

Лабораторный практикум по информатике

Емельянов Эдуард Владимирович

Содержание

1	Графический интерфейс Linux. Представление ФС.	5
1.1	Знакомство с клавиатурой	5
1.2	Вход в Linux	7
1.2.1	Графический интерфейс пользователя (ГИП)	7
1.2.2	Текстовая консоль и файловая система Linux	9
1.3	Отображение файловой системы в ГИП	12
2	Служебные программы.	14
2.1	Содержимое K-меню	14
2.2	Работа в виртуальной консоли	17
2.3	Дополнительные возможности файловой системы	18
2.4	Подключение сменных носителей	20
2.4.1	Подключение сменных носителей при помощи Konqueror	20
2.4.2	Подключение сменных носителей из консоли	21
2.4.3	Подключение сканеров и камер	21
3	Форматирование текстового документа.	22
3.1	Знакомство с Open Office Writer	23
3.1.1	Запуск Open Office Writer	23
3.1.2	Навигация по тексту	23
3.1.3	Методы ввода текста	24
3.1.4	Форматирование текста	25
3.1.5	Открытие и сохранение документов	26
3.2	Ввод и редактирование текста	26
3.2.1	Использование стилей в документе	27
3.2.2	Таблицы	28
3.2.3	Колонтитулы	29
3.3	Автоматические элементы	30
3.3.1	Использование автоматических списков	30
3.3.2	Сноски	32
3.3.3	Оглавление	32
3.3.4	Алфавитный указатель	33
3.4	Формулы	34
3.4.1	Простые	34
3.4.2	Сложные	35
4	Создание визитных карточек и писем	35
4.1	Создание визитных карточек	35
4.2	Создание деловых писем	37

5	Создание, редактирование и форматирование таблиц.	39
5.1	Простые таблицы	40
5.2	Таблицы с формулами	42
5.3	Диаграммы и графики	43
5.4	Ссылки	44
5.5	Расчеты по электронным таблицам	45
6	Системы счисления.	47
6.1	Перевод чисел в 10-ю СС	48
6.2	Перевод чисел из 10-й СС	48
6.3	Арифметические операции	49
7	Компьютерные презентации. Разработка презентации.	50
7.1	Создание простой презентации	51
7.2	Оформление спецэффектов	53
7.3	Создание автоматического показа презентации	55
8	Работа с электронными СУБД Open Office Base	55
8.1	Ознакомление с СУБД. Создание таблицы	55
8.1.1	Создание таблицы в режиме дизайна	56
8.1.2	Создание таблиц с помощью «Мастера таблиц»	58
8.2	Работа с таблицами, создание запросов	59
8.2.1	Отбор и сортировка данных при помощи запросов	59
8.3	Сортировка и поиск данных	60
8.3.1	Сортировка данных в таблице	60
8.3.2	Поиск данных	60
8.4	Создание форм для ввода данных	60
8.5	Создание и печать отчетов	61
9	Работа с растровой графикой.	62
9.1	Исправление дефектов фотографии	63
9.2	Редактирование фотографий, фотомонтаж	66
9.3	Создание анимаций	68
10	Работа с векторной графикой.	70
10.1	Двумерные векторные фигуры	70
10.2	Трехмерные фигуры	72
10.3	Другие векторные редакторы	74

11 Сетевые протоколы http, ftp.	75
11.1 Протокол http	75
11.1.1 Консольные браузеры: lynx и links.	75
11.1.2 Графические браузеры.	76
11.2 Протокол ftp	78
12 Web-страницы	80
12.1 Mozilla	80
12.2 Open Office Writer/Web	83
12.3 Публикация страницы	83
13 Компьютерная модель и ее исследование.	84
13.1 Исследование математической модели.	85
13.2 Исследование физической модели.	86
14 Основы программирования в shell.	89
14.1 Простейшие скрипты.	89
14.1.1 Подсчет файлов 1.	89
14.1.2 Подсчет файлов 2.	90
14.2 Более сложные скрипты.	92
14.2.1 Сбор файлов из разных папок в одну 1.	92
14.2.2 Сбор файлов из разных папок в одну 2.	93
15 Основы программирования на языке C.	95
15.1 Знакомство с языком C.	95
15.1.1 Введение.	95
15.1.2 Структура файлов на языке C.	96
15.2 Простейшие программы.	98
15.2.1 Hello World! (Здравствуй, мир!).	98
15.2.2 Ввод, условный оператор.	99
15.2.3 Префиксные и постфиксные операции. Циклы.	101
15.2.4 Операции с файлами.	104
16 Программирование на C.	107
16.1 Простейшие интерактивные программы.	107
16.1.1 Приветствие пользователя.	107
16.1.2 Приветствие пользователя 2.	108
16.2 Файловые операции.	108
Список литературы	111

1 Графический интерфейс Linux. Представление ФС.

1.1 Знакомство с клавиатурой

Прочитайте этот раздел, даже если вы знакомы с клавиатурой. Наверняка узнаете что-нибудь полезное :-).

Рассмотрите клавиатуру. Начнем с верхнего левого угла. Здесь расположена клавиша `<Esc>` (произносится — «искейп») — «Отмена». Ее нажатие зачастую равносильно нажатию кнопки «Отмена» диалоговых окон.

Далее идет ряд **функциональных клавиш** `<F1> ... <F12>`, которые имеют особое назначение в различных программах. По негласно принятому стандарту не меняется значение только клавиши `<F1>` — ее нажатие вызывает справочную систему данной программы.

Следующая группа из трех клавиш не представляет особого значения, т.к. они не используются практически ни в каких программах. Стоит только упомянуть полезную клавишу «PrtSc» (Print screen — печать экрана). При нажатии на нее в буфер обмена заносится текущее изображение, которое вы видите на экране. Это — быстрый способ создавать снимки экрана при подготовке методической литературы.

Нижняя часть клавиатуры разделена на три горизонтальных группы: текстовая клавиатура, клавиши редактирования текста / управления курсором, цифровая клавиатура, подобная клавиатуре калькулятора.

Текстовая клавиатура подобна клавиатуре печатной машинки за исключением дополнительных клавиш. Клавиша «Забой» (`← Backspace`) — крайняя справа в верхнем ряду предназначена для удаления символа слева от курсора. Во втором ряду есть еще одна особая клавиша — «Табуляция» или «Tab» (`↵`). Ее значение важно при форматировании текстовых документов (если вы хотите получить ровные столбцы из цифр. По краям третьего ряда расположены клавиши `<Caps Lock>` и «Ввод» (`Enter ↵`).

Клавиша `<Caps Lock>` служит для фиксирования верхнего регистра при вводе букв. Если ее нажать, загорится соответствующий светодиодный индикатор, и вы будете печатать заглавными буквами. Если необходимо ввести 1-2 символа строчными буквами — достаточно нажать одну из клавиш `<Shift>` (они расположены по краям второго ряда снизу) и, удерживая ее — нажать клавишу с нужным символом. Для окончания ввода заглавными буквами, достаточно повторно нажать клавишу `<Caps Lock>`. Если вы печатаете строчными буквами и вам необходимо набрать 1-2 символа заглавными, нет необходимости нажимать `<Caps Lock>` — достаточно нажать нужную клавишу, удерживая `<Shift>`. Аналогично вводятся спецсимволы, расположенные на цифровых кла-

вишах (!@#\$%^&*()_+).

При нажатии клавиши <Enter> в командной строке выполняется только что введенная команда, в текстовом редакторе она заканчивает текущую строку (абзац).

Для смены раскладки клавиатуры (английская/русская) нужно нажать одновременно клавиши <Ctrl> и <Shift>.

Клавиши <Ctrl> и <Alt> используются только в сочетании с другими клавишами для выполнения *клавиатурных макросов*.

Основные сочетания клавиш:

- <Ctrl> + <Fn> — переключение на n-й рабочий стол;
- <Alt> + <F1> — вызов К-меню (в оболочке KDE);
- <Alt> + <F2> — вызов меню «Запуск программы»;
- <Alt> + <F3> — меню окна (в графической оболочке);
- <Alt> + <F4> — закрыть окно;
- <Alt> + <↔> — переключение между открытыми окнами;
- <Ctrl> + <↔> — переключение между рабочими столами;
- <Ctrl> + <Shift> — смена раскладки клавиатуры;
- <Ctrl> + <w> — закрыть дочернее окно в многооконном приложении / закрыть окно;
- <Ctrl> + <q> — закрыть все окна многооконного приложения / закрыть окно;
- <Ctrl> + <s> — сохранить текущий документ;
- <Ctrl> + <a> — выделить все (в текстовых и подобных им документах);
- <Shift> + <курсорные клавиши> — выделить блок текста (можно выделять, нажав левую клавишу мыши и передвигая курсор от начала нужного блока до его конца);
- <Ctrl> + <c> — скопировать выделенный блок в буфер обмена (специальная область памяти);
- <Ctrl> + <v> — вставить в данное места блок из буфера;
- <Ctrl> + <x> — вырезать выделенный блок в буфер обмена;
- <Ctrl> + <z> — отменить предыдущее действие;
- <Ctrl> + <Home> — переместиться в начало документа;
- <Ctrl> + <End> — переместиться в конец документа;
- <Ctrl> + <Page Up>/<Page Down> — пролистать 1 страницу назад/вперед.

Следует упомянуть, что в Linux все можно настроить под себя (вы даже можете настроить клавиатуру, чтобы, например, символы на ней шли по алфавиту — правда, при этом будет намного сложнее набирать текст вслепую). Поэтому указанные сочетания клавиш справедливы только для «голой» системы, установленной по умолчанию.

1.2 Вход в Linux

Теперь вы можете включить компьютер, если он был выключен.

- ▶ **Включение компьютера** производите в следующей последовательности: сначала нажмите кнопку на сетевом фильтре (он расположен около системного блока), затем включите системный блок, а после этого — монитор. ◀

Через некоторое время появится окно, информирующее о загрузке Linux. По экрану будет пробегать текст, информирующий о том, какие модули операционной системы загружаются в данный момент.

- ▶ Не удивляйтесь, что Linux загружается очень долго: необходимо время для запуска большого количества различных сетевых служб и прочих сервисов, чтобы вы могли полноценно работать. Linux — надежная и стабильная система. Компьютер с Linux может без выключения работать годами (пока не сгорит какая-нибудь деталь), существует лишь одна операция, требующая перезагрузки компьютера с Linux — установка нового ядра («сердца» операционной системы). ◀

1.2.1 Графический интерфейс пользователя (ГИП)

Сразу после загрузки Linux возникает экран приглашения входа в систему. В поле «Имя пользователя» нужно ввести свой **логин** — регистрационное имя в системе (англ. «log in» — «регистрация для входа»). Далее система попросит ввести свой пароль. Пока вы его не смените, его нет — просто нажмите <Enter> (если вы входите как пользователь по умолчанию, ваше регистрационное имя — **user**). После входа в систему вы сможете сменить свой пароль командой **passwd**.

Если введенные логин и пароль были правильными, появится заставка входа, информирующая о процессе загрузки **X** и **оболочки** (X — общеупотребительное название для сервера графического интерфейса пользователя, оболочкой называется выбранный вид интерфейса.) Существует множество графических оболочек. Наиболее популярные среди них — **KDE** и **Gnome**. По умолчанию вашей оболочкой будет KDE, но во время входа в систему можно определить свою оболочку по умолчанию (для этого внизу экрана входа расположены соответствующие кнопки).

После окончания загрузки оболочки вы можете приступать к работе в X. Как можете заметить на первый взгляд, X почти ничем не отличаются от графической оболочки Windows, но, например, KDE имеет куда больше возможностей, чем ГИП W.

Область экрана, которую вы видите, называется **рабочий стол**. Внизу экрана расположена **панель управления**. Если она сокрыта, подведите курсор

сор мыши к нижней части экрана, и панель появится. На панели расположены значки быстрого доступа к некоторым программам, а также — кнопка К-меню, панель задач и системный лоток (см. рис. 1). Щелкните левой клавишей мыши по кнопке К-меню. Вы увидите меню, в котором содержатся ярлыки доступа к программам, установленным на данном компьютере.

Linux — система, где все можно настроить на свой вкус, в том числе и оформление окон и рабочего стола. Доступ к настройкам можно получить, запустив из К-меню «Центр управления».

По умолчанию KDE поддерживает работу с 4-мя рабочими столами, однако, их количество можно изменить в «Центре управления». Переключаться на *i*-й рабочий стол можно либо при помощи апплета переключения рабочих столов, расположенного на панели управления, либо нажав $\langle \text{Ctrl} \rangle + \langle \text{Fi} \rangle$.

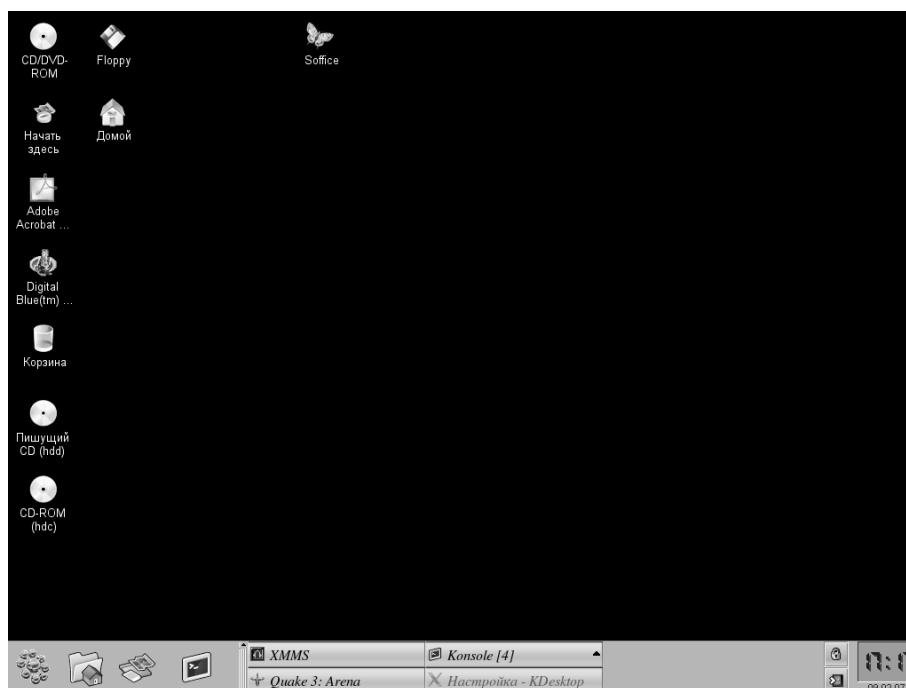


Рис. 1: Рабочий стол Linux.

Выполните задание 1:

Ознакомьтесь с основными элементами ГИП, запуская некоторые программы из К-меню. Ответьте на вопросы:

1. Для чего предназначена панель управления?
2. Как называется область, закрывающая бóльшую часть экрана?
3. Какие основные группы программ сосредоточены в К-меню?
4. Опишите основные элементы окон ГИП на примере окна какой-либо программы (например, «Центр управления»).

Выполните задание 2:

Войдите в KDE, посмотрите, как выглядит эта оболочка. Затем выйдите из нее, выбрав в К-меню пункт «Завершить сеанс». Войдите в Gnome и посмотрите, как выглядит эта среда. Выйдя из Gnome, установите понравившуюся вам оболочку по умолчанию. Кроме того, установите по умолчанию русский язык (выбрать эти настройки можно в любой момент до нажатия <Enter> после ввода пароля).

1.2.2 Текстовая консоль и файловая система Linux

Linux — многопользовательская операционная система, одновременно на ПК с Linux могут работать много человек (естественно, сидя за разными ПК, но выполняя команды на данном), чего нельзя сделать в W. Изначально в Linux не было поддержки X, с тех времен сохранилась возможность работать в реальной консоли (позже вы узнаете, что в X есть ее аналоги — `Xterm` и `Konsole`, которые чем-то похожи на режим сессии DOS в W.)

В Linux по умолчанию существует 6 консолей (терминалов). Перейти на какую-нибудь из них из X можно, нажав сочетание клавиш <Ctrl> + <Alt> + <F1...6>. Монитор переключится в текстовый режим и вы увидите приглашение в консоль:

```
ASPLinux release ...
Kernel ...
... login:
```

Вы можете ввести свой логин, а затем пароль. После этого появится приглашение к работе вида `igor_k at /home/101/igor_k`. Это — приглашение **командного процессора**, обрабатывающего вводимые команды и выполняющего соответствующие действия.

Теперь можно работать в консоли, выполняя различные команды. Конечно, рядовому пользователю это покажется страшным, но истинные *линуксоиды* очень уважают консоль и даже в X работают обычно в виртуальном терминале. Это действительно очень удобно: зная всего 100 – 200 основных команд проще набрать их в консоли, чем искать значки этих приложений в меню. Кроме того, некоторые команды можно выполнить только в консоли.

- ▶ В Linux всем устройствам соответствуют файлы в каталоге `/dev`, поэтому работа с устройством сводится к открытию соответствующего файла и чтению/записи информации из него. Работу по непосредственной передаче данных устройству выполняет его **драйвер**. ◀

Для переключения между консолями уже можно не нажимать клавишу <Ctrl> (используйте сочетания <Alt> + <F1...6>), а для переключения в X нажмите <Alt> + <F7>.

Выполните задание 3:

Переключитесь в консоль. Выполните вход в систему. Посмотрите, как выглядит стандартное приглашение. Просмотрите содержимое своего домашнего каталога (сеанс консоли начинается в этой папке), набрав команду `ls`.

- ▶ В Linux существует понятие **стандартного ввода, стандартного вывода и стандартного потока ошибок**. Их назначение в следующем: стандартным вводом называют файл, из которого программа читает входные данные, стандартный вывод — файл, в который программа помещает выходные данные, а в стандартный поток ошибок записываются все ошибки, возникающие при выполнении программы. Если не определено иначе, все три потока по умолчанию работают с одним и тем же файлом — консолью. ◀

Любая файловая система содержит два основных элемента: файл (для компьютера это просто связанный в одно целое набор единиц и нулей, записанных в разные участки на поверхности жесткого диска) и папка (это — особый файл, в котором хранится описание названий содержащихся в нем файлов и папок и указаний, как собрать эти папки и файлы из хранящихся на диске данных). Вся информация хранится на диске кусками — блоками. Если файл не вмещается в один блок, он разбивается на несколько частей и в таком виде записывается на диск. Писать на диск файлы в непрерывном виде (как, например, на магнитную ленту или оптический диск) нельзя, т.к. в этом случае после удаления мелких файлов останется много незанятого места, а большой файл записать будет некуда. Такой способ записи сильно замедляет дисковые операции. Чем больше размер блока, тем быстрее доступ к файлу, но и больше дисковые потери (если у вас много файлов размером меньше 1 кБ, а размер блока — 4кБ, на каждом файле будет теряться более 3кБ, что выльется в огромную потерю места, если количество таких файлов зашкалит за несколько десятков тысяч).

Выполните задание 4:

Создайте папку `Work`, набрав `mkdir Work`. В этой папке вы будете сохранять все свои задания. Постепенно, по мере их выполнения, вы освоите Linux, а тогда освоить любую другую ОС вам будет очень просто.

Зайдите в папку `Work`: `cd Work` и создайте в ней файл, куда в дальнейшем будете заносить информацию о новых командах: `touch Help.txt`.

Просмотреть содержимое файла можно командой `cat`: `cat Help.txt` покажет вам содержимое только что созданного файла (но, так как он пуст, ничего не отобразится). Занесите в этот файл новые команды, завершая каждую строку нажатием `<Enter>`:

(продолжение задания 4)

```
echo "ls <папка> - вывод содержимого текущей или указанной папки" >> Help.txt
echo "mkdir <имя> - создание папки" >> Help.txt
echo "cd <папка> - смена текущей папки" >> Help.txt
echo "touch <имя> - создание файла" >> Help.txt
echo "cat <имя файла> - вывод содержимого файла на экран" >>
echo "echo - вывод на экран текста, стоящего после команды" >> Help.txt
```

Обратите внимание на то, что после каждой команды `echo` мы ставили два символа «больше» и название файла: `>> Help.txt`. Это — **перенаправление вывода команды**. Вместо того, чтобы писать текст на экран, он отображается в файл.

Если после команды, делающей вывод результатов на экран, указать один символ «больше»: `> filename`, командный процессор запишет весь вывод этой программы в файл `filename`, а если такой файл существовал, прежнее его содержимое будет затерто. Если же после команды набрать `>> filename`, то если файл `filename` существовал, вывод команды будет дописан в его конец.

- ▶ Чтобы не набирать одинаковые команды, полезно использовать клавиши перемещения курсора вверх и вниз, расположенные внизу на клавиатуре между текстовой и вспомогательной цифровой клавиатурами. Эти клавиши помогают передвигаться в истории введенных команд на 1 пункт. Кроме того, клавиши `<Page Up>` и `<Page Down>` позволяют передвигаться на 30 команд в истории. ◀

Однако, неудобно работать с файловой системой из консоли. Более удобным является использование файловых менеджеров. В консоли наиболее популярен `Midnight Commander`. Для его запуска щелкните правой клавишей мыши в свободном месте рабочего стола и вызовите `Konsole` (из контекстного меню), затем в открывшемся окне наберите `mc` и нажмите `<Enter>`. Второй вариант: нажмите `<Alt> + <F2>` и в появившемся окне запуска программы наберите `mc` и нажмите `<Enter>`. `MC` состоит из двух панелей, в каждой из которых отображается содержимое одной из папок. Для выполнения файловых операций используются сочетания клавиш. Под панелями менеджера расположены подсказки, что выполняется при нажатии какой-либо функциональной клавиши.

Выполните задание 5:

Откройте файловый менеджер `mc`. Создайте при его помощи в своей домашней папке папку «Test» (нажмите `<F7>`, введите имя папки и нажмите `<Enter>`).

(продолжение задания 5)

Создайте в этой папке текстовый файл (`touch file.txt`), в котором напишите «Работа с файлами и папками» (чтобы открыть файл, переместите клавишами управления курсора выделение на этот файл, затем нажмите `<F4>`). Сохраните файл (`<F2>`).

1.3 Отображение файловой системы в ГИП

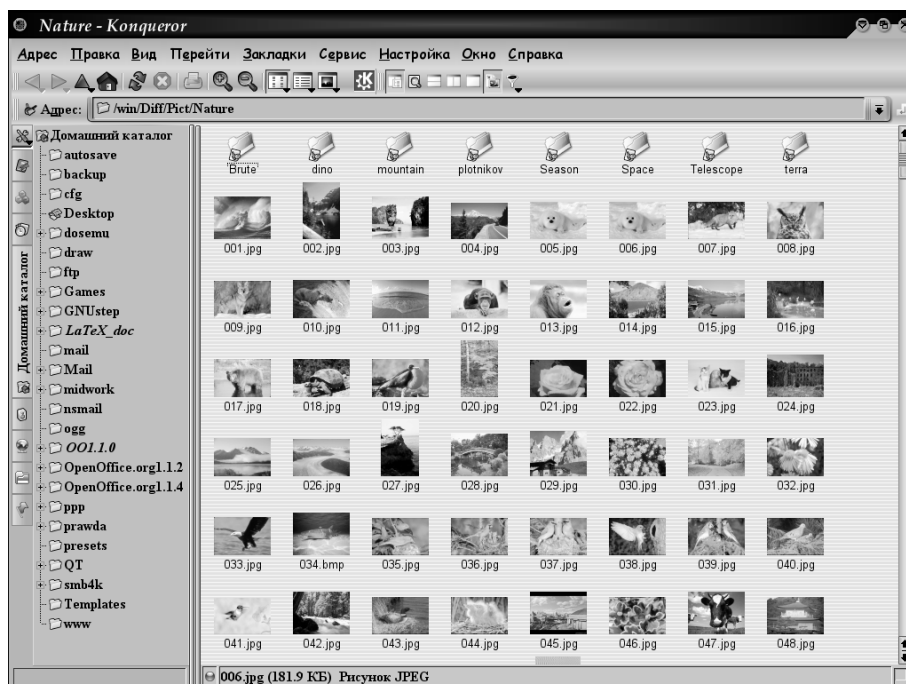


Рис. 2: Отображение файловой системы в Konqueror.

Для отображения древовидной структуры файловой системы удобно использовать графический файловый менеджер. В KDE есть собственный файловый менеджер — Konqueror (см. рис. 2), однако, просмотреть содержимое ФС из X можно при помощи любого браузера.

На панели управления (в К-меню или на рабочем столе) есть кнопка «Домой». Нажмите ее. Откроется браузер Konqueror, в котором вы сможете просмотреть содержимое своей домашней папки.

► Запустить Konqueror можно и командой `konqueror` ◀

Переместиться в какую-нибудь папку можно, щелкнув по ней два раза левой клавишей мыши (если вы не меняли настроек действия мыши в «Центре управления»). Кроме того, на левой панели Конкуерора находится **дерево каталогов** для быстрого доступа к ним. Если в каталоге есть подуровни, он отображается с крестиком слева, нажав на который можно увидеть вложенные в него папки.

- ▶ Дважды щелкнув на каком-нибудь файле левой клавишей мыши можно открыть его программой по умолчанию для данного типа файлов. Если вы хотите открыть его другой программой, щелкните по файлу правой клавишей мыши и из контекстного меню «Открыть в» выберите программу. ◀
- ▶ Можно настроить Konqueror так, чтобы папки открывались одиночным нажатием левой кнопки мыши, тогда чтобы выделить элемент ФС, необходимо навести на него мышью и немного подождать. ◀

Выполните задание 6:

Запустите Konqueror, ответьте на вопросы:

1. Какие основные элементы входят в состав любой файловой системы?
2. Как можно попасть в какую-либо папку?
3. Как выйти из нее на уровень вверх?
4. Как открыть файл программой по умолчанию для данного типа файлов?
5. Как можно открыть файл нестандартной программой?
6. Что является в Linux самым верхним уровнем файловой иерархии?

Выполните задание 7:

Создайте в своей домашней папке папку <1-й_урок>.

- ▶ Для того, чтобы создать папку, запустите файловый менеджер konqueror, активировав пункт К-меню «Домой». Щелкните правой кнопкой мыши в открывшемся окне и из контекстного меню выберите «Создать» → «Папку». Однако, удобнее и быстрее сделать это при помощи МС. ◀

Выполните задание 8:

Запустите любой текстовый редактор (например, Gedit, Kate, Kedit или Open Office Writer).

Наберите ответы на любые 3 вопроса из списка:

1. Для чего предназначен системный лоток и где он расположен?
2. Как называется область, закрывающая большую часть экрана?
3. Какие основные группы сосредоточены в К-меню?
4. Чем отличается флажок от переключателя?
5. Чем отличается список от текстового поля?

Сохраните файл в своей рабочей папке. Не закрывайте его пока.

Выполните задание 9:

Скопируйте в свой домашний каталог пару песен из папки /Big/Music. Сделайте это при помощи:

(продолжение задания 9)

1. mc — в левой панели откройте папку /Big/Music, в правой — домашнюю папку. Затем в панели слева выделите нужные файлы клавишей <Insert>, в домашней папке создайте каталог Music (клавиша <F7>), зайдите в него, переместитесь в левую панель и, нажав <F5>, скопируйте файлы в свою папку.

2. При помощи браузера — нажмите кнопку «Домой», войдите в папку /Big/Music. Выделите несколько песен нажатием левой клавиши мыши.

- Нажмите правую клавишу мыши на любом из выделенных файлов. Из контекстного меню выберите пункт «Копировать». Перейдите в папку, куда нужно скопировать файлы и нажмите правую клавишу в свободном месте папки. Из контекстного меню выберите «Вставить». Файлы скопируются.

- Прделайте то же самое, но вместо вызова контекстного меню нажмите сочетание <Ctrl> + <C>. Перейдите в папку, куда нужно скопировать и нажмите <Ctrl> + <V>. Файлы скопированы.

- ▶ Переключаться между панелями можно при помощи клавиши <Tab> (↔). ◀
- ▶ Если при выделении держать нажатой клавишу <Ctrl>, можно выделить объекты, расположенные в произвольном порядке. ◀
- ▶ Прослушать песни можно при помощи программы Xmms (xmms). В К-меню: «Звук и видео» → «Музыкальный проигрыватель». ◀

2 Служебные программы. Стандартные программы общего назначения.

2.1 Содержимое К-меню

Рассмотрите содержимое К-меню. Отсюда можно получить доступ к большинству установленных в системе программ. Однако, можно запустить программу и при помощи окна запуска программы или из консоли Konsole (см. предыдущее занятие).

Выполните задание 1:

Запишите названия некоторых программ из К-меню в файл, который сохраните в своей рабочей папке (создайте в ней папку «Урок 2», куда и сохраняйте все, что сделаете сегодня).

- ▶ На каждое задание создавайте отдельную папку, куда и сохраняйте свою работу. Этим вы облегчите себе поиск недоделанных упражнений. Кро-

ме того, возьмите в привычку давать файлам обдуманное имя. Как можно реже используйте русский язык при именовании файлов и папок: существует слишком много различных кодировок для отображения кириллицы, и есть вероятность, что человек, которому вы пошлете свой файл, не сможет его открыть из-за проблем с названием. ◀

Запустите перечисленные программы и выполните предложенные задания.

Выполните задание 2:

`gedit` – подобие «Блокнота» в W. Программа позволяет набирать простейшие тексты. Для набора и форматирования сложных текстов необходимо установить дополнительное ПО. Наберите в `gedit` названия стандартных программ. Сохраните этот файл под именем «1.txt» в папке «Урок 2».

Выполните задание 3:

`kolourpaint` – программа для создания примитивных растровых изображений. Запустите программу. Ознакомьтесь с содержанием меню и левой панели инструментов.

Вы имеете возможность, выбирая на левой панели необходимый инструмент, рисовать различные фигуры, задавать их цвет, заливать краской.

Создайте несколько замкнутых фигур разного цвета (цвет можно выбрать на нижней панели инструментов). Залейте их другими цветами.

Выделите часть изображения и перенесите его в другое место. Выделение производится при перемещении мыши с нажатой правой кнопкой. Для вырезания выделенного участка используйте сочетание клавиш `<Ctrl> + <X>` или `<Shift> + <Delete>`. Скопировать выделение можно сочетанием `<Ctrl> + <C>` или `<Ctrl> + <Insert>`. Вставить скопированный фрагмент можно при помощи сочетания `<Ctrl> + <V>` или `<Shift> + <Insert>`.

Нарисуйте на изображении несколько разноцветных отрезков и кривых Безье. Обратите внимание на особенности построения этих кривых: они строятся по нескольким опорным точкам и касательным.

Посмотрите, что произойдет при заливке незамкнутой фигуры. Для этого можете, например, построить замкнутую фигуру, удалить небольшую часть ее границы «ластиком», а затем попробовать залить ее каким-нибудь цветом. *«Откат» операции можно сделать при помощи сочетания клавиш `<Ctrl> + <Z>`.*

Вставьте в изображение несколько надписей. Используйте разные шрифты, размеры и цвет текста.

Полученный рисунок сохраните под именем `test.bmp` в своей рабочей папке.

Выполните задание 4:

Создайте новый рисунок размером 800x600 точек. Нарисуйте, что захотите. Желательно использовать при рисовании как можно больше инструментов, а не только «карандаш» и «кисть».

Второй рисунок тоже сохраните в домашней папке.

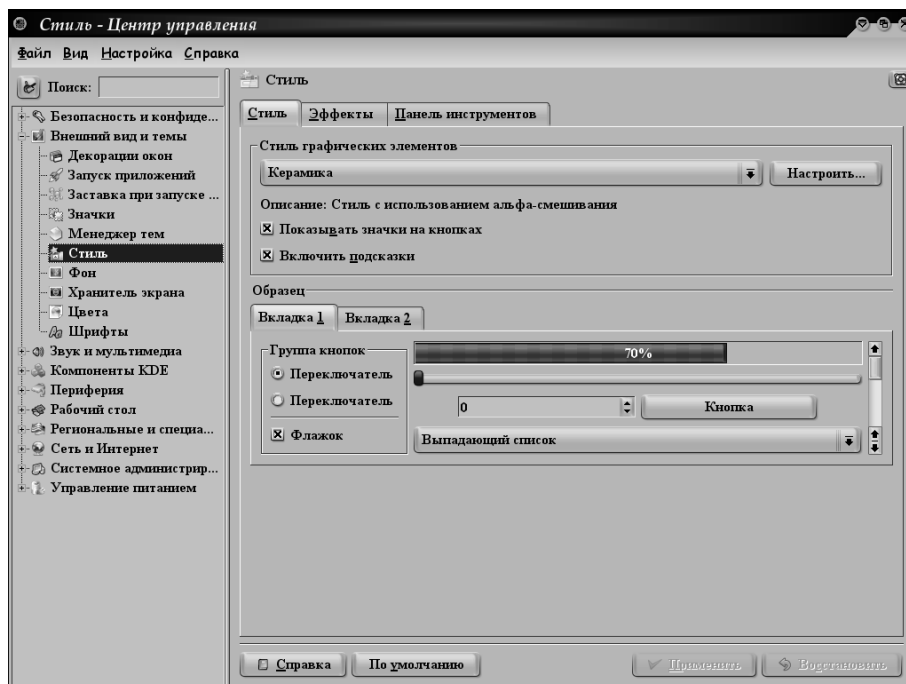


Рис. 3: Центр управления.

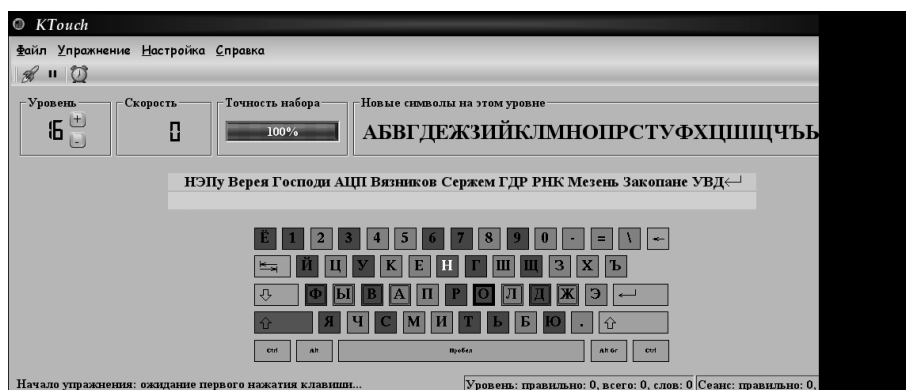


Рис. 4: Программа-тренажер слепой печати K-Touch.

Выполните задание 5:

`konsole` – текстовая консоль. Запустите сеанс. Просмотрите содержимое папки, в которой открылся сеанс командой `ls`.

Ответьте на вопрос(ы):

- Какой тип графики используется в редакторе `kolourpaint`: векторный или растровый?
- Каковы принципиальные отличия растровой графики от векторной?
- Увеличьте или уменьшите изображение. Можно ли рисовать чертежи в этом редакторе?
- Как приблизительно вычислить, сколько места займет на диске картинка, нарисованная в `xpaint`?

Выполните задание 6:

Ознакомьтесь с содержанием других вложенных папок. Запустите 2-3 программы и опишите их назначение.

Запустите «Центр управления» (см. рис. 3) и попытайтесь настроить внешний вид KDE по своему усмотрению.

Выполните задание 7:

После того, как выполните все задания, запустите программу `ktouch` («К-меню» → «Образование» → «Разное» → «К-Touch» или из консоли или окна запуска программ, набрав название приложения). Эта программа помогает научиться слепой печати. Занимаясь на каждом занятии в этой программе после выполнения основного задания вы сможете более привыкнуть к клавиатуре и повысить скорость набора текста.

2.2 Работа в виртуальной консоли

Для того, чтобы не переключаться постоянно из графической оболочки в консоль существуют **виртуальные консоли**. Самая примитивная из них — вызывается командой `xterm` (см. рис. 5).



Рис. 5: Виртуальный терминал (консоль) Xterm

Виртуальная консоль практически аналогична консоли реальной. В ней вы также можете выполнять команды, редактировать файлы и т.п.

Более «красивая» виртуальная консоль — `konsole` (см. рис. 6).

Она уже более приспособлена к X, чем `xterm`, и позволяет пользователю менять свое оформление.

Выполните задание 8:

Запустите программу `konsole` (либо из окна «Выполнить программу», либо из К-меню → «Система» → «Терминал», либо же из панели быстрого запуска, нажав на иконку с изображением монитора).

Настройте на свой вкус тему консоли (нажмите правую клавишу мыши, затем выберите из контекстного меню пункт «Настройка»).



Рис. 6: Виртуальная консоль Konsole (слева), ее «прозрачная» тема (справа)

2.3 Дополнительные возможности файловой системы

Продолжим работу с файловыми системами. Для того, чтобы работать в Linux, можно и не знать подробностей функционирования его деталей, как не задумываемся мы, например, о функционировании компьютера, когда работаем за ним. Однако, зная особенности операционной системы, можно значительно повысить скорость и продуктивность своей работы.

Linux поддерживает много типов ФС. Но наиболее распространены три: `xfs`, `reiserfs` и `ext3fs`. На вашем компьютере разделы жесткого диска отформатированы в `reiserfs`. В любом случае, в какой бы ФС ни были отформатированы разделы, в Linux существует следующая организация файловой структуры: существует каталог верхнего уровня (**корневой каталог** — `/`), в котором размещены каталоги следующего уровня.

Другие разделы жесткого диска можно **монтировать** в любой каталог командой `mount`. Вам разрешается монтировать только сменные накопители информации. Остальные разделы файловой системы в целях безопасности разрешено монтировать только суперпользователю `root` (администратор компьютера). Более простая процедура монтирования диска — через его иконку на рабочем столе (пункт «Подключить» контекстного меню). Для **отмонтирования** диска используется команда `umount` или пункт «Отмонтировать» (или «Извлечь» — для компакт-диска) контекстного меню иконки диска.

В файловых системах Linux помимо понятий файл и каталог существуют особые файлы: **символическая и жесткая ссылки**. Создать жесткую ссылку можно командой `ln <имя файла> [<имя ссылки>]`, а символическую — `ln -s <имя файла> [<имя ссылки>]`. Символическая ссылка — это файл, в котором записан полный путь к другому файлу. Жесткая ссылка возможна *только внутри того же раздела, на котором размещена цель ссылки*, она представляет собой запись в каталоге, содержащую *файловый дескриптор цели*. Можно сказать, что жесткая ссылка — это копия файла, которая практически не занимает места на диске. Ссылка отличается от копии тем, что если редактировать цель ссылки, то по всем ссылкам получим измененный файл. По понятиям безопасности нельзя создавать жесткие ссылки на папки (иначе может получиться циклическая запись ссылок, в итоге которой из папки нельзя будет выйти).

- ▶ Привыкайте, что многие команды требуют указания **параметров** — специальных ключей, перед которыми расположен символ одного или двух минусов. Эти ключи необходимы для расширения возможностей команды. Например, `ls` выводит содержимое текущего каталога, а `ls -l` выводит подробные сведения о каждом файле и каталоге. Узнать возможности команды можно, набрав `man <имя команды>`. ◀

Выполните задание 9:

Создайте в своем домашнем каталоге `/home/<ваш логин>` файл `link` (если забыли — командой `touch link`). Создайте на него жесткую и символическую ссылки (`ln link h_link` и `ln -s link s_link`). Внесите в этот файл запись: `echo "lalala" > link`. Посмотрите, как изменились файлы `s_link` и `h_link`: `cat s_link; cat h_link`.

- ▶ Не забывайте после ввода каждой команды нажимать `<Enter>`. Можно написать несколько команд в одну строку, нажав `<Enter>` в самом конце. Для этого разделяйте команды символом точки с запятой (`;`). ◀

Удалите файл `link`: `rm link`. Теперь посмотрите опять файлы `s_link` и `h_link`. Как видите, в файле — жесткой ссылке ничего не изменилось, а при попытке открыть символическую появилась ошибка: `No such file or directory`, говорящая, что объект ссылки удален.

В каждой папке *всегда существуют по крайней мере две других*: ссылка на саму себя (`.`) и ссылка на родительскую папку (`..`). Ссылка на саму себя существует для запуска программ, находящихся в данной папке, а ссылка на родительский каталог — для удобства перемещения (команда `cd ../` перемещает вас на один уровень файловой системы выше).

Выполните задание 10:

Откройте для редактирования свой файл `Help.txt`. Запишите в него все новые команды, которые вы узнали за сегодняшний урок.

2.4 Подключение сменных носителей

Для удобства подключения сменных носителей в Linux была разработана среда `hal`, следящая за подключением различных устройств. Как только вы вставляете компакт-диск или флэш-карту, `hal` монтирует это устройство, и дает сигнал KDE к запуску этого устройства. Появляется диалоговое окно, в котором можно выбрать действия, которые вы хотите произвести с устройством (открыть для просмотра содержимого в файловом менеджере, запустить просмотр видео или другие действия).

- ▶ При работе с флэш-картами или дискетами следует помнить, что **перед извлечением сменного носителя его следует отмонтировать**. Несоблюдение этого непременно приведет к повреждению файлов, которые вы сохраняли перед извлечением, а в некоторых случаях вы можете таким образом повредить носитель информации. Особенно осторожно следует обращаться со сменными жесткими дисками. CD-ROM, в отличие от других носителей, без отмонтирования извлечь не получится. ◀

Если `hal` не открыл окно выбора действий с подключенными устройствами, можно открыть его либо в графическом, либо в текстовом файловом менеджере.

2.4.1 Подключение сменных носителей при помощи Konqueror

Подключите к компьютеру сменный носитель и откройте Konqueror (можно запустить его, активировав иконку «Домой»). В левой боковой панели выберите вкладку «Устройства». Справа отобразится список подключенных к компьютеру устройств хранения информации (см. рис. 7).

Щелкните правой кнопкой мыши по значку нужного носителя информации и из контекстного меню выберите пункт «Монтировать» (он также может называться «Подключить»). Можно сразу дважды щелкнуть по этому значку, тогда устройство подмонтируется, и в файловом менеджере откроется папка, в которую оно подмонтировано.

Отмонтировать устройство можно таким же образом, выбрав из контекстного меню пункт «Отмонтировать» (или «Извлечь» для CD-, DVD- или BlueRay-ROM).

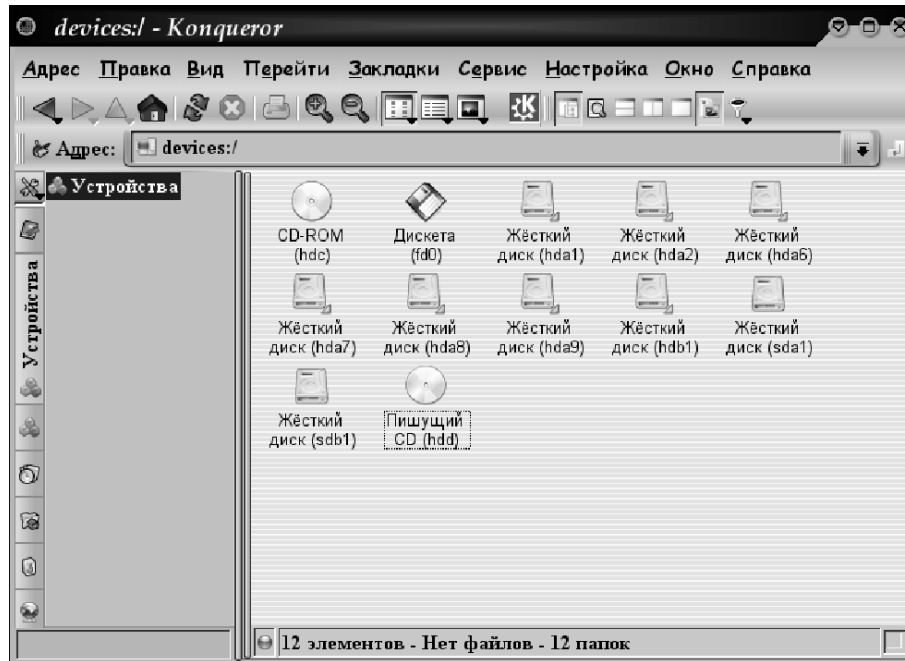


Рис. 7: Подключение сменных носителей при помощи Konqueror

2.4.2 Подключение сменных носителей из консоли

Для выполнения этой операции, запустите терминальный сеанс (например, командой `konsole`). Все сменные носители `hal` монтирует к папкам, динамически создаваемым в директории `/media`. Зайдите в эту папку и просмотрите ее содержимое (`cd /media`, затем `ls`). Вы увидите следующие папки: `floppy` для дисководов, `cdrom` (`dvdrom`, `cdwriter` или `dvdwriter` в зависимости от типа устройства) для CD- или DVD-ROM, `usbdevice` для флэш-карт или переносных жестких дисков.

Наберите `mount <папка>`, где вместо `<папка>` укажите папку, соответствующую устройству, которое вы хотите подключить. Просматривать файловые ресурсы удобнее при помощи `Midnight Commander`.

После выполнения всех необходимых операций, перед тем как извлечь сменный носитель, не забудьте отмонтировать его командой `umount`.

2.4.3 Подключение сканеров и камер

Подключение сканеров и камер тоже автоматизировано донельзя. Если вы хотите просмотреть и/или сбросить на жесткий диск фотографии с цифровой камеры, подключите ее к USB-порту компьютера. Затем запустите программу `gtkam`. Она проведет сканирование всех USB-портов, и если подключенная камера совместима с этой программой, `GtKam` опознает ее. В левой панели программы можно будет найти пункт, соответствующий вашей камере. Выбирая папки с фотографиями в левой панели, вы можете просматривать их содержимое в

правой. При помощи меню можно выбрать нужные действия с фотографиями.

Сканирование в Linux тоже осуществляется довольно просто: подключите к USB-порту свой сканнер. Затем запустите программу `xsane`. Если ваш сканнер поддерживается программой, откроется окно параметров сканирования и предварительного просмотра отсканированного. Интерфейс программы, как и интерфейс GtKam, достаточно интуитивный, т.е. достаточно поверхностного знакомства с работой в графических интерфейсах пользователя, чтобы осуществлять простейшие операции в этих программах.

3 Форматирование текстового документа.

В текстовых редакторах имеется множество дополнительных функций, ускоряющих и облегчающих работу с текстами:

- расстановка переносов;
- изменение масштаба просмотра документа;
- автоматическая нумерация страниц;
- создание границ вокруг текста и страниц;
- разработка своего стиля и использование его в дальнейшем для быстрого форматирования текста;
- поддержка нескольких словарей для проверки правильности написания;
- работа с макросами;
- поддержка нескольких окон;
- удобная справочная система;
- проверка орфографии, грамматики и синтаксиса;
- настройка панелей инструментов (добавление и удаление кнопок и команд);
- возможность предварительного просмотра перед печатью;
- настройка параметров страницы и поддержка нестандартных форматов бумаги;
- вставка в текст специальных символов.

В Linux есть много разных текстовых редакторов: от примитивных вроде `gedit` до сложных, как Open Office (OO). Gedit предназначен для набора простых текстов, которым не нужно содержать сложного форматирования и т.п. (например, компьютерных программ).

OO — сложная офисная система, предназначенная для набора, редактирования и форматирования текстовых документов любой сложности. Конечно, при помощи этой программы нельзя создавать профессиональные текстовые документы для печати в типографии (такие документы набирают в специальных типографских системах — $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$), но для офисного использования

ОО подходит как нельзя лучше.

Помимо текстовых документов Open Office позволяет создавать электронные таблицы, презентации, векторные рисунки и диаграммы, документы html, сложные составные документы, формулы, наклейки, визитные карточки, письма, базы данных.

3.1 Знакомство с Open Office Writer

3.1.1 Запуск Open Office Writer

Зайдите в К-меню и выберите из подменю «Офис» → «Open Office.org» пункт «Writer». Запустится текстовый редактор Open Office Writer.

В открытом редакторе всегда отображаются две панели инструментов: «Стандартная» и «Форматирование». Панели содержат кнопки редактирования и форматирования текста, а также выпадающие списки, в которых можно выбрать стиль текста, его гарнитуру (шрифт) и размер. Для просмотра описания пунктов панели нужно привести на них курсор мыши, не нажимая никаких клавиш. Через небольшой промежуток времени под курсором загорится описание данной кнопки или списка.

Чтобы добавить или удалить панель инструментов, используется меню «Вид» → «Панели инструментов».

В случае возникновения вопросов по нажатию кнопки <F1> появляется окно помощи, где можно найти описания и советы. Кроме того, в новых версиях офисных пакетов существует т.н. «Помощник», дающий советы в процессе работы.

На рис. 8 изображено окно программы с написанным в нем текстом.

Выполните задание 1:

Ознакомьтесь с меню и панелями инструментов. Сохраните сразу пустой документ в своей рабочей папке, чтобы не потерять набранный текст в случае сбоя компьютера (каждые 15 минут ОО Writer автоматически сохраняет документ).

- Сохранить документ можно, по крайней мере, тремя способами: во-первых, самый простой — нажать <Ctrl> + <S>; во-вторых — щелкнуть по значку дискеты на панели инструментов; а в-третьих — выбрав из меню «Файл» пункт «Сохранить». ◀

3.1.2 Навигация по тексту

Под навигацией понимают перемещение курсора внутри текста. Для навигации можно использовать колесо мыши, прокручивающее документ построчно; пол-

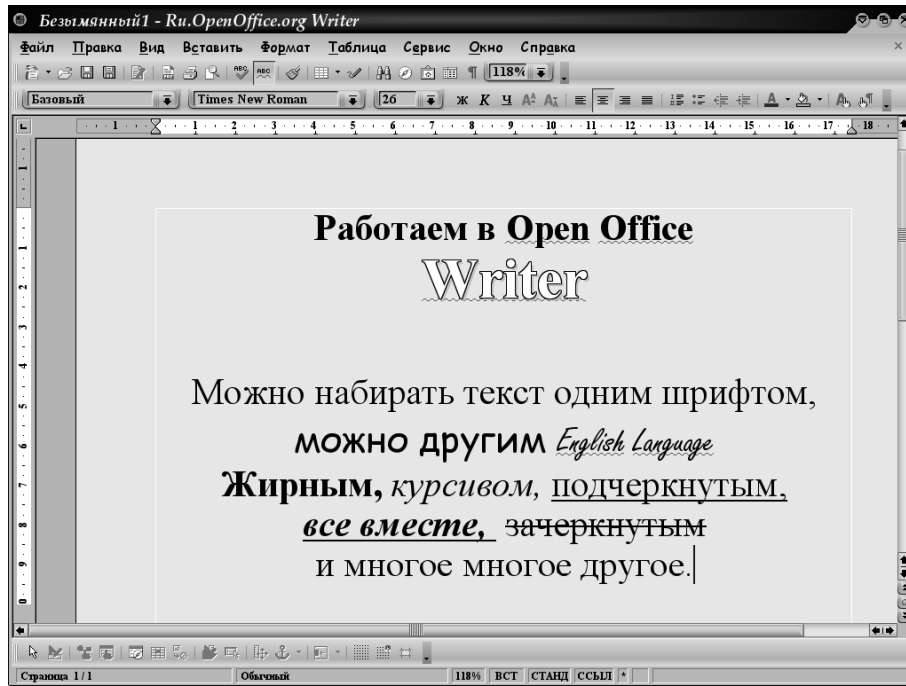


Рис. 8: Окно программы Open Office Writer.

зунок навигации, расположенный справа от окна ввода текста (ухватившись за него мышью); или же клавиши управления курсором.

Следует учитывать, что прокручивая текст при помощи мыши, для перемещения курсора в нужное положение следует щелкнуть в этом месте левой клавишей мыши.

Клавиатурная навигация использует следующие клавиши. Клавиши управления курсором позволяют перемещаться на одну строку вверх или вниз, или на один символ влево или вправо. Клавиши <Page Up> и <Page Down> перемещают курсор на страницу вверх и вниз, соответственно. Клавиши <Home> и <End> перемещают курсор в начало или конец строки, соответственно.

Можно использовать клавиши управления курсором с нажатой клавишей <Ctrl>. Тогда курсор перемещается влево или вправо на одно слово, а клавиши <Page Up> и <Page Down> перемещают его в начало и конец документа, соответственно.

3.1.3 Методы ввода текста

Существует два режима **ввода текста**: вставка (ВСТ) и замена (ЗАМ), которые отображаются в строке статуса программы (внизу окна). Переключение между режимами осуществляется клавишей <Insert>. По умолчанию действует режим вставки, когда новые символы при вводе «раздвигают» старые. В режиме замены новые символы заменяют символы, расположенные справа от курсора.

Удобная функция Open Office — **автодополнение**: при вводе длинных слов Open Office предлагает их окончание. Если оно вас устраивает, нужно нажать клавишу <Enter>. Параметры автодополнения, также как и многие другие, можно настроить в свойствах Open Office.

Удаление текста осуществляется клавишами <Backspace> (она удаляет символ, расположенный слева от курсора) и <Delete> (удаляется символ справа). Кроме того, можно выделить мышью или при помощи клавиатуры (удерживая нажатой клавишу <Shift> и перемещая курсор) фрагмент текста и удалить его весь, нажав <Delete> или <Backspace>.

Иногда бывает необходимо скопировать или переместить фрагмент текста в другую часть документа или в другой документ. Для этого необходимо выделить нужный фрагмент, а затем скопировать его в **буфер обмена**. В Linux существует два буфера обмена: в первый автоматически помещается выделенный мышью текст. Для того, чтобы вставить этот текст в другое место, необходимо нажать **среднюю клавишу мыши** (колесо). Однако, такой способ неудобен, если выделенный фрагмент необходимо вставлять несколько раз. Для этого используется второй буфер обмена. Чтобы поместить в него выделенный фрагмент, нажмите сочетание <Ctrl> + <C>. Чтобы вырезать фрагмент в буфер обмена, нажмите <Ctrl> + <X>. Затем установите курсор в позицию, куда вы хотите вставить этот фрагмент, и нажмите <Ctrl> + <V>. Содержимое второго буфера остается в нем до тех пор, пока вы не поместите в него новую информацию.

3.1.4 Форматирование текста

При вводе текста бывает необходимо задать различные способы отображения символов: жирный, курсив, подчеркнутый. Для этого можно использовать кнопки **Ж**, **К** и **П** панели инструментов, либо нажав клавишу <Ctrl> и, не отпуская ее, для жирного, <U> для подчеркнутого и <I> для курсивного текста. Вернуть обратно нормальное начертание можно, еще раз нажав соответствующую кнопку, или сочетание клавиш.

Помимо изменения способа начертания вы можете изменять гарнитуру (отображаемый шрифт) текста, его размер и цвет. Для этого используйте соответствующие элементы панелей инструментов.

Помимо символов, существует более крупная единица измерения объема в текстовом документе — абзац. Форматирование абзацев заключается в возможности задания межстрочного интервала (Open Office предлагает три размера: единичный, полуторный и двойной, но из меню «Файл» → «Абзац» можно выбрать и другие интервалы); величины абзацного отступа; выравнивания (по левому краю, по правому краю, по середине или по ширине).

3.1.5 Открытие и сохранение документов

Для сохранения документа щелкните изображение дискетки на панели инструментов, выберите из меню «Файл» пункт «Сохранить», или же нажмите сочетание клавиш <Ctrl> + <S>. Если документ сохраняется первый раз, появится диалоговое окно, в котором необходимо выбрать путь к сохраняемому файлу и ввести его имя.

Открыть документ можно выбрав из меню «Файл» пункт «Открыть», щелкнув по иконке с изображением открытой папки, либо же нажав сочетание <Ctrl> + <O>.

Если файл требуется сохранить в другом формате (например, в формате MS-Word), используется пункт меню «Файл» → «Сохранить как».

3.2 Ввод и редактирование текста

Выполните задание 2:

Наберите текст:

Я набираю этот текст не только для того, чтобы получить оценку. Это, прежде всего, необходимо **мне самому**. Я хочу *научиться работать с компьютером* не на уровне рядового пользователя, которого программисты **снисходительно** называют **ламером**, а на уровне продвинутого пользователя.

Практические занятия имеют огромное значение: я смогу *освоить различные программы*, которые могут пригодиться мне в дальнейшей жизни не только на работе, но и дома. Ведь в наше время очень тяжело представить себе человека, никоим образом не относящимся к информационно-вычислительной технике.

Выполнение практических заданий *носит характер самообучения*. Я сам постигаю основы работы с ПК, преподаватель же выполняет роль **консультанта** в случае, когда что-то мне не понятно. Но стоит заметить, что я учусь все лучше и лучше, и все реже прошу подсказать мне, **КАК ВЫПОЛНИТЬ ТУ ИЛИ ИНУЮ ОПЕРАЦИЮ^a**.

^aНапример, я уже знаю как вставлять сноски

- Кегль (размер символов), а также гарнитуру (вид шрифта) можно задать, изменяя значения в выпадающих списках на панели инструмен-

тов. Начертание символов можно изменить либо нажав кнопки «Ж», «К», «П» на панели инструментов, либо сочетания клавиш: <Ctrl> + , <Ctrl> + <I>, <Ctrl + U> соответственно для жирного, курсивного или подчеркнутого начертания. ◀

Выполните задание 3:

Сделайте надпись, использующую нижние и верхние индексы, спецсимволы, например, $X_i^k = \sum X_{ij}^k \cdot y_i$.

- ▶ Для того, чтобы введенный символ был верхним или нижним индексом, необходимо выделить его и в контекстном меню (по нажатию правой кнопки мыши) выбрать пункт «Формат ⇒ Символ», где пометить нужный атрибут символа. Второй способ — выделить символ и щелкнуть на панели инструментов кнопку «Верхний индекс» или «Нижний индекс». ◀

3.2.1 Использование стилей в документе

Большое удобство представляют **стили**, позволяющие упростить процесс форматирования документа. Чтобы применить к тексту тот или иной стиль, необходимо выбрать этот стиль из выпадающего меню панели инструментов, либо из окна «Стили», появляющегося при нажатии <F11>.

Основным стилем документа является «Основной» (или «Базовый»). Также существуют специальные стили для содержимого таблиц, заголовков, перечней и т.п. Изменяя определенный стиль, вы сразу же изменяете отображение всего текста, имеющего такой же стиль.

Выполните задание 4:

Создайте стиль на основе стиля «Основной» со следующими параметрами:

- Кегль (размер шрифта) — 14 пт;
- Первая строка — с отступом 1 см;
- Отступы слева и справа — 0 см;
- Выравнивание — «По ширине».
- Гарнитура — «Urw Bookman»

Дайте стилю название «Мой стиль».

Таким же образом на основе стиля «Заголовок 1» и «Заголовок 2» создайте свои собственные стили заголовков 1-го и 2-го уровня.

Выравнивание обоих уровней заголовков создайте по центру, гарнитура — «Urw Bookman», кегль у заголовка 1-го уровня — 30pt, у заголовка 2-го уровня — 20pt. Заголовок 1-го уровня сделайте жирным, а 2-го — курсивным. До и после заголовков оставьте пространство, равное кеглю заголовков.

(продолжение задания 4)

Назовите заголовки «Уровень 1» и «Уровень 2».

Перед набранным до этого текстом создайте заголовок 1-го уровня: «Практическая работа 1», затем, следом за ним — заголовок 2-го уровня: «Форматирование текста». Весь остальной текст оформите в стиле «Мой стиль».

3.2.2 Таблицы**Выполните задание 5:**

Начните задание с новой страницы. Для этого вставьте в документ разрыв страницы.

Наберите стилем «Уровень 2» заголовок «Форматирование таблиц».

Создайте и заполните таблицу — часть расписания ваших занятий. Для этого в меню «Вставить» выберите пункт «Таблица». Появится окно, изображенное на рис. 9.

«Имя таблицы» — необязательное поле. В поле «Размер таблицы» установите количество столбцов — 6, количество строк — 9. Поле «Верхний колонтитул» необходимо отметить галочкой, если вы хотите, чтобы заголовок таблицы выделялся жирным шрифтом. Поле «Повторять верхний колонтитул» отметьте, если таблица будет переноситься на следующую страницу, чтобы заголовок повторялся. Поле «Не разбивать таблицу» следует отметить, если вы не хотите, чтобы она переносилась между страницами. Если снять отметку с поля «Обрамление», таблица создастся с прозрачными границами.

- Разрыв страницы можно создать двумя способами: либо выбрав из меню «Вставка» пункт «Разрыв» → «Разрыв страницы», либо нажав сочетание клавиш <Ctrl> + <Enter>. ◀

№пары	Понедельник	Вторник	Среда	Четверг	Пятница
1	математика	—	физика	—	рус. язык
2	химия	технология	спецпредметы	—	физика
3	информатика	спецпредметы	спецпредметы	—	литература
4	биология	—	химия	—	экология
5	—	—	—	информатика	—
6				физика	
7				рус. язык	
8				литература	

- Для объединения нескольких ячеек выделите их, щелкните внутри выделения правой кнопкой мыши, из контекстного меню выберите «Ячейка» → «Объединить». Аналогично, чтобы разбить ячейку на несколько, выберите пункт «Разбить». Для того, чтобы изменить вид границ таблицы, выделите ячейки, в которых необходимо сменить обрамление, из кон-

текстного меню выберите «Таблица». На появившемся окне откройте вкладку «Обрамление». Здесь можно настроить толщину линии в выбранных ячейках и ее положение (щелкните в нужном месте на схеме, расположенной в левой части окна).

Выполните задание 6:

Создайте таблицу:

Расписание дежурства

Время	Понедельник	Вторник	Среда	Четверг	Пятница
8.00–10.00	Савельев А.В.	Иванов Б.П.	Петров Г.А.	Жуков С.В.	Кузнецов А.А.
10.00–12.30	Иванов Б.П.	Петров Г.А.	Савельев А.В.	Кузнецов А.А.	Жуков С.В.
12.30–13.30	Перерыв				
13.30–15.00	Петров Г.А.	Жуков С.В.	Кузнецов А.А.	Савельев А.В.	Иванов Б.П.
15.00–17.00	Жуков С.В.	Кузнецов А.А.	Иванов Б.П.	Петров Г.А.	Савельев А.В.



Рис. 9: Окно «Вставить таблицу».

Чтобы задать какое-то выравнивание в ячейках таблицы необходимо выделить эти ячейки, а затем нажать на панели инструментов кнопку, соответствующую желаемому типу выравнивания. Кроме того, можно нажать на выделении правой кнопкой мыши и из контекстного меню выбрать нужный тип выравнивания.

- ▶ Сложные границы можно получить, выделив нужные ячейки, и щелкнув внутри них правой клавишей мыши. Из контекстного меню выберите пункт «Таблица», в открывшемся окне выберите «Обрамление». Слева располагается макет ячейки. Щелкните по тому краю макета, у которого вы хотите изменить тип линии. Затем выберите из выпадающего списка нужный тип линий (или «Нет линий», если хотите сделать линию невидимой). Нажмите ОК.

3.2.3 Колонтитулы

Колонтитулы — элементы, располагающиеся в самой верхней и самой нижней частях страницы. Обычно в колонтитулы заносится такая информация, как

номера страниц, названия разделов, сведения об авторе, название книги.

Колонтитулы отображаются на каждой странице, однако, можно настроить их поведение так, чтобы они не отображались на первых нескольких страницах (где обычно располагаются титульный лист и, например, оглавление).

Можно использовать колонтитулы и в других целях. Например, вы хотите, чтобы на каждой странице у вас отображалась чертежная рамка. Для этого необходимо создать в нижнем колонтитуле таблицу, которая будет также выходить и на основной текст. В клетку «страница» рамки можно занести поле «номер страницы» для автоматической нумерации.

Выполните задание 7:


Добавьте в документ верхний и нижний колонтитул («Вставить» → «Верхний/Нижний колонтитул» → «Обычный»). На верхнем колонтитуле отобразите общее количество страниц документа («Вставить» → «Поля» → «Количество страниц»), на нижнем — номер страницы (аналогично предыдущему).

Отредактируйте документ, задав размер левого поля 2.5 см, остальных — по 1 см. На колонтитулы отведите по 1.5 см. Для этого в меню «Формат» активируйте пункт «Страница».

3.3 Автоматические элементы

3.3.1 Использование автоматических списков

Выполните задание 8:

Создайте список, содержащий несколько уровней. Для создания списков существуют кнопки . Левая — для создания нумерованного списка, правая — маркированного.

Используйте различные стили.

После ввода каждого элемента списка нажимайте <Enter>. Тогда автоматически будет увеличиваться счетчик текущего уровня списка.

Менять уровень списка можно клавишей TAB или shift+TAB. Кроме того, можно воспользоваться кнопками изменения уровня списка панели инструментов «Список».

- ▶ Для изменения стиля списка из меню «Формат» в верхней части окна выберите пункт «Нумерация и маркировка». На вкладке «Настройки» появившегося окна можно настроить стиль многоуровневого списка. Выберите из списка слева номер уровня, в стилях «Нумерация», «Стиль символа» и, если нужно, «Символы» выберите оформление данного уровня. На вкладке «Положение» можно настроить отступы от края страницы

до начала маркера и от маркера до текста (см. рис. 10). Затем нажмите «ОК».

Пример:

Животные:

1. Домашние:

1. *грызуны*
 - мышь
 - крыса
2. *хищники*
 - собака
 - кошка

2. Дикие:

1. *млекопитающие*
 - кенгуру
 - дельфин
2. *земноводные*
 - крокодил
 - черепаха

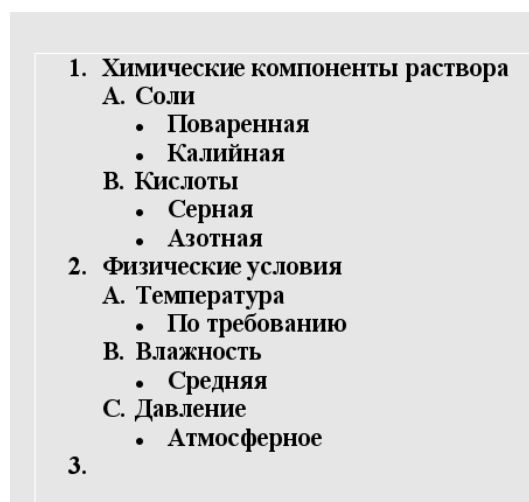
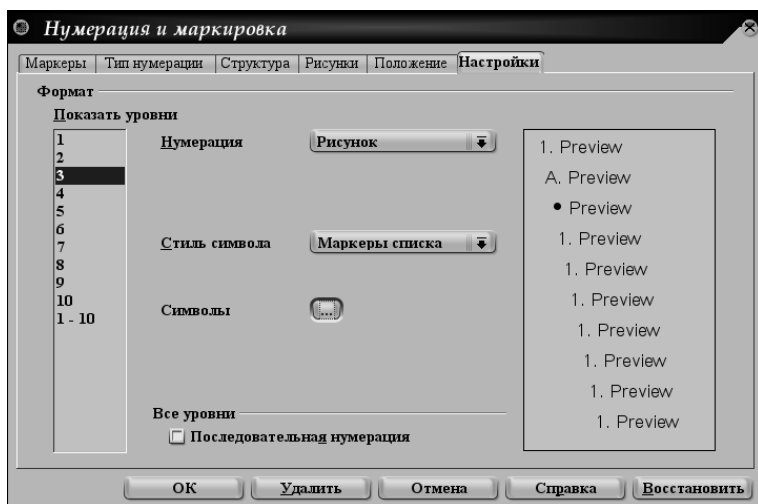


Рис. 10: Изменение стиля списка, внешний вид списка после изменения параметров.

Выполните задание 9:

Теперь постройте еще один список. Образец — далее. Здесь необходимо пропускать некоторые уровни списка.

Химические элементы:

1. Кислоты:

1. *Неорганические*
 - соляная¹

¹широко используется в народном хозяйстве

- серная

2. *Органические*

- аскорбиновая²
- уксусная³

2. **Щелочи:**

- едкий натр
- едкое кали

3. **Нейтральные вещества**

1. *Растворители*

- вода
- этанол

3.3.2 Сноски

Выполните задание 10:

Добавьте к некоторым элементам автоматические сноски⁴.

Чтобы вставить сноску, установите курсор в нужном месте текста. Затем из меню «Вставить» выберите пункт «Сноски». На появившемся диалоговом окне выберите обычные автоматические сноски (концевые сноски со всего документа печатаются в его конце). Когда вы нажмете «ОК», автоматически появится символ сноски, а курсор переместится в нижнюю область страницы, где вы сможете ввести текст сноски. Чтобы вернуться к редактированию текста, щелкните по основному тексту левой клавишей мыши.

3.3.3 Оглавление

Если в вашем документе не применялись стили «Уровень 1» и «Уровень 2», то создайте в нескольких местах документа заголовки. Желательно, чтобы документ состоял из нескольких страниц, и на каждой было хотя бы по одному заголовку 1-го уровня и 2-3 заголовка 2-го уровня.

Выполните задание 11:

Вставьте в самом начале текста автоматический элемент «Оглавление». Для этого воспользуйтесь меню «Вставка» → «Оглавления и указатели» → «Оглавления и указатели». Выберите какой-нибудь стиль оглавления, а также количество уровней, которые будут в нем отображаться.

²а это — всем известный витамин С

³куда без нее на кухне?

⁴Вроде такой

(продолжение задания 11)

Можно сменить заголовок (в левой части появившегося окна отображается предварительный просмотр получившегося оглавления). После этого нажмите «ОК» (см. рис. 11). На странице появится оглавление.

Попробуйте внести в документ новый заголовок, или изменить старый. Теперь, чтобы оглавление обновилось, щелкните на его тексте правой кнопкой мыши и из контекстного меню выберите «Обновление указателя». Все изменения сразу отразятся в оглавлении.

Добавьте в документ несколько страниц (для создания новой страницы нажмите <Ctrl> + <Enter>). На каждой странице создайте по 1...2 заголовка, оформленных разными стилями (от «Заголовок 1» до «Заголовок 3»). Обновите оглавление.

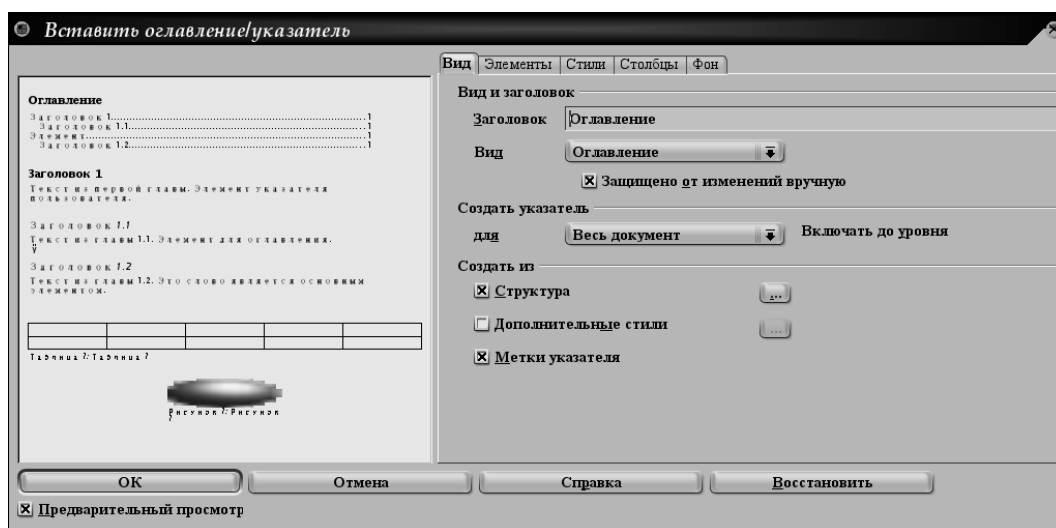


Рис. 11: Меню вставки оглавления и указателей.

3.3.4 Алфавитный указатель

Выполните задание 12:

Выделите какое-нибудь слово, например, если вы пользовались классификацией животных, выделите «дикие». Выберите в меню «Вставка» → «Оглавления и указатели» → «Элемент указателя».

В качестве 1-го ключа укажите «Дикие животные». Укажите, что это — основной элемент.

Затем выделите, например, «Кошка». Опять повторите операцию с выделением указателя, но кроме 1-го ключа укажите 2-й ключ — «Хищники».

Проделайте подобное еще с несколькими словами, затем перейдите в конец документа и вставьте алфавитный указатель также, как вставляли оглавление.

Обратите внимание, что элементы указателя содержатся в особой базе данных, из которой сначала в алфавитном порядке собираются все текстовые элементы, а затем для них выбираются соответствующие страницы. Важно не ошибиться при выборе названия элемента указателя, особенно если он встречается на нескольких страницах. Если в одном месте документа вы назовете этот элемент не так, как в другом, он войдет в алфавитный указатель под двумя именами.

3.4 Формулы

3.4.1 Простые

Выполните задание 13:

На новой странице стилем «Заголовок 1» наберите: «Набор и редактирование формул». Ниже вставьте формулу:

$$\lim_{x \rightarrow 0} \int_0^{\pi} \frac{\sin x}{x} dx = \int_0^{\pi} dx = \pi$$

Для того, чтобы вставить формулу, выберите в меню «Вставить» → «Объект» → «Формула». Экран разделится на две части. Вверху отображается ваш документ, внизу расположено командное окно. В него необходимо вводить команды для создания элементов. Однако, некоторые элементы можно вводить, выбирая из плавающего меню. Если его нет, выберите в меню «Вид» → «Выделенное».

Плавающее окно разделено на две части: вверху расположены кнопки групп символов, внизу — кнопки для вставки конкретного символа. После нажатия кнопки, соответствующей выбранному элементу, необходимо в области, помеченные как <?> ввести текст. Изменения автоматически отображаются в верхней части окна Writer. Если вы все сделали правильно, в нижней части экрана должно быть написано: `lim from {x toward 0} int from 0 to %pi {sin x over x} dx = int from 0 to %pi dx = %pi.`

На первый взгляд кажется сложным создавать формулы таким образом, однако, имея небольшие знания английского и немного практики в наборе формул, вы будете набирать их при помощи клавиатуры значительно быстрее, чем выбирая из меню нужные элементы при помощи мыши.

Обратите внимание, что для обозначения пределов интегрирования, а также условия предела мы использовали `from` для нижнего и `to` для верхнего индекса. Эти команды — полные аналоги соответствующих команд `csub` и `csup`, однако, в отличие от последних, применимы только к математическим операторам с

пределами.

3.4.2 Сложные

Выполните задание 14:

Вставьте в текст «многоэтажную» формулу:

$$F(\omega, N) = \int_{-\infty}^{\infty} E(t) \cdot \frac{\sin \frac{N\omega t}{2\pi}}{\sin \frac{\omega t}{2\pi}} e^{-i\omega t} dt.$$

Код, соответствующий данной формуле:

```
F(%omega,N) = int_{-infinity}^infinity E(t) cdot
{sin {N %omega t} over {2 %pi}} over
{sin {%omega t} over {2 %pi}} e^{-i%omega t} dt.
```

4 Создание визитных карточек и писем

4.1 Создание визитных карточек

Выполните задание 1:

Запустите OO Writer. Вставьте таблицу, состоящую из двух столбцов и шести строк. Обратите внимание на формат страницы. Для того, чтобы на одном листе А4 уместилось 12 визиток, необходимо установить размеры всех полей по 1 см. Зайдите в меню «Формат» → «Страница», активируйте вкладку «Страница» и установите размер всех полей по 1 см, а формат листа — А4.

Теперь для того, чтобы задать фиксированный размер визиток, выделите созданную таблицу целиком и щелкните внутри нее правой клавишей мыши. Из появившегося контекстного меню выберите пункт «Строка» → «Высота». Появится диалоговое окно (см. рис. 13), в котором необходимо задать высоту визитки: 4.5 см. Нажмите «ОК». Затем точно так же щелкните правой клавишей мыши внутри таблицы и задайте ширину столбцов (9 см) из меню «Столбец» → «Ширина».

- Выделить всю таблицу можно, **дважды** нажав <Ctrl> + <A>, когда курсор находится внутри нее. ◀

В итоге у нас получится таблица — эскиз для 12-ти визиток. Перейдем к их заполнению.

Если вы хотите, чтобы текст внутри визиток был отцентрирован, выделите всю таблицу и задайте выравнивание по центру (аналогично для выравнивания по ширине или по правому краю).

Нет необходимости заполнять все ячейки. Для начала заполним одну, а затем, выделив ее содержимое, скопируем во все остальные.



Рис. 12: Создание визитной

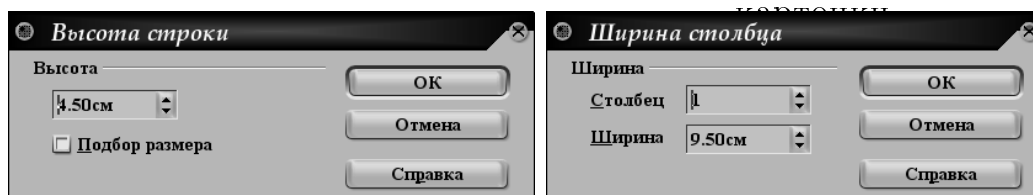


Рис. 13: Задание размеров ячеек таблицы.

- Можно сделать так, чтобы содержимое ячеек было отцентрировано и по вертикали. Для этого, выделив всю таблицу, щелкните внутри нее правой клавишей мыши и из контекстного меню «Ячейка» выберите пункт «По центру».

Выполните задание 2:

Напишите самой первой ячейке крупным жирным шрифтом свое имя. Затем ниже — род деятельности и контактную информацию. В каждую ячейку можно добавить изображение, чтобы сделать визитку более красочной.

Вы можете залить буквы разными цветами, тогда при печати на цветном принтере визитка будет выглядеть значительно лучше, чем простая черно-белая.

У вас получится что-то вроде изображенного на рис. 12.

Теперь в ячейке с базовой визиткой нажмите $\langle \text{Ctrl} \rangle + \langle \text{A} \rangle$, чтобы выделить все содержимое ячейки. Оно сразу попадет в буфер обмена, но для верности можно еще нажать $\langle \text{Ctrl} \rangle + \langle \text{C} \rangle$ (скопировать выделенное в буфер).

- В графической оболочке KDE существует поддержка двух независимых буферов обмена: в один помещается то, что вы выделяете мышью, чтобы вставить его содержимое, необходимо нажать среднюю клавишу мыши; во второй помещается скопированное при помощи сочетания $\langle \text{Ctrl} \rangle + \langle \text{C} \rangle$, чтобы вставить его содержимое, необходимо нажать $\langle \text{Ctrl} \rangle + \langle \text{V} \rangle$. Очень удобно работать с первым буфером: вы выделяете часть текста, которую надо скопировать в другое место документа, и щелкаете средней клавишей мыши там, куда необходимо вставить выделенный

4.2 Создание деловых писем

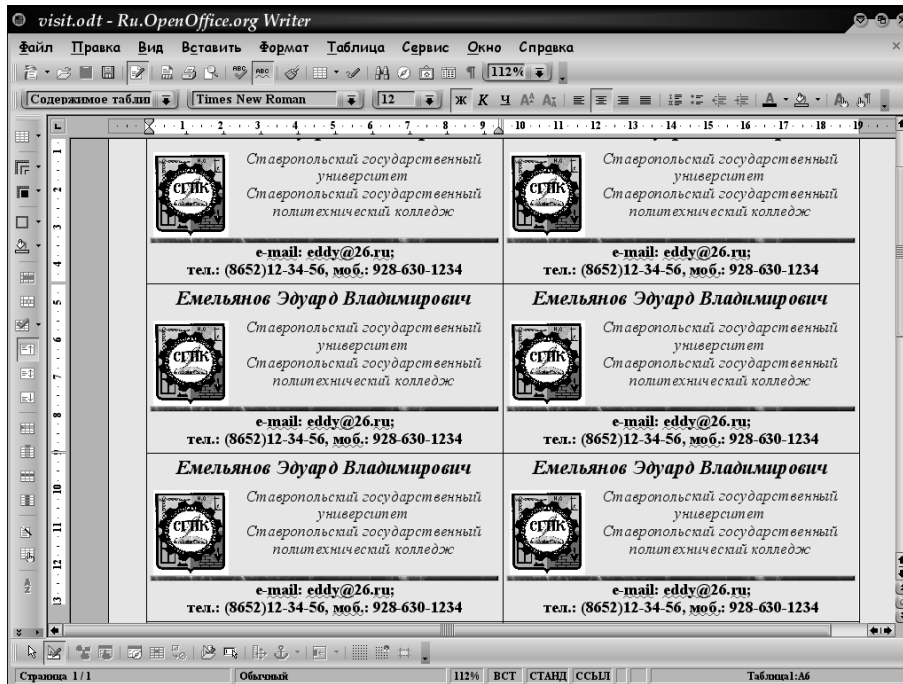


Рис. 14: Копирование визитных карточек.

элемент.

Теперь щелкните мышью внутри пустых ячеек и нажимайте среднюю клавишу мыши (или $\langle \text{Ctrl} \rangle + \langle \text{V} \rangle$). Содержимое буфера обмена будет вставлено в данную ячейку. В результате вы получите 12 визиток на одном листе (см. рис. 14). В дальнейшем достаточно будет распечатать их на хорошей бумаге на лазерном или струйном принтере.

4.2 Создание деловых писем

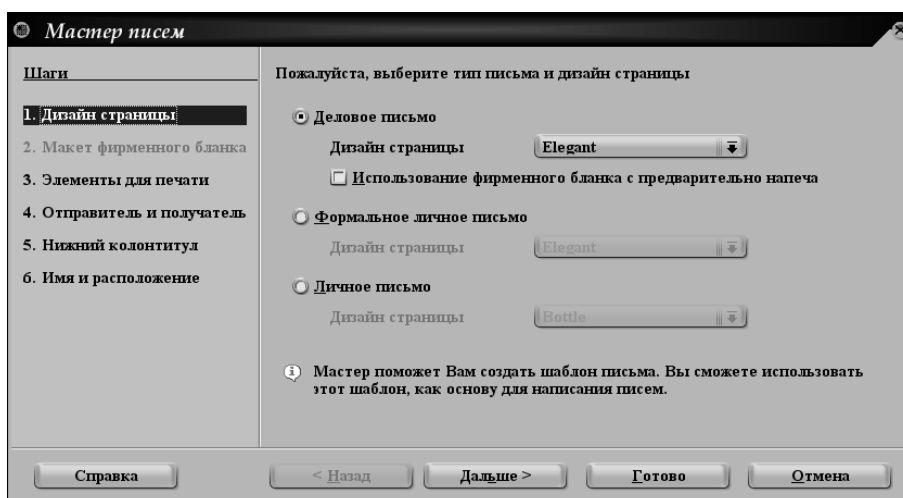


Рис. 15: Мастер писем.

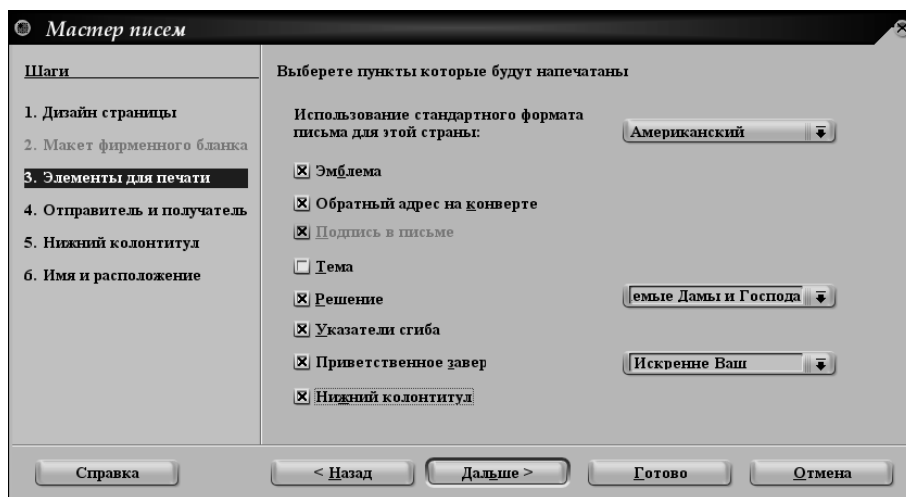


Рис. 16: Второй шаг мастера писем.

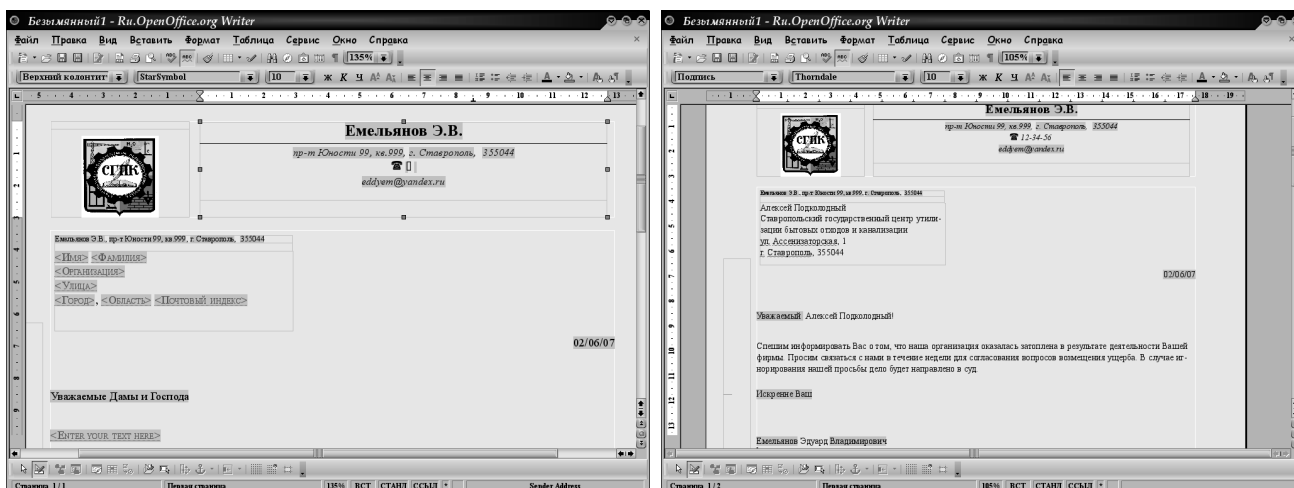


Рис. 17: Шаблон письма (слева) и готовое письмо (справа).

Выполните задание 3:

Деловое письмо легче всего создать при помощи мастера писем. Запустите его из меню «Файл» → «Мастер» → «Письмо». Появится диалоговое окно мастера (см. рис. 15).

Оставьте тип страницы «Деловое письмо». Выберите дизайн страницы (при его смене автоматически обновляется образец письма на фоне). Нажмите «Далее». Установите стандартный формат «Русский» (если такового не окажется, оставьте американский). Установите флажки на элементах, которые должны отображаться в письме (см. рис. 16)

Далее заполните поля «Адрес отправителя». В поле «адрес получателя» оставьте отметку на первом пункте. Нажмите «Далее». Вам предложено будет оформить нижний колонтитул. Если вы хотите, чтобы на каждой странице письма отображался какой-нибудь текст, занесите его в поле нижнего колонтитула, для оформления номера, отметьте соответствующий флажок.

(продолжение задания 3)

После этого будет предложено изменить стандартный шаблон письма. Можете ничего не выбирать, а просто нажать «Готово». По шаблону создастся новое письмо (см. рис. 17).

Выполните задание 4:

Теперь заполните письмо. Придумайте какой-нибудь текст для официального послания. Окончательный вариант письма сохраните в своей рабочей папке.

5 Создание, редактирование и форматирование таблиц.

Для набора табличных данных существует множество специальных программ — электронные таблицы. Конечно, можно набирать таблицы и в текстовом редакторе, но намного удобнее делать это в специальной программе, позволяющей не только набирать и красиво оформлять таблицы, но и производить в них несложные расчеты. Благодаря этому очень удобно создавать различные сметы и отчеты.

Запустите Open Office Calc. Ознакомьтесь с меню программы. Основное рабочее пространство программы покрыто сеткой, в каждую из ячеек которой можно заносить различные данные. Ячейки можно объединять между собой (надо выделить несколько ячеек и выбрать пункт меню «Формат» → «Объединить ячейки»). На рис. 18 изображено окно программы.

Самая важная функция электронных таблиц — создание формул, включающих функцию и диапазон вычисляемых ячеек. Благодаря этому электронные таблицы получили такое широкое применение. Необходимо отметить также возможность прогнозирования значения с помощью анализа "что-если". Это означает, что можно пересмотреть различные варианты данных для достижения результата.

По имеющимся данным можно построить диаграмму.

Диаграммы настраиваются по многим параметрам (тип, размер, шрифт, цвет, оформление и др.). Кроме всего перечисленного в OO Calc имеется множество дополнительных функций, ускоряющих и облегчающих работу с таблицами:

- автоматический подбор высоты строк и ширины столбцов;
- выполнение условного форматирования таблиц;
- наложение защиты на документ;
- поддержка большинства математических, экономических и других функ-

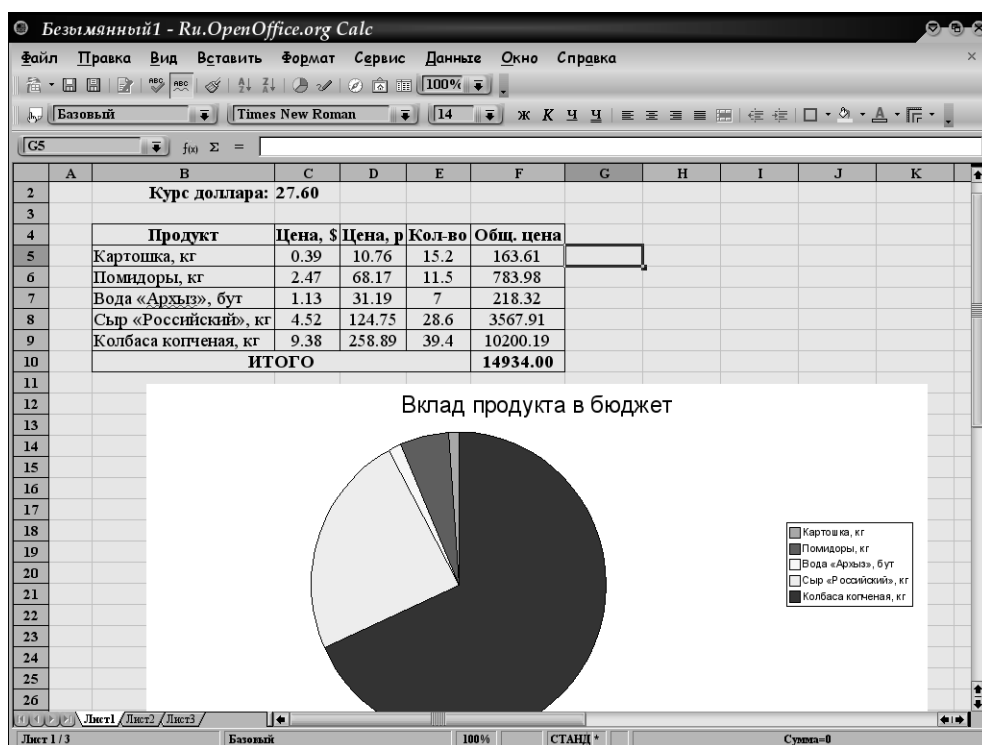


Рис. 18: Окно программы Open Office Calc.

ций;

- возможность сортировки данных таблицы;
- изменение масштаба просмотра документа;
- создание границ вокруг ячеек и таблицы;
- разработка своего стиля и использование его в дальнейшем для быстрого форматирования текста;
- поддержка нескольких словарей для проверки правильности написания;
- работа с макросами;
- удобная справочная система;
- проверка орфографии, грамматики и синтаксиса;
- настройка панелей инструментов (добавление и удаление кнопок и команд);
- возможность предварительного просмотра перед печатью;
- настройка параметров страницы и поддержка нестандартных форматов бумаги.

5.1 Простые таблицы

Выполните задание 1:

Создайте таблицу по образцу:

(продолжение задания 1)

Экзаменационные оценки				
№п/п	Ф.И.О.	Математика	Физика	Химия
1.	Пупкин Вася	2	3	2
2.	Иванов Коля	3	4	4
3.	Сергеев Олег	5	5	4
4.	Никитин Стас	3	3	4
5.	Аверьянов Саша	4	3	3

Отсортируйте таблицу по алфавиту. Оформите ее границы.

- Для создания автоматической нумерации поставьте цифру 1 в 1-й ячейке списка, затем выделите ее одним щелчком левой кнопки мыши и протяните на нужное количество ячеек за правый нижний угол. Курсор мыши при этом примет вид крестика. В появившемся окне «Заполнить ряды» в поле «Тип рядов» следует указать «Линейный» и выбрать приращение равным 1. Аналогично можно заполнять любые ряды с равноотстоящими значениями.

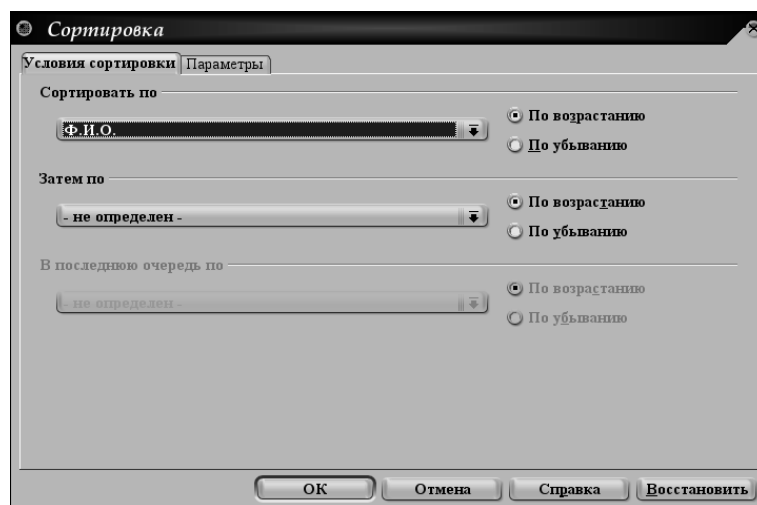


Рис. 19: Сортировка данных в ОО Calc.

- Чтобы отсортировать данные таблицы, выделите ее (вместе с заголовками). Выделять колонку с нумерацией не нужно. В окне параметров сортировки (см. рис. 19) укажите, по каким колонкам следует сортировать данные. Т.к. все ряды у нас связаны, будем сортировать только по Ф.И.О. В элементе «Затем по» должно быть значение «– не определен –». Переключатель направления сортировки — в положение «По возрастанию».

Выполните задание 2:

Создайте таблицу по образцу:

<i>Расход материалов при строительстве</i>				
№п/п	наименование	по плану	фактически	примечание
1.	кирпич (м ³)	3455	3789	брак - 120 шт
2.	цемент (м ³)	1756	1997	
3.	песок (м ³)	3545	4748	
4.	перекрытия ж/б (шт)	6	6	

Отсортируйте содержимое таблицы по алфавиту. Оформите ее границы, сделайте цветную заливку ячеек таблицы.

5.2 Таблицы с формулами

- ▶ В нижней левой части окна программы расположены переключатели рабочих листов. По умолчанию существует 3 листа, однако, можно добавлять, удалять и переименовывать листы по своему усмотрению. ◀

Выполните задание 3:

Создайте на другом листе документа таблицу 3x31.

Занесите в ячейки первой строки данные: в A1 – №п/п, в B1 – X, в C1 – Y. В первый столбец занесите номер ячейки с данными в таблице (как в предыдущих таблицах), во второй столбец занесите значения аргумента X, начиная от -1.5 и заканчивая +1.5 с шагом 0.1.

Для этого введите в первую ячейку -1.5, в ячейке под ней напишите формулу $=Xx + 0.1$, где Xx – координаты предыдущей ячейки. Например, в ячейке B2 должна быть запись $=B1+0.1$.

Затем протяните ячейку с формулой на нужное число клеток вниз, выбрав в окне «Заполнить ряды» тип «формула».

Можно вместо этого воспользоваться удобными возможностями OO Calc: протянуть вниз на 35-40 ячеек содержимое ячейки B1 и в окне «Заполнить ряды» указать тип – линейный, нач. значение – -1.5, кон. значение – 1.5, приращение – 0.1.

В третий столбец вставьте какую-нибудь формулу, например, $=Xx^2+3*Xx^3$, где Xx – координаты соседней слева ячейки (т.е. для C1 это – B1). Протяните эту ячейку до конца таблицы.

- ▶ Вводить цифры удобнее при помощи дополнительных цифровых клавиш, расположенных в левой части клавиатуры, так как десятичный разделитель обозначается либо точкой, либо запятой. При вводе с дополнительной клавиатуры отобразится правильный десятичный разде-

5.3 Диаграммы и графики

литель. ◀

- ▶ Вместо того, чтобы вписывать координаты ячеек или диапазонов, достаточно просто щелкнуть левой кнопкой мыши по данной ячейке (либо выделить левой кнопкой диапазон данных) в процессе ввода формулы. ◀

Выполните задание 4:

Создайте на другом рабочем листе таблицу 2x101. В первый столбец занесите координату X от -5 до $+5$ с шагом 0.1 , аналогично тому, как вы делали в предыдущем задании.

Во второй столбец занесите координату Y — какую-нибудь сложную функцию от X , например, $\exp(\sin Xx)$.

- ▶ Если с введением формулы возникают проблемы (например, вы не можете найти, как данная функция записывается в этой версии электронных таблиц), можно воспользоваться пунктом меню «Вставка» → «Формула», откуда и выбрать требуемую функцию из предлагаемого списка. ◀

5.3 Диаграммы и графики

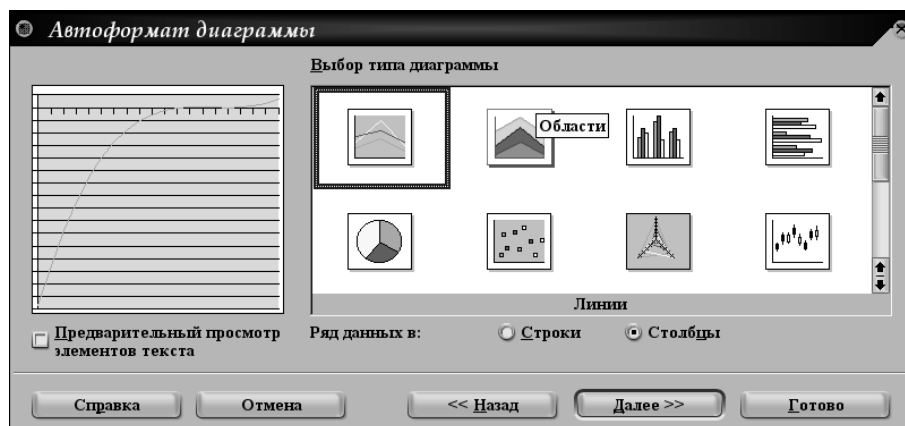


Рис. 20: Выбор типа диаграммы.

Построим по предыдущим двум таблицам график.

Выполните задание 5:

Выделите в таблице из задания 3 второй и третий столбцы и постройте по ней гистограмму и график.

Выполните задание 6:

Аналогичным образом постройте график к таблице из задания 4. Учтите, что из-за больших размеров таблицы график может строиться долго.

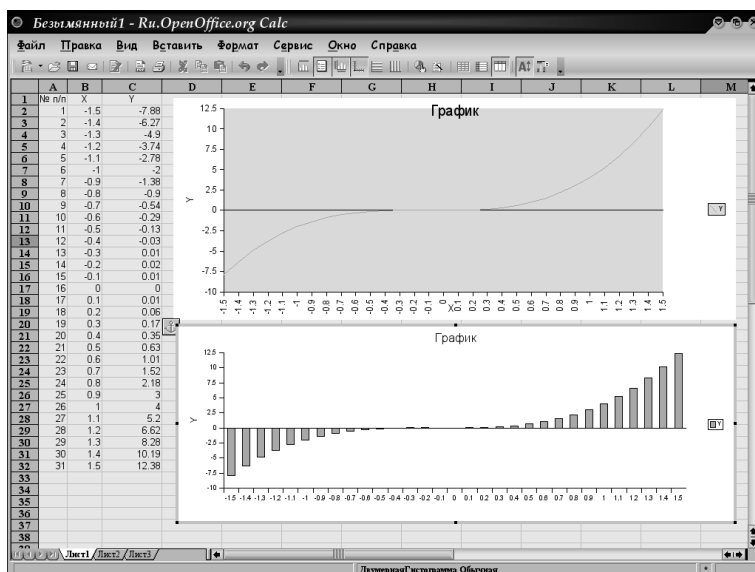


Рис. 21: Выбор типа диаграммы.

Для построения диаграмм и графиков активируйте пункт меню «Вставить» → «Диаграмма». В появившемся окне «Автоформат таблицы» отметьте галочкой «Первая строка как подпись» и «Первый столбец как подпись», чтобы первый столбец понимался как данные оси X, а первая строка таблицы — как подписи осей в легенде. Щелкните «Далее». Выберите тип диаграммы (см. рис. 20). Сначала — «Гистограмма», затем, когда вы будете строить график, выберите «Линии» (см. рис. 20). Переключатель «Ряды данных» должен находиться в положении «Столбцы». Нажмите «Далее». В окне «Выбор варианта» укажите «Обычная». Снимите галочки с пунктов «Линии сетки». Нажмите «Далее». Заполните пункты «Название диаграммы», «Заголовок оси». Нажмите «Готово» и на листе появится диаграмма.

Свойства диаграммы можно редактировать и после построения, выбирая нужные пункты контекстного меню диаграммы. Если дважды щелкнуть левой кнопкой мыши по диаграмме, можно будет редактировать ее объекты: менять их параметры, перемещать их внутри области диаграммы. В итоге ваш рабочий лист будет выглядеть как на рис. 21.

5.4 Ссылки

Ссылки на ячейки необходимы для выполнения сложных вычислений, особенно когда рассчитывается функция от содержимого нескольких ячеек.

Вы уже использовали ссылку на соседнюю ячейку при вычислении формул. Здесь мы ознакомимся с видами ссылок: абсолютными и относительными.

Выполните задание 7:

На следующем листе документа создайте таблицу «Цены на продуктовые товары». В первый столбец занесите значение номера по порядку. Включите в таблицу несколько товаров (второй столбец), в третьем столбце запишите стоимость товаров в «у.е.», а в четвертом — в рублях.

Для автоматического пересчета введите в 1-ю ячейку третьего столбца формулу $=Xx*Yy$. Где Xx — соседняя слева ячейка с ценой в рублях, Yy — ячейка, содержащая курс валюты (поместите ее рядом с таблицей).

Теперь протяните ячейку с формулой вниз. У вас получились нули. Это произошло из-за того, что при протягивании ячеек все незащищенные ссылки на другие ячейки изменяются. Т.е. курс «у.е.» во второй строчке будет братья уже из ячейки под действительным курсом.

Чтобы этого не происходило, необходимо защитить знаком \$ координату, которая должна оставаться неизменной. В данном случае перед координатой строки (т.е. цифрой) необходимо будет поставить знак доллара.

Пример таблицы для данного задания:

<i>Цены на продуктовые товары</i>			
№п/п	Продукт	Цена (\$)	Цена (р.)
1.	Сметана (1кг)	1.22	
2.	Хлеб (1 булка)	0.31	
3.	Молоко (1л)	0.43	
4.	Колбаса копченая (1кг)	4.55	
5.	Вареники (0.5кг)	0.95	

Чтобы значение цены в «у.е.» бралось каждый раз из ячейки слева от ячейки с ценой в рублях, координаты этой ячейки защищать не нужно. Следовательно, если цена в долларах записывается у нас в столбец В и первое значение расположено в ячейке В1, а курс доллара располагается в ячейке D1, то в ячейку С1 надо будет записать: $=B1*D\$1$.

5.5 Расчеты по электронным таблицам

Выполните задание 8:

Постройте статистическую таблицу разрядки конденсатора:

(продолжение задания 8)

t, мс	I, мкА	t, мс	I, мкА	t, мс	I, мкА	t, мс	I, мкА
0	150.5	6	7.3	11	1.0	16	0.1
1	100.4	7	5.0	12	0.5	17	0.1
2	55.7	8	3.8	13	0.3	18	0.1
3	35.6	9	2.6	14	0.2	19	0.0
4	19.4	10	2.3	15	0.2	20	0.0

Зная, что конденсатор разряжается по закону $I = I_0 e^{-\frac{t}{RC}}$ (*), а также, что $R = 22\text{кОм}$, найдите емкость конденсатора.

- Таблицу следует набирать **в два столбца**, здесь она дана в восемь столбцов для экономии места. ◀

Для выполнения задания сделаем кое-какие расчеты:

- Т.к. данные таблицы содержат ошибки, нам придется искать средние значения I_0 и C . Прологарифмируем формулу (*): $\ln I = \ln I_0 - \frac{t}{RC}$.
- Построим справа от нашей таблицы дополнительный столбец: $\ln I$ (в первую ячейку которого введем формулу $=\ln(Xx)$, где Xx — значение тока в 0-й момент времени, и растянем до конца таблицы).

В наших вычислениях не надо использовать последние два значения (при $t = 19$ и 20с), иначе они внесут серьезную ошибку в вычисления.

- Искомое значение $C = \frac{t}{R(\langle \ln I_0 \rangle - \ln I)}$, усреднив все значения C получим близкую к истине величину. ($\langle X \rangle$ — обозначение средней величины). Добавим столбец, в который запишем эту формулу: $= \langle t \rangle / 22000 / (\langle I_0 \rangle - \langle I \rangle)$, здесь $\langle t \rangle$, $\langle I_0 \rangle$ и $\langle I \rangle$ — координаты ячеек, в которых содержатся соответствующие значения времени, логарифма нулевого тока и логарифма тока. Т.к. I_0 нам пока не известно, введите вместо $\langle I_0 \rangle$ координаты любой пустой ячейки. Запишите в эту ячейку $=\ln(150.5)$. Потом мы внесем в нее более точное значение.

Не забывайте, что ссылка на ячейку с адресом $\ln I_0$ не должна изменяться, т.е. это должна быть **абсолютная ссылка**. Например, если Эта величина находится у вас в ячейке D25, то ссылку на нее в формуле следует писать так: $\$D\29 .

Чтобы были видны значимые цифры, следует установить цифровой формат ячеек с шестью знаками после запятой (из контекстного меню «Формат ячейки»).

- Под заполненным только что столбцом посчитайте среднее значение емкости: $=\text{average}(xX:yY)*1000000$. Здесь $(xX:yY)$ — весь диапазон емкостей (написав $\text{sum}()$, выделите столбец сверху), а функция average возвраща-

ет среднее арифметическое значение из выбранного диапазона. Умножать на 1000000 необходимо, чтобы получить значение емкости конденсатора в микрофарадах.

- Рядом создайте столбец, в котором нужно посчитать среднюю ошибку. В первую его ячейку запишите $=(xX*1000000-\$y\$Y)^2$, где Xx — соседняя слева ячейка, $\$y\Y — ячейка, в которой записано среднее значение емкости. В ячейке снизу запишите сумму ошибок.
- Перебирайте значения I_0 от 145 до 155. Остановитесь на том значении, при котором будет минимальной сумма ошибок.

Теперь вы получили значение емкости конденсатора. Это был пример использования электронных таблиц в физике и математике.

Вообще, для физических и математических расчетов существуют более «продвинутые» электронные таблицы (вроде Origin или LabPlot), в которых намного больше специфических функций: построение и анализ графиков по сложным функциям, выполнение сложных расчетов и т.п.

Для сложных вычислений существует такой «монстр», как Matlab — программа, которая удовлетворит практически все вычислительные и визуализационные потребности современного ученого.

6 Системы счисления.

Система счисления (СС) — знаковая система, в которой числа записываются по определенным правилам с помощью символов некоторого алфавита (цифр). СС бывают позиционными и непозиционными.

В **непозиционной СС** значение цифры не зависит от ее положения в числе. Пример — римская СС. В ней если меньшая цифра стоит до большей в числе, она вычитается, а если после — то прибавляется к остальным. М — 1000, D — 500, С — 100, L — 50, X — 10, V — 5, I — 1. Например, $MCMXCIX_{\text{рим}} = 1999_{10}$.

В **позиционной СС** значение цифры зависит от ее положения в числе — разряда. **Основанием** позиционной СС называется количество цифр в ней. Например, арабская десятичная СС, 16-чная и т.д. Если развернуть число в позиционной СС в сумму, получим:

$$A_x = a_{n-1}a_{n-2} \cdots a_0.a_{-1}a_{-2} \cdots a_x = a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \dots + a_0 + a_{-1} \cdot \frac{1}{x} + a_{-2} \cdot \frac{1}{x^2} + \dots \quad (6.1)$$

Здесь a_n — составные цифры числа, n — номер разряда.

6.1 Перевод чисел в 10-ю СС

Этот перевод осуществляется по формуле 6.1.

Примеры:

- Из 2-й:

$$10.11_2 = 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 2.75_{10}$$

- Из 8-й:

$$67.5_8 = 6 \cdot 8^1 + 7 + 5 \cdot 8^{-1} = 55.625_{10}$$

- Из 16-й:

Шестнадцатеричная СС использует следующие цифры: 1, 2, 3, ..., 9, А (10), В (11), С (12), D (13), Е (14), F (15).

$$19F_{16} = 1 \cdot 16^2 + 9 \cdot 16 + 15 = 415_{10}$$

6.2 Перевод чисел из 10-й СС

Для этого используется метод последовательного деления числа на основание новой СС. Последовательность остатков, записанная в обратном порядке, и будет искомым числом.

Примеры:

- В 2-ю: $19_{10} = X_2$

$$\begin{array}{r|l} 19 & 2 \\ \hline 18 & 9 \\ \hline 1 & 8 \\ \hline & 1 \\ & \frac{1}{4} \\ & \frac{2}{4} \\ & \frac{4}{0} \\ & \frac{0}{2} \\ & \frac{2}{0} \\ & \frac{0}{1} \end{array}$$

Следовательно, $19_{10} = 10011_2$.

- В 8-ю: $35_{10} = X_8$

$$\begin{array}{r|l} 35 & 8 \\ \hline 32 & 4 \\ \hline & 3 \end{array}$$

Следовательно, $35_{10} = 43_8$.

- В 16-ю: $127_{10} = X_{16}$

$$\begin{array}{r|l} 127 & 16 \\ 112 & 7 \\ \hline 15 & \Rightarrow F \end{array}$$

Следовательно, $127_{10} = 7F_{16}$.

Для перевода дробей из десятичной СС в другую, умножаем их на основание другой СС, записывая после десятичной точки получающиеся целые числа вплоть до достижения требуемой точности (или достижения нулевого результата). При умножении мы считаем, что целая часть дроби всегда равна нулю.

Пример:

$$0.75_{10} = X_2; \quad 0.75 \cdot 2 = 1.5 \cdot 2 = 1.0 \Rightarrow 0.75_{10} = 0.11_2$$

$$0.12_{10} = X_8; \quad 0.12 \cdot 8 = 0.96 \cdot 8 = 7.68 \cdot 8 = 5.44 \dots \Rightarrow 0.12_{10} \approx 0.075_8$$

Перевод чисел между 2-й, 8-й и 16-й СС.

Так как $8 = 2^3$, а $16 = 2^4$, то 1 разряд 8-й СС содержит 3, а 1 разряд 16-й СС – 4 разряда 2-й СС. Следовательно, перевод между ними облегчается путем разбиения двоичного числа на триады (для 8-й СС) или тетрады (для 16-й СС).

Триады двоичных чисел и соответствующие восьмеричные:

X_8	0	1	2	3	4	5	6	7
X_2	000	001	010	011	100	101	110	111

Тетрады двоичных чисел и соответствующие шестнадцатеричные:

X_{16}	0	1	2	3	4	5	6	7	8	9
X_2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
X_{16}	A	B	C	D	E	F				
X_2	1010	1011	1100	1101	1110	1111				

Например, $72_8 = X_2$, разложим каждую цифру восьмеричного числа по триадам, тогда получим: $72_8 = 111010_2$; $A3_{16} = X_2$, здесь разложим каждую цифру по тетрадам: $A3_{16} = 10100011_2$.

6.3 Арифметические операции

Арифметические операции в других СС осуществляются в столбик, так же, как и в привычной нам десятичной СС. Необходимо только помнить, что суммирование производится в другой СС, так что $5 + 5$ не всегда равно 10, а может быть равно и 12_8 , и A_{16} .

В выполнении этой работы вам поможет калькулятор. Выберите в нем инженерный режим. Теперь все свои вычисления вы сможете проверить при его

помощи. Например, для перевода числа 124 из десятичной системы счисления в двоичную, выберите режим десятичного ввода (decimal), наберите 124, а затем — выберите двоичный (binary) режим. Калькулятор покажет 1111100 — число 124 в двоичной системе.

Выполните задание 1:

Переведите числа в другую систему счисления.

$$125_{10} = X_2$$

$$1111_{10} = X_8$$

$$12321_{10} = X_{16}$$

$11001101_2 = X_{16}$. Подсказка: переводить число из двоичной в 16-ю систему удобно при помощи тетрад.

$$A3B9_{16} = X_2$$

$1100011100101_2 = X_8$. Подсказка: в восьмеричную систему из двоичной удобно переводить числа при помощи триад.

$$2756_8 = X_2$$

$A44_{16} = X_8$. Подсказка: переведите из 16-й в двоичную (при помощи тетрад), а затем полученное число — в 8-ю (при помощи триад).

$$7423_8 = X_{16}$$

$110_2 + 143_{10} = X_8$. Подсказка: переведите число 143_{10} в двоичную систему, в ней выполните сложение, ответ переведите в восьмеричную систему при помощи триад.

$$200_{10} + 10011_2 = X_{16}$$

$A4_{16} - 65_8 = X_2$. Подсказка: здесь удобнее сразу перевести оба слагаемых в двоичную систему.

$$300_{10} + 400_8 = X_{16}$$

$$1.22_{10} = X_2 \text{ (С точностью до 3-х знаков после запятой)}$$

$$6.234_8 = X_{10} \text{ (- // -)}$$

$$1.45_{10}/1.11_2 = X_8 \text{ (- // -)}$$

7 Компьютерные презентации. Разработка презентации.

Для подготовки мультимедийных презентаций можно использовать удобную программу — Open Office Impress, которая позволяет:

- создавать профессиональные слайдовые демонстрации, включающие графики, изображения, текст, анимации и прочие объекты;
- добавлять спецэффекты к элементам слайда: тексту, анимациям и изображениям, а также — настраивать эффекты появления самих слайдов на

экране;

- создавать в презентациях векторные изображения;
- создавать слайды на основе профессионально оформленных шаблонов;
- создавать различные виды презентации, позволяющие удобно ориентироваться среди уже созданных слайдов в больших презентациях;
- сохранять презентации не только в «родном» формате, но и экспортировать презентацию в формат MS PowerPoint, pdf-формат (являющийся мировым стандартом файлов документации) и другие форматы.

7.1 Создание простой презентации

Выполните задание 1:

Запустите OO Impress. Появится окно Мастера презентаций. Отметьте «Пустая презентация» и нажмите кнопку «Готово». Откроется пустая презентация (см. рис. 22).

Заполните ее, как описано ниже.

Основная область программы состоит из трех частей. Слева расположена область предварительного просмотра пиктограмм слайдов, позволяющая выбирать произвольный слайд. В центре — область создания слайда с закладками, позволяющими выбирать различные представления режима отображения презентаций. Справа расположена дополнительная область, из которой можно выбрать фон конкретной страницы, ее оформление, спецэффекты, тип смены слайдов при демонстрации презентации.

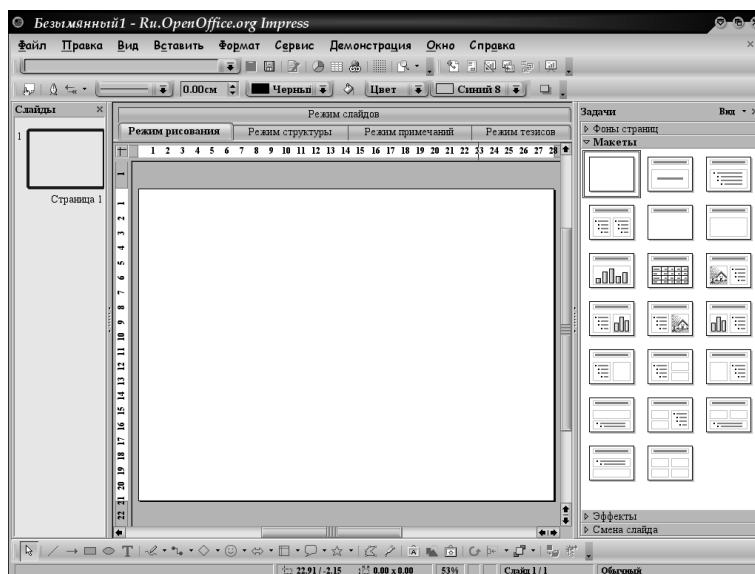


Рис. 22: Презентация Open Office Impress.

Выберите какой-нибудь из фоновых рисунков, выберите оформление слайда

«Только заголовок». Запишите заголовок презентации. Например, «Мой колледж», или «Дикие звери».

Далее необходимо создать следующий слайд. Для этого либо щелкните правой кнопкой мыши в области пиктограмм и выберите из контекстного меню пункт «Новый слайд», либо активируйте пункт меню «Вставить» → «Слайд».

Выберите оформление слайда по своему усмотрению. Добавьте заголовок слайда, текст, изображение. Вставить картинку можно двумя способами. Если вы выбрали оформление слайда, содержащее изображение или объект, дважды щелкните по этому объекту левой кнопкой мыши. Появится диалоговое окно «Вставка OLE объекта», в котором выберите «Создать из файла». Нажмите кнопку «Обзор» и выберите нужный рисунок. Снимите флажок «Связать с файлом», чтобы ваша презентация не зависела от компьютера, на котором ее показывают.

Второй вариант, наиболее подходящий — из меню «Вставить» → «Изображение» → «Из файла». В этом случае окно выбора изображения будет содержать область предварительного просмотра картинки.

- ▶ Изображения можно найти в сети на компьютере SGRPK в сетевой папке /Big/Pict. Здесь много фотографий различных природных объектов, животных и т.п. Кроме того, можно скачать изображения при помощи браузера Mozilla с адреса www.sgrpk → «Разные картинки». ◀

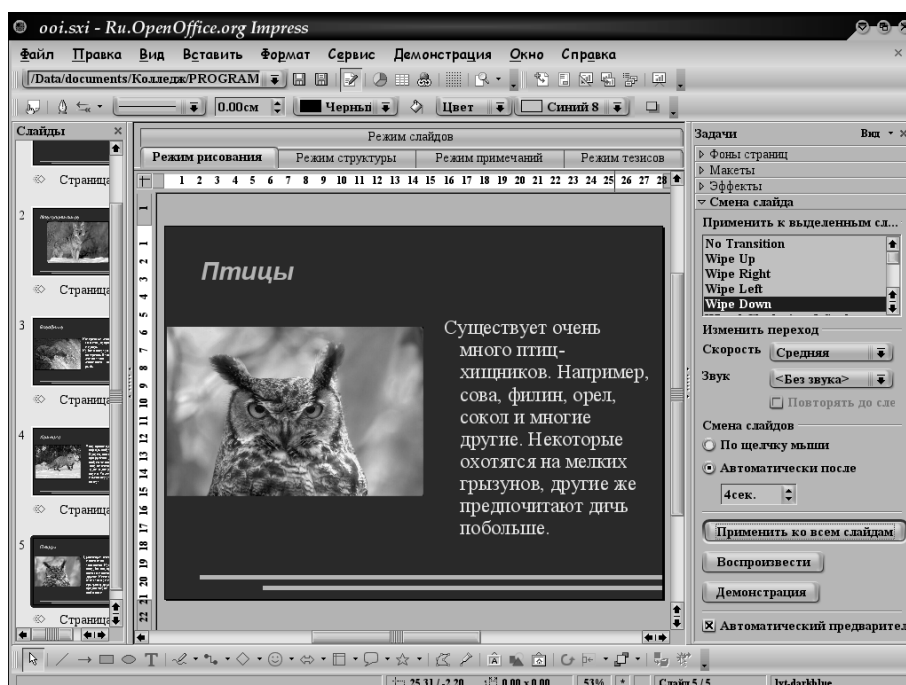


Рис. 23: Настройка смены слайдов.

Создайте не менее пяти слайдов. Соблюдайте основные требования к презентациям: фон обязательно должен быть темным. Наилучший вариант — черный

фон. В этом случае зрение не будет напрягаться. Текст должно быть хорошо видно на фоне. Не стоит делать красный фон с желтыми буквами — подобные цвета очень плохо сочетаются. И вообще, красным и желтым цветами лучше выделять только важные элементы текста, либо заголовки. Основной текст лучше сделать белым, светло-серым или голубым.

7.2 Оформление спецэффектов

Для эффективного показа презентации можно настроить особый тип смены слайдов, а также спецэффекты появления и исчезновения отдельных элементов.

Вызвать настройку смены слайдов можно из меню «Демонстрация» → «Смена слайдов», или же активировав соответствующую закладку в правой области окна.

Выполните задание 2:

Выберите желаемый тип смены слайдов, настройте время, если хотите сменять слайды автоматически. Нажмите «Применить ко всем слайдам», если хотите, чтобы все слайды сменялись одинаково (это — наилучший вариант, не утомляющий зрителя) (см. рис. 23).

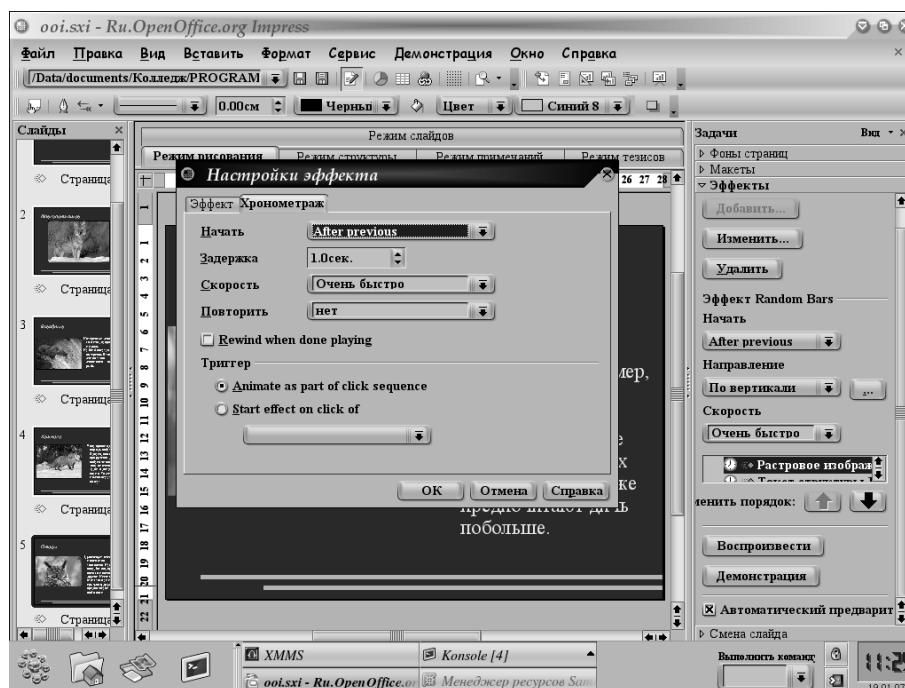


Рис. 24: Настройка спецэффектов.

Для настройки спецэффектов появления или исчезания элементов слайда активируйте вкладку «Эффекты» в правой области окна или меню «Демонстрация» → «Эффекты». Щелкните кнопку «Добавить». Выберите тип спец-

эффекта и скорость. Нажмите «ОК». Чтобы эффекты появлялись автоматически, выберите в пункте настроек «Начать» параметр «After previous» (после предыдущего), укажите желаемые параметры спецэффекта и скорость.

Каждому спецэффекту будет соответствовать одна строчка в списке эффектов. Если щелкнуть правой кнопкой мыши по желаемому эффекту, можно настроить дополнительные параметры: хронометраж, свойства. Настройка хронометража необходима для создания последовательного появления спецэффектов в желаемой последовательности (см. рис. 24).

Если в окне с перечнем спецэффектов щелкнуть по выбранному эффекту правой клавишей мыши, из контекстного меню можно выбрать пункт «Хронометраж», где можно задать временные задержки при появлении спецэффекта (см. рис. 25).

Постарайтесь не перегружать презентацию спецэффектами, а также не размещать на одном слайде большого числа элементов. Элементы должны размещаться рационально, между ними должно быть свободное пространство (около 10% размера соответствующей стороны слайда).

Анимация не должна утомлять зрение (т.е. элементы не должны «скакать» — они должны перемещаться плавно).

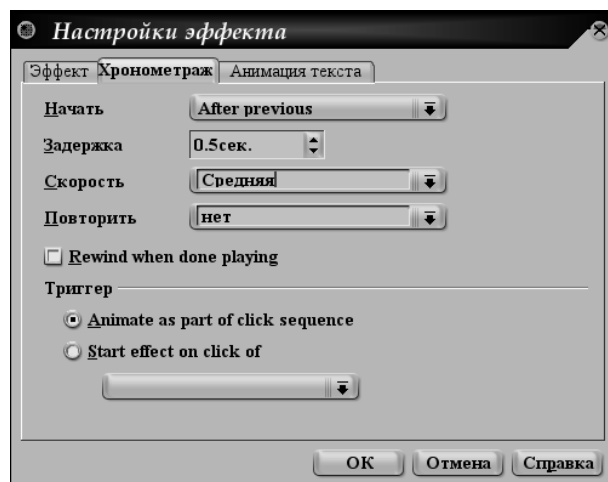


Рис. 25: Выбор временных задержек спецэффектов.

Для создания оглавления презентации можно воспользоваться меню «Вставить» → «Обзорный слайд».

Выполните задание 3:

Добавьте спецэффекты к элементам отдельных слайдов.

Запустите презентацию (для этого надо нажать <F5> или <F9>). Подправьте спецэффекты, которые работают не так, как вам бы хотелось. Сохраните работу в домашней папке.

7.3 Создание автоматического показа презентации

Активируйте пункт меню «Демонстрация». Настройте автоматическую смену слайдов по времени — подпункт «Настройки демонстрации» и вид смены слайдов — подпункт «Смена слайда». Вид смены слайдов можно настроить отдельно для каждого слайда, также как и паузу между слайдами.

Сделайте задержку между появлением слайдов 3..5 секунд. Задержку спецэффектов для элементов слайдов создайте в диапазоне 1..3 секунд, чтобы можно было успеть прочесть текст слайда и разглядеть изображения. Не переусердствуйте со спецэффектами: для каждого слайда их должно быть не более 3..5, но и не пренебрегайте разнообразить спецэффекты. Например, если заголовков слайда вылетает сверху, текст слайда может выплывать снизу, а изображения — плавно появляться.

8 Работа с электронными СУБД Open Office Base

Системы управления базами данных (СУБД) служат для организации и хранения больших массивов данных, поиска в них необходимой информации и генерации отчетов по заданным критериям. Пакет Open Office содержит встроенные средства для работы с базами данных различных типов. Помимо функций обслуживания установленных на компьютере баз данных, в Open Office есть свой собственный сервер баз данных.

8.1 Ознакомление с СУБД. Создание таблицы

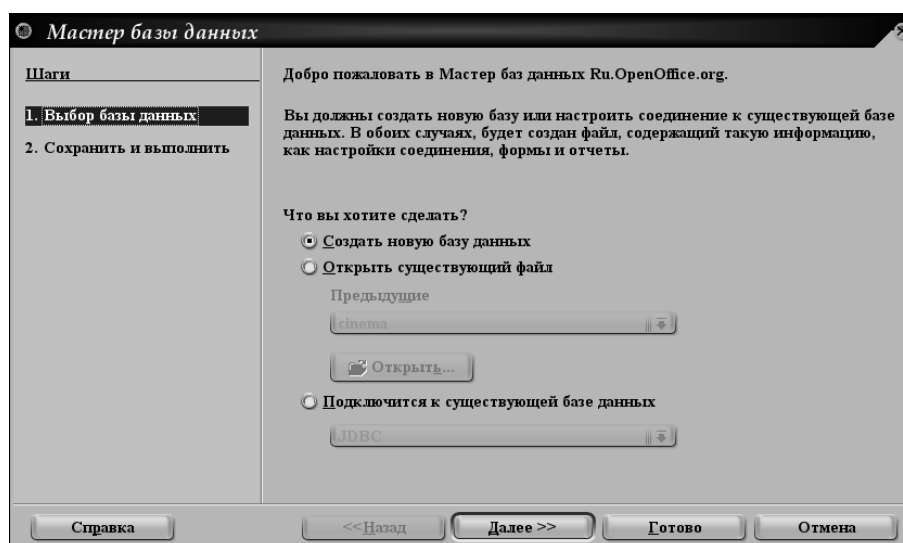


Рис. 26: Мастер баз данных.

Выполните задание 1:

Создайте пустую базу данных, как это описано ниже.

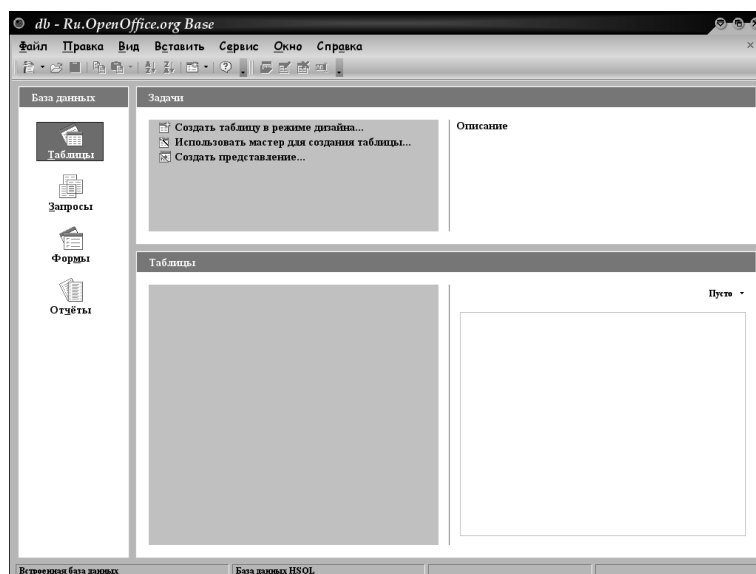


Рис. 27: Окно пустой базы данных.

Запустите Open Office Database (Базы данных), либо откройте окно любого приложения Open Office и из меню «Файл» → «Создать» выберите пункт «База данных». Появится диалоговое окно мастера баз данных (см. рис. 26).

На первом шаге оставьте переключатель «Что вы хотите сделать?» на пункте «Создать новую базу данных». Нажмите «Далее». Переключатель регистрации базы данных переведите на «Нет». из флажков в поле «Что вы хотите сделать после этого» оставьте отмеченным только флажок «Открыть базу на редактирование». Нажмите «Готово». Появится окно сохранения базы данных, где вам необходимо указать путь к базе (ваша домашняя папка) и ее имя. Нажмите кнопку «Сохранить». Работа с мастером окончена.

Теперь у вас открыта пустая база данных (см. рис. 27).

Обратите внимание на кнопки панели инструментов слева. Первой расположена кнопка «Таблицы». Таблицы представляют собой массивы данных, с которыми вы будете работать. Далее расположен пункт «Запросы». Запросы позволяют организовать доступ к данным по различным критериям. «Формы» позволяют как создавать новые данные, так и отображать существующие. Поле «Отчеты» позволяет вам генерировать документы по имеющимся данным.

8.1.1 Создание таблицы в режиме дизайна

Активируйте кнопку «Таблицы». В списке задач активируйте пункт «Создать таблицу в режиме дизайна». Появится окно создания таблицы. Создадим базу данных учета кадров.

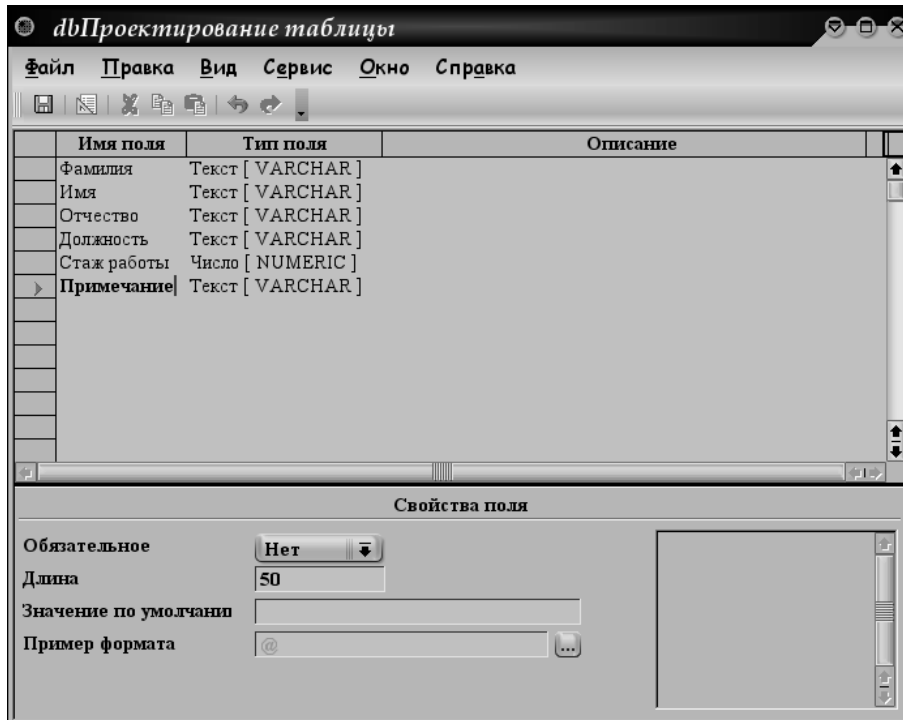


Рис. 28: Проектирование таблицы.

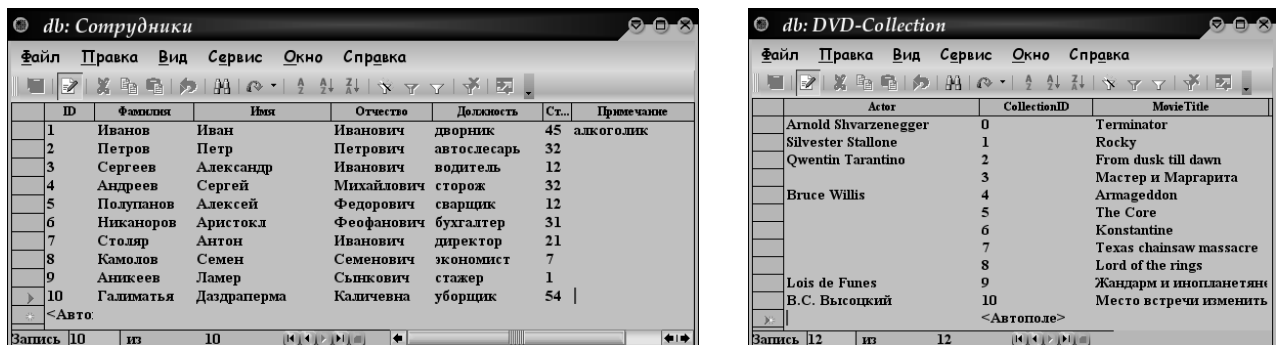


Рис. 29: Заполнение таблиц.

В столбце «Имя поля» занесите поля «Фамилия», «Имя», «Отчество», «Должность», «Стаж работы», «Примечание». Для всех полей укажите тип «Текст», кроме поля «Стаж работы», для которого укажите «Число». В свойствах полей укажите «Обязательное» для всех полей, кроме «Примечания» (см. рис. 28).

Активируйте пункт меню «Файл» → «Сохранить». Назовите базу данных «Сотрудники». Программе понадобится первичный ключ, и она запросит подтверждения для его создания. Ответьте «Да». Таблица готова.

Дважды щелкните по таблице «Сотрудники» в списке таблиц. Она откроется для редактирования.

Заполните вручную 10-15 полей. Чтобы не заполнять первичный ключ вручную, можете щелкнуть на таблице «Сотрудники» правой клавишей мыши и выбрать в поле свойств «Автоназначение». Тогда вам не понадобится изменять

данное поле вручную. После заполнения ваша таблица будет выглядеть как на рис. 29. Сохраните базу данных.

Выполните задание 2:

Создайте таблицу «Сотрудники», заполните ее.

8.1.2 Создание таблиц с помощью «Мастера таблиц»

Теперь мы создадим вторую таблицу при помощи мастера таблиц. Активируйте задачу «Использовать мастер для создания таблицы». Появится диалоговое окно мастера таблиц.

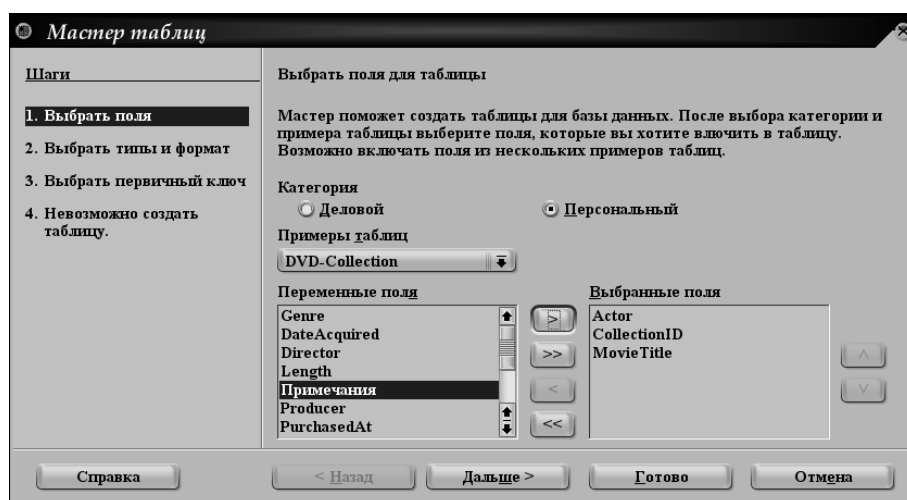


Рис. 30: Мастер создания таблиц.

Создадим таблицу «Коллекция DVD». Переключатель «Категория» установите в положение «Персональный» (см. рис. 30). Из списка «Примеры таблиц» выберите «DVD-Collection», а из переменных полей перекиньте в область выбранных полей поля «Actor» (главная роль), «CollectionID» (идентификатор в коллекции) и «Movie Title» (название фильма). Нажмите «Дальше». На следующем шаге сделайте обязательными поля «CollectionID» и «Movie Title». Для поля «CollectionID» можете указать автоназначение. Нажмите «Дальше». Для первичного ключа укажите поле «CollectionID». Нажмите «Дальше». На последнем шаге нажмите «Готово».

Заполните данную таблицу. Не забывайте сохранять базу данных после каждого изменения.

Выполните задание 3:

Создайте таблицу «Коллекция DVD».

8.2 Работа с таблицами, создание запросов

Для быстрого поиска данных удобно использовать запросы. Создадим запросы, отображающие в алфавитном порядке список сотрудников и список коллекции фильмов.

8.2.1 Отбор и сортировка данных при помощи запросов

Выполните задание 4:

Активируйте задачу «Использовать мастер для создания запроса». Откроется диалоговое окно мастера. На первом шаге необходимо выбрать поля, которые будут отображаться в запросе. Выберите Ф.И.О. сотрудников и должность. Нажмите «Дальше».

На втором шаге необходимо выбрать режимы сортировки данных. В первом поле выберем «Фамилия», затем «Имя» и третьим ключом укажем «Отчество». Везде укажем «По возрастанию». Нажмите «Дальше».

На этом шаге предлагается создать запрос поиска по конкретным данным. Пропустим его. Нажмите «Дальше».

Можете заполнить поля альтернативных названий, а можете сразу нажать «Дальше».

Здесь также можно оставить все как есть, но при желании можете поменять название запроса. Нажмите «Готово». Появится таблица запроса.

Фамилия	Имя	Отчество	Должность
Андреев	Сергей	Михайлович	сторож
Аникеев	Ламер	Сынкович	стажер
Галиматья	Даздраперма	Каличевна	уборщик
Иванов	Иван	Иванович	дворник
Камолов	Семен	Семенович	экономист
Никаноров	Аристокл	Феофанович	бухгалтер
Петров	Петр	Петрович	автослесарь
Полупанов	Алексей	Федорович	сварщик
Сергеев	Александр	Иванович	водитель
Столяр	Антон	Иванович	директор

MovieTitle	Coll..
Armageddon	4
From dusk till dawn	2
Konstantine	6
Lord of the rings	8
Rocky	1
Terminator	0
Texas chainsaw massacre	7
The Core	5
Жандарм и инопланетяне	9
Мастер и Маргарита	3
Место встречи изменить нельзя	10

Рис. 31: Создание запросов.

Выполните задание 5:

Аналогично создайте запрос по коллекции DVD с сортировкой по названию. Таблицы запросов изображены на рис. 31.

Если база данных содержит несколько сотен или тысяч записей, без запросов довольно сложно найти в ней данные, отвечающие определенным критериям.

8.3 Сортировка и поиск данных

Для сортировки и поиска данных удобно использовать инструментальную панель таблиц. На ней содержатся значки поиска информации (в виде бинокля) и ее сортировки (стрелка вниз с надписью $A \rightarrow Z$ означает сортировку по алфавиту, стрелка с надписью $Z \rightarrow A$ — обратную сортировку).

8.3.1 Сортировка данных в таблице

Откройте одну из таблиц. Щелкните по заголовку любого поля. Для сортировки данных нажмите на одну из кнопок сортировки: по алфавиту или в обратном порядке. Вся таблица будет отсортирована соответственно выбору. Кроме того, можно указать сложную сортировку, если нажать кнопку $A \ Z$ без стрелки. Будет предложено выбрать несколько критериев сортировки.

Выполните задание 6:

Отсортируйте данные в таблицах по разным критериям.

8.3.2 Поиск данных

Выполните задание 7:

Нажмите кнопку поиска данных. Появится окно поля записи. Попробуйте, вводя различную информацию из вашей таблицы в поле поиска, найти ячейку, где эта информация хранится.

8.4 Создание форм для ввода данных

Зайдите на закладку «Формы» и выберите пункт задач «Использовать мастер для создания формы». Появится диалоговое окно мастера.

На первом шаге мастера из таблицы «Сотрудники» выберите все поля, кроме «ID». На втором шаге просто нажмите «Дальше». На третьем шаге необходимо задать расположение полей. Наилучшим будет расположение «Столбцы — подписи слева», но можете выбрать и другое расположение. На следующем шаге оставьте все как есть. Далее выберите стиль оформления формы (можно просматривать его на создаваемой таблице за окном мастера). Задайте имя формы, если не хотите оставить имя по умолчанию.

Появится форма, позволяющая редактировать и создавать записи в более привлекательном виде. Обратите внимание на нижнюю инструментальную па-

8.5 Создание и печать отчетов

нель. Здесь содержатся кнопки навигации между записями и создания новых записей. Создайте несколько записей при помощи формы.

Выполните задание 8:

Создайте формы для редактирования записей таблиц (см. рис. 32).

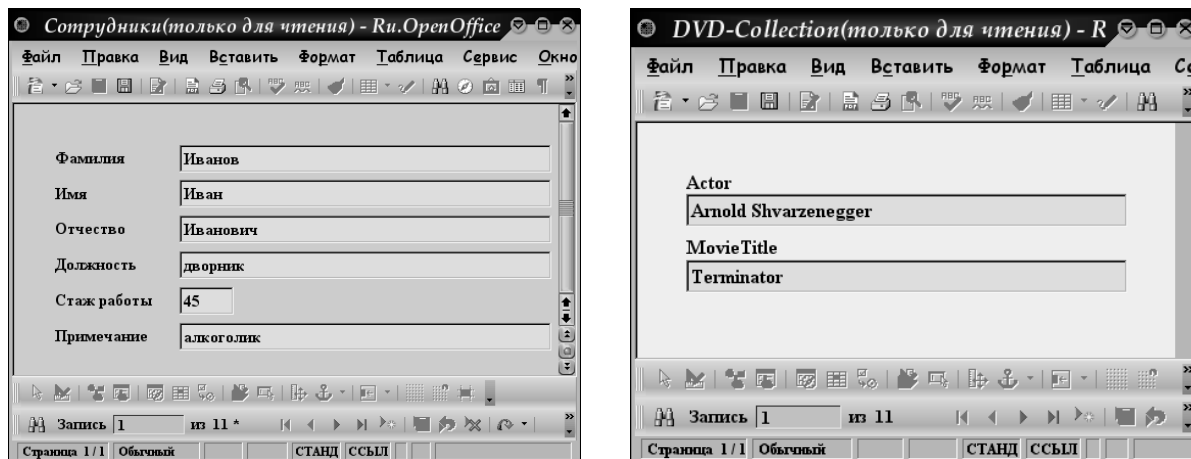


Рис. 32: Формы для редактирования и создания записей.

8.5 Создание и печать отчетов

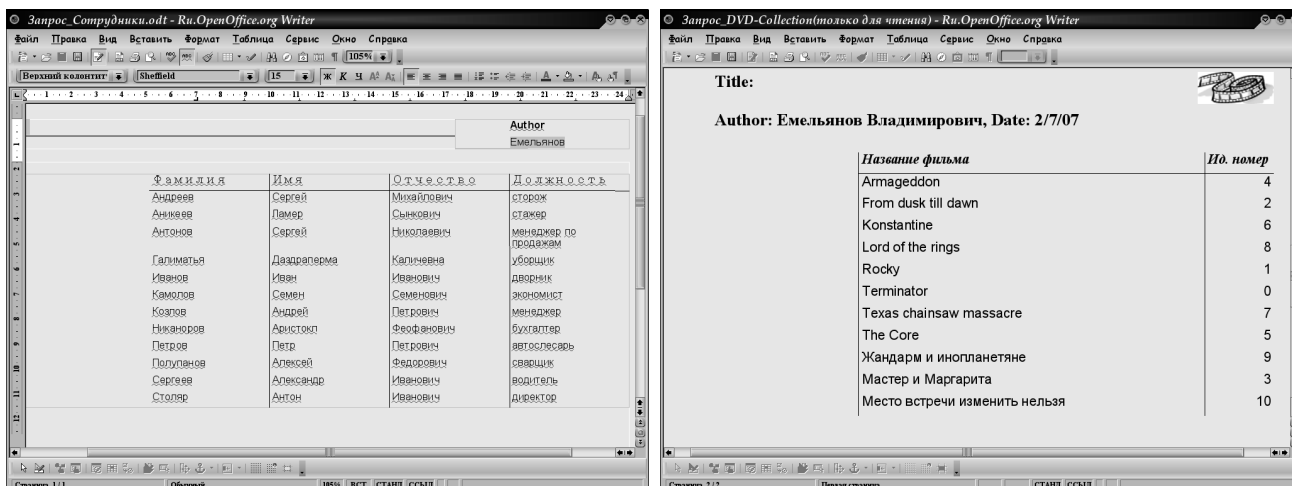


Рис. 33: Генерирование отчета.

Отчет — окончательный этап поиска данных, необходимый для их распечатки или иной работы с ними. Зайдите на вкладку «Отчеты» и активируйте задачу «Использовать мастер для создания отчета». Появится диалоговое окно мастера отчетов.

Для начала создадим отчет, отображающий отсортированный по алфавиту список сотрудников и их должностей (т.е. отчет создадим на базе **запроса** «Сотрудники»). Выберите все поля запроса. На следующем шаге оставьте все как есть. Точно так же поступите на следующем шаге. Следующий шаг — выбор стиля. Выберите понравившийся стиль (просматривать как он будет выглядеть можно в окне предварительного просмотра OO Writer, которое возникло в начале работы мастера. Потом нажмите сразу «Готово». Сгенерируется текстовый документ с требуемым отчетом. Сохраните его под названием «Отчет_сотрудники» в своей рабочей папке.

Выполните задание 9:

Создайте отчеты по обеим таблицам (см. рис. 33).

- ▶ Здесь уместно в разметке колонтитулов выбрать «Сinema», но задать книжную ориентацию листа. ◀

Теперь измените несколько записей в обеих таблицах при помощи форм. Посмотрите, что при этом происходит с запросами.

9 Работа с растровой графикой.

Это задание мы будем выполнять в программе Gimp. Запустите программу. Появится главное окно (см. рис. 34), а также, возможно, некоторые другие окна. Главное окно содержит строку файлового меню; панель инструментов, где вы можете выбрать необходимый инструмент рисования или создания спецэффектов; меню выбора цвета; меню параметров инструмента.



Рис. 34: Главное окно программы Gimp.

Откройте какое-нибудь изображение (меню «Файл» → «Открыть» или сочетание клавиш <Ctrl>+<O>). Изображения можно взять с сетевой папки

9.1 Исправление дефектов фотографии

/Big/Pict. Появится диалоговое окно открытия файла (см. рис. 35). Слева расположено дерево каталогов, где вы можете выбрать нужную папку. В центре располагается список файлов и папок текущего каталога. Справа — область предварительного просмотра изображения.

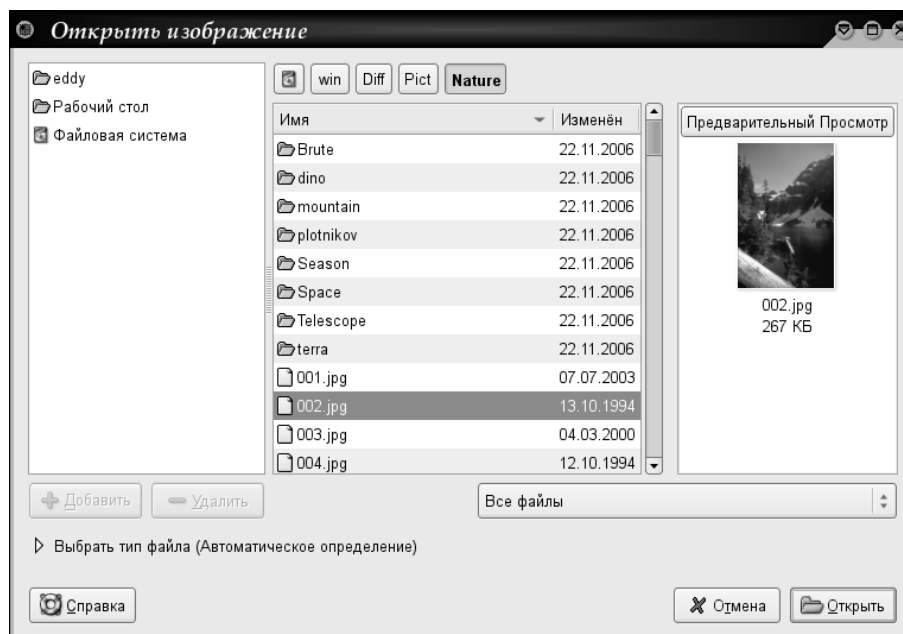


Рис. 35: Диалоговое окно открытия файла.

Когда вы найдете понравившуюся картинку, нажмите «ОК». Откроется окно редактирования с выбранным изображением (см. рис. 36). В верхней части этого окна располагается строка меню. Изучите его внимательно. Многие из этих функций мы будем сегодня использовать.

Изучите панель выбора инструмента. Обратите внимание, что при наведении курсора мыши на кнопку, всплывает подсказка, помогающая понять назначение данного инструмента. Кроме того, во всплывающей подсказке указано, при нажатии какой клавиши на клавиатуре активируется данный инструмент. Использование клавиш быстрого доступа очень удобно, особенно если вы часто занимаетесь редактированием фотографий.

9.1 Исправление дефектов фотографии

Выбранная фотография (см. рис. 36) имеет некоторые дефекты, которые надо будет исправить, если мы хотим распечатать качественную фотографию. Хорошо заметно, что изображенные на заднем плане горы получились размытыми и засвеченными. Кроме того, можно было бы увеличить общую яркость изображения, повысить контраст, улучшить некоторые детали.



Рис. 36: Образец картинка.

Сделаем горы на заднем плане более четкими. Для этого нам необходимо выделить участок изображения. Для выделения областей существует множество инструментов. Нам необходимо выделить область сложной формы. Увеличим участок, где будем выделять нужную область (для изменения масштаба используются клавиши $\langle + \rangle$ и $\langle - \rangle$, либо инструмент «Изменение масштаба»). Менять масштаб можно и при помощи колеса мыши, удерживая нажатой клавишу $\langle \text{Shift} \rangle$. Теперь активируем инструмент «Умные ножницы» (клавиша $\langle I \rangle$). Будем проставлять точки на границе контура, который необходимо выделить. Чем ближе точки друг к другу, тем точнее будет выделение. Построенные контуры можно редактировать: если навести на узел контура мышью, значок около курсора сменится на крест со стрелками. Теперь если нажать левую кнопку мыши, можно перемещать узел. Кроме того, если навестись на построенный контур и нажать левую кнопку мыши, на данный участок можно добавить узловые точки. Обведите контур и щелкните левой кнопкой мыши на самую первую точку контура. Контур замкнется. Чтобы преобразовать его в выделение, необходимо щелкнуть левой кнопкой мыши внутри контура.

- ▶ В процессе выделения меняйте масштаб изображения, чтобы правильно обвести мелкие детали (нажмите $\langle \text{Shift} \rangle$ и, удерживая его, крутите колесико мыши. Перемещайте отображаемый участок изображения при помощи нижнего и бокового ползунков. ◀

Для того, чтобы граница выделения не бросалась в глаза после коррекции фотографии, сделаем размытие. Щелкните правой кнопкой мыши на рисунке

9.1 Исправление дефектов фотографии

и из контекстного меню выберите «Выделение» → «Растушевка». Выберите радиус размытия от 3 до 10 пикселей в зависимости от размера выделения (можно оставить 5 пикселей, которые Gimp предлагает по умолчанию).

Теперь скопируем выделенное и вставим его в виде нового слоя. Для копирования выделения нажмите сочетание клавиш <Ctrl>+<c>. Теперь вставим выделение при помощи сочетания клавиш <Ctrl>+<v>. Выше выделение появится. В окне «Слои» выделение будет обозначено как «Плавающее выделение». Щелкните по этой надписи правой кнопкой мыши и из контекстного меню выберите «Новый слой». Выделение появится в новом слое. Если выделение не совпадает с предыдущим положением, вызовите инструмент «Перемещение слоев и выделенных областей» (или нажмите клавишу <m>). Перемещайте выделение вплоть до совпадения его границ с исходным изображением.

Осветлим вырезанный участок. Активируйте пункт меню «Инструменты» → «Инструменты цвета» → «Яркость - контрастность» окна редактирования изображения. Появится окно «Настройка яркости и контрастности». Обратите внимание, чтобы был отмечен флажок «Предварительный просмотр» (см. рис. 38).

- ▶ Перемещение мышью иногда может быть довольно грубым, для более точной подводки положения изображения используйте клавиши управления курсором. ◀

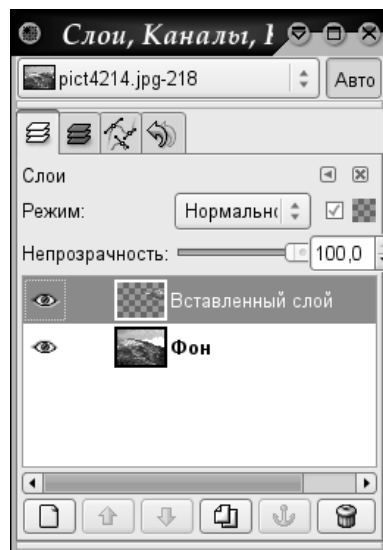


Рис. 37: Окно «Слои, каналы, контуры».

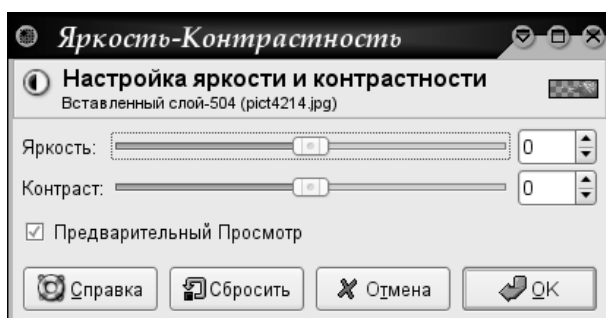


Рис. 38: Окно настройки яркости и контраста.

Изменяйте яркость и контраст выделенного участка вплоть до достижения желаемого значения. Сохраните изображение в своей рабочей папке. При изменении яркости и контраста может измениться окраска. Для ее коррекции из подменю «Инструменты цвета» пункт «Цветовой баланс». Настройте желаемый цветовой оттенок.

Аналогично можно добавить оттенок озеру. Для его выделения можно использовать инструмент «Волшебная палочка», который выделяет участки изображения, имеющие похожие цвета. Чтобы сгладить выделение сначала увеличьте его на $10 \div 15$ пикселей, а затем уменьшите. Сделайте у выделения резкие края (по аналогии с тем, как мы размывали выделение).

В итоге у вас получится скорректированное изображение. Можете, используя другие инструменты, дополнительно скорректировать изображение. Сохраните результат в свою папку. У вас может получиться картинка, как на рис. 39.



Рис. 39: Итоговое изображение.

Выполните задание 1:

Откройте какую-нибудь фотографию с явными дефектами. Исправьте эти дефекты, акцентируйте внимание на главных деталях фотографии, немного размойте второстепенные.

9.2 Редактирование фотографий, фотомонтаж

Загрузите в редактор какую-нибудь фотографию.

Вы можете выполнить с изображением следующие манипуляции:

- улучшите фокусировку в центре изображения и создайте дефокусировку на краях;
- ▶ Для этого лучше всего выделить овальную область в центре изображения, затем улучшить в ней фокусировку, инвертировать выделение и размыть окружающую область. Для того, чтобы между областями не было резких границ, можно, сняв выделение, немного размыть эту область инструмен-

том «размытие». Кроме того, выделенную таким образом область можно скопировать и вставить новым слоем поверх основного рисунка, это значительно улучшит переходы между редактируемой областью и остальным изображением. ◀

- замените цвет некоторых цветных участков изображения;
- ▶ Откройте диалог «Каналы и слои». В нем можно задавать отображаемые слои, редактировать их, менять их порядок. Менять один цвет на другой можно при помощи инструмента «Заливка», если выбрать в его свойствах «Заполнять похожие области». ▶
- создайте вокруг фотографии красивую рамку;
- ▶ Для создания рамки выделите все изображение. Затем выберите в диалоге «Новое выделение» пункт «Вычесть из выделения» и выделите прямоугольную область так, чтобы вокруг нее осталась узкая рамка. Залейте эту рамку полупрозрачной заливкой, либо проведите с ней какую-нибудь трансформацию. ▶
- разрежьте изображение посередине, удалите одну половину, а копию второй отразите зеркально и вставьте на место первой;
- создайте некоторую размытость второстепенных участков изображения, сделайте более резкими основные участки;
- попробуйте сделать фотомонтаж из пары фотографий. Для этого необходимо открыть сразу обе фотографии. Одна из них будет фоном — ее следует сохранить в своей рабочей папке под каким-нибудь другим именем. Выделите на второй фотографии элемент, который планируете вставить на первую, нажмите <Ctrl> + <C>, а затем, щелкнув мышью на первой фотографии — <Ctrl> + <V>. Переместите этот элемент в нужное место, масштабируйте его, отредактируйте для наилучшего сочетания с подложкой.

Так как меню русифицировано, особой трудности в редактировании изображений у вас не возникнет. Стоит лишь обратить внимание, что для изменения какой-то одной области изображения без затрагивания другой области, стоит выделить изменяемую область. Тогда все изменения будут проходить лишь в ней, не выходя за рамки выделения.

Сохраните полученное изображение в своей домашней папке.

Выполните задание 2:

Откройте две-три фотографии. Прочелайте с ними некоторые из указанных манипуляций. Создайте пустой рисунок с размерами фотографии. Используйте его как холст для фотомонтажа, смонтировав вместе какие-нибудь детали фотографий. Например, взяв за основу фото городского пейзажа, добавьте в его верхнюю часть полупрозрачное изображение сказочного замка.

9.3 Создание анимаций

Выполните задание 3:

Создайте анимацию по описанному ниже алгоритму. Сохраните ее в своей рабочей папке.

Создайте новое изображение («Файл» → «Создать») размером 800x600 пикселей (можно выбрать из пункта «Образцы» появившегося диалогового окна) (см. рис. 40). Щелкните по надписи «Расширенные параметры» и выберите из выпадающего меню «Фон»

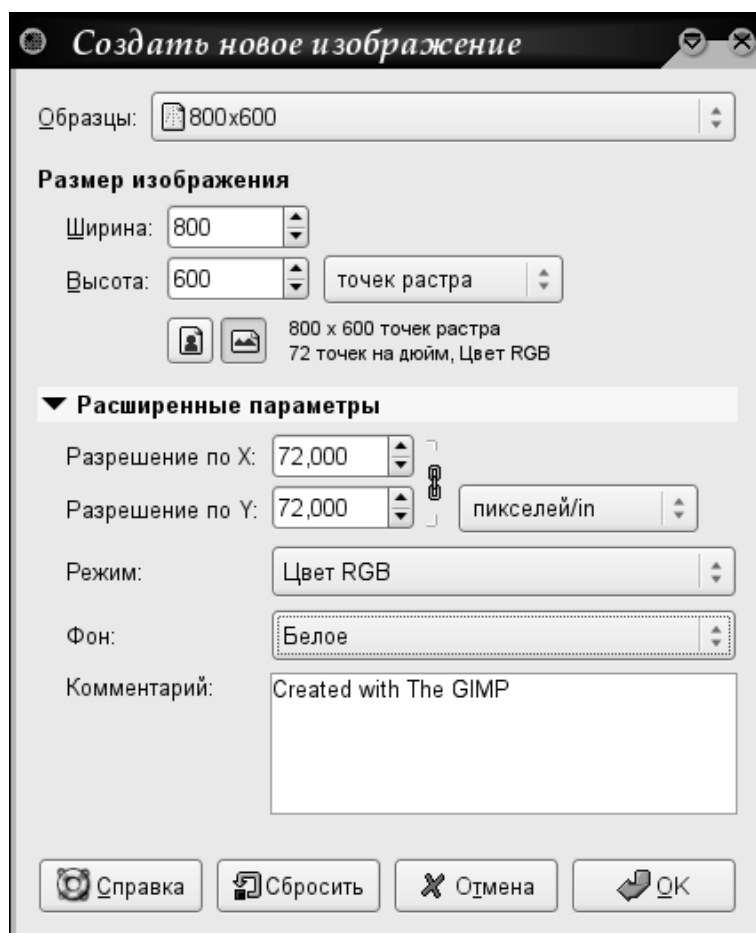


Рис. 40: Создание нового изображения.

По умолчанию изображение содержит один слой. Для создания анимации нам необходимо добавить несколько слоев. Щелкните кнопку «Новый слой» (в левом нижнем углу окна «Слои, каналы, контуры», рис. 41 слева). Откроется окно «Новый слой». Выберите тип заливки «Белое» и нажмите «ОК». Появится второй слой с белым фоном. Теперь, удерживая <Shift>, щелкайте левой клавишей мыши по кнопке «Новый слой». Будут появляться новые слои с предыдущими параметрами. Создайте от семи до десяти слоев.

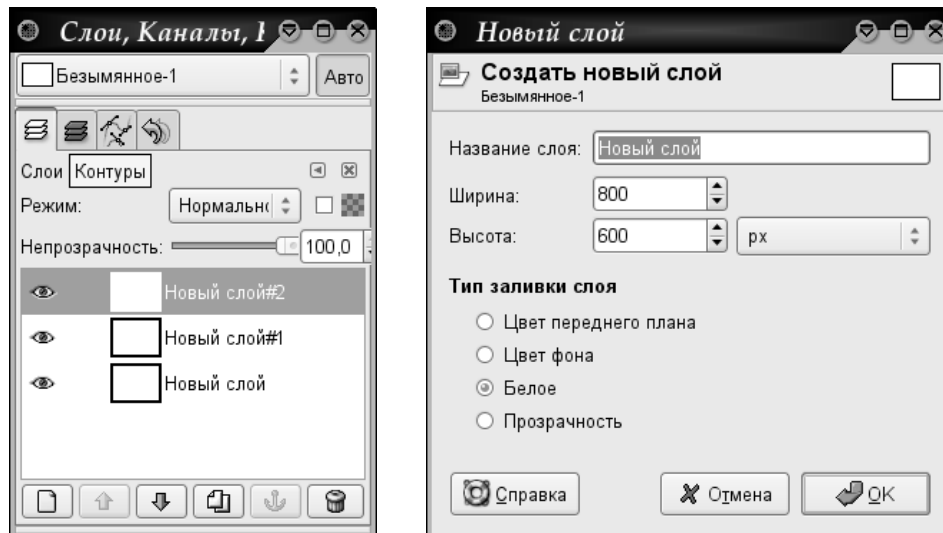


Рис. 41: Диалоговое окно «Слои» (слева). Окно «Новый слой» (справа).

Теперь щелкните по самому нижнему слою в списке слоев и нарисуйте основу первого кадра. Можно вставить изображение.

- Для того, чтобы сделать фоном рисунок, откройте его в другом окне Gimp, нажмите $\langle \text{Ctrl} \rangle + \langle \text{A} \rangle$, $\langle \text{Ctrl} \rangle + \langle \text{C} \rangle$. Перейдите в свое окно и нажмите $\langle \text{Ctrl} \rangle + \langle \text{V} \rangle$. В самом верху списка слоев появится надпись «Плавающее выделение»). Щелкните по ней правой клавишей мыши и выберите «Прикрепить слой». Переместите слой в нужную позицию. ◀

Для того, чтобы видеть, что вы рисуете на первом слое, уберите отметку в виде глаза со всех остальных слоев в окне «Слой...».

Теперь создайте над вашей основой новый слой с прозрачной заливкой (это необходимо, чтобы было видно фон). Нарисуйте здесь первый элемент. Например, пчелу. Создайте следующий прозрачный слой. Те элементы предыдущего слоя, которые не должно быть видно, необходимо замаскировать фоном. Подумайте, как это сделать.

Дорисуйте детали, которые должны появиться в третьем кадре. Создавайте новые слои с прозрачным фоном и дорабатывайте свой рисунок, пока не возникнет необходимость сменить фон. Тогда перейдите на созданный новый слой с белой заливкой. Нарисуйте на нем новый фон.

Повторяйте предыдущие операции, пока не получите 30-40 кадров. Теперь необходимо сохранить анимацию. Для этого воспользуйтесь меню «Файл» → «Сохранить как». В появившемся диалоговом окне наберите имя файла с расширением gif, например, `anim.gif`. Нажмите «ОК». Появится диалоговое окно сохранения в формат gif. Чтобы файл сохранился не как одна картинка, а как анимация, укажите «Сохранить как анимацию». Нажмите «Экспорт». В следующем окне ничего не изменяйте, кроме пункта «Расположение кадра», где

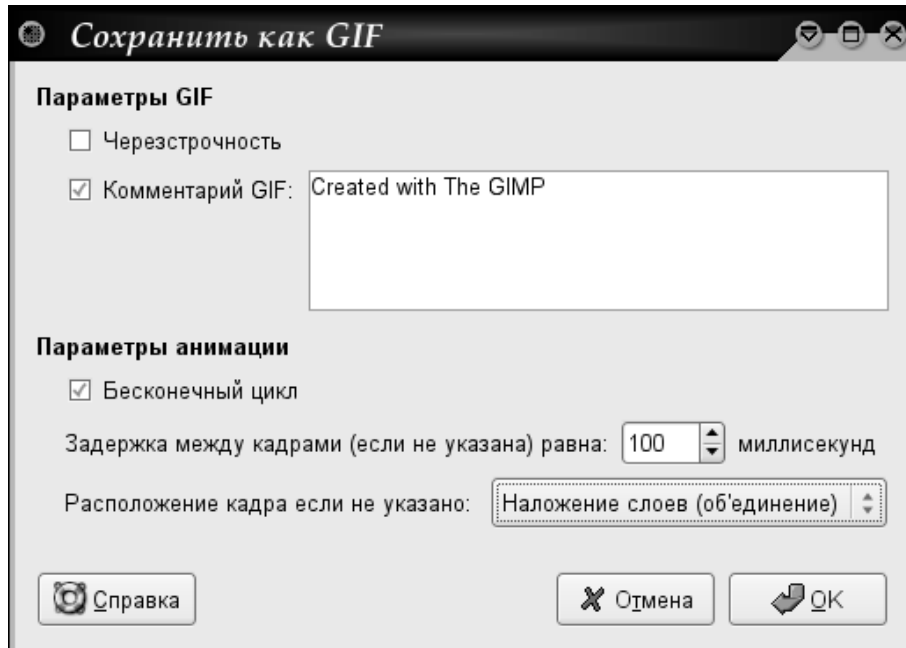


Рис. 42: Сохранение анимации.

выберите «Наложение слоев (объединение)» (см. рис. 42). Теперь можно нажимать «ОК». Gimp сохранит вашу анимацию. Откройте сохраненный файл любой программой для просмотра изображений (или браузером) и посмотрите, как выглядит ваша анимация.

10 Работа с векторной графикой.

Запустите векторный редактор Open Office Draw. Кроме того, создать векторный рисунок можно из меню Open Office «Создать» → «Рисунок». Окно примет вид, подобный изображенному на рис. 43.

По умолчанию у вас открываются стандартная панель инструментов и панель «Рисование». Из меню «Вид» → «Панели инструментов» можно выбрать дополнительные панели, например, «Стрелки» или «3D-Объекты».

Ответьте на вопрос(ы):

1. Чем векторная графика отличается от растровой?
2. Рационально ли преобразовывать цифровые фотографии в векторный формат?
3. Почему технические чертежи выполняют в векторном формате?

10.1 Двумерные векторные фигуры

Для выравнивания объектов можно воспользоваться специальными параметрами: направляющими и сеткой. Доступ к ним возможен из контекстного меню

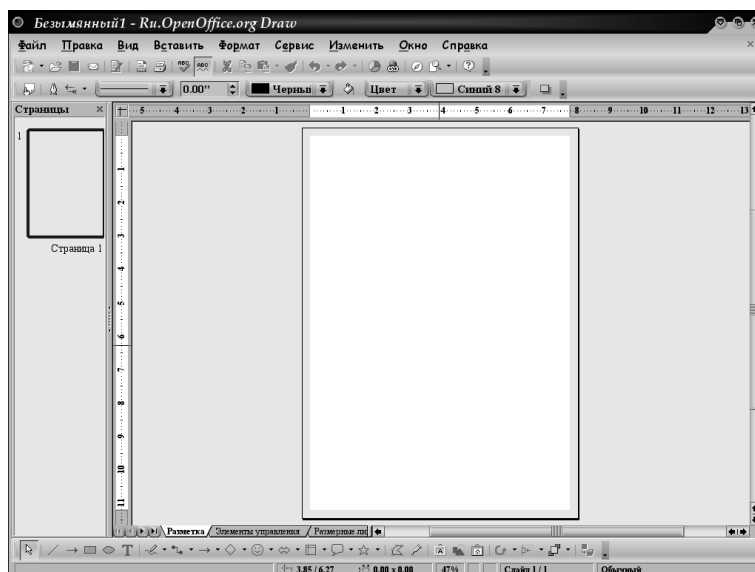


Рис. 43: Окно программы Open Office Draw.

листа, возникающем при нажатии правой клавиши мыши. Если вы захотите нарисовать диаграмму, элементы которой должны быть выровнены, следует создать нужное количество направляющих и активировать привязку к направляющим и к сетке. В этом случае, если вы придвинете фигуру близко к направляющей или узлу сетки, она автоматически встанет на свое место.

На панели меню внизу окна располагаются кнопки рисования: линий, прямоугольников и эллипсов, соединительных линий, наиболее часто использующихся фигур, а также различных эффектов. Создадим небольшую блок-схему, а также нарисуем несколько трехмерных фигур.

Для создания блок-схем используйте кнопку «Схемы» на панели рисования. Справа от кнопки расположен черный треугольник, если щелкнуть по нему левой клавишей мыши, откроется меню выбора элементов блок-схем. Нарисуем блок-схему решения квадратного уравнения.

Вначале нам необходимо ввести данные, выберите элемент «Данные» («Flowchart: Data»). Поместите ее в верхней части листа и выровняйте по центру (выравнивание доступно из контекстного меню фигуры или из панели рисования). Дважды щелкните левой клавишей мыши по фигуре и введите текст: a, b, c — это параметры уравнения. Теперь нам необходимо построить блок вычисления дискриминанта. Расположите его ниже («Процесс» или «Flowchart: Process»). Если ваша фигура окажется залита цветной заливкой, на панели инструментов в верхней части окна выберите в выпадающем меню «Стиль/Заливка области» пункт «Невидимая». Все элементы выравнивайте по центру.

В построенную фигуру занесите формулу вычисления дискриминанта. Подумайте, как правильно ее записать.

Ниже постройте элемент «Выбор» («Flowchart: Decision»). В него занесем

проверку условия неотрицательности дискриминанта ($D < 0$). От него будут отходить стрелки к двум блокам: один будет сообщать об отсутствии действительных корней, второй — вычислять корни и выводить их на экран. Создайте необходимые элементы и внесите в них нужные данные.

Для соединения всех блоков используйте элемент «Соединительная линия»: выберите соединительную стрелку, затем щелкните левой клавишей мыши в месте, откуда вы ее хотите начать, не отпуская левую клавишу, протяните курсор до конечной точки. Автоматически построится соединительная линия. Если вам не понравится, как она будет проходить, можно вручную задать ей необходимое размещение, потянув мышью за опорные точки, выделенные красным цветом.

В итоге у вас получится блок-схема, похожая на изображенную на рис. 44.

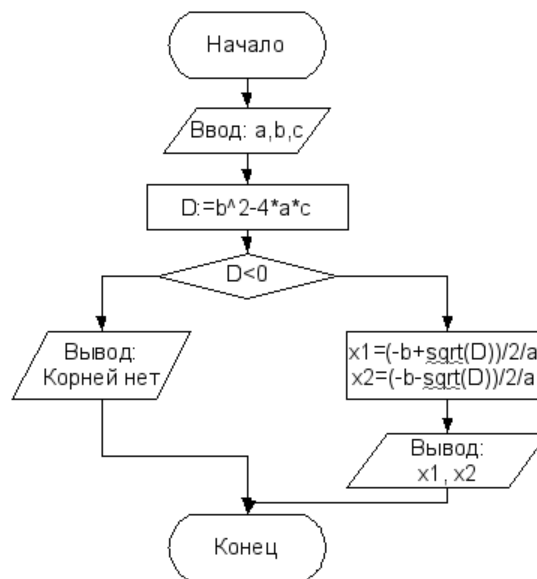


Рис. 44: Блок-схема программы для решения квадратных уравнений.

Выполните задание 1:

Нарисуйте блок-схему, изображенную на рис. 44. Модифицируйте эту схему так, чтобы в случае равенства дискриминанта нулю на экран выводился только один корень.

10.2 Трехмерные фигуры

Теперь вставим в наш документ еще один рисунок («Вставить» → «Слайд»), на котором изобразим трехмерный векторный рисунок. Активируйте панели инструментов «3D-объекты» и «Настройки 3D» (из меню «Вид» → «Панели инструментов»). При помощи этих панелей мы сможем рисовать простейшие трехмерные фигуры и настраивать их параметры.

Выполните задание 2:

Постройте рисунок, содержащий различные трехмерные объекты. Поэкспериментируйте с настройками параметров. Помните, что расположение источника освещения для получения красивого и правдоподобного изображения для всех элементов должно быть одинаковым. Используйте свою фантазию по максимуму.

В результате может получиться что-то вроде изображенного на рис. 45.

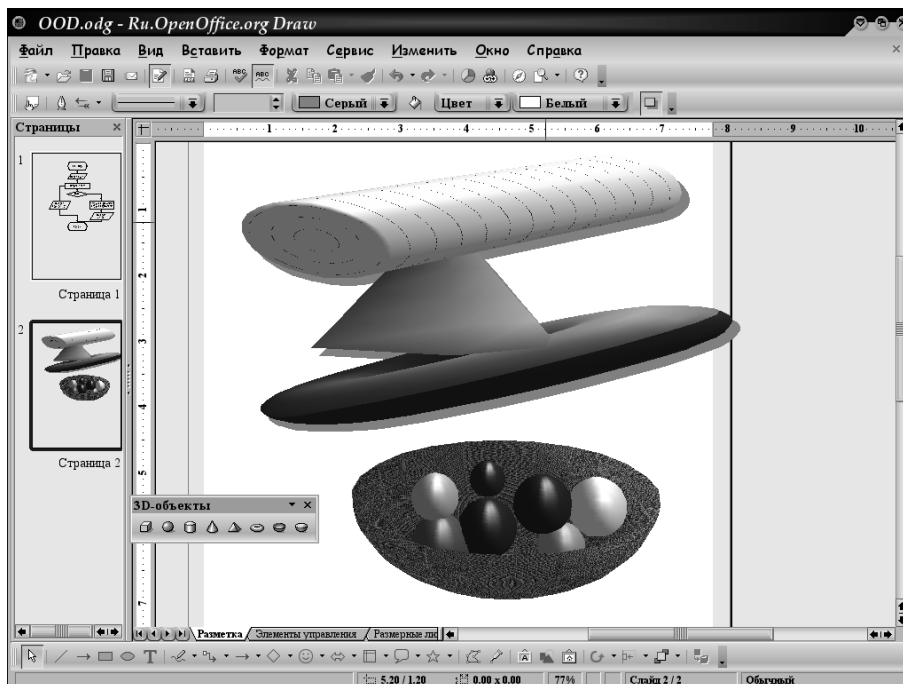


Рис. 45: Векторный рисунок в OO Draw.

Группировка объектов применяется для того, чтобы они вели себя как единое целое. Это удобно, когда вы нарисовали сложный векторный рисунок или элемент рисунка и не хотите, чтобы он в дальнейшем менялся. Для группирования объектов выделите те из них, которые вы хотите сгруппировать (если между ними находятся объекты, которые не должны входить в группу, выделите один из объектов, затем нажмите <Ctrl> и, не отпуская этой клавиши, щелкните левой клавишей мыши по остальным объектам). Выделив нужные объекты щелкните на любом из них правой клавишей мыши и из появившегося контекстного меню выберите «Группировать».

Сгруппированные объекты ведут себя как одно целое. Так вы можете нарисовать сложный элемент изображения, сгруппировать его части и, копируя их, создать составное изображение.

Выполните задание 3:

Добавьте в свой документ еще один лист и нарисуйте на нем трехмерную модель шарикового подшипника. Поэкспериментируйте с расположением и группировкой объектов (из контекстного меню, которое можно вызвать нажатием правой клавиши мыши при наведении ее на объект).

Выполните задание 4:

Создайте еще один лист и начертите сложный объект, состоящий из нескольких повторяющихся группированных объектов по 5-7 примитивов.

Обратите внимание на то, что чертить замкнутые или граничащие друг с другом фигуры проще, установив привязки курсора при построении новой детали к краю уже нарисованной детали, сетке, середине детали и т.п.

Сохраните изображение в домашней папке.

Ответьте на вопрос(ы):

1. В чем основное различие векторной и растровой графики?
2. Какое изображение лучше напечатается на принтере: растровое или векторное?
3. Что такое векторный примитив? Назовите 3-4 примитива.
4. Чему равен (примерно) размер файла растрового изображения в формате bmp, если изображение имеет размеры 250x300 пикселей с трехбайтной глубиной цвета?
5. Почему точные схемы и чертежи подготавливают только в векторном формате?

10.3 Другие векторные редакторы

В Linux существуют такие удобные векторные графические редакторы, как Xfig и Dia. Если вы не можете найти их в меню, нажмите сочетание клавиш <Ctrl> + <F2>, в появившемся окне наберите dia (xfig) и нажмите <Enter>.

Выполните задание 5:

Нарисуйте в другом векторном редакторе простую схему или чертеж, например, план расположения мебели в компьютерном классе. Сохраните изображение в домашней папке.

11 Сетевые протоколы http, ftp.

11.1 Протокол http

Если вы хотите обмениваться файлами с другим человеком, удаленным за много тысяч километров от вас, придется использовать в качестве посредника очень много компьютеров (серверов и шлюзов).

Для обмена файлами и информацией в глобальной сети интернет существует значительное количество **сетевых протоколов**, один из них (и самый распространенный) — http. Это — протокол обмена гипертекстовыми документами, которые могут включать в себя текст, изображения, музыку, видео и различные другие мультимедийные объекты. Просмотр гипертекстовых документов осуществляется при помощи **браузера** — специальной программы. Большинство браузеров может работать с большим количеством протоколов, но пока мы изучаем протокол http, и будем работать с ним.

Для просмотра гипертекстовых документов в Linux существует множество браузеров, как консольных, так и графических. Они работают обычно не только с протоколом http, но и с другими протоколами, позволяя, например, скачивать файлы по протоколу ftp, заходить на защищенные сайты по протоколу https, читать новости и т.п.

В интернете существует множество разнообразных ресурсов, содержащих огромное количество документов. Кроме того, интернет предлагает средства текстового общения (форумы, блоги, ICQ, IRC и многое другое), рассылки писем, новостей, телефонии, видеотелефонии.

11.1.1 Консольные браузеры: lynx и links.

Откройте окно виртуальной консоли (например, `konsole` или `xterm`). Запустите в нем сначала браузер Lynx командой `lynx http://www.sgpk`. Полистайте страницы. Обратите внимание, как браузер реагирует на нажатие тех или иных клавиш. Активация гиперссылки происходит при нажатии клавиши `Enter`, перемещение между ссылками — клавишами «вверх» и «вниз».

Выполните задание 1:

Откройте страницу с фреймами (например, ссылку «разные картинки») при помощи браузера lynx. Обратите внимание, что фреймы можно просматривать только по отдельности.

- «Фрейм» — это часть html-документа, ведущая себя как отдельная страница. Из нескольких фреймов можно собрать сложную web-страницу, позволяющую, например, просматривать в одном из фреймов гиперссыл-

ки, открываемые во втором фрейме. Так, например, устроен просмотр файлов и изображений на сайте www.sgpk. ◀

Выполните задание 2:

Теперь откройте тот же сайт при помощи браузера Links ([links http://www.sgpk](http://links)). Посмотрите, как теперь выглядят страницы с фреймами (см. рис. 46). Скажите, в чем недостаток текстовых браузеров, а в чем их преимущества перед графическими.



Рис. 46: Браузер Links.

11.1.2 Графические браузеры.

Откройте браузер Mozilla. Ознакомьтесь с его меню (см. рис. 47).

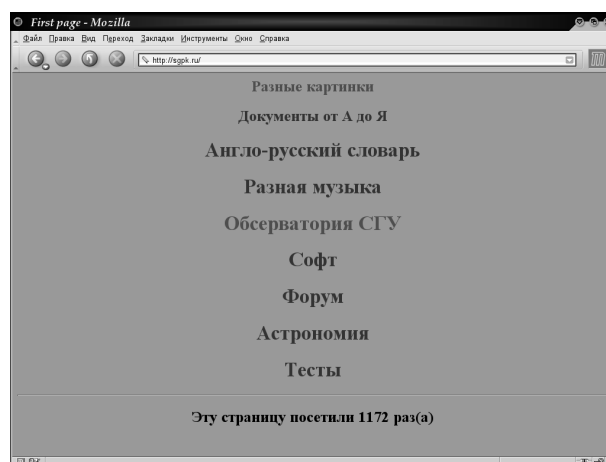


Рис. 47: Браузер Mozilla.

Выполните задание 3:

Посмотрите ресурсы сайта www.sgpk.ru. Для этого наберите в адресной строке www.sgpk.ru. По умолчанию используется протокол http, который и добавится перед набранным адресом в адресной строке.

Зайдите на форум и оставьте в нем какую-нибудь запись. Сможете ли вы удалить запись после того, как ее добавили?

Зайдите на хранилище документации. Как сохранить документ в свою домашнюю папку? Как открыть его или найти после сохранения?

Зайдите на хранилище изображений. Как сделать изображение заставкой на рабочем столе?

Для доступа к какому-нибудь сетевому ресурсу при помощи браузера в адресной строке необходимо набрать адрес этого ресурса. По умолчанию браузер обращается к ресурсу по протоколу http, если вы хотите обратиться к ресурсу, доступному по другому протоколу (ftp, snmp и т.п.), необходимо явно указать этот протокол в адресной строке перед адресом ресурса.

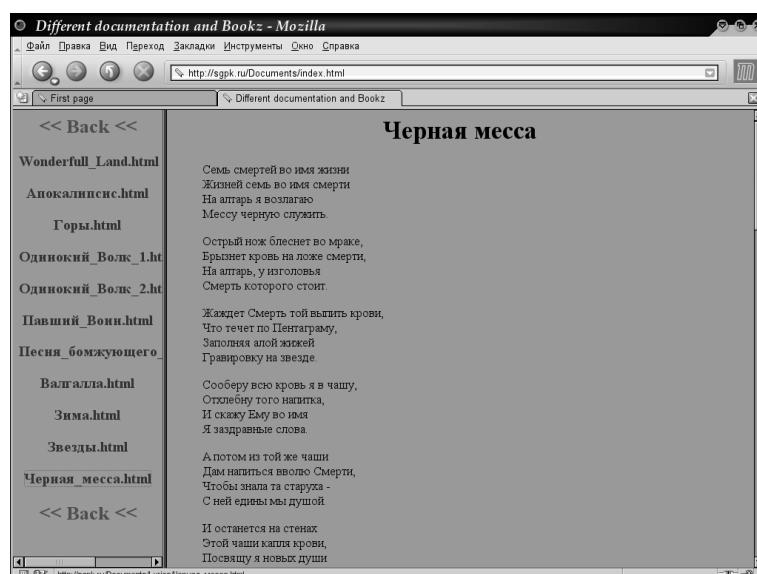


Рис. 48: Просмотр нескольких страниц при помощи вкладок.

Если браузер не может открыть web-страницу, расположенную по данному адресу, это означает сбой в настройке сервера доменных имен, который по символному имени сервера возвращает его IP-адрес (именно он используется браузером, символное имя только облегчает запоминание адресов).

Откройте файл pdf-формата с лекциями по информатике. Как видите, в браузере установлен плагин для просмотра pdf документов (иначе этот файл открылся бы в отдельном окне, либо браузер предложил бы на выбор список программ, которыми можно открыть данную ссылку).

В браузере Mozilla можно открывать гиперссылки не только в том же или другом окне, но и в новой вкладке. Для этого надо щелкнуть по ссылке **средней** кнопкой мыши. Откройте некоторые ссылки в новых вкладках (см. рис. 48).

Обратите внимание на то, что гиперссылки можно присвоить не только текстовым элементам, но и изображениям (это происходит, например, когда вы смотрите картинки на сайте www.sgpk.ru: в правом фрейме открывается уменьшенное изображение, содержащее ссылку на это же изображение «в полный рост».)

Выполните задание 4:

Зайдите на ссылку «Словарь». Как видите, помимо самих html файлов широко используются cgi-скрипты, позволяющие высылать пользователю различную информацию, в зависимости от того, какие параметры примет скрипт.

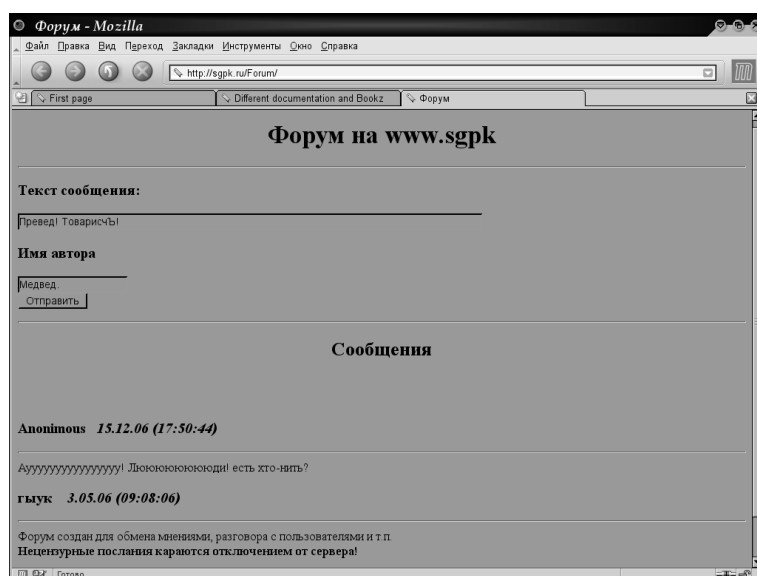


Рис. 49: Форум на локальном сервере.

Ответьте на вопрос(ы):

1. Для чего используется протокол TSP?
2. Чем протокол http отличается от протокола ftp.
3. Можно ли с помощью браузера слушать музыку, смотреть кино?

11.2 Протокол ftp

Для обмена файлами через вычислительную сеть существует более приспособленный для этих целей протокол — ftp. Напишите в командной строке Mozilla `ftp://sgpk`. Вы сможете просматривать ресурсы сервера, доступные по протоколу ftp. Однако, браузер не позволяет закачивать (upload) файлы на сервер. Для этого необходимо использовать файловый менеджер, например, mc.

Запустите `mc`. Подключимся к серверу `sgrk`. Для этого необходимо выполнить следующие действия:

1. Нажмите `<F9>`, появится меню текущей активной панели панели (см. рис. 50 слева). Нажмите `<F>` или же переместите курсор на пункт «FTP-соединение». Откроется окно настройки соединения.
2. В поле «ftp» занесите название сервера: `sgrk`; в остальные поля ничего не заносите. Нажмите `<Enter>` (см. рис. 50 справа).
3. Если вы все сделали правильно, появится содержимое удаленной папки, доступной по протоколу `ftp`.

► В общем случае формат адресной строки `ftp`-соединения выглядит так: `ftp://<имя пользователя>:<пароль>@<адрес сервера>`. ◀

Вы можете закачивать свои файлы на сервер, чтобы хранить их там, либо же чтобы обмениваться ими с другими пользователями. Для этого выберите в другой панели файлового менеджера (нажав клавишу `<Tab>`) файл, который хотите закачать на сервер. Чтобы скопировать выбранный файл, нажмите `<F5>`. Обратите внимание, что для закачивания файлов пользователей на сервер доступна только папка `pub`, остальные папки доступны только для чтения.

Вы зашли в анонимном режиме. Это аналогично тому, как если бы вы записали в поле «Имя пользователя» `anonymous`. Однако, существует и вариант авторизованного доступа к `ftp`-серверу. В этом случае администратор сервера может выделить ресурсы, доступные для данного пользователя и не доступные для других, чтобы этот человек мог хранить важную информацию, не опасаясь несанкционированного доступа к ней.

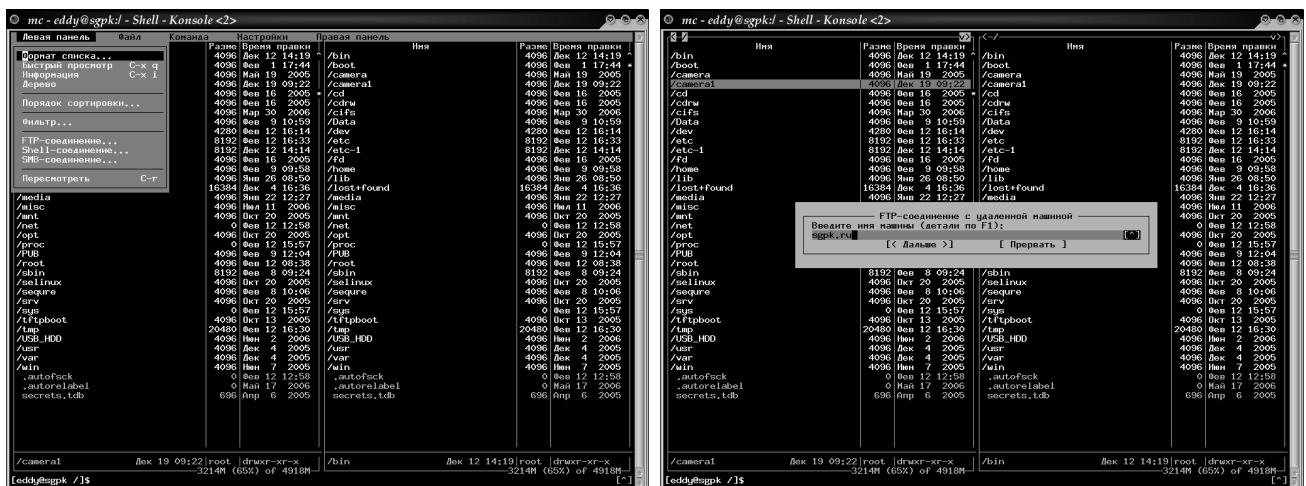


Рис. 50: Создание подключения `ftp` при помощи `mc`.

Выполните задание 5:

Зайдите на ресурс `ftp://sgpk` при помощи `ms`. Скачайте какой-нибудь файл к себе в домашнюю папку. Отошлите на сервер какой-нибудь свой файл.

Точно так же зайдите на тот же ресурс при помощи графического браузера. Для этого в строке адреса перед адресом ресурса следует указать протокол: `ftp://sgpk`. Браузер будет выступать в роли графического файлового менеджера.

12 Создание и редактирование web-страниц, размещение их на сайте.

Создать простой гипертекстовый документ можно при помощи любого текстового редактора. Но проще сделать это при помощи специализированных средств — компоновщиков.

12.1 Создание гипертекстового документа при помощи компоновщика Mozilla

С браузером Mozilla вы уже знакомы, однако, помимо браузера, существует и инструмент создания гипертекстовых документов — компоновщик (Mozilla Composer).

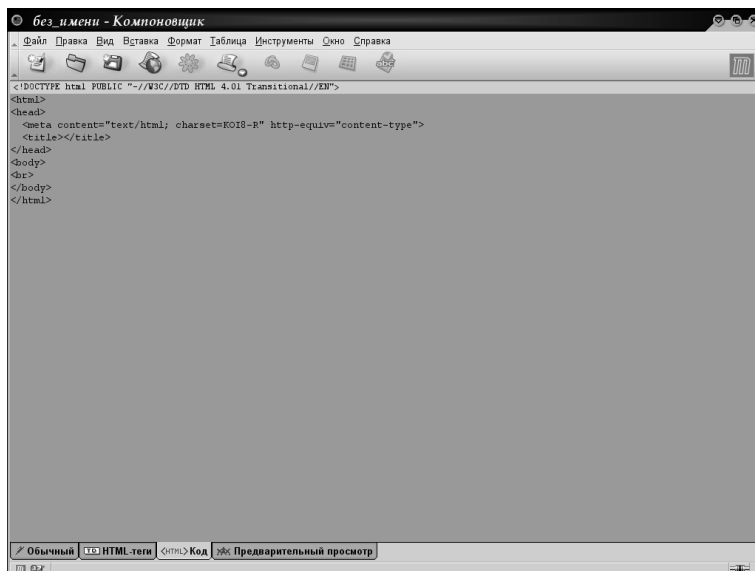


Рис. 51: Компоновщик Mozilla.

Выполните задание 1:

Запустите Mozilla, выберите в меню «Файл» → «Создать» → «Страницу компоновщика» («File» → «New» → «Composer page»). Теперь вы можете редактировать эту страницу в WYSIWYG режиме.

Внизу компоновщика расположены закладки, активируя которые можно просмотреть, как будет выглядеть ваша страница в браузере, а также — редактировать исходный текст страницы.

Активируйте закладку «HTML-Код». Обратите внимание, что даже пустой документ уже содержит несколько стандартных **тэгов** — управляющих последовательностей, своего рода команд (см. рис. 51). Первым тэгом в документе является парный тэг `<html>`, означающий, что все содержимое документа рассматривается как гипертекст. Далее следуют теги `<meta>`, в которых записывается вспомогательная информация о тематике страницы, ее создателе и т.п. В парный тэг `<body>` заключено тело гипертекстового документа.

Каждый парный тэг закрывается тэгом с таким же именем, но начинающимся с символа косой черты (слэша). Например, тэг `<body>` закрывается тэгом `</body>`.

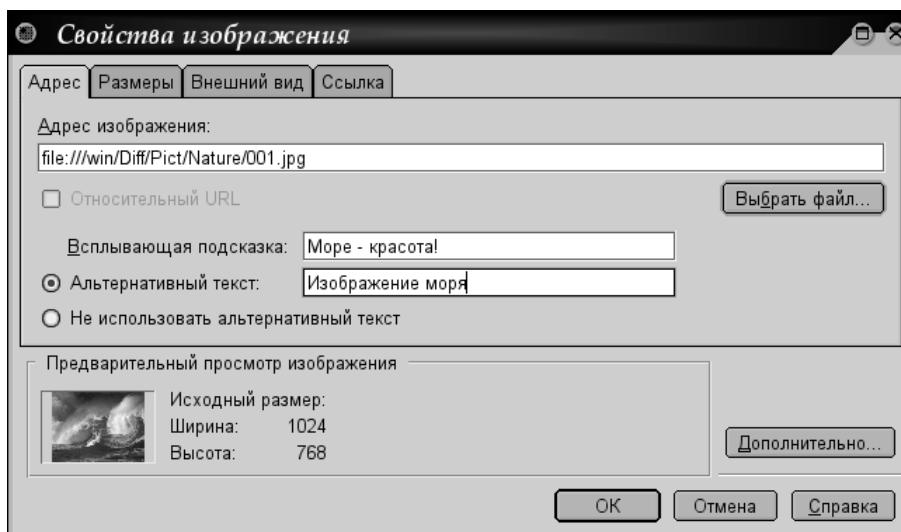


Рис. 52: Окно «Свойства изображения».

Редактировать документ удобнее всего в режиме WYSIWYG (вкладка «Обычный»). Этот режим мало чем отличается от создания текстового документа при помощи Open Office.

Выполните задание 2:

Создайте два документа и сохраните один под именем 1.html, второй — под именем 2.html. Заполните документы содержанием, как описано ниже.

Придумайте, что разместить на странице (можно обратиться за образцами по адресу sgpk.ru). Создайте рациональное оформление страницы, сделайте цветной фон (или разместите на нем фоновый рисунок), заполните документ содержанием.

Например, в первом документе напишите что-нибудь про себя. Можете добавить несколько картинок с локальной папки, или с сервера sgpk.ru. Для этого активируйте пункт меню «Вставка» → «Изображение». В поле «Адрес изображения» введите адрес картинки (протокол `http`), если вы знаете его. Например, `http://sgpk.ru/pict/Nature001.jpg`. Если же вы не знаете адреса, нажмите кнопку «Выбрать файл» (см. рис. 52) и укажите на понравившуюся картинку. Кроме того, можно заполнить поля «Всплывающая подсказка» (текст, который будет выводиться при наведении курсора мыши на картинку) и «Альтернативный текст» (текст, отображающийся на месте картинки, если просматривать данную страницу с отключением изображений или в текстовом браузере).

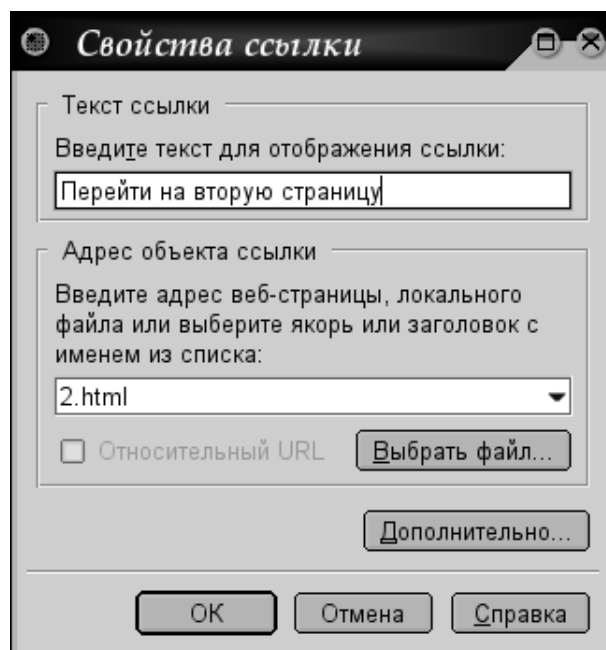


Рис. 53: Окно «Свойства ссылки».

Если изображение будет слишком большим, измените его размеры. Это можно сделать как из окна «Свойства изображения» (появится, если щелкнуть по нему левой клавишей мыши два раза), так и протягивая край изображения.

Используйте меню «Формат» для изменения параметров текста, а также стилей заголовка. Так, основной заголовок лучше всего сделать, активировав в меню «Формат» → «Абзац» пункт «Заголовок 1». Для задания выравнивания текста, а также изменения его атрибутов (жирный, курсивный, подчеркнутый) используйте инструментальную панель.

После этого создайте гиперссылку на второй документ. Для этого либо вос-

пользуйтесь сочетанием клавиш `<Ctrl> + <L>`, либо активируйте пункт меню «Вставка» → «Ссылка». Появится окно, изображенное на рис. 53.

В поле «Текст для отображения ссылки» введите «Перейти на вторую страницу», в поле «Адрес» — `2.html`, или же воспользуйтесь кнопкой «Выбрать файл».

Аналогично поступите с документом `2.html`, только в поле «Текст...» введите «На страницу 1», а в качестве адреса укажите `1.html`. Сохраните оба файла.

Теперь просмотрите свои файлы в браузере. Это можно сделать либо открыв эти файлы, либо прямо из компоновщика, выбрав пункт меню «Файл» → «Просмотр страницы». Щелкните по гиперссылке средней кнопкой мыши, и в новой вкладке откроется ваш второй файл. Активируя гиперссылку на нем вы сможете вернуться на первый файл.

12.2 Создание гипертекстового документа в Open Office Web

Open Office позволяет экспортировать созданные в нем документы в формат `html` для возможности публикации на web-сервере.

Выполните задание 3:

Экпортируйте любой текстовый документ и любую презентацию из своей рабочей папки в формат `html`. Для этого воспользуйтесь меню «Файл» → «Экспорт». Откройте полученный документ при помощи браузера.

Отредактируйте экспортируемый текстовый документ в Mozilla Composer, для чего во время просмотра документа нажмите сочетание клавиш `<Ctrl> + <E>`н или выберите из меню «Файл» пункт «Редактировать».

Удалите из документа все `<meta>` тэги, оформите основные заголовки стилем `<h1>`, а подзаголовки — `<h2>`. Сделайте выравнивание текста по правому краю, а выравнивание заголовков — по центру.

Удалите содержимое одной из ячеек таблицы. Измените содержимое перечней, добавив новый пункт. Обратите внимание, какими тэгами обрамляются нумерованные, а какими — маркированные списки.

12.3 Публикация страницы

Для того, чтобы разместить свою домашнюю страницу на сайте, необходимо иметь к нему авторизованный доступ на запись (чаще всего — по протоколу `ftp`). Вы можете размещать свои страницы, помещая их в папку `/home/user/www`. Для того, чтобы кто-то смог просмотреть ваши ресурсы, он должен набрать в адресной строке браузера `http://206-<номер вашего компьютера>/~user`.

Однако, мало поместить файл в эту папку. По умолчанию браузер ищет файл `index.html`. Если такого не найдется, он выдаст ошибку. Поэтому вы должны сначала создать индексный файл. Если ваша страница занимает единственный документ, можно смело переименовать его в `index.html` и сохранить в папку `www`. Если же файлов много, необходимо будет в индексном файле создать гиперссылки на каждый из этих файлов.

Выполните задание 4:

Опубликуйте свою домашнюю страницу. Посмотрите, как она отображается, набрав в браузере `http://localhost/~user`. `localhost` — удобный синоним имени компьютера, за которым вы работаете.

Если страница отобразится верно, попросите друзей посмотреть со своих компьютеров, как она отображается. Посмотрите 2-3 странички, созданных другими.

13 Компьютерная модель и ее исследование.

Моделирование — метод познания, состоящий в создании и исследовании моделей. **Модели** позволяют представить *в наглядной форме* объекты и процессы, недоступные для непосредственного восприятия. Модели играют важную роль в проектировании и создании различных технических устройств, механизмов, электрических цепей и т.д.

Все модели можно разбить на два класса: *предметные* (материальные) и *информационные*. Предметные модели воспроизводят свойства объектов в материальной форме, а информационные — в образной или знаковой.

С помощью формальных языков строятся *формальные информационные модели* (математические, логические и др.), а с помощью математических понятий и формул строятся *математические модели*. Исследование модели становится более наглядным, если использовать ее **визуализацию** — представить модель в виде чертежа, схемы, изображения и т.п.

Информационные модели делятся на следующие типы:

- Табличные модели представляют собой перечень объектов и их свойств. Типичным примером такой модели является преЙскурант, таблица истинности логической схемы и т.п. Визуализация такой модели наиболее наглядна в форме диаграммы или графика.
- Иерархические модели имеют иерархическую структуру: объекты делятся на классы, которые распределяются по степени подчинения друг другу. Наиболее наглядно и естественно представить такую модель в виде иерархического дерева (например, генеалогическое древо).
- Сетевые модели применяются для отражения систем со сложной структу-

рой, в которых связи между элементами имеют произвольный характер. Визуализация такой модели имеет вид графа, в котором отражены взаимосвязи элементов модели в виде стрелок или прямых.

Для исследования модели можно использовать такой алгоритм:

1. Содержательная постановка задачи. Задание необходимых для составления модели условий и параметров.
2. Создание качественной описательной модели. Словесное описание модели и допущений, возможных при формализации.
3. Составление формальной модели. Запись всех необходимых уравнений и условий, составление набросков графиков и т.п.
4. Визуализация модели. Графическое отображение модели.

13.1 Исследование математической модели.

Выполните задание 1:

Найдите графическим методом корень уравнения $x^3 = \sin x$, не имеющего точного алгебраического решения.

Выполнять это задание следует, воспользовавшись следующим алгоритмом:

1. Создайте электронную таблицу Open Office Calc. Сохраните ее под названием Lesson_7 в своей рабочей папке.
2. В самой первой ячейке поставьте -1, в ячейке под ней запишите формулу =A1+0.05 и протяните эту ячейку за правый нижний угол до 41-й строки (в появившемся диалоговом окне отметьте «Формула» и нажмите «ОК».
3. В ячейке **B1** запишите формулу =A1^3, а в **C1** — =SIN(A1) и тоже протяните их до 41-й строки.
4. Выделите всю получившуюся таблицу и выберите в меню «Вставить» → «Диаграмма».
5. В открывшемся диалоге выбора типа диаграммы поставьте флажок у пункта «Первый столбец как подпись»; нажмите «Далее».
6. Выберите тип диаграммы — «Линии»; укажите, что ряды данных расположены в столбцах; нажмите «Далее».
7. Выберите «Обычная»; нажмите «Далее».
8. Введите название диаграммы и укажите подписи осей; нажмите «Готово».

На построенном графике найдите координаты точек пересечения двух кривых. Это и будут искомые корни уравнения. Для более точного определения корня, постройте еще один столбец, в который внесите формулу =A1^3-sin(A1). Выделите в этом столбце две строки так, чтобы в одной значение было меньше, а в другой больше нуля. Удерживая <Ctrl>, выделите соответствующие ячейки первого столбца. Постройте график по этим Значениям. Увеличив его вы смо-

жете определить корни уравнения со значительно большей точностью (см. рис. 54).

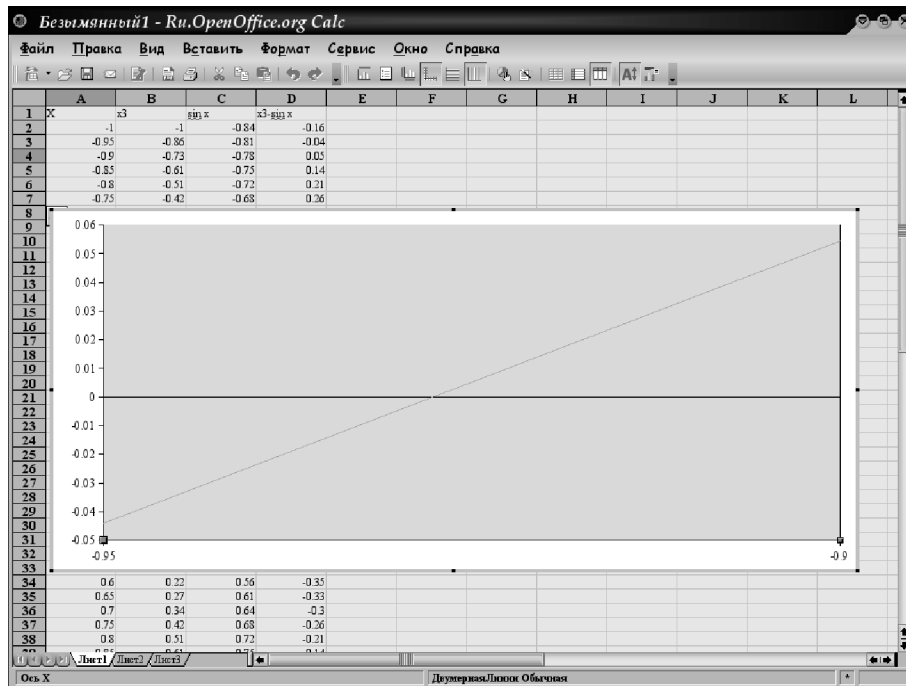


Рис. 54: Решение уравнения $x^3 = \sin x$ графическим способом

13.2 Исследование физической модели.

Выполните задание 2:

Построить модель полета тела, брошенного под углом α к горизонту (см. рис. 55).

Постановка задачи: в процессе тренировки теннисистов используются автоматы для бросания мяча. Требуется задать автомату необходимую скорость и угол бросания мячика для попадания в определенное место площадки.

Качественная описательная модель. Здесь можно сделать следующие допущения:

- из-за малых, по сравнению с Землей, размеров мяча его можно считать материальной точкой;
- т.к. мячик взлетает на высоту, пренебрежимо малую по сравнению с радиусом Земли, можно считать, что на него действует постоянная сила тяжести, а его движение вдоль оси Y происходит равноускоренно с ускорением $g = 9.81 \text{ м/с}^2$;
- скорость бросания мяча невелика, так что сопротивлением воздуха можно пренебречь.

Формальная модель. Опишем математически движение мяча. По оси X

он движется равномерно, а по оси Y — равноускоренно. Проекции начальной скорости на координатные оси равны:

$$v_x = v_0 \cos \alpha; \quad v_y = v_0 \sin \alpha.$$

Соответственно, координаты будут меняться по законам:

$$x = v_0 \cos \alpha \cdot t; \quad y = v_0 \sin \alpha \cdot t - \frac{gt^2}{2}.$$

Когда мяч упадет на площадку, его координата y станет равна 0, следовательно, можно получить время полета мяча:

$$v_0 \sin \alpha \cdot t = \frac{gt^2}{2} \quad \Rightarrow \quad t_1 = 0, \quad t_2 = \frac{2v_0 \sin \alpha}{g}.$$

t_1 нас не интересует, т.к. это — время вылета мяча из автомата. Теперь, подставив найденное t в выражение для x , получим:

$$x = \frac{v_0^2}{g} \sin 2\alpha. \quad (*)$$

Пусть аппарат обладает возможностью задать мячу одну из трех скоростей: $v_1 = 5$ м/с, $v_2 = 10$ м/с, $v_3 = 20$ м/с. Согласно формуле (*), какими бы ни были углы наклона, для конкретной скорости v_0 расстояние, которое пролетит мяч, лежит в диапазоне $\left[0, \frac{v_0^2}{g}\right]$. Таким образом, если необходимо бросить мяч на расстояние x , вначале следует выбрать необходимую начальную скорость из условия

$$\frac{v_0^2}{g} \geq x, \quad \Rightarrow \quad v_0 \geq \sqrt{gx}.$$

Итак, если начальная скорость v_1 , v_2 или v_3 удовлетворяет этим условиям, вторым шагом будет выбор угла наклона автомата. Следует учесть, что почти для всех точек таких углов будет 2: $\alpha_1 = 45^\circ - \beta$ и $\alpha_2 = 45^\circ + \beta$, где β — некоторый угол.

Вычислить угол можно из формулы (*):

$$\alpha = \frac{1}{2} \arcsin \frac{gx}{v_0^2}.$$

Вполне естественно, что для некоторых расстояний будет возможным задать все три начальные скорости, и для каждой скорости задать по два угла, т.е. попасть в эти места можно при различных шести значениях скорости/угла наклона автомата.

Это задание также будем выполнять в OO Calc.

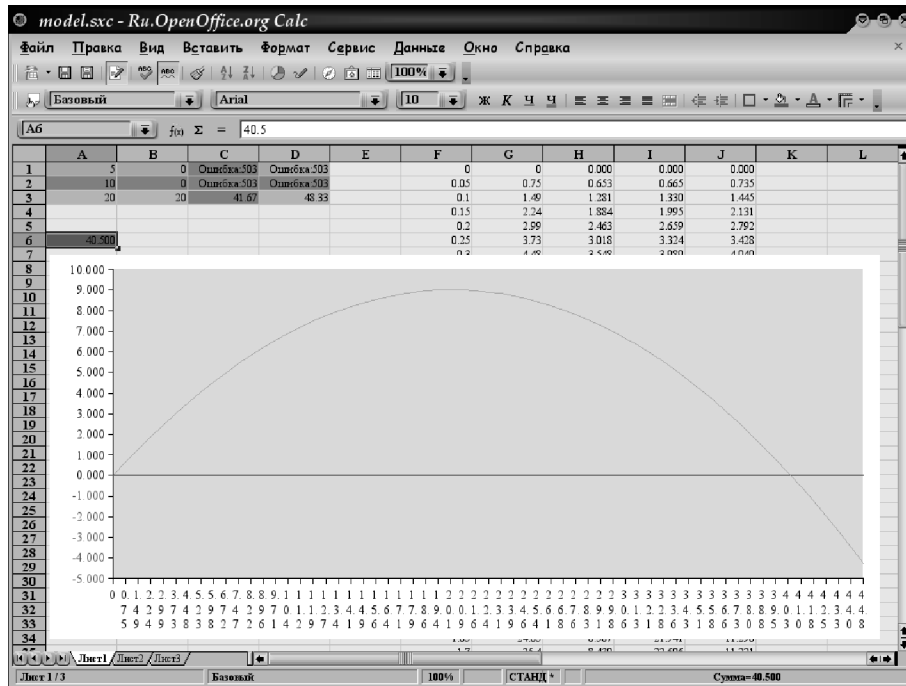


Рис. 55: Модель полета тела, брошенного под углом к горизонту

1. Создайте новый документ или используйте другой лист имеющегося документа.
2. В первый столбец занесите скорости: 5, 10 и 20 м/с. Выделите ячейку A6 каким-нибудь цветом. В нее вы будете заносить требуемую дальность полета. В первую ячейку второго столбца занесите формулу $=IF(A1 \geq \sqrt{\$A\$6 * 9,81}; A1; 0)$ и протяните ее вниз до третьей строки. Это — условие, проверяющее, сможет ли при данной начальной скорости мяч долететь до цели.
3. В первую строку столбца C занесите формулу $=0,5 * ASIN(9,81 * \$A\$6 / (B1^2)) * 180 / PI()$. Здесь будет рассчитываться значение первого угла. Значение второго угла будет рассчитываться в столбце D, куда следует занести формулу $=90 - C1$. Выделите обе ячейки и протяните вниз до третьей строки.
4. В столбец F занесите значения времени от 0 до 4 с интервалом 0,05.
5. В столбцах G и I будем рассчитывать координату X при полете мяча с максимальной скоростью. Занесите в G1 формулу $=\$A\$6 * COS(\$C\$3 * PI() / 180) * F1$. В I1 занесите формулу $=\$A\$6 * COS(\$D\$3 * PI() / 180) * F1$.
6. В столбцах H и J будем рассчитывать координату Y мяча. Занесите в ячейку H1 формулу $=\$A\$6 * SIN(\$C\$3 * PI() / 180) * F1 - 4,905 * F1^2$, а в ячейку J1 — формулу $=\$A\$6 * SIN(\$D\$3 * PI() / 180) * F1 - 4,905 * F1^2$.

7. Протяните первую строку от столбца G до столбца J до цифры 4 столбца F.
8. Теперь внесите в ячейку A6 число 15 и постройте траекторию полета мяча для двух углов. Для этого выделите столбцы G и H и выберите в меню «Вставить» → «Диаграмма». Укажите, что подписи оси X расположены в первом столбце. Выберите тип диаграммы «Линии», «Обычная». Аналогично поступите для столбцов I и J.
9. Теперь измените значение расстояния на 10 и посмотрите, как изменятся графики полета мяча. Для построения графика выделите колонки, в которую занесены координаты X и Y (например, G и H. Затем выберите пункт меню «Вставка» → «Диаграмма». В появившемся окне не забудьте поставить флажок «Первый столбец как подпись», далее выберите нужный тип диаграммы и нажмите «ОК».

14 Основы программирования в shell.

Shell (оболочка) — командный процессор, принимающий команды пользователя, анализирующий их и выполняющий необходимые действия. Наиболее часто в Linux употребляется оболочка Bash (`bash`), она же вызывается по команде `sh`. Некоторые команды оболочки вы уже знаете, но бывают случаи, когда приходится по многу раз вводить одни и те же команды, меняя лишь малую их часть. Для упрощения работы пользователя в таких случаях используются командные файлы (**скрипты shell**).

14.1 Простейшие скрипты.

14.1.1 Подсчет файлов 1.

|| Выполните задание 1:

|| Создайте скрипт, подсчитывающий количество файлов в текущей папке.

Создайте в своей рабочей папке файл `count` и сделайте его исполняемым (наберите команду `chmod 775 count`). Теперь откройте этот файл для редактирования любым текстовым редактором (если вы работаете в `mc`, наведите выделение на этот файл и нажмите клавишу F4, если командной строке — наберите, например, `kwrite count`).

В самом начале файла наберите строку

```
#!/bin/sh
```

Она необходима для того, чтобы командный процессор распознал данный файл как скрипт. Далее наберите непосредственно текст скрипта:

```

COUNT=-2                                # Не считаем папки . и ..
for FILE in `ls -a`                        # Начало цикла
do                                         # по списку файлов в папке
    COUNT=`expr $COUNT + 1`             # Увеличить счетчик на 1
done                                       # Конец цикла
echo "Папка содержит $COUNT файлов и папок" # Вывод результата

```

- ▶ Обратите внимание, что кавычки, в которые заключены команды `ls` и `expr`, получают при нажатии буквы ``` (слева от 1) в английской раскладке. Они означают, что заключенный в них текст будет выполнен как команда, а результат ее — подставлен вместо кавычек. ◀
- ▶ Знаком `#` отмечаются комментарии, все, что написано в строке после него, не будет считано командным интерпретатором, поэтому, в принципе, комментарии можно не записывать. Они нужны только для лучшего понимания скрипта. Единственный комментарий, который обязательно следует записать — указание командной оболочке, что для исполнения файла необходимо использовать `bash` (`#!/bin/sh`). ◀

Здесь `COUNT` — счетчик, который будет содержать количество файлов и папок в текущей папке. Далее в программе следует цикл, в котором переменная `FILE` принимает все значения из листинга текущей папки. На каждый файл переменная `COUNT` увеличивается на 1.

Знак `$` перед именем переменной позволяет получить ее значение, а команда `expr` представляет собой простейший калькулятор командной строки, возвращая значение выражения, стоящего после нее.

Набрав и сохранив программу, выполните ее, для чего в `ms` просто наведите на нее выделение и нажмите `Enter`, а в командной строке ее можно выполнить двумя способами: набрав либо `sh count`, либо `./count`.

14.1.2 Подсчет файлов 2.

Выполните задание 2:

Теперь заставьте программу не просто считать файлы и папки вместе, а определять отдельно количество файлов, папок и исполняемых файлов.

Для этого нам понадобится три счетчика. В начале программы обнулим счетчики файлов и исполняемых файлов, а счетчику папок присвоим значение `-2`, чтобы не считать папки `."` и `".."`.

Кроме того, сделаем так, чтобы программа могла не только считать файлы в текущей папке, но и в любой другой, указанной в параметре команды.

- ▶ **Параметром** команды называют текст, который идет после ее имени.

Обычно параметры используются для того, чтобы задавать различные варианты исполнения команды, либо передавать ей какие-либо данные. Например, в предыдущем тексте для вывода списка всех файлов данной папки мы передали команде `ls` параметр `-a`. Для того, чтобы просмотреть содержимое не текущей папки, необходимо передать ее имя как параметр команды `ls`: например, `ls /usr/lib`. ◀

- ▶ **Ключом** команды называют параметр, влияющий на ее выполнение. Обычно все ключи начинаются со знака `'-'` или же `'--'`. ◀

Организуем цикл по списку файлов и папок в данной папке, а затем будем проверять каждый элемент этого списка, прибавляя к соответствующему счетчику 1. Проверка типа выполняется с помощью условного оператора `[]`, проверяющего истинность значения, заключенного в скобках. Оператор `&&` выполняет следующую за ним команду, если предыдущая вернула истинное значение. Каждому типу файлов соответствует свой флаг проверки: `-f` — для любых файлов, `-d` — для папок, `-x` — для исполняемых файлов. Когда будет необходимо применить сложную проверку, будем пользоваться операторами `if — then — else — fi`, означающими «если» — «то» — «иначе» — «конец проверки».

- ▶ Цикл в скриптах `bash` организуются командой `for`, после которой следует имя переменной, в которую будут заноситься очередные значения цикла, затем — слово `in`, означающее, что значения будут браться из последующего списка, и, наконец, сам список значений, чаще всего его заменяют выводом какой-нибудь программы, как мы и поступим. ◀

После инициализации переменных проверим, ввел ли пользователь имя папки, в которой необходимо подсчитать количество файлов. Если это имя не введено, будем считать файлы в текущей папке.

- ▶ Доступ к параметрам, сообщенным скрипту в командной строке, осуществляется так же, как и доступ к значениям переменных, за тем исключением, что каждый параметр обозначается цифрой, соответствующей его номеру. Нулевой параметр соответствует названию команды. Пример: `$2` означает доступ ко второму параметру введенной команды. ◀

После завершения цикла выведем содержимое счетчиков на экран.

```
#!/bin/sh

countf=0          # счетчик файлов
countd=-2        # счетчик папок
countx=0         # счетчик исполняемых файлов
[ "$1" != "" ] && dir="$1/" # по умолчанию будем считать в текущей папке
for file in `ls -a $1`
do
```

```

[ -f "$dir$file" ] && countf='expr $countf + 1'
[ -d "$dir$file" ] && countd='expr $countd + 1'
[ -x "$dir$file" -a ! -d "$dir$file" ] && countx='expr $countx + 1'
done
echo -e "Подсчет файлов в $PWD/$1:\n$countd\t \
папок\n$countf\tфайлов\n$countx\tисполняемых"
echo "Всего: 'expr $countf + $countd + $countx' файлов и папок"

```

- Параметр `-e` говорит команде `echo`, что она должна учитывать специальные параметры, управляющие текстом: `\n` — начать новую строку, `\t` — вставить символ табуляции. ◀

Набрав и сохранив программу под именем `count1`, выполните ее, для чего в `ms` просто наведите на нее выделение и нажмите Enter, а в командной строке ее можно выполнить двумя способами: набрав либо `sh count1`, либо `./count1`.

14.2 Более сложные скрипты.

14.2.1 Сбор файлов из разных папок в одну 1.

Выполните задание 3:

Создайте программу, собирающую в одну папку файлы, раскиданные по большому числу папок. Это полезно, например, при создании фотоальбома с цифрового фотоаппарата.

Так как при чтении списка файлов программа будет регулярно находить папки, файлы из которых надо извлечь, нам придется вызывать эту программу рекурсивно, то есть из самой себя, передавая параметром папку, в которую надо «складывать» файлы. Таким образом, у нашей команды обязательно должен быть один параметр. По умолчанию «складывать» файлы будем в папке `GATHER`, поиск будем начинать с текущей папки.

Вначале проверим, является ли параметр нашей команды полным именем, то есть проверим, начинается ли оно со знака `/`. Для этого нам поможет команда `grep`, показывающая все вхождения заданного ей параметра в файле или входных данных. Данные на вход команды `grep` перенаправим при помощи *конвейера* — оператора `|`, который переводит стандартный вывод (выходные данные) первой программы на стандартный вход второй (входные данные).

- **Полным именем** (путем) файла называется весь путь к файлу, начиная с корневой папки. Например, `/var/log/messages/1.msg` — полное имя файла, `messages/1.msg` — его имя (путь к нему) относительно папки `log`, а `1.msg` — краткое имя файла (или же — собственно имя файла). ◀

При проверке имя папки сохраним в переменной `tmp`. Если оно является полным, присвоим его переменной `where`, в которой будет храниться полный путь к искомой папке. Если имя является неполным, сделаем его таким, подставив префиксом текущую папку (она хранится в переменной `PWD`), а если `gather` была запущена вообще без параметров, присвоим переменной `where` значение `GATHER`.

Файлы будем копировать командой `cp`, если вы хотите, чтобы файлы не копировались, а перемещались, необходимо заменить ее на команду `mv`.

```
#!/bin/sh

tmp='echo "$1" | grep "^/'
if [ "$tmp" != "" ]; then
    where="$1"
else
    if [ "$1" != "" ]; then
        where="$PWD/$1"
    else
        where="$PWD/GATHER"
    fi
fi
[ ! -e $where ] && mkdir $where

for file in `ls`
do
    if [ "$PWD/$file" != "$where" \
    -a -d "$file" ]; then
        cd $file
        gather $where
        cd ../
        continue
    fi
    cp $file $where
done
```

Проверка, является ли
параметр \$1 полным именем
файла.
Если не является, то
если он не пустой
приводим его к полному виду,
иначе -
будем складывать в GATHER.
Конец проверки
Конец проверки
Если папки назначения нет,
создаем ее.
Цикл по списку файлов в
текущей папке (без скрытых).
(перенос строки)
Если наткнулись на папку,
входим в нее
и собираем из нее файлы,
а потом выходим назад,
прерывая цикл, чтобы
не обработать случайно папку
Копируем файл в место назначения
Конец цикла

14.2.2 Сбор файлов из разных папок в одну 2.

Выполните задание 4:

Усложните программу: пусть она собирает только картинки `jpg` и переименовывает их цифрами, начиная с 1. Программу назовите `gather1`

Для этого нам понадобится счетчик, в котором будет находиться номер текущей картинки, назовем его `count`. Кроме того, т.к. программа работает рекурсивно, необходимо передавать дочерним процессам (вызываемым основной программой) номер текущей картинки (будем передавать его вторым параметром).

Основная программа также должна знать, до какого номера «досчитался» дочерний процесс, т.е. он должен возвращать текущее значение счетчика после завершения своей работы (сделаем это при помощи команды `echo $count`, выводящей в конце программы текущее значение счетчика).

Увеличивать значение счетчика будем опять при помощи команды `expr`. Проверить, является ли файл картинкой `jpg` можно по его расширению при помощи команды `grep`, перенаправив на ее стандартный вход имя файла, выведенное командой `echo`. Будем искать вхождение выражения `.jpg` в конец имени файла, для этого используем знак `$` в конце выражения, означающий конец строки (начало строки обозначается знаком `^`). Кроме того, следует учесть, что имя файла может быть не только в нижнем регистре, но и в верхнем, т.е. следует проверять как малые, так и большие буквы в расширении. Для того, чтобы подставить одно из нескольких значений в строчном выражении, их пишут в квадратных скобках, не разделяя запятой.

```
#!/bin/sh

if [ "$2" != "" ]; then          # Если есть второй параметр,
    count="$2"                  # присваиваем счетчику его значение,
else                             # иначе
    count=1                     # присваиваем счетчику 1.
fi
tmp='echo "$1" | grep "^/'
if [ "$tmp" != "" ]; then
    where="$1"
else
    if [ "$1" != "" ]; then
        where="$PWD/$1"
    else
        where="$PWD/GATHER"
    fi
fi
[ ! -e $where ] && mkdir $where
for file in `ls`
do
    if [ "$PWD/$file" != "$where" \
-a -d "$file" ]; then
        cd $file
        count='gather $where $count'
        cd ../                      # Присвоим счетчику значение дочернего процесса.
        continue
    fi
    tmp='echo "$file" | \
grep ".[jJ][pP][gG]$" '        # Проверим, картинка ли наш файл
    if [ "$tmp" != "" ]; then    # если картинка, то
        cp $file $where/$count.jpg# скопируем ее в папку назначения,
```

```
        count='expr $count + 1'    # и увеличим на 1 значение счетчика.  
    fi  
done  
echo $count                        # В конце выведем текущее значение счетчика.
```

15 Основы программирования на языке C.

15.1 Знакомство с языком C.

15.1.1 Введение.

Для написания компьютерных программ используется большое число языков программирования. Для того, чтобы запустить написанную на одном из них программу, следует **транслировать** ее в понятные компьютеру коды. Программ-трансляторов намного больше, чем языков программирования, каждый программист использует тот транслятор, который ему больше нравится и лучше всего удовлетворяет требованиям к готовой программе.

Трансляторы делятся на **компиляторы** и **интерпретаторы**.

Интерпретаторы представляют собой трансляторы реального времени, читающие программный код строка за строкой, анализирующие его и передающие операционной системе необходимые команды. Примером языка программирования, использующего интерпретатор для выполнения программ является BASIC. Недостатки таких программ в том, что для их запуска всегда необходимо запускать в начале программу-интерпретатор, кроме того, на анализ кода тратится большое количество времени, что сильно замедляет выполнение программы.

Более «продвинутыми» являются языки программирования, использующие компиляторы. Компилятор анализирует программный код и создает **исполняемый файл**, содержащий **машинные коды**, понятные операционной системе. Исполняемый файл не нужно анализировать в ходе его выполнения — он загружается в память компьютера и выполняется ОС сразу, что значительно повышает скорость выполнения программы.

Большая часть кодов всех операционных систем написана на языках C и C++. С первым мы и познакомимся. Второй представляет собой усовершенствованный вариант языка C, значительно упрощающий программисту написание сложных программ.

Файлы, содержащие текст программ на языке C, обычно имеют расширение `c`, а на языке C++ — `cpp`, `cc` или `c++`. Кроме них существуют **заголовочные файлы**, имеющие расширение `h` (в C++ — `hpp`, `H` или `h++`). Заголовочные файлы содержат информацию о функциях и переменных, хранящихся в соответствующих `c` файлах.

15.1.2 Структура файлов на языке С.

Для трансляции программного кода, написанного на С и С++ в системе Linux используется компилятор `gcc` (аббревиатура, означающая GNU C compiler — компилятор языков С и С++).

Программы на языке С имеют модульную структуру, где каждый модуль заключен в фигурные скобки `{}`. Модулями могут быть как функции, так и тела циклов, операторов выбора и пр.

Функции на языке С имеют следующий вид:

```
TYPE FNAME(ARGLIST){
    FBODY;
    return RET;
}
```

где **TYPE** — **тип данных** **RET**, возвращаемых функцией; **FNAME** — имя функции, оно должно начинаться с буквы или символа `_`, может содержать латинские буквы (компилятор различает прописные и строчные буквы, так что `myFn` и `myfn` — разные имена функций), цифры и символ `_`; **ARGLIST** — **список аргументов**, представляющий собой набор переменных (и описаний их типов), которые должны быть переданы в функцию и/или возвращены ею (если не хватает одного возвращаемого значения **RET**; **FBODY** — **тело функции**, представляющее собой набор команд, которые должна выполнять эта функция.

Все функции в программе разделяются знаком « ; », который говорит компилятору, что действие данного оператора закончилось.

Главной функцией программы является функция `main`, возвращающая обычно целое значение `int`. Если компилятор не находит такой функции, он выдает ошибку. Функция `main` может быть единственной в программе, а может и вызывать другие функции.

Более широким является понятие **оператора**, представляющего собой указание ОС совершить какое-то действие. Оператором может быть и функция, и, например, арифметические операции (`+`, `-`, `*`, `/`). Существует множество различных операторов, часть из них вы изучите.

Типы данных. Язык С использует много типов данных, нас будут интересовать следующие:

`int` — целое число;

`float` — число с плавающей точкой (дробное);

`char` — «буква» — целое число, занимающее 1 байт, обычно его используют для хранения букв;

`void` — значение-ничто, этот тип используется для обозначения функций, не возвращающих никакого значения и в некоторых других случаях;

`TYPE*` — указатель на тип `TYPE`, представляющий собой адрес, по которому в памяти размещены данные; чтобы ОС знала, как трактовать эти данные, перед указателем и указывается их тип.

Ответьте на вопрос(ы):

В языке Pascal помимо функций существует понятие процедуры, выполняющей какое-либо действие, но не возвращающей никаких значений. Как будет выглядеть запись процедуры на языке С?

Чтобы использовать в своей программе переменные, необходимо заранее дать компилятору знать о них, **объявив их** до их использования. Объявление переменной имеет вид, подобный `int A1`, означающий, что далее будет использоваться целая переменная `A1`. Если объявить переменную вне всех функций, она будет **глобальной**, т.е. ее значение во всех функциях программы будет одним и тем же. Если же объявить ее внутри какого-то модуля, при завершении его программа сотрет эту переменную, поэтому ее называют **локальной**.

Если необходимо объявить несколько переменных, их можно перечислить через запятую: `float a, b, c`; эквивалентно записи `float a; float b; float c;`.

Библиотечные функции. В стандартном С существует множество функций, но все же их обычно недостаточно для совершения всего многообразия операций, которое может понадобиться программисту. Большую часть необходимых функций содержат **библиотеки**, представляющие собой уже скомпилированные файлы, содержащие определенный набор функций. Чтобы выполнить такую функцию, необходимо подключить данную библиотеку, чтобы в процессе компиляции специальная программа (`linker`) включила в код нашей программы код, содержащий данные функции.

Подключение библиотек осуществляется путем указания компилятору ее имени, например, чтобы подключить математическую библиотеку, при компиляции надо указать `gcc Имя программы -lm`, т.е. поставить ключ `-lИмя библиотеки`.

Каждая библиотека содержит набор большого количества функций, собранных по пакетам, подключение которых осуществляется при помощи специальной записи в начале файла, имеющей вид `#include <Имя заголовочного файла>` для стандартных библиотек или `#include "Имя заголовочного файла"` для

подключаемых файлов, созданных пользователем (т.е. большой проект можно разбить на несколько мелких файлов и указать главному файлу, что их надо подключить на завершающем этапе компиляции — связке).

- ▶ Стандартная библиотека C подключается по умолчанию, поэтому в простых программах не надо указывать ключ `-lstd`. ◀

Основные функции, необходимые для преобразования типов данных и некоторых других действий, содержатся в модуле `stdlib`. Модуль `stdio` позволяет работать с файлами, и, как следствие, многими устройствами (которые, как вы помните, представляются в ОС Linux как файлы), а также с консолью (набранный вами текст попадает в файл стандартного ввода, а отображаемый на экране получается из файла стандартного вывода). Для математических вычислений используется библиотека `math`. Это — основные библиотеки, которые нам понадобятся. Кроме них существует множество специальных библиотек, но для познания основ C они не обязательны.

Конечно, хорошим вариантом могло бы показаться собрать все библиотечные файлы в одну библиотеку и подключать ее ко всем компилирующимся программам, но в этом случае размер полученного исполняемого файла был бы огромен.

15.2 Простейшие программы.

15.2.1 Hello World! (Здравствуй, мир!).

Итак, приступим к познанию C со стандартной программы, выводящей на экран надпись "Hello World!". В C существует множество операторов вывода, в данном случае нам больше всего подойдет `printf` (вывод на экран по формату). Если вы хотите узнать что-либо по конкретной стандартной функции, вызовите ее man-страничку командой `man <команда>` (так же, как это можно узнать для любой команды Linux).

|| Выполните задание 1:

|| Создайте программу, печатающую надпись "Hello, world".

Создайте в своей рабочей папке папку «C». Создайте в ней текстовый файл `HelloWorld.c` и откройте его любым редактором (в Bash можно, войдя в эту папку, сразу набрать `mcedit HelloWorld.c` или `kedit HelloWorld.c`, тогда редактор сам создаст нужный файл и откроется для его редактирования).

- ▶ Создание папки — `mkdir Имя`, создание файла — `touch Имя`. ◀

Для того, чтобы вывести текст на экран монитора, нам понадобится только одна библиотека `stdio`, которую мы объявим в начале файла. Программа

выполняет простейшие действия, т.е. она будет содержать одну-единственную функцию `main`, которая вызовет функцию вывода на печать и завершится:

```
#include <stdio.h>           // Подключим стандартные функции ввода-вывода

int main(){                  /* Главная функция */
    printf("Hello, World!\n"); // Выведем на экран надпись <<Привет, мир!>>
    return 0;                // Выход из программы
}
```

- ▶ Обратите внимание на то, что в языке C комментариями является весь текст строки, стоящий после символа двойного слэша (`//`), а также весь текст, независимо от количества занимаемых им строк, стоящий между символами `/*` и `*/`. ◀

Запись `\n` говорит о том, что после фразы «Hello, World!» необходимо вставить знак переноса строки, чтобы после окончания программы приглашение Bash не оказалось на одной с ней строчке.

- ▶ Программа после своего выполнения должна возвращать 0 или положительное число, что говорит о нормальном ее завершении, если в программе произошла ошибка, возвращается код ошибки (отрицательное число). ◀

Набрав и сохранив текст программы, нам следует откомпилировать его. Можно просто набрать `gcc HelloWorld.c`, тогда компилятор создаст исполняемый файл `a.out`, но лучше указать компилятору ключ `-o`, который поместит готовую к выполнению программу в файл, указанный после ключа:

```
gcc HelloWorld.c -o Hello
создаст файл Hello. Запустите его командой ./Hello.
```

- ▶ Указывать `./` перед именем программы нужно для того, чтобы Bash знала, что данный файл нужно искать в текущей папке, а не в папках, в которых по умолчанию хранятся программы. ◀

|| **Выполните задание 2:**
|| Составьте блок-схему программы.

15.2.2 Ввод, условный оператор.

Обычно программа должна выполнять какие-то действия в ответ на действия пользователя или введенные им данные. Ввести данные в программу можно как при ее запуске (параметрами командной строки), так и в процессе ее выполнения при помощи функций `getchar`, `scanf` и им подобных.

Функция `getchar` считывает одну букву и возвращает ее код, функция

`scanf` имеет такой формат: `scanf("формат список переменных)`, где `формат` — описание типа данных, вводимых пользователем (например, `%d` означает целое число), `список переменных` — разделенный запятыми список переменных, которые должен ввести пользователь (на каждую переменную должен быть указан ее формат, форматы разделяются через пробел).

- ▶ Более подробно о стандартных функциях можно узнать из их `man`-страниц (`man функция`). ◀

Выполните задание 3:

Составьте программу, запрашивающую у пользователя 2 числа и сообщающую, какое из них больше, или же они равны.

Для этого нам понадобится еще и изучить условный оператор, выполняющий определенные действия при выполнении/невыполнении какого-либо условия.

В C структура условного оператора выглядит так:

```
if (Условие) действия1;
else действия2;
```

Если `Условие` истинно, выполняются `действия1` (если их много, следует заключить их в блок `{}`), иначе выполняются `действия2`.

Выполните задание 4:

Составьте блок-схему алгоритма программы. Если возникнут затруднения, внимательно просмотрите ее текст (ниже).

Теперь создайте файл `if.c` и наберите в нем текст программы.

```
#include <stdlib.h>
int a,b; // Резервируем память для переменных

int main(){
    printf("Введите значения a и b (через пробел)\n");
    scanf("%d %d",&a,&b); // Запрашиваем данные
    if(a>b) printf("a больше b\n"); // И проверяем условие
    else{
        if(a<b) printf("b больше a\n");
        else printf("a и b равны\n");}
    return 0;
}
```

Обратите внимание на знак `&` перед именем переменной. Он означает, что функции передается значение адреса этой переменной, а не ее копия. Если бы

мы передавали в функцию саму переменную, то введенное пользователем значение останется внутри функции, а сами переменные не изменятся.

Откомпилируйте файл и запустите его. Проверьте, как он работает при различных вариантах введенных значений.

15.2.3 Префиксные и постфиксные операции. Циклы.

Допустим, у нас есть какой-то счетчик i и нам надо увеличивать его на 1 каждый раз, когда происходит какое-то событие. Можно, конечно, записать $i = i + 1$; , но более наглядно (да и короче) было бы написать, используя оператор **инкремента** $++$: $++i$; . Кроме того, существует еще оператор **декремента** $--$, уменьшающий на 1 значение счетчика: $--i$; .

Для сокращения записей можно использовать операторы $+=$, $-=$, $*=$, $/=$ и $%=$. Запись $x @= y$ означает то же самое, что $x = x @ y$, где знак $@$ — одна из операций $+ - * / \%$. Оператор $%$ возвращает остаток от деления первого числа на второе.

Операторы инкремента и декремента можно ставить как перед переменной (**префикс**), так и после нее (**постфикс**). Важно понять разницу в префиксных и постфиксных операциях. Для этого напишем программу, выдающую числа и результаты префиксных и постфиксных операций с ними. Назовите ее `prefpost.c`. Разберитесь с текстом программы самостоятельно.

```
#include <stdlib.h>
int x,y;

int main(){
    x = 1; y = 1;
    printf("x = %d, y = %d\n", x, y);           // x=1, y=1
    printf("++x = %d, y++ = %d\n", ++x, y++); // x=2, y=1
    printf("x = %d, y = %d\n", x, y);         // x=2, y=2
    printf("--x = %d, y-- = %d\n", --x, y--); // x=1, y=2
    printf("x = %d, y = %d\n", x, y);         // x=1, y=1
    return 0;
}
```

|| Выполните задание 5:

Наберите программу, откомпилируйте ее и посмотрите результат выполнения.

- Здесь вы заметили, что `printf` имеет такой же формат, как и `scanf`, с той лишь разницей, что первая функция выводит данные на дисплей, а вторая — считывает их с него. ◀

По результатам программы видно, что префиксная операция выполняется

сразу же, а постфиксная только после выполнения определенных действий с этой переменной.

Циклы. Их предназначение — в многократном выполнении однообразных действий вплоть до выполнения определенных условий.

Для выполнения многократных вычислений с изменяющейся с определенным шагом переменной используется цикл `for`, а для действий, завершающихся лишь после выполнения какого-либо условия — цикл `while`.

Цикл `for` имеет следующий вид: `for (Нач.действия; Усл.; Операция){ Тело цикла }`, где **Нач.действия** — действия, которые выполняются при самом первом запуске цикла, чаще всего это присваивание переменной цикла определенного значения; **Усл.** — условие, которое проверяется каждый раз в конце цикла, если оно не выполняется, цикл завершается; **Операция** — действия, производящиеся каждый раз после невыполнения условия **Усл.**; **Тело цикла** — действия, выполняющиеся в цикле. Любой из параметров в скобках цикла может отсутствовать (даже одновременно все три).

Выполните задание 6:

Создайте программу, которая будет вычислять факториал числа n , введенного пользователем.

► Факториал n ($n!$) — произведение всех чисел от 1 до n . ◀

Первый вариант `factor1.c` создадим по всем правилам организации цикла, т.е. вначале запросим пользователя ввести число n , затем организуем цикл от 1 до n , в котором будем умножать переменную-ответ на счетчик цикла.

Выполните задание 7:

Составьте блок-схему программы.

Вы на собственном опыте убедитесь, что для одной и той же блок-схемы можно создать множество реализаций программы.

Выполните задание 8:

Наберите, откомпилируйте и посмотрите результаты выполнения всех вариантов программы.

Итак, первый вариант:

```
#include <stdio.h>
int n;

long factor(int n){
    int i;
```

```
    long f = 1;
    for(i = 2; i <= n; i++) f *= i;
    return f;
}

int main(){
    printf("Введите значение n\n");
    scanf("%d", &n);
    printf("Факториал %d равен %d\n", n, factor(n));
    return 0;
}
```

Здесь вводится тип данных `long`, позволяющий вместить большее число, чем тип `int`. Факториал — быстрорастущая функция, поэтому уже при $n = 32$ наша функция вернет неверное значение из-за того, что ответ будет больше, чем может вместиться в тип `long`.

Запустите программу и посмотрите, как она работает.

Теперь модифицируем нашу функцию `factor`. Скопируйте файл `factor1.c` в `factor2.c` командой `cp factor1.c factor2.c` или же сохраните его в текстовом редакторе под новым именем.

Сделаем так, чтобы факториал вычислялся в теле функции. Воспользуемся тем, что в разделе Операция цикла `for` можно писать любую функцию.

```
long factor(int n){
    long f = 1;
    for(; n > 0; f *= i--);
    return f;
}
```

Теперь мы воспользовались тем, что в функцию передается копия аргумента, т.е. внутри нее мы можем делать с ним все, что угодно.

Цикл `for(; n > 0;)` эквивалентен циклу `while(n > 0)`. Сделаем еще одну модификацию, использующую этот цикл:

```
long factor(int n){
    long f=1;
    while(n > 0) f *= i--;
    return f;
}
```

- Досрочно можно выйти из цикла командой `break`, а заставить его перейти к началу — командой `continue`. Зная английский язык легко можно разобраться с программами. ◀

Выполните задание 9:

Напишите программу, которая будет считать сумму квадратов первых n чисел. Сначала составьте ее блок-схему. Подумайте, сколько переменных понадобится для программы. Не забывайте, что нужно по крайней мере две переменных: для хранения введенного числа n и результата вычисления. Подумайте, нужна ли отдельная переменная для хранения счетчика цикла (посмотрите предыдущие примеры).

- ▶ Для вычисления степени числа используется оператор \wedge . Например, чтобы вычислить x^2 , необходимо записать $x\hat{2}$. Кроме того, для небольших степеней можно использовать вместо оператора возведения в степень многократное умножение ($x*x$), работающее быстрее. ◀
- ▶ Обратите внимание на представление целых чисел в памяти компьютера. Если результат вычисления будет очень большим (например, когда вы введете 10000, на экране отобразится отрицательное число. Это не ошибка. Просто старший бит числа отражает его знак, а для хранения большого числа может понадобиться заполнить знаковый бит. Чтобы результат отображался всегда правильно, необходимо указать, что данная переменная беззнаковая, указав в ее определении `unsigned`, например, для беззнакового целого `unsigned int`, и для беззнакового длинного целого — `unsigned long`. ◀

15.2.4 Операции с файлами.

Для операций с файлами существует структура `FILE`. Команда `fopen(char* n, char* m)` (n — имя файла — указатель на символьную строку, m — режим доступа к файлу — см. `man fopen`) возвращает указатель на эту структуру, который будет использоваться в командах ввода/вывода `fscanf` и `fprintf`. Синтаксис последних команд аналогичен `scanf` и `printf` за исключением того, что первым их аргументом идет указатель на структуру `FILE`.

- ▶ **Указатель** — это адрес переменной в памяти компьютера. С указателями следует обращаться осторожно, чтобы не уйти за пределы дозволенной области. Указатели очень удобно передавать вместо аргументов функций, если они занимают много места в памяти, т.к. на аргумент выделяется столько места, сколько он занимает, а указатель требует всего 2-4 байта. Чтобы получить по указателю содержимое данной величины, т.е. для **разыменования указателя**, используется оператор `*`, а чтобы получить адрес переменной — оператор `&`. ◀

В этом задании вы научитесь совершать простые операции с файлами, кроме того, узнаете как передавать данные в программу посредством аргументов

командной строки.

Функцию `main` мы всегда записывали, опуская аргументы. На самом деле, полный ее формат — `int main(int argc, void **argv)`, где `argc` — количество аргументов функции, `**argv` — указатель на указатель на структуру, хранящую аргументы командной строки. Доступ к указателю можно получить как через элемент массива (`char* name = argv[2]`), так и путем разыменования указателя (`char* name = *argv++`). Тип `void*` (указатель на ничто) необходимо явным образом преобразовать к любому необходимому типу данных, указав, например, `int main(int argc, char **argv)`.

В нашей программе будет использоваться **массив** — индексируемый набор данных одного типа. В нем будет храниться имя файла, с которым будет работать программа. Объявление `char name[128]` означает, что в памяти выделится блок в 128 байт (т.к. `char` занимает 1 байт), а его имя является указателем на первый элемент (`name[0]`). Обратите внимание, что нумерация массивов начинается с 0, а не с 1.

Итак, напишем программу, которая будет открывать файл (имя которого можно ввести как первый аргумент программы), записывать в него оставшиеся аргументы (если таковые есть), затем спрашивать у пользователя радиус круга и рисовать этот круг в том же файле.

После работы с файлом его обязательно нужно закрыть при помощи функции `fclose`. Вообще-то, при выходе из программы будет очищена вся память и закрыты все ее файлы, но лучше закрывать их явным образом, чтобы в них не попал какой-нибудь «мусор».

Выполните задание 10:

Наберите данную программу (назовите ее `file.c`). Т.к. программа использует математическую функцию `sqrt`, возвращающую квадратный корень ее аргумента, необходимо сообщить компилятору, что объявление этой функции находится в файле `math.h`. Кроме того, этот файл не входит в стандартную библиотеку, поэтому компилятору нужно указать явно, что необходимо подключить математическую библиотеку ключом `-lm` (ключ пишется до имени файла с программой).

```
#include <stdio.h>
#include <math.h>

FILE *f;
char name[128];

void printstar(int r, int n){
    int i;
    fprintf(f, " ");
    for (i = 0; i < n; i++) fprintf(f, " ");
```

```

    if (n != r){
        fprintf(f, "*");
        for (i = 2*(r-n)-1; i > 0; i--) fprintf(f, " ");
    }
    fprintf(f, "*\n");
}

int main(char argc, char** argv){
    if (argc < 2){
        printf("Введите имя файла для записи\n");
        scanf("%s", name);
    }
    else strcpy(name, argv[1]);
    if ((f = fopen(name, "w")) == NULL){
        printf("Ошибка открытия файла\n");
        return -1;
    }
    fprintf(f, "Мой файл:\n");
    int a = 1;
    while (++a < argc) fprintf(f, "%s\n", argv[a]);

    unsigned char r;
    printf("Введите радиус круга\n");
    scanf("%d",&r);
    for(a = r; a > 0; a--) printstar(r, r - sqrt((double)(r*r - a*a)));
    for(a = 0; a <= r; a++ ) printstar(r, r - sqrt((double)(r*r - a*a)));
    fclose(f);
    return 0;
}

```

Скомпилируйте программу (`gcc -lm file.c -o file`). Запустите ее сначала без аргументов (`./file`), а затем — с несколькими аргументами (`./file myfile arg1 arg2 arg3`). Посмотрите выходной файл (в первом случае он будет носить имя, которое вы ему дадите в диалоге с программой, а во втором его именем будет первый аргумент командной строки).

Ответьте на вопрос(ы):

- Почему «круг» вытянут вдоль вертикальной оси?
- Почему для присваивания имени `name` первого аргумента командной строки использовалась функция `strcpy`, побуквенно копирующая строки, а не более короткая запись `name=argv[1]`? (подсказка: `name` — статический указатель, жестко привязанный к массиву.)
- Что будет, если в функции `printstar` заменить `fprintf(f, ...)` на `printf`?
- Какова роль функции `printstar`?

Выполните задание 11:

Теперь скопируйте файл с текстом программы в файл `file1.c`, откройте этот файл в текстовом редакторе, и измените его так, чтобы функция печатала в файл сумму квадратов целых чисел от 1 до n , беря значением n первый аргумент командной строки, а именем файла — второй аргумент. Если же программа была запущена без аргументов, число n и имя файла следует запросить у пользователя.

Для получения числа используется функция `atoi` (напр., `int i = atoi(argv[1])`).

16 Программирование на С. Создание интерактивных консольных программ.

Теперь, когда мы научились составлять простые программы, попробуем более сложные. Для взаимодействия пользователя и компьютера используется множество элементов. Самый простой способ — консольный ввод/вывод: программа запрашивает у пользователя необходимые данные, которые тот набирает на клавиатуре. Более сложный — графический. Для запуска графических программ требуется графическая оболочка (X), а также подключение к программе графических библиотек. Программирование графических программ несколько сложнее, чем консольных.

16.1 Простейшие интерактивные программы.

16.1.1 Приветствие пользователя.

Выполните задание 1:

Создайте программу, запрашивающую у пользователя его имя и приветствующую его по имени. Для начала составьте блок-схему программы, после этого — приступайте к ее написанию.

Чтобы упростить себе работу, используйте готовые блоки из предыдущих программ.

- ▶ Редактировать несколько файлов сразу можно при помощи редактора `gedit`: зайдите в свою рабочую папку с файлами и наберите в командной строке `gedit *.c`. Откроется редактор, в котором каждый файл можно редактировать, активировав соответствующую вкладку. ◀

Итак, для имени вам понадобится символьный массив длиной около 128 сим-

волов, ввод имени будет осуществляться при помощи команды `scanf`, а вывод данных — `printf`.

16.1.2 Приветствие пользователя 2.

Наверное, вы заметили, что при вводе имени при помощи функции `scanf` ввод производится до первого пробела, т.е. фамилию, имя и отчество так ввести не получится. Функция `fgets` позволяет вводить данные до символа «конец файла» или символа новой строки (`enter`). Формат команды таков: `fgets(char* BUFFER, int NUM, FILE *NAME)`, где `BUFFER` — указатель на символьный массив (его длина должна быть не меньше `NUM`), `NUM` — максимальное количество введенных символов, `NAME` — указатель на файл, с которого следует производить считывание.

В Linux при запуске программы сразу открывается три файла: `stdin` (ввод), `stdout` (вывод) и `stderr` (поток ошибок). Для ввода имени вам необходимо считывать с файла `stdin`, который и надо будет указать третьим аргументом. Итак, необходимо будет строку считывания имени заменить на `fgets(name, sizeof name, stdin)`, если символьный массив вы назвали `name`.

Выполните задание 2:

Исправьте программу, как предложено выше.

16.2 Файловые операции.

Выполните задание 3:

Создайте программу, позволяющую составлять «картотеку» пользователей, куда будет заноситься их фамилия, имя и дата рождения.

В начале программы объявите переменные для хранения фамилии, имени и даты рождения, например, так: `char firstname[32], lastname[32], date[32]`. Не забудьте подключить библиотеку стандартного ввода/вывода `stdio.h`. Еще нам понадобится переменная-символ, в которой будет храниться введенная команда (`char command`) и указатель на файл-базу данных (`FILE* file`).

Далее, нам понадобятся две функции: распечатывающая весь список (`void printcard()`) и добавляющая пользователя (`void newuser`).

Функция, добавляющая пользователя, должна запросить ввести его фамилию, имя и дату рождения, а затем оператором `fprintf` записать эти данные в файл-базу данных.

Функция, печатающая «картотеку», должна для начала «перемотать» файл-базу данных на начало (функция `rewind(file)`), затем считать из него первую

запись (функция `fscanf`) и в цикле печатать/считывать информацию, вплоть до достижения конца файла. Здесь удобнее использовать цикл `do ... while`:

```
do printf("%s\t%s\t%s\n", firstname, lastname, birthdate);
while(fscanf(file,"%s %s %s\n", firstname, lastname, birthdate)!=EOF);
```

Этот цикл сначала выполняет свое тело, а потом проверяет условие. Нам необходимо считать данные вплоть до конца файла, для этого в операторе проверки условия выполнения цикла `while` мы считываем следующую запись. Если предыдущая запись — последняя, функция `fscanf` возвращает признак конца файла `EOF`, тогда цикл прекращается.

Теперь дошла очередь и до основной программы. Во-первых, откроем файл базы данных (назовите его как угодно, в данном случае откроется файл `card`). Так как при первом запуске файла может не существовать, необходимо, чтобы в этом случае он создавался. Если же запуск не первый, файл должен открыться для дописывания данных в его конец. Для этого используем функцию `open` со вторым аргументом `"a+"` — открыть для дописи и чтения: `file = fopen("card", "a+")`;

Теперь создадим цикл считывания символа и проверки, является ли введенный символ командой. Этот цикл продолжается до тех пор, пока не будет введено `q`. Для такого бесконечного цикла удобно использовать скобки `do ... while`.

Проверку введенной команды проще осуществлять при помощи функции `switch`, за которой следует последовательность вариантов `case ... : ... ; break`. Если аргумент функции совпадает с аргументом какого-либо `case`, выполняются все действия после этого `case` вплоть до оператора `break`, выходящего из оператора проверки.

Итак, последний участок функции `main` можно оформить так:

```
do{
    printf("Введите команду\n(n - новый пользователь,
    p - отобразить картотеку, q - выход):\n");
    command = getchar();
    switch(command){
        case 'n': newuser(); break;
        case 'p': printcard(); break;
        case 'q': fclose(file); exit(0);
    }
}
while(true)
```

Выполните задание 4:

Составьте блок-схему данной программы, наберите ее текст, откомпилируйте и запустите. Создайте несколько записей, проверьте реакцию программы на ввод посторонних символов, введите команду «р» и посмотрите, как выглядят все записи.

Выйдите из программы, посмотрите содержимое файла, в котором хранилась ваша база данных.

Список рекомендуемой литературы

1. А. Богатырев. Хрестоматия по программированию на Си в Unix.// 1992–1995. *Учебник по программированию, доступный для скачивания в сети Internet.*
2. А. Ионов и др. OpenOffice.org Руководство пользователя.// www.openoffice.ru *Первая попытка написать учебник по Open Office на русском языке.*
3. Б.В. Керниган, Д.М. Ричи. Язык C.// (множество различных изданий) *Классический учебник по языку C.*
4. В. Костромин. Linux для пользователя.// БХВ-Петербург, 2002. *Полезная книга для изучения основ работы в Linux.*
5. S. Figgins, E. Siever, A. Weber. Linux in a Nutshell.//O'Reilly, 2003. *Основы работы в Linux.*
6. The Linux Bible.// Yggdrasil Computing Incorporated, 1994–1995. *Наиболее полный сборник вопросов и ответов по Linux.*
7. Linux Client Migration CookBook.// IBM.com – Redbooks, 2004. *Книга, помогающая полностью перейти на Linux с альтернативных операционных систем.*
8. Linux documentation project.// www.linuxdoc.org *Проект, посвященный документации, касающейся Linux и программного обеспечения для этой операционной системы.*

Сайты, посвященные Linux:

1. www.asplinux.ru. Сайт разработчиков российского дистрибутива ASP Linux.
2. firststeps.ru. «Первые шаги» — самоучители разных языков программирования и не только.
3. www.linuxcenter.ru. Здесь можно найти документацию, ПО для Linux, а также заказать товары почтой.
4. www.linux.org. Сайт, всецело посвященный Linux.

Много разнообразной литературы, посвященной Linux, можно заказать почтой в агентстве «Питер» (www.piter.com), а также найти в Internet (пользуйтесь поисковыми системами Google, Yandex, Aport, AltaVista, FileSearch, а также пиринговыми сетями).