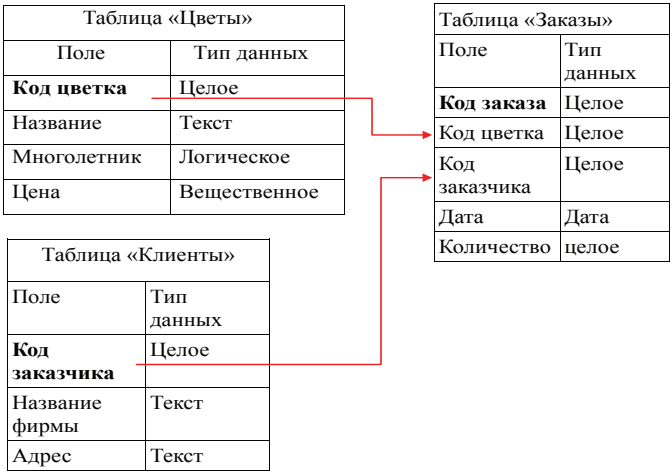


Рис. 3.54. Схема базы данных «Цветочный магазин»

Важным понятием является понятие *ключевого поля* – поля или набора полей, значение которого однозначно определяет запись таблицы. Ключевые поля выделены на схеме жирным шрифтом. Стрелками показаны связи между таблицами данных.

Мастер базы данных

1. Запустите OpenOffice.org Base. Откроется окно «Мастер баз данных». Создание новой базы состоит из нескольких этапов (в простом случае – из двух).

2. На первом шаге можно выбрать одну из возможностей:

- создать новую базу данных (выберите сейчас этот вариант);
- открыть существующую;
- подключиться к существующей базе данных.

3. На втором шаге, в случае создания новой БД, указывается ее имя и выбирается, нужно ли регистрировать БД в OpenOffice.org. (Регистрация позволит использовать поля БД в других документах, например текстовых, или при создании рассылок.) В нашем случае регистрация не нужна.

4. Появится диалоговое окно «База данных».

Диалоговое окно «База данных» является основным и позволяет переходить от одного способа представления данных к другому (рис. 3.55).

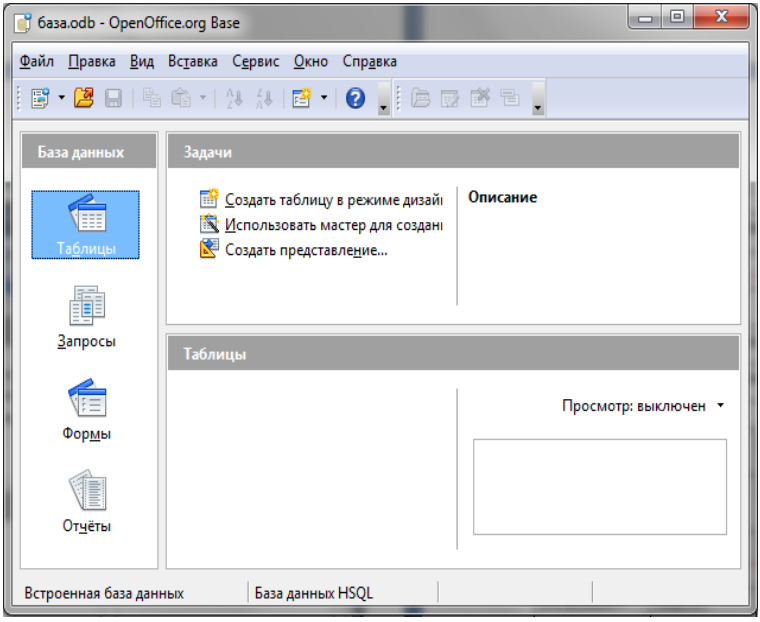


Рис. 3.55. Окно «База данных»

В левой части расположены вкладки «Таблицы», «Запросы», «Формы», «Отчеты», с помощью которых можно выбирать те объекты БД, с которыми будете работать.

В центре сверху располагаются «Задачи» – действия, которые можно выполнить с выбранными объектами, например, «Создать форму в режиме дизайна...» или «Использовать мастер для создания форм...».

В центре снизу располагается собственно список таблиц, запросов, форм или отчетов – в зависимости от того, какая вкладка открыта. Этот список может быть пуст.

3.11.2. Создание таблиц данных

Начнем создавать первую таблицу нашей БД. Таблица «Цветы» содержит поля: «Код цветка» (целое), «Название» (текст), «Многолетник» (логическое), «Цена» (вещественное).

1. В окне базы данных перейдите на вкладку «Таблицы».
2. Выберите задачу «Создать таблицу в режиме дизайна».
3. Откроется окно «Проектирование таблицы» (см. рис. 3.56).

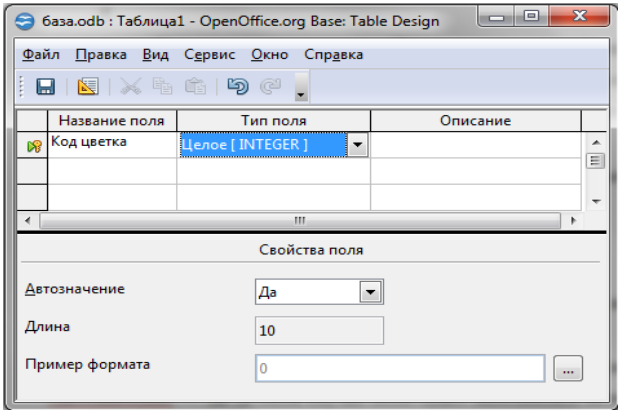


Рис. 3.56. Окно «Проектирование таблицы»

Верхнюю часть окна «Проектирование таблицы» занимает таблица, состоящая из трех столбцов (Название поля, Тип поля, Описание). Каждая строка этой таблицы предназначена для описания одного поля создаваемой таблицы (текущая строка помечена стрелкой).

Нижняя часть окна предназначена для описания свойств поля.

4. Введем первую запись в таблицу:
   - в столбце «Название поля» запишите: **Код цветка**;
   - в столбце «Тип поля» выберите из списка – **Целое**;
   - столбец «Описание» является не обязательным и служит для пояснения смысла данного поля;
   - не меняя текущую строку, в окне «Свойства поля» укажите «Автозначение» – **«Да»** (в этом случае поле будет заполняться автоматически по порядку — 1, 2, 3...).

5. Аналогично опишите другие поля. Поле «Многолетник» имеет тип Логическое. В описании к этому полю можно указать, что значение «да» оно будет принимать, если цветок многолетний.

6. После описания всех полей надо **указать ключевое поле таблицы** (в нашем случае это поле код цветка). Для этого необходимо:
   - выделить строку, описывающее это поле (щелкнуть мышью слева от строки);
   - открыть контекстное меню (правая кнопка мыши) и выбрать «Первичный ключ». Слева от строки таблицы появится ключ, что означает, что это поле в таблице ключевое.

7. Теперь нужно сохранить таблицу и присвоить ей имя – «Цветы» (с помощью меню Файл – Сохранить или кнопки на панели инструментов).

8. Аналогично создайте все остальные таблицы.

Примечание:

- Не забывайте указывать обязательные поля в таблицах и задавать описание полей, особенно в тех случаях, когда содержимое поля не очевидно из его названия.
- Задавая свойства поля, уменьшайте его размер до необходимого (например, для текстового поля размер поля по умолчанию – 50, а на фамилию достаточно 20). Это позволит уменьшить объем необходимого дискового пространства.

Связывание таблиц

1. Откройте основное окно «База данных».

2. Выберите пункт меню Сервис – Связи. Откроется окно «Проектирование связей».

3. В открывшемся окне «Добавить таблицы» (если оно не открылось, нажмите на кнопку «Добавить таблицу») последовательно выбирайте из списка все таблицы, после каждой таблицы нажимая на кнопку «Добавить».

4. Закройте окно добавления таблиц.

Между таблицами «Цветы и Заказы» имеется связь по полю «Код цветка». Организуем ее:

- В таблице «Цветы» щелкните по полю «Код цветка».
- Не отпуская мыши, перетащите поле «Код цветка» из таблицы «Цветы» на поле «Код цветка» в таблице «Заказы».

Аналогично свяжите между собой поля «Код заказчика» из таблицы «Клиенты» и «Код заказчика» из таблицы «Заказы».

Сохраните изменения макета данных.

Примечание: связывать между собою можно поля с одним типом данных, при этом они могут иметь разные имена.

Заполнение таблиц

1. В окне «База данных» выделите вкладку «Таблицы».

2. В списке таблиц выбрать дважды щелкните на таблице «Клиенты». Откроется таблица «Клиенты». Она пока пуста.

В столбце «Код клиента» написано <автополе>, что значит, что это поле будет заполняться автоматически.

3. Установите курсор в поле «Название фирмы» и впишите туда какое-нибудь название фирмы-клиента.

4. Перейдите в поле «Адрес» и впишите туда город, в котором находится фирма (один из трех: Москва, Курск, Ростов). Нажмите Enter. Обратите внимание, что заполнилось поле «Код клиента» и появилась новая строка таблицы для следующей записи.

Строки таблицы – **записи** – представляют информацию об одном объекте. Столбцы таблицы – **поля** – характеризуют свойства объектов. **Текущая запись** выделяется треугольником. В нижней строке таблицы указывается номер текущей записи, их общее количество, а также расположены кнопки перехода между записями (на одну вперед, на одну назад, к первой, к последней) и кнопка создания новой записи. **Перейти к другой записи** (сделать ее текущей) можно щелкнув в любом месте нужной строки, с помощью кнопок перехода или указав ее номер.

1. Измените ширину столбцов (или высоту строк), чтобы все содержимое было видно (аналогично Calc, перетащив разделитель полей (строк)).
2. Заполните таблицу «Клиенты» (3-5 записей).
3. Аналогично заполните таблицу «Цветы» (5-7 записей).
4. Заполните таблицу «Заказы» (10-15 записей).

Обратите внимание, что значения в полях «Код цветка» и «Код клиента» должны совпадать со значениями из таблиц «Цветы» и «Клиенты», соответственно (в противном случае появится сообщение об ошибке).

3.11.3. Отбор необходимых данных из таблиц

Поиск данных

Допустим, мы хотим найти заказчика с каким-то определенным названием. В случае, если таблица «Клиенты» имеет много записей, осуществить поиск нужного клиента «визуально» бывает трудно. Можно использовать инструмент «Поиск».

1. Откройте таблицу «Клиенты».
2. Поставьте курсор в начале таблицы.
3. Щелкните на кнопке «Поиск» на панели инструментов (бинокль).
4. В поле «Найти текст» введите название клиента.
5. Установить область поиска – «Все поля», положение – в начале поля.
6. Нажмите «Найти».

После нахождения первой записи можно продолжить поиск (снова нажать «Найти»).

Примечание. В случае поиска по полю «Дата» надо установить флажок «Применить формат поля».

Фильтрация данных

Не всегда для работы необходимы все данные, содержащиеся в таблице. Для того чтобы отобрать нужные данные, можно использовать ***фильтр***.

Создавая фильтр, вы сообщаете перечень условий отбора, который описывает необходимые вам записи. При помощи фильтра можно рассортировать записи.

Допустим, нас интересуют заказы, сделанные позже определенного числа. Отберем необходимые записи.

1. Откройте таблицу «Заказы».
2. На панели инструментов нажать кнопку «Фильтр по умолчанию». Откроется окно «Фильтр по умолчанию».
3. В имени поля укажите «Дата». В условии – знак больше (>). В значении – нужную дату. Нажмите ОК.
4. Чтобы вернуть представление всех записей, снова нажмите на кнопку «Фильтр по умолчанию».

В фильтре можно использовать несколько условий, в этом случае надо указать, какой оператор (AND или OR) связывает эти условия.

Сортировка

Сортировать записи можно с помощью кнопок (сортировка по возрастанию и сортировка по убыванию) на панели инструментов.

3.11.4. Запросы

Запрос – это команда, которая формулируется для СУБД и требует представить определенную информацию.

Запрос служит для отбора записей по некоторому критерию. При этом, в отличие от фильтра, запрос может работать с несколькими таблицами и результат его работы может быть сохранен и в дальнейшем использован.

Типы запросов:

- Запрос-выборка – средство отбора данных, хранящихся в таблицах и отвечающих некоторым критериям.
- Перекрестный запрос предназначен для группировки записей и представления их в компактном виде.
- Запросы на добавление, изменение и удаление данных позволяют добавить, изменить или удалить данные из таблицы (одной или нескольких), создать новую таблицу, отобрав туда данные, отвечающие некоторым критериям.

Запросы можно создавать в режиме дизайна (запросы по образцу) или записывать на специальном языке (SQL-запросы). Запрос, созданный в конструкторе, автоматически транслируется в SQL-запрос.

Вначале рассмотрим создание запросов в режиме дизайна.

Примечание: во всех примерах, следующих далее, используйте те значения полей, которые встречаются в ваших таблицах.

Создание запроса в Режиме дизайна

В основном окне База данных откройте вкладку Запросы и выберите создание запроса в режиме дизайна.

Откроется окно дизайнера запросов.

Пример 1. В нашей базе данных цветочного магазина нужно найти всех заказчиков из Москвы. Для построения запроса нам понадобиться поля «Название фирмы» и «Адрес» из таблицы «Клиенты»:

1. В окне «Добавление таблицы» выбрать из списка таблицу «Клиенты» и нажать «Добавить». Закрыть это окно.
2. В бланке запроса в первом столбце в строке «Поле» выбрать из списка поле «Название фирмы».
3. Во втором столбце – поле «Адрес». В строке «Критерий» указать – «Москва».
4. Сохранить запрос под именем «Клиенты из Москвы».
5. На панели инструментов (или в пункте меню «Правка») нажмите кнопку «Выполнить запрос». Вы увидите результат работы этого запроса.

Примечание. Обратите внимание, что при выводе результата запроса в поле «Адрес» выводится одно и то же – «Москва» (так как мы ищем поставщиков из Москвы), поэтому поле «Адрес» можно не выводить, для этого в режиме дизайна (в основном окне БД пункт меню Правка – Правка или контекстное меню «Правка») снять галочку «Видимый» для поля «Адрес».

Запрос может основываться на нескольких таблицах.

Пример 2. Требуется узнать, какие цветы заказали клиенты из Москвы.

1. Создайте запрос на основе таблиц «Цветы», «Клиенты» и «Заказы» (нам нужно все три таблицы для сохранения связи между ними, так как «Клиенты» и «Цветы» напрямую между собой не связаны).
2. В запросе необходимы поля: «Адрес» (из таблицы «Клиенты») с условием отбора – «Москва»; название (из таблицы «Цветы»).
3. Сохраните запрос под именем «Цветы в Москву» и выполните его.

Параметрический запрос

Позволяет задавать условия отбора с клавиатуры в режиме диалога.

Пример 3. Допустим, нам надо отбирать названия клиентов из разных городов; сегодня – из Москвы, завтра – из Санкт-Петербурга. Можно создать отдельный запрос на каждый город, а можно создать параметрический запрос:

1. Создайте новый запрос в режиме дизайна на основе таблицы «Клиенты».

2. Внесите в бланк запроса поля: «Название фирмы» и «Адрес».

3. Для поля «Адрес» в строке «Критерий» напишите **=: [город]**.

4. Сохраните запрос под именем «Клиенты параметрический» и выполните его.

Появится окно, в котором будет написано имя запрашиваемого параметра (город) и поле для ввода параметра. Укажите значение параметра (например, Москва) и получите список поставщиков из Москвы. При новом запуске запроса можно указать другой параметр.

Сложные условия в запросах на выборку данных

Пример 4. Требуется создать запрос, который отберет клиентов из Москвы или Курска, заказавших астры.

1. Откройте вкладку «Запросы». Выберите «Создать запрос в режиме дизайна».
2. Добавьте в запрос все таблицы.
3. В бланк запроса поместите поля «Название фирмы» и «Адрес» (из таблицы «Клиенты») и «Название» (из таблицы «Цветы»).
4. Укажите условия отбора: в поле «Адрес» – «Москва», строчкой ниже – «Курск», в поле «Название» – «астра», строчкой ниже – «астра».

Примечание. Условия, связанные связкой И, записываются в одной строке, условия, связанные связкой ИЛИ – в разных строках.

1. Сохраните запрос.
2. Выполните его.

Обратите внимание, что условие «Название» (цветка) – «астра» должно повторяться в двух строчках, так как критерий в целом выглядит так «Адрес» = «Москва» и «Название» = «астра» или «Адрес» = «Курск» и «Название» = «астра»

Задание

Создайте запрос, который отберет заказы из Москвы на многолетние цветы (условие отбора для поля «Многолетник» – «Да» или «True»).

Использование в запросах вычисляемых полей

В запросе можно не только отбирать существующие данные из таблиц, но и производить вычисления, отображая их результаты в результирующей таблице. Таким образом, можно получать данные, которых нет в исходных таблицах.

Пример 5. Требуется создать запрос, в котором вычисляется стоимость заказов.

1) Откройте бланк нового запроса.

2) Добавьте таблицы «Заказы» и «Цветы».

3) В бланк добавьте поля «Код заказа», «Код цветка», «Количество», «Цена».

4) В следующем столбце в строке «Поле» надо задать выражение, которое будет вычислять стоимость заказа (см. рис. 3.57):

«Цветы». «Цена» * «Заказы». «Количество»

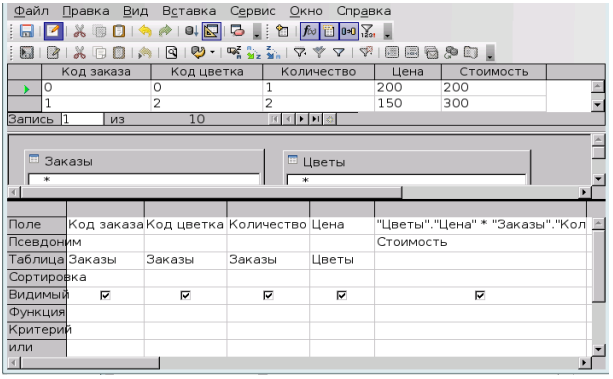


Рис. 3.57. Запрос с вычисляемым полем

Сохраните запрос как «Стоимость заказов».

Просмотрите результат. Обратите внимание на название поля, в котором вычисляется стоимость. Вернитесь в режим дизайна и задайте для этого поля псевдоним – «Стоимость».

Правила построения выражения:

- Название таблиц и полей указываются в двойных кавычках.
- Название таблицы отделяется от названия поля точкой.
- В выражении могут быть использованы знаки арифметических операций.

Задание

Создайте запрос, в котором вычислите скидку в 10% на заказы со стоимостью больше 100 р.

1) Используйте запрос «Стоимость заказов».

1) В строке «Условия отбора» для поля «Стоимость» укажите >100.

2) Добавьте поле «Скидка», в котором вычислите скидку.

3) Сохраните запрос под именем «Скидка на заказы».

Запрос с группировкой

Запрос с группировкой позволяет группировать записи по значениям полей и проводить некоторые статистические расчеты.

Пример 6. В качестве примера рассмотрим запрос, в котором вычислим количество заказов каждого заказчика.

Для этого надо будет сгруппировать данные по каждому заказчику и вычислить количество его заказов.

Запрос будем строить на основе таблицы «Заказы».

1. Создайте новый запрос на основе таблицы «Заказы».
2. Нам понадобятся поля: «Код клиента» и «Номер заказа». Добавьте их в бланк запроса.
3. Нажмите кнопку «Функция» на панели инструментов. В бланке запроса появится новая строка: «Функция».
4. Для поля «Код клиента» в качестве групповой операции выберите Group (Группировка), а для поля «Номер заказа» – Count (Количество) (выбор операции производится из выпадающего списка).
5. Сохраните запрос. Просмотрите результат.
6. Вернитесь в режим дизайна и для поля «Номер заказа» в строке «Псевдоним» задайте новое имя «Количество заказов».
7. Сохраните изменения. Просмотрите результат.

Задание

Создайте запрос, в котором найдите количество многолетников и однолетников. Сохраните запрос.

3.11.5. Создание форм

Форма является одним из способов представления информации и позволяет сделать работу с отдельными записями более удобной. Экранная форма позволяет объединить поля в группы, что облегчает восприятие информации.

Способы создания форм:

1) Мастер форм. Позволяет создавать формы в режиме диалога с мастером.

2) Дизайнер форм. Позволяет самостоятельно разрабатывать формы с заданными свойствами.

Мастер для создания форм

Формы, созданные с помощью мастера, более разнообразны по стилю оформления, могут содержать данные из разных таблиц.

Создадим форму, в которую включим все поля из таблицы «Заказы» и добавим из таблицы «Клиенты» поля «Название фирмы» и «Адрес». Так как в форму надо включить поля из разных таблиц, создадим в начале запрос, в который войдут все необходимые поля и форму будем создавать на основе этого запроса.

1) Откройте вкладку запросы и в режиме дизайна создайте запрос, в который включите все поля из таблицы «Заказы»

и поля «Название фирмы» и «Адрес» из таблицы «Клиенты». Сохраните запрос как «Запрос для формы».

2) Откройте вкладку «Форма» и выберите «Использовать мастер для создания формы».

3) В качестве исходной таблицы выберите «Запрос для формы».

4) На первом шаге выбираются поля, которые будут включены в форму (рис. 3.58).

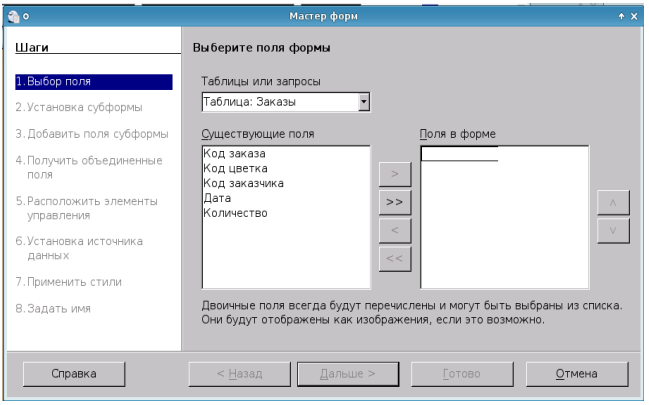


Рис. 3.58. Мастер форм

В поле со списком «Таблицы/Запросы» указана таблица или запрос, которая была выбрана как основная (в нашем случае это «Запрос для формы»). Ниже в окне приведены существующие поля этой таблицы.

С помощью кнопок со стрелками, направленными вправо, переместите в правое окно «Поля в форме» все поля из запроса. (Кнопка с одной стрелкой перемещает одно поле, кнопка с двойной стрелкой – все поля.) Если поле было выбрано ошибочно, его можно удалить из списка полей кнопками со стрелками, направленными влево.

Нажав кнопку «Далее», перейти к следующему шагу.

1) На втором шаге определяем, нужно ли создавать субформу (вложенную форму). В нашем случае – вложенная форма не нужна. Поэтому третий и четвертый шаг будут пропущены.

2) Пятый шаг – расположение элементов управления формы.

3) «Установка источника данных» – здесь определяется, как форма будет связана с исходными таблицами – выбрать «Форма для отображения всех данных». При этом можно установить ряд ограничений:

- запретить изменение существующих данных;
- не разрешать удаление существующих данных;
- не разрешать добавление новых данных.

В нашем случае эти ограничение не нужны.

4) «Применить стили» – цветовое оформление формы.

5) «Имя формы» напишем «Заказы». «Вариант дальнейшей работы» – работа с формой.

Примечание. На любом шаге создания формы можно вернуться назад и внести изменения.

Конструктор форм

Конструктор форм позволяет создавать формы любой степени сложности и более удобные для конечного пользователя. Также в Конструкторе форм можно изменить уже созданную форму.

Открыть окно конструктора можно как для новой, еще не созданной формы, так и для созданной. Для этого надо:

- на вкладке Формы выбрать название формы открыть контекстное меню и выбрать Изменить;
- для создания новой формы в режиме Конструктора, выбрать Создать форму в режиме дизайна.

Откройте форму Заказы в режиме Конструктора.

Панель инструментов Элементы управления

Используется для размещения в форме объектов.

Рис. 3.59. Панель «Элементы управления»

Панель «Элементы управления» содержит следующие инструменты (рис. 3.59):

1. Выбор объектов – осуществляет выделение объекта.
2. Режим разработки – позволяет переходить из режима разработки в режим просмотра и обратно.
3. Элемент управления – для выделенного элемента открывает окно свойств.
4. Свойства формы.
5. Флажок.
6. Текстовое поле.
7. Поле форматированного ввода.
8. Кнопка.
9. Переключатель.
10. Список.
11. Поле со списком.
12. Метка.
13. Дополнительные элементы управления.
14. Режим дизайна.
15. Мастер.

Свойства объектов

Все объекты формы имеют свойства, которые можно изменить в соответствии со своими требованиями. Для этого надо выделить объект и нажать кнопку «Элемент управления» на панели инструментов «Элементы управления».

Окно свойств зависит от объекта и может содержать следующие вкладки:

- Общие – свойства, связанные с оформлением.
- Данные – свойства, связанные с источником данных.
- События – свойства, связанные с необходимостью реакции на события (например, наведение курсора на объект, щелчок мышью на объекте и др.).

Для того чтобы изменить свойства объекта, его необходимо выделить (инструментом выбор объекта) и открыть для него окно свойств.

Просмотрите свойства объектов (полей), находящихся в области данных формы Заказы.

Управление объектами

В процессе создания формы можно перемещать, удалять и изменять размеры объектов.

Выделение объектов. С помощью мыши – щелчком на объекте. Если надо выделить несколько объектов – SHIFT+щелчок мыши.

С помощью инструмента Выбор объекта.

Изменение размеров объекта. Выделенный объект имеет маркеры выделения, перетаскивая которые можно изменить размер объекта.

Перемещение объекта:

- Выделите объект.
- Подведите указатель мыши к границе выделения.
- Удерживая кнопку мыши, передвиньте объект на нужное место.

Примечание. Если объект состоит из двух частей (например, поле и его название), то для перемещения одного объекта отдельно от другого его необходимо разгруппировать (в контекстном меню команда «Группировка» – Разгруппировать).

Удаление объектов. Для удаления объекта его надо выделить и нажать кнопку Delete.

3.11.6. Отчеты

Отчет – это гибкое и эффективное средство для организации данных при выводе на печать. С помощью отчета имеется возможность вывести необходимые сведения в том виде, в котором требуется.

Данные, помещаемые в отчет, могут быть отобраны из нескольких таблиц и представлены в наиболее удобном виде. При этом, как и при создании форм, удобнее перед созданием отчета создать запрос, содержащий все необходимые в отчете поля из разных таблиц.

Создать отчет можно с помощью мастера.

Создадим отчет, в котором будут расположены данные о заказах и клиентах, сгруппированные по датам заказов.

1. Создайте вспомогательный запрос, в который включите поля «Код заказа», «Дата», «Код клиента» (из «Заказы») и «Название фирмы», «Адрес» (из «Клиенты»).
2. Перейдите на вкладку «Отчеты» и запустите мастер создания отчетов.
3. Шаг первый – Выбор полей – все поля из созданного вспомогательного запроса (см. п. 1).
4. Шаг второй – Поля меток – можно задать названия полей отчета. Это важно в том случае, если в отчет вошли поля из разных таблиц с одинаковым названием. Этот шаг пропускаем.
5. Шаг третий – Группировка – позволяет сгруппировать данные по значениям полей. В левом столбце «Поля» выбираем «Дата» и перемещаем в правый столбец Группировка.
6. Шаг четвертый – параметры сортировки. Можно задать сортировку данных в отчете. Пропускаем.

7. Шаг пятый – Выбор стиля – задает внешний вид отчета. Можно выбрать любой.

8. Шаг шестой – Создать отчет – выбор типа отчета (нужен динамический) и действия после создания – Создать отчет сейчас.

Мы закончили создание учебной базы данных «Цветы», в процессе которого познакомились с основными возможностями системы управления базами данных OpenOffice.org Base.

ДЛЯ ЗАМЕТОК

Н. Ю. Иванова, В. Г. Маняхина

СИСТЕМНОЕ И ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Учебное пособие

Управление издательской деятельности
и инновационного проектирования
МПГУ
117571 Москва, Вернадского пр-т, д. 88, оф. 446
Тел.: (499) 730-38-61
E-mail: izdat.innov@mpgu.edu

Издательство «Прометей»
129164 Москва, ул. Кибальчича, д. 6, стр. 2
Тел.: (495) 683-15-65
E-mail: info@prometej.su

Выполнено при техническом содействии
ИП Заика А.А.

Подписано в печать 01.12.2011 г.
Формат 60x90/16. Объем 12,625 п.л.
Тираж 500 экз. Заказ № 201.

ISBN 978-5-4263-0078-1