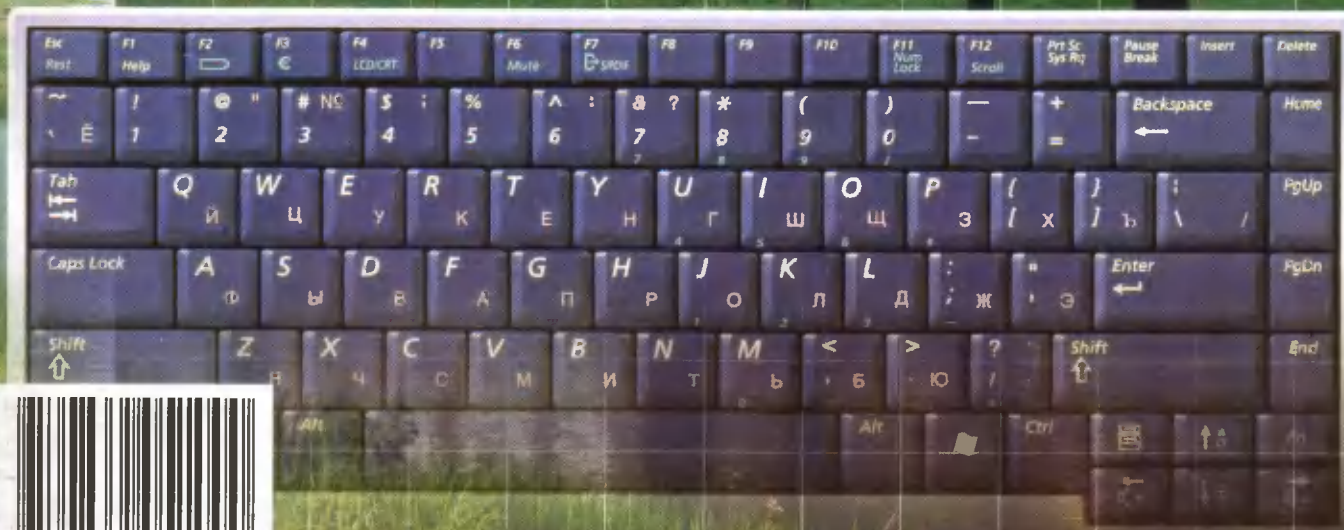


radio mir

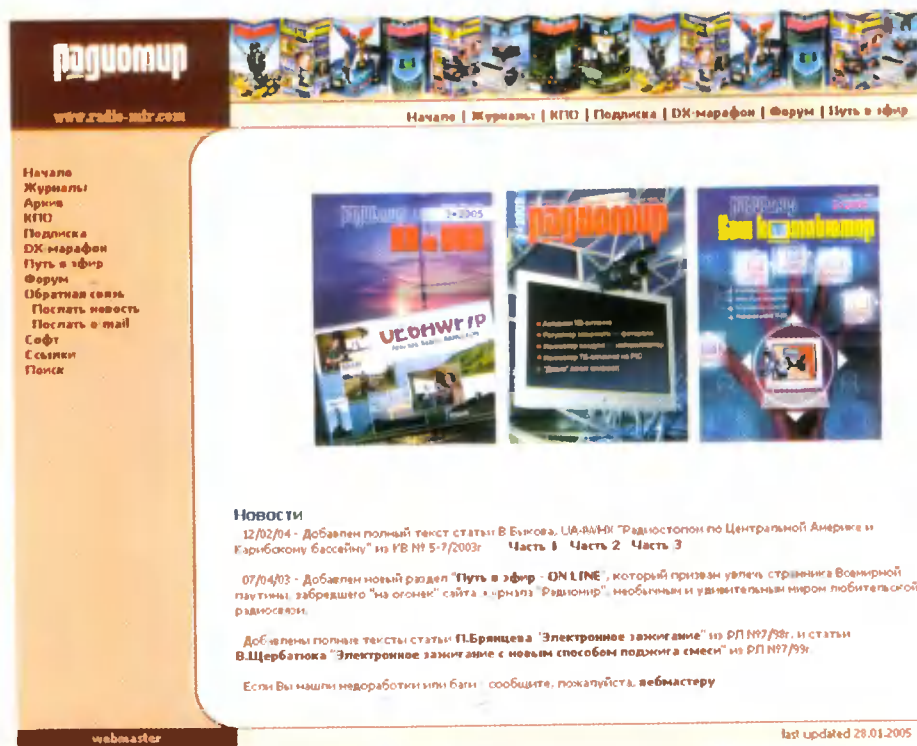
5 • 2005

Ваш компьютер

- Слово о словаре
- Шаманские пляски вокруг CD-ROM
- Разговаривая с оппонентом — всегда улыбайся
- Структурные исключения в C++ Builder
- Работа с консольными приложениями в Visual C++ 6.0



Официальный сайт журнала "Радиомир"



Журналы

Аннотированное содержание журналов за предыдущие годы, начиная с 2000 г.

Архив

Содержание журналов за 1998 — 2000 г.г.

КПО

Наиболее свежие объявления о купле-продаже радиодеталей, бытовой и радиолюбительской радиоаппаратуры, опубликованные в журнале "Радиомир"

Подписка

Данные о наличии и цене журналов при заказе через редакцию

DX-марафон

Таблица достижений коротковолновиков ex USSR по числу подтвержденных стран по списку диплома DXCC

Форум

Доска объявлений для обмена информацией между радиолюбителями

Путь в эфир

Избранные статьи для начинающих радиолюбителей

Обратная связь

Наиболее простой способ поделиться своим мнением или любопытной новостью с редакцией

Поиск

Система поиска поможет Вам в поиске информации о нужной публикации

Учредитель и издатель: ООО "Радиомир Пресс"

Свидетельство о регистрации
ПИ №77-9841 от 17.09.2001

Главный редактор
Ольга СТРИЖАНКОВА

Зам. гл. редактора
Иван БЕЛЬСКИЙ

Редколлегия:
Сергей ДРОЗДОВСКИЙ,
Владимир КУЦЕНКО,
Елена ЛЕВИТМАН,
Геннадий ПЕЧЕНЬ,
Геннадий ШУЛЬГИН

Контактные телефоны:
в Москве (095) 105-99-89,
в Минске (017) 249-41-47

Компьютерная верстка —
Янина БЕЛЬСКАЯ

Техническая графика —
Наталья БЕЛЬСКАЯ,
Мария КАЛАБУХОВА

Оформление обложки —
Наталья БЕЛЬСКАЯ,
Надежда БОГОМОЛОВА

Адреса для писем:
119454, РФ, г.Москва, а/я 37,
факс (095) 432-62-04;
220095, РБ, г.Минск-95, а/я 199

E-mail: rl@radiopage.by
rm@radio-mir.com
WWW: <http://radio-mir.com>

Наши платежные реквизиты:
получатель: ООО "Радиомир Пресс",
ИНН 7729404741, КПП 772901001,
р/с 40702810900010000084
в ООО КБ "Агропромкредит"
ф-л Центральный, г.Москва,
корр. счет 3010181050000000109
БИК 044525109
Адрес банка: 125315, г.Москва,
Ленинградский пр., дом 76, корп. 4

Материалы для публикации принимаются
в рукописном, печатном и электронном
вариантах

Требования к графическим материалам рек-
ламного характера в электронном виде:
CorelDRAW до 10.0, все шрифты в кривых;
bitmaps 300 dpi; TIFF 300 dpi; CMYK.
Приложить печатную копию.

За достоверность рекламной и другой публи-
куемой информации несут ответственность
рекламодатели и авторы. Мнение редакции
не всегда совпадает с мнениями авторов

Адрес редакции: 119454, РФ, г.Москва,
ул.Коштовяна, 6 — 233, тел. (095) 105-99-89

Подписано к печати 25.03.2005 г.
Формат 60 x 84 1/8.
Печать офсетная, 6 печ. л.
Цена свободная.

© ООО "Радиомир Пресс". Воспроизведение матери-
алов журнала в любом виде без письменного разре-
шения редакции запрещено. При цитировании ссылка
на "Радиомир. Ваш компьютер" обязательна.

Отпечатано в типографии
ООО "Красногорская типография".
Заказ № 839

радиомир Ваш компьютер

ЕЖЕМЕСЯЧНЫЙ МАССОВЫЙ ЖУРНАЛ
С 1995 г. по 6/2001 г. издавался под названием
"Радиолобитель. Ваш компьютер"

№5/май/2005

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

НЕ ТОЛЬКО НОВИЧКУ

А.ГРИНЧУК. Windows XP для пользователя Файлы и папки	2
Е.ЗАЙЦЕВА. Эффективная работа с Microsoft Word XP Вставка и управление рисунками. Печать документа	4
Е.ШАКЕЛЬ. Эффективная работа с табличным процессором Microsoft Excel XP Основные приемы форматирования	8
С.ГРИНЧУК. Разработка web-сайтов в Microsoft Office FrontPage 2003 Вставка и управление параметрами встроенных фреймов	11
В.КУЦ. Слово о словаре	13
Б.КИСЕЛЕВ. MATLAB. Решение обыкновенных дифференциальных уравнений	16
С.РЮМИК. Интернет-калейдоскоп, freeware #18	20

КОММУНИКАЦИИ

А.КАШКАРОВ. Разговаривая с оппонентом — всегда улыбайся.....	22
--	----

ДАЙДЖЕСТ

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

О.ВАЛЬПА. Borland C++ Builder 6 для начинающих	27
А.КОРБИТ. Интегрированная среда VISUAL C++ 6.0 и работа с консольными приложениями	30

ДИАЛОГ ПРОГРАММИСТОВ

В.КАРЦЕВ. Исключения C++ и структурные исключения C в C++ Builder	33
CyberManiac / HI-TECH. Теоретические основы крэкинга	35
Е.ЛЕБЕДЕВ. Знакомство с WinAsm Studio	38

РЕЦЕПТЫ

Р.КАРПАЧ. Шаманские пляски вокруг CD-ROM TEAC CD-55a	41
И.РОЩИН. Онлайн-распаковка архивов	43

МИР 8 БИТ

Возвращаясь к напечатанному №12/2000, С.36; №9/2001, С.37. И.РОЩИН. Расширения файлов TR-DOS	44
--	----

ИГРОТЕКА

И.ШИРКО. TimeZero — online-игра нового поколения	46
--	----

ДОСКА ОБЪЯВЛЕНИЙ

Куплю, продам, обменяю	48
------------------------------	----

Полные листинги некоторых программ расположены по адресу
<http://radio-mir.com>

А.ГРИНЧУК,
E-mail: gav@nihe.niks.by

Windows XP для пользователя

Файлы и папки

Следует помнить, что любая ошибка при работе с Windows XP может стать последней. По крайней мере, пока не сделаете следующую.

Из неофициального руководства пользователя Windows XP

Упорядоченное размещение файлов и папок требует некоторого времени и усилий, но все это с лихвой окупается в дальнейшей работе. Также не нуждается в доказательстве и другой факт — ошибки неизбежны. Например, при создании папки достаточно не вовремя щелкнуть мышкой,

одного правила — щелкать по желтым значкам папок, а не по текстам с именем. А все потому, что можно случайно войти в режим переименования. Проверим это на эксперименте. Двойной щелчок по значку папки (в области файлов и папок) должен привести к открытию папки. А щелчки по имени

ла. Например, документ, созданный в Microsoft Word, будет называться "Отчет.doc", созданный в блокноте — "привет.txt", а графический файл может носить имя "рисунок.bmp". Первая часть ("Отчет") — это собственно имя файла, после точки находится расширение ("doc" — чаще всего трехбуквенное). Проводник можно настроить на разные режимы работы, с отображением или без отображения расширения. Изменить настройки можно, выбрав в меню

Сервис пункт Свойства папки, в открывшемся окне (рис.3) щелчком мыши указывается, скрывать или нет расширения для зарегистрированных (известных системе) типов файлов. Если расширения оставить, то Проводник будет давать более полную информацию о документах, но процесс переименования немного усложнится — придется следить за неприкосновенностью расширения. Выглядит это следующим образом:

- выделяется файл;
- **Файл/Переименовать**;
- курсор мыши подводит-

Рис. 1

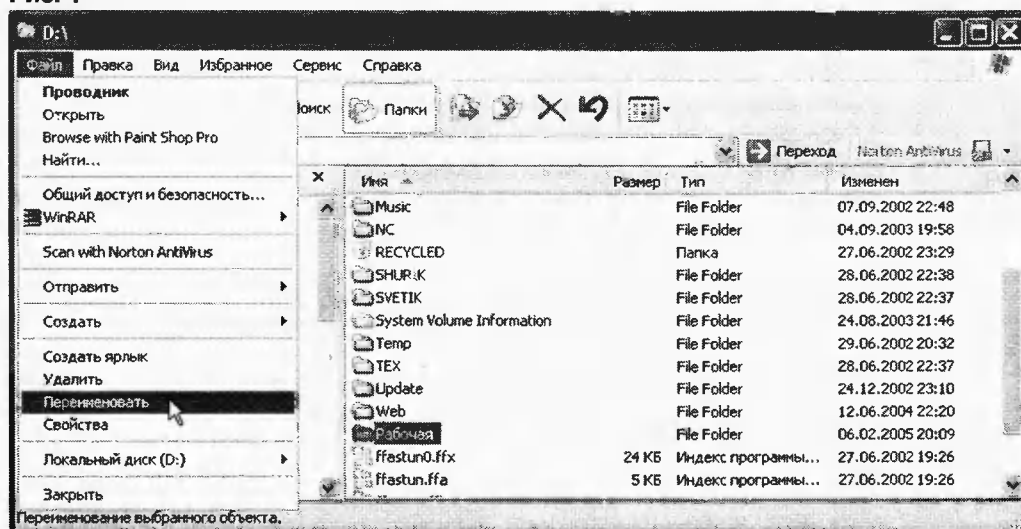


Рис. 2



и папка приобретет имя "Новая папка". Другой распространенный источник ошибок — сохранение файла "не по адресу". В общем, если ошибок избежать не удастся, необходимо научиться их исправлять.

Начнем с переименования папки. Для примера изменим имя нашей папки "Рабочая" на "Учебная". В Проводнике ее необходимо найти и выделить одиночным щелчком мыши. Обязательно следует проверить раскладку клавиатуры — убедитесь, что вы переключились на русский язык. А затем в Проводнике в меню "Файл" выбирается команда "Переименовать" (рис.1). После этого имя папки оказывается выделенным и окруженным рамкой (рис.2). Никакие дополнительные манипуляции с мышкой не требуются — достаточно нажать клавишу **Delete** и ввести новое имя. При работе с Проводником стоит придерживаться еще

папки, да еще с паузой между ними, приведут к режиму переименования. Снова нажимаем **Delete** и вводим имя "Рабочая". Возможно, такой способ переименования вам даже понравится, но надежнее первый вариант:

- выделить папку;
- **Файл/Переименовать**;
- удалить старое имя (**Delete**);
- ввести новое.

У операции переименования файлов есть некоторые нюансы. Файл, кроме имени, имеет еще одну характеристику — расширение имени файла (или просто расширение). Фактически, это информация о типе фай-

Рис. 3

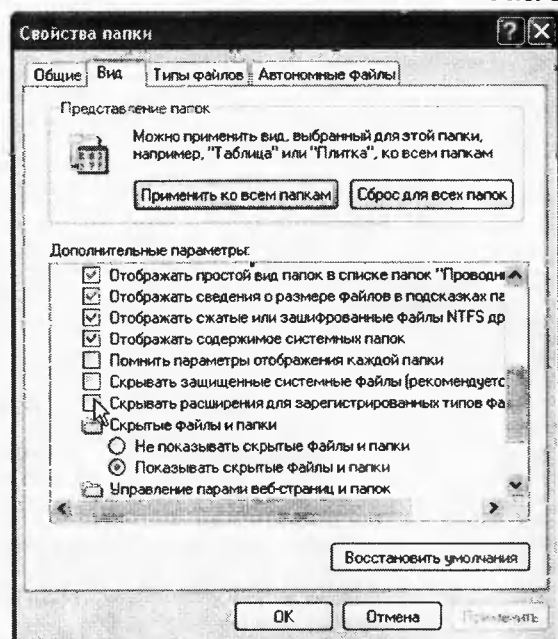
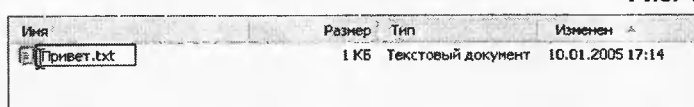


Рис. 4



ся к началу имени файла, производит щелчок левой клавишей (рис.4) и нажатиями на клавишу **Delete** удаляется имя файла до точки с расширением;

- вводится новое имя.

Итак, файлы и папки приобрели нужные нам имена. Следующий шаг — размещение нужных файлов по нужным папкам. Необходимо различать две операции: копирование и перемещение файлов. При копировании с файла снимается копия и размещается в новом месте. В итоге получаются два одинаковых файла: исходный и копия. Перемещение же файла есть просто перемещение: в исходном месте исчезает, в конечном — появляется. В результате остаемся по-прежнему с одним файлом. Путаница возникает потому, что эти две операции уж очень похожи в исполнении. Принцип один — указываем Проводнику объект, говорим, что с ним нужно сделать (копировать или вырезать), и по-

Перемещение файла отличается только в одном пункте — вместо команды копирования файла используется команда **Вырезать**. Переместим наш многострадальный файл **привет.txt** из папки **D:\Рабочая** прямо на диск **D**. Пусть вас не смущает тот факт, что работать будем напрямую с диском, при копировании файлов алгоритм действий от этого не зависит:

- открывается папка с файлом и выделяется файл (**D:\Рабочая** и **привет.txt**);

- **Правка/Вырезать**;

- открывается папка назначения (**D:**);

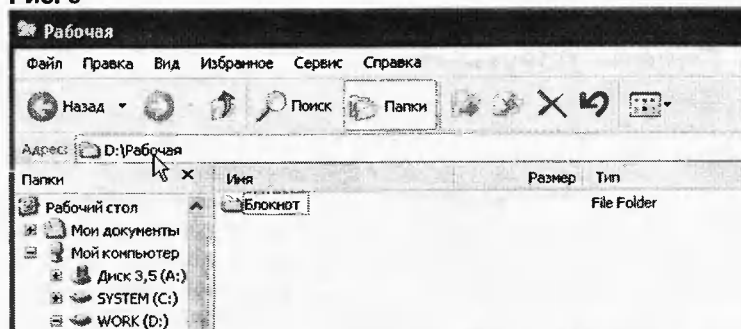
- **Правка/Вставить**.

А чтобы пользователю не приходилось мучительно вспоминать, какую же из команд он недавно выбрал, при вырезании файла дается дополнительная подсказка. Значок вырезаемого файла становится полупрозрачным (рис.6) — полупрозрачный нарек на то, что после завершения всех операций файла в папке **D:\Рабочая** не останется.

щелчком уже папка, а все остальное — один к одному.

Копируем ли мы файл или папку, мы все равно работаем только с одним объектом. Возникает другой вопрос — как же применить изученные операции к группе объектов? Опять-таки, компания Microsoft позаботилась, чтобы пользователю пришлось изучать как можно меньше новых операций. Алгоритм тот же — выделяются объекты, выбирается операция, открывается папка назначения и выполняется вставка. Значит, проблема в одном — выделить сразу несколько объектов. Простые щелчки мышкой не подходят — когда производится щелчок по второму объекту, с первого выделение снимается. Есть два варианта действий. Во-первых, при щелчках удерживать нажатой клавишу **Ctrl**. Тогда выделения с файлов сниматься не будут, и этот вариант больше подходит, когда требуется переместить или скопировать несколько расположенных не подряд объектов. Без оговорок не обойтись — щелчки мышкой должны быть аккуратными, без движения мышки во время щелчка (рис.7). Иначе можно получить никому не нужный набор копий файлов и папок. Во-вторых, если стоит задача выделить группу файлов от и до, то стоит воспользоваться клавишей **Shift**. Тогда выделяется первый файл, нажимается и удерживается клавиша **Shift**, производится щелчок по последнему из нуж-

Рис. 5



казываем место назначения. Ваш компьютер выступает просто в роли глуповатого помощника, которому приходится все подробно объяснять.

Перейдем от общей информации к конкретной реализации, и с этой целью скопируем файл **привет.txt** из папки **D:\Рабочая\Блокнот** в папку **D:\Рабочая**. Для этого:

- открываем папку с файлом и одиночным щелчком выделяем файл (соответственно — **D:\Рабочая\Блокнот** и **привет.txt**);

- **Правка/Копировать** (в меню Проводника);

- открываем папку назначения (**D:\Рабочая**);

- **Правка/Вставить**.

Все достаточно просто и логично, ошибка здесь может случиться только на третьем шаге — конечная папка должна быть именно открыта, а не просто выделена одиночным щелчком на области файлов и папок (рис.5). Остается проверить исходную папку и убедиться в том, что и там у нас осталась еще одна копия файла.

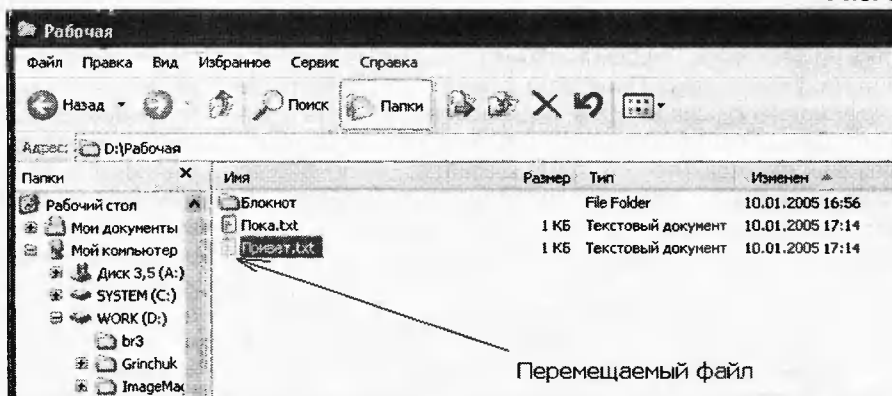


Рис. 7

Одно из достоинств Проводника — единообразное выполнение многих операций. Только что мы убедились, что в папку и прямо на диск файлы копируются совершенно одинаково. Можно предположить, что и папки копируются точно так же как и файлы. Предположение верное, только выделяется одиночным

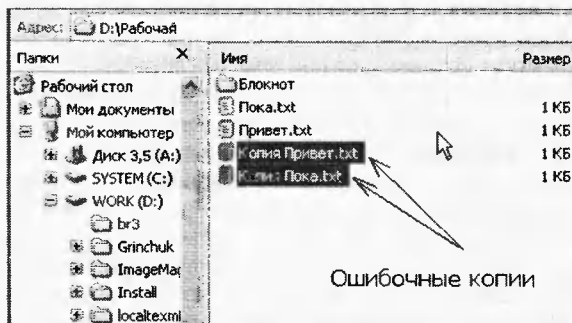
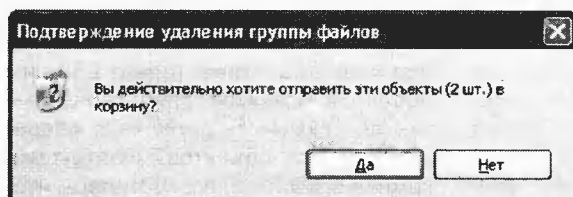


Рис. 8



ных файлов. Все находящиеся между ними объекты будут выделены, можно приступать к "оптовым" операциям.

Наши файлы оказались в нужных папках. По прошествии нескольких месяцев счет им пойдет на десятки-сотни. Возможно, часть из них придется удалить. Это проще всего — выделить и нажать клавишу **Delete**. Система дает шанс передумать, выводя предупреждающее сообщение (рис.8). Но и при

просмотре папок, и при удалении сначала нужно сориентироваться в общей массе. Для этого лучше перейти в режим таблицы (**Вид/Таблица**), а затем приступить к сортировке — **Вид/Упорядочить значки**. Варианты выбора:

- **имя** — файлы и папки сортируются по названиям. Сначала идет список папок, затем — файлов. И те, и другие расположены по алфавиту, причем латинские буквы имеют приоритет над кириллическими. Повторный выбор этой команды приведет к тому, что список отобразится в обратном порядке;

- **размер** — файлы (но не папки) будут отсортированы по размеру;

- **тип** — сортировка по типам файлов. Например, файлы, созданные в Блокноте или в MS Word, будут сгруппированы отдельно от остальных;

- **изменен** — сортировка по времени последнего изменения. Может оказаться очень полезной, если вы помните, что последний раз с этим документом вы работали где-то в начале февраля, и больше никакой информации. Приятно, что учитывается не только дата, но и время создания документа.

И файлы, и папки, естественно, хранятся на конкретных физических носителях информации. В следующей статье мы двинемся именно в этом направлении и займемся программой по обслуживанию дисков.

(Продолжение следует)

Эффективная работа с Microsoft Word XP

(Продолжение. Начало в №№1-4/2005)

Е.ЗАЙЦЕВА,

г.Минск, РИВШ БГУ,

E-mail: helena-zay@yandex.ru

Вставка и управление рисунками. Печать документа

Итак, мы почти готовы к распечатке нашего документа. Осталось навести окончательный лоск. Критически рассмотрим последний вариант, для чего выполним наши любимые процедуры:

1. **Запуск Word**. Без проблем!

2. **Загрузка файла Urok_4.doc**. С этим также справляемся без особых усилий. А теперь посмотрим, что можно сделать. Для начала, нужно удалить два из трех вариантов оформления списка "Класс 1...", "Класс 2...". Лучше оставить первый или второй. Затем следует завершить письмо логически. Введем заключительный абзац, набрав коротенькую фразу "Приятного отдыха!". Оформите ее полужирным курсивом. Итак, содержательная часть документа готова. Но если посмотреть на экран прищурившись, все еще кажется, что чего-то не хватает. Для солидности следовало бы снабдить письмо каким-либо изображением. Поэтому мы сейчас и приступим к изучению приемов вставки в документ рисунков из сопровождающей Word специальной галереи картинок.

3. **Вставка и управление рисунками в документе**

Несколько слов об изображениях в документах Word. Текстовый процессор предоставляет возможности встраивать, создавать и редактировать изображения в виде графических объектов или рисунков.

Графический объект — некое целостное изображение, чаще всего геометрическая фигура. Графические объекты могут создаваться при помощи встроенного в Word графического редактора средних возможностей, управление которым осуществляется посредством специальной панели **Рисование**. Такого рода работу мы обсудим позднее, здесь только заметим, что графические объекты могут располагаться произвольным образом, в том числе и в тексте документа, поверх него или на заднем плане.

Рисунок может содержать несколько графических объектов, объединенных общей композицией в одно целое. Рисунок обязательно требует для своего размещения отдель-

ной области прямоугольной формы. Эту область можно перемещать по документу, менять ее размеры, обрамлять.

Word допускает импорт в документ изображений различных графических форматов с применением фильтрации. Своими же силами текстовый процессор справляется с графикой растрового (bitmap — **.bmp**) и векторного (windows metafile — **.wmf**) форматов. В растровом формате изображение представляется совокупностью элементарных точек изображения — пикселей. Изображения такого формата выходят из-под пера графического редактора Paint. Недостатки растровых картинок происходят из их точечного характера — они могут искажаться при изменении масштаба изображения. В векторном формате изображение кодируется посредством математического описания его элементов некоторым вектором набора параметров. Векторную графику (**.wmf**) и растровую графику (**.bmp**) в рамках Word вы сможете сами и создавать, и импортировать. Импортирование изображений может выполняться в нескольких режимах. Можно заказать встраивание (хранение) рисунка в документе без связи с исходным местом его хранения, а также связывание (с хранением и без). Можно использовать коллекцию картинок Clip Gallery, присоединенную к пакету Microsoft Office, а также импортировать графическое изображение, хранящееся в файле, с несколькими вариантами его вставки в документ. На этом уроке мы рассмотрим только вставку картинок из коллекции, а технику работы с другими изображениями обсудим позже.

Сначала выберем место для вставки изображения. Это может быть, например, промежуток между "шапкой" с названием туристической фирмы и адресом клиента. Убедитесь, что у вас включен режим отображения непечатаемых символов. Тогда вы увидите, что в указанном месте у нас располагается пустой абзац. Для определенности сделаем там еще один пустой абзац, щелкнув мышкой перед маркером абзаца и нажав клавишу **Enter**. Установим курсор в первый пустой абзац. Для вставки изображения из

коллекции воспользуемся меню **Вставка/Рисунок/Картинки**. В результате рядом с вашим документом откроется вспомогательное диалоговое окно с заголовком **Вставка картинки** (рис.1). Чтобы убедиться в доступе к коллекции, раскроем список **Просматривать** и проверим наличие флажка у позиции **Коллекции Microsoft Office**. Если его там нет, установите щелчком мыши. Далее щелчком по гиперссылке синего цвета **Коллекция картинок** внизу окна **Вставка картинки**. Это приведет к разворачиванию отдельного окна **Избранное (коллекция картинок (Microsoft))**, где собраны все доступные изображения. Подвинув вниз полосу прокрутки левой вертикали окна, найдем папку **Коллекции Microsoft Office** и развернем ее содержимое щелчком по плюсику слева от названия. Теперь появятся папки с тематическими названиями, в которых представлены изображения по категориям на со-

Рис. 1

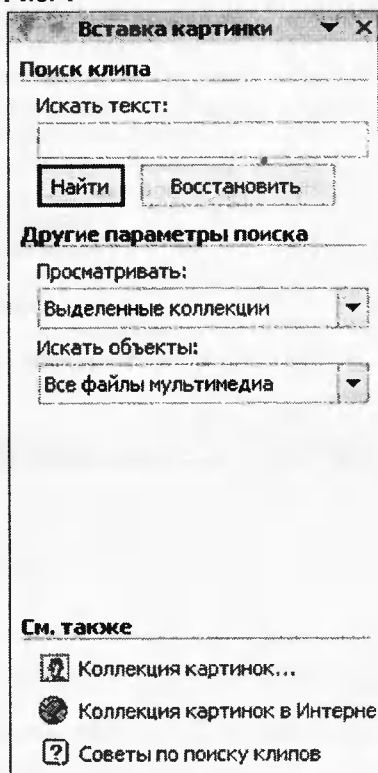
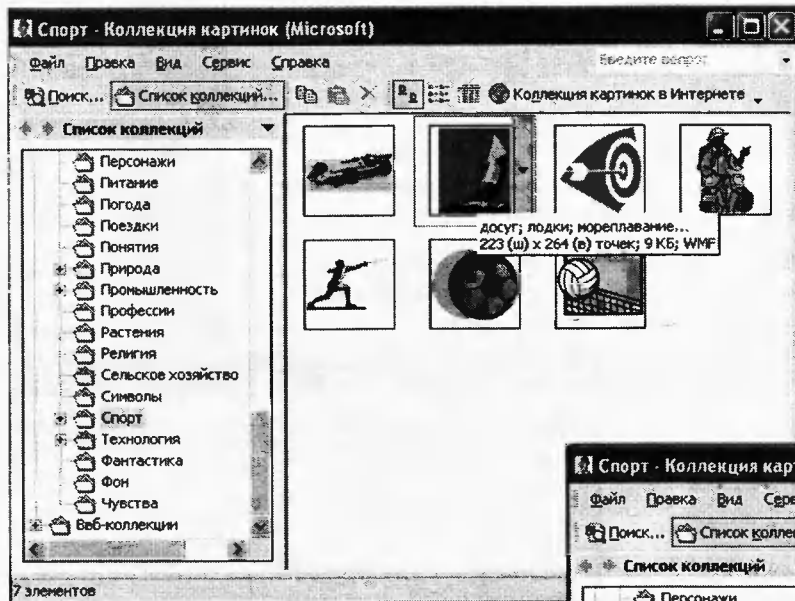


Рис. 2



ответствующую тему. Давайте докрутим список до категории **Спорт** (названия расположены в списке по алфавиту) и раскроем ее содержимое щелчком по значку в виде желтой папки левее названия (рис.2). Находящиеся там рисунки будут представлены справа от списка категорий в виде миниатюр. Наведя указатель мыши на миниатюру, например, изображение яхты, получим подсказку, включающую ключевые слова для поиска рисунка, его геометрические размеры, тип и размер в байтах соответствующего файла. Чтобы вставить картинку в документ, наведем мышью на серую панель с кнопкой раскрытия списка у правого края миниатюры, щелчком вызовем выпадающее

меню, в котором нужно выбрать команду **Копировать** (рис.3). Так мы отправим копию картинки в универсальный буфер обмена операционной системы. Теперь останется только внедрить ее в документ при помощи, например, инструмента **Вставить** или такой же команды меню **Правка**. Если других рисунков в документе не требуется, при помощи инструмента **Заккрыть** управления окнами закроем окно **Избранное (коллекция картинок (Microsoft))** и **Вставка картинки**.

А теперь займемся отделкой рисунка в документе. Для работы с рисунком его предварительно выделяют щелчком мышки где-нибудь внутри изображения. Признаками выделения служат маленькие квадратные маркеры на углах и сторонах прямоугольника границ рисунка. Сразу после вставки рисунка эти маркеры имеют контрастирующий цвет по отношению к фону рисунка. Выделенный рисунок можно перемещать, менять его размеры и удалять. Для перемещения подведите мышку в центр выделенного рисунка и выполните прием "протаскать и бросить". Заметим, что перемещать рисунок в таком состоянии как теперь можно не далее места расположения маркера конца абзаца. Попробуйте перетаскать рисунок на второй свободный абзац перед адресом клиента и обратно. Обратите внимание, что как только вы начали процедуру перетаскивания, при перемещении мышки с нажатой кнопкой по тексту документа будет передвигаться курсор в виде небольшого вертикального штриха. Он указывает вам место позиционирования перетаскиваемого объекта. Для изменения размеров рисунка укажите мышкой на один из квадратиков-маркеров, удобнее всего на угловой. Когда указатель примет форму двуправленной стрелки, нажмите левую кнопку мышки и тащите по диагонали рисунка. Появившаяся штриховая рамка покажет вам, как изменяются размеры рисунка. В нашем примере давайте подгоним размеры рисунка так, чтобы он занимал простран-

Рис. 3

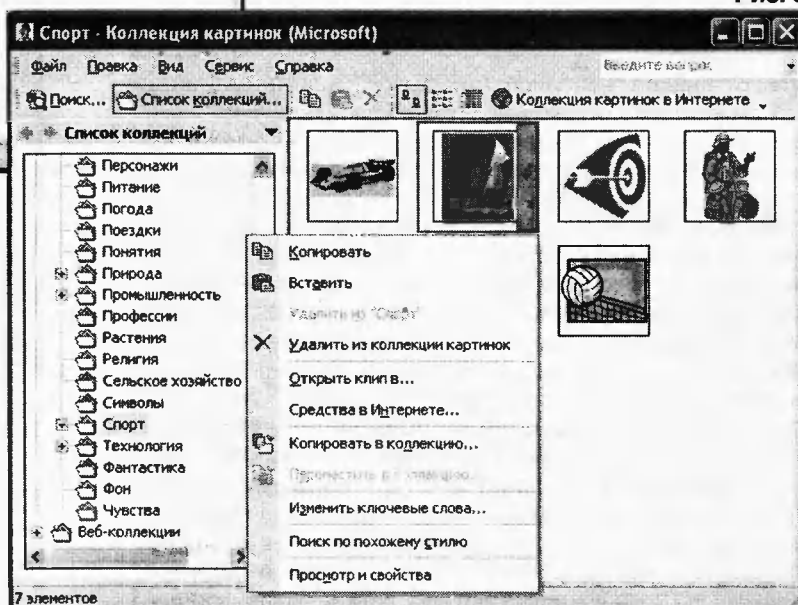
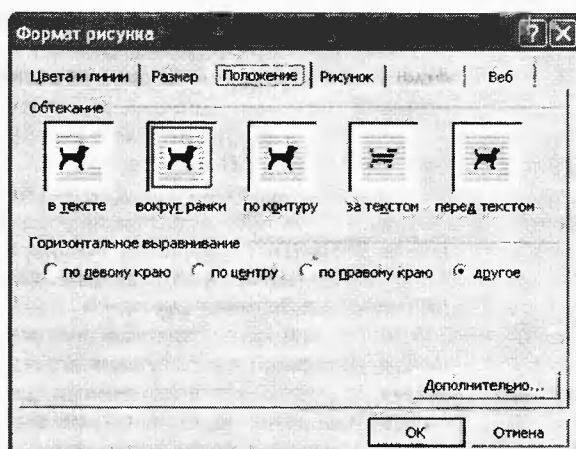


Рис. 4



ка рисунка) на левом поле. Все это признаки установки обтекания рисунка текстом. Конечно, это еще далеко не предел в смысле обработки рисунка в документе, но не все сразу — оставим что-нибудь на следующие уроки. Завершающие штрихи подготовки письма сделает

4. Установка параметров страниц

- Это выполняется посредством команды **Файл/Параметры страницы**. На вкладке **Поля** (рис.6) в соответствующих полях зададим следующие параметры: верхнее, нижнее, левое — 2 см, правое — 1 см. Поле **Переплет** служит для задания дополнительного зазора слева для подшивки листа. Поле **Применить** позволяет заказать область использования данных установок. Выберем здесь значение **Ко всему документу**. В разделе **Ориентация** оставим выбранным по умолчанию образец книжная. Теперь вытащим

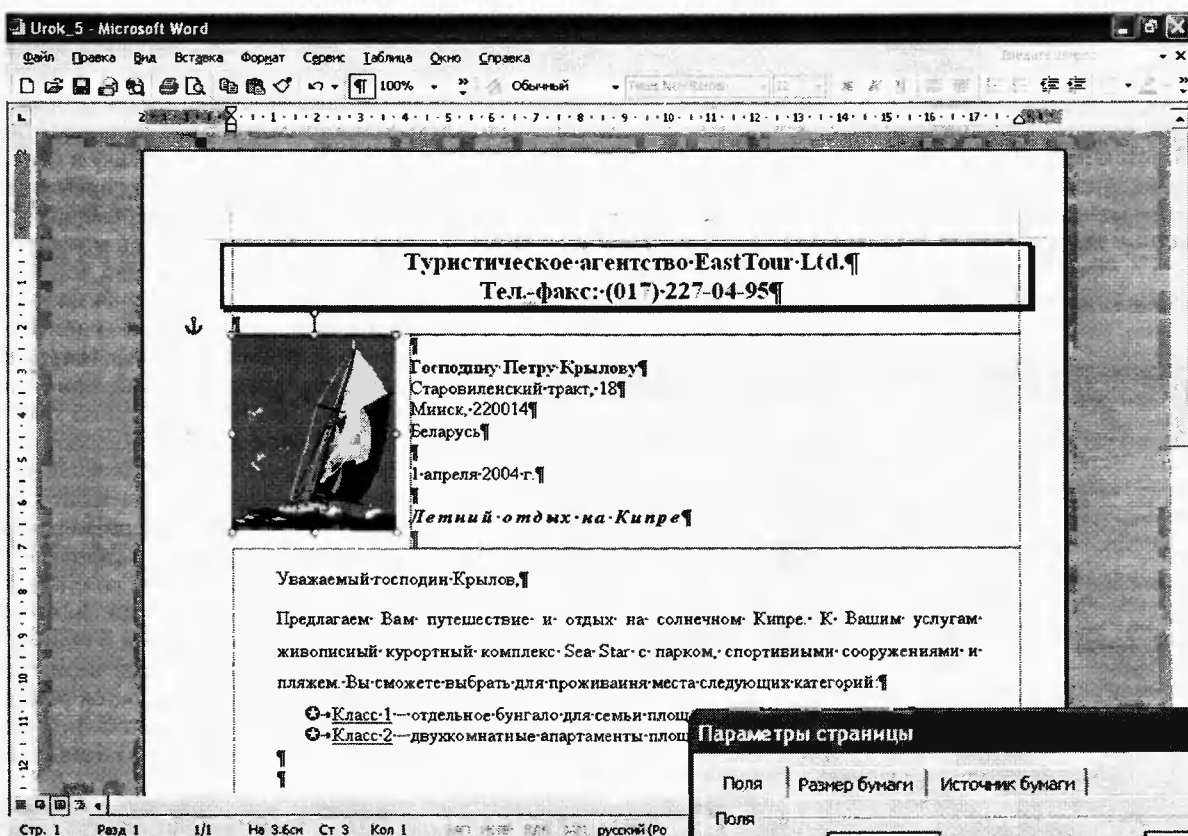
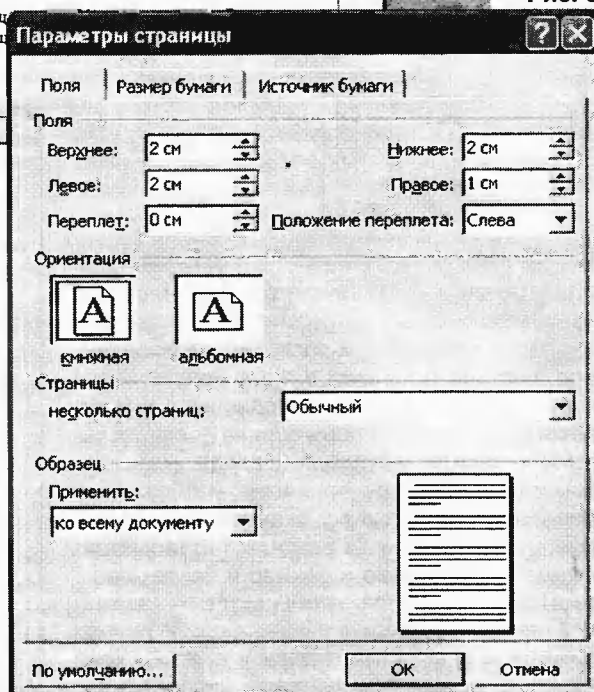


Рис. 5

Рис. 6

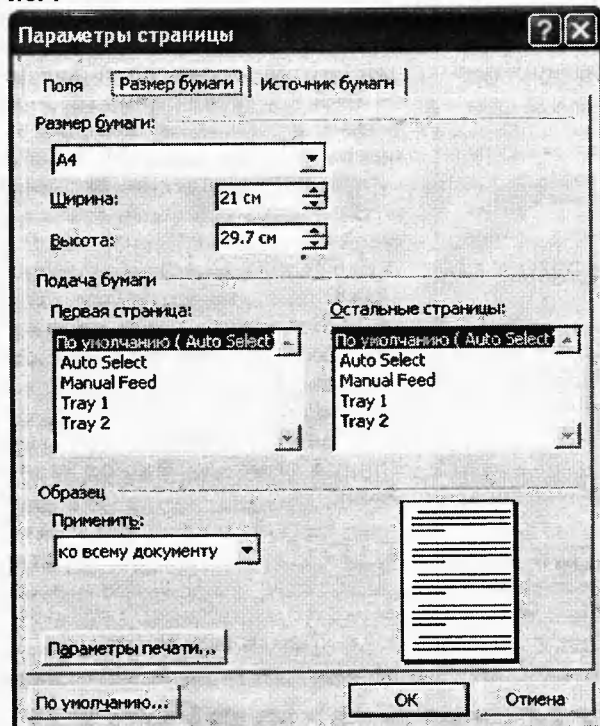


ство от первого свободного абзаца под "шапкой" до строки "Летний отдых...". Наконец, чтобы удалить выделенный рисунок, достаточно нажать клавишу **Del**, а затем можно его восстановить при помощи инструмента **Отменить**....

Обращаю ваше внимание на то, что пока рисунок ведет себя как символ — раздвигает текст, если попытаться разместить его между словами. Конечно, такой вариант взаимодействия с текстом неудобен. Лучше изменить параметры расположения рисунка так, чтобы текст как бы "обнимал" изображение. Выделим рисунок и вызовем диалог **Формат/Рисунок** (рис.4). Отметим, что если вы не щелкнете рисунок мышью, этой команды вы не найдете днем с огнем. На вкладке **Положение** диалога **Формат/Рисунок** щелчком мыши выберем образец **вокруг рамки** в разделе **Обтекание** и **ОК**. Посмотрим, что получилось. Теперь текст обнимает рисунок справа и снизу, что выглядит гораздо интереснее (рис.5). Обратите внимание и на то, что изменилась форма и прозрачность маркеров выделения рисунка, а также появился значок в форме якоря (привяз-

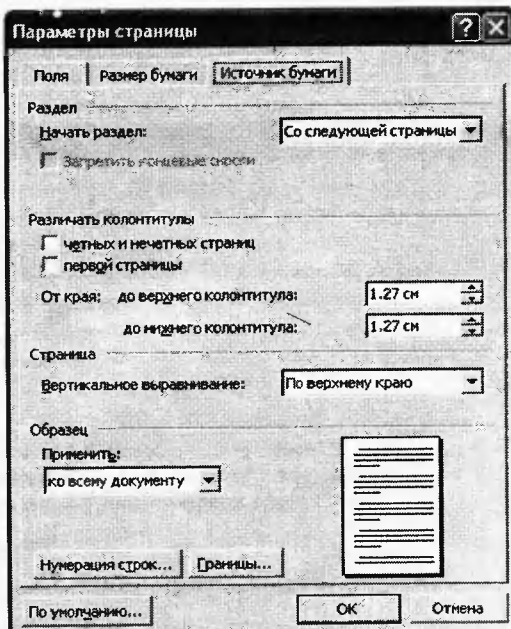
вкладку **Размер бумаги** (рис.7). Здесь выберем размер бумаги A4, а в разделе **Подача бумаги** оставим позицию **По умолчанию** в полях **Первая страница** и **Остальные страницы**. Отдельное задание параметров для первой страницы предназначено для случаев, когда первые страницы документов печатаются на фирменных бланках. Тип подачи зависит от принтера. Поинтересуйтесь этим воп-

Рис. 7



росом перед печатью и установите нужные значения. Если же вы не уверены, выберите параметры по умолчанию, и диспетчер печати справится сам. На вкладке **Источник бумаги** (рис.8) заказывают специфические тонкости, связанные с печатью листов, и она понадобится позднее, когда будем обсуждать работу с большими документами. Щелкнем **ОК** и вернемся к документу. Полюбуемся результатами нашей работы и запишем их под новым именем.

Рис. 8



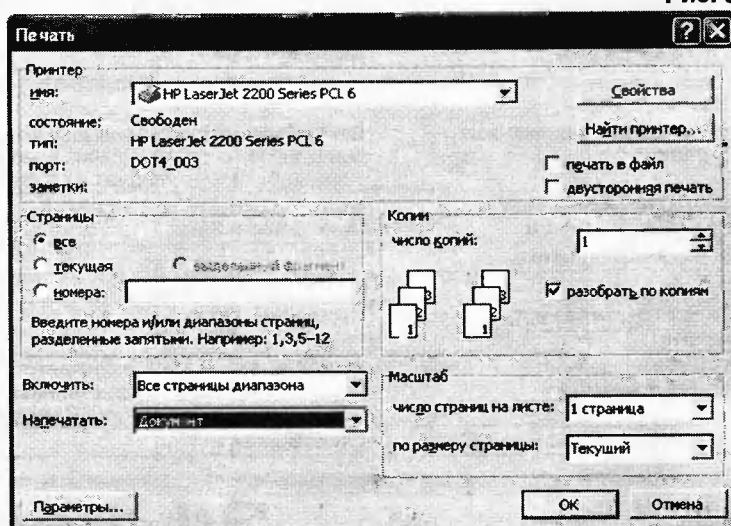
5. Сохранение файла под новым именем

Сохраним документ под новым именем, выполнив команду **Файл/Сохранить как**. В поле **Имя файла** поверх старого имени введем **Urok_5.doc**, **Enter**, а теперь выпустим наш документ в свет, напечатав его.

6. Печать документа

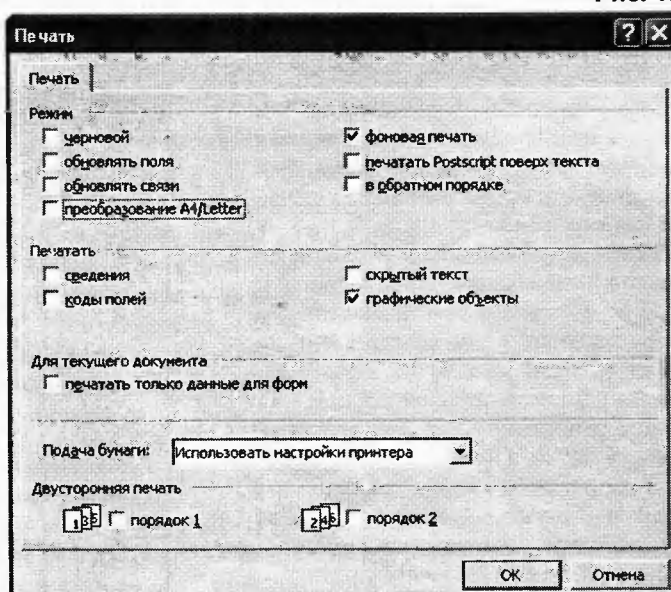
Не забудьте включить принтер (если это нужно) и вставить бумагу! Процесс печати запускается по команде **Печать** меню **Файл** или щелчком мыши по значку принтера на стандартной панели. В первом случае вы сможете задать параметры печати в соответствующем диалоговом окне (рис.9), во втором — печать сразу пойдет с установками по умолчанию. Для нашего примера зададим следующие установки в диалоге **Печать**. В разделе **Страни-**

Рис. 9



цы можно включить переключатель **Все** или **Номера** и впечатать 1. В разделе **Копии** установим 1 в поле **число копий**. Обратите внимание на поля-списки **Включить** и **Напечатать**. Здесь можно заказать распечатку сначала нечетных, а потом четных страниц (что удобно при печати на обеих сторонах листа бумаги). Можно также, помимо печати собственно документа, распечатывать всякую всячину типа примечаний, сведений, стилей и т.д. Но нам

Рис. 10





Краткое содержание Урока 5	
Действие	Реализация
1. Запуск Word	Щелчок по значку
2. Загрузка файла Urok_4.doc	Файл/Открыть, указать маршрут и имя файла
3. Вставка и управление рисунками в документе	
Вставка и управление рисунком	
Вставка рисунка из коллекции	Вставка/Рисунок/Картинки, гиперссылка Коллекция картинок, развернуть папку Коллекции Microsoft Office, раскрыть папку категории, навести мышь на миниатюру, щелчок по кнопке раскрытия списка команд и Копировать, затем Правка/Вставить в позиции курсора в документе
Выделение рисунка	Щелчок внутри изображения
Перемещение/копирование рисунка	Выделить, подвести указатель и, удерживая кнопку, перетащить/+Ctrl мышкой
Изменение размеров рисунка	Выделить, указать на маркер и перетащить рамку мышкой
Удаление рисунка	Выделить и Del
Настройка расположения рисунка в тексте	Выделить и Формат/Рисунок/Положение
4. Установка параметров страниц	Файл/Параметры страницы и указать на вкладке Поля значения полей и ориентацию листа, на вкладке Размер бумаги — формат и способ подачи бумаги
5. Сохранение файла под новым именем	Файл/Сохранить как, впечатать имя файла Urok_5.doc и Enter
6. Печать документа	
Печать документа	
Запуск печати без установки параметров	Инструмент
Установка параметров и запуск печати документа	Файл/Печать, указать в соответствующих полях-списках печатаемый объект, страницы, число копий, задать режимы печати (кнопка Параметры) и ОК
7. Завершение работы Word	Файл/Выход

пока нужен документ, поэтому остановимся именно на этом пункте списка **Напечатать**. Теперь включим кнопку **Параметры**, что отправит нас к вкладке **Печать** панели **Параметры** меню **Сервис** (рис.10). Здесь задаются режимы печати. Установим флажки **фоновая печать** и **графические объекты**. В этом случае вы сможете работать над другим документом в процессе печати, а также будете гарантированы в отношении передачи на печать рисунка. Щелкнем на обеих панелях последовательно кнопки **ОК** и ждем завершения процесса печати.

Ура! Теперь мы можем подержать в руках плоды нашего кропотливого труда. Но как много потрудиться нам еще предстоит впереди! А этот урок закончен, и мы, просмотрев итоговую таблицу.

7. Завершаем работу Word, выполнив команду **Выход** меню **Файл**.

(Продолжение следует)

Е.ШАКЕЛЬ,
г.Минск, РИВШ БГУ,
E-mail: shakel@nihe.niks.by

Эффективная работа с табличным процессором Microsoft Excel XP

(Продолжение. Начало в №№1-4/2005)

Основные приемы форматирования

Продолжая знакомство с приложением Microsoft Excel XP, посвятим наш сегодняшний урок дальнейшему знакомству с приемами форматирования.

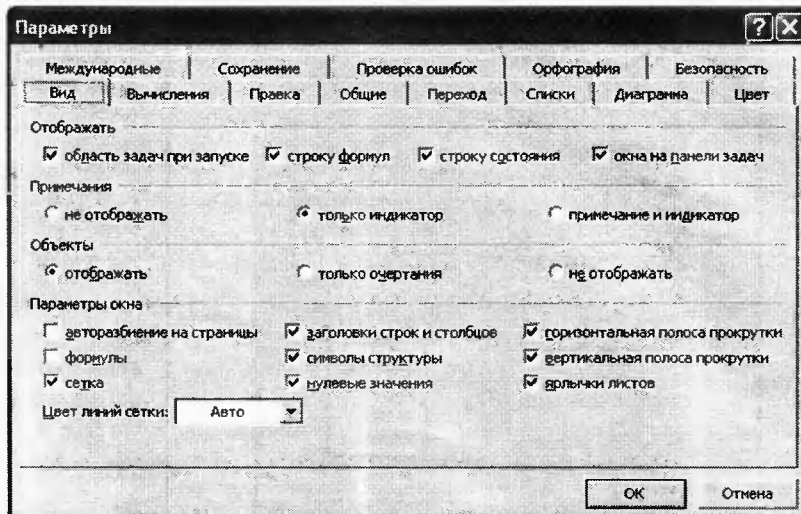
Итак, осуществим уже знакомые вам процедуры запуска MS Excel и загрузки рабочего файла **Учебный_файл.xls** и обсудим процесс дальнейшего оформления ячеек рабочего листа.

Обратите внимание на линии сетки, разграничивающие ячейки на рабочем листе. Их удобно использовать при вводе данных, но не всегда стоит оставлять после того как процесс создания и оформления электронной таблицы завершен. С одной стороны, линии сетки зрительно разделяют строки и столбцы вашей таблицы, а с другой — далеко не всегда упрощают восприятие. Поэтому, как правило, нужные области рабочего листа заключают в рамки (выполняют оформ-

ление границ), а линии сетки отключают. Воспользуйтесь командой **Параметры** меню **Сервис** и снимите фла-

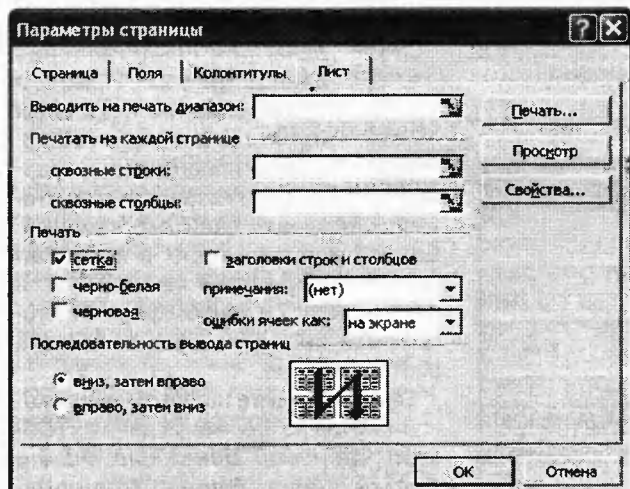
жок **Сетка** на вкладке **Вид** (рис.18), что приведет к изменению вида рабочей области. Обратная процедура ус-

Рис. 18



тановки флажка **Сетка** вернет отображение разделительных линий. С другой стороны, разделительные линии сетки могут быть удалены с экрана только для вывода на печать. В данном случае необходимо выполнить команду **Параметры страницы...** меню **Файл** и снять флажок **Сетка** на вкладке **Лист** (рис.19). При этом, независимо от того, присутствуют линии раздела ячеек или нет на экране, они не будут выводиться на печать.

Рис. 19



Функцию установки рамок (границ) электронной таблицы чаще всего выполняет кнопка **Границы** на панели **Форматирование**. Выделите область рабочего листа, которую вы хотите заключить в рамки, откройте список на кнопке **Границы** , щелчком мыши выберите нужный образец. В случае если у вас появилось желание поэкспериментировать, можете воспользоваться кнопкой **Нарисовать границы**, доступ к которой открывается из списка того же инструмента **Границы**. Нажатие ее приведет к подключению панели **Граница** (рис.20), инструмента которой помогут выбрать тип и цвет граничных линий, а затем нарисовать их.

Рис. 20



Откройте список инструмента **Вид линии** и щелчком мыши выберите подходящий образец, затем щелкните по кнопке **Цвет линии** и воспользуйтесь цветовой палитрой для выбора цвета. Далее кнопкой **Нарисовать границу** определите, что вы будете рисовать: отдельные линии (**Границы рисунка**) или сетку для всех ячеек выделенной области (**Сетка по**

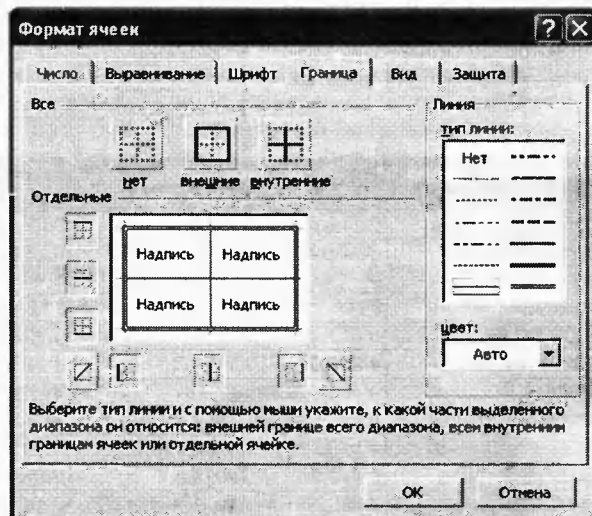
границе рисунка). А теперь приступим. Нажмите мышью и протяните по любой линии раздела ячеек — граница готова. Если нарисовали лишнее, нажмите инструмент **Стереть границу** и протяните мышью по нарисованной линии. Готово.

С другой стороны, выбор границ удобнее осуществлять с помощью команды **Ячейки...** меню **Формат**, так как на вкладке **Границы** диалогового окна **Формат ячеек** (рис.21) вам пре-

ки на панели **Форматирование** или через команду **Ячейки...** меню **Формат**. Воспользовавшись меню, на вкладке **Вид** (рис.22) вы сможете не только установить требуемый цвет заливки, но также выбрать узор и цвет узора для заполнения ячеек из соответствующего списка.

Что ж, применим наши знания на практике. Оформим электронную таблицу рабочей книги **Учебный_файл.xls** следующим образом:

Рис. 21



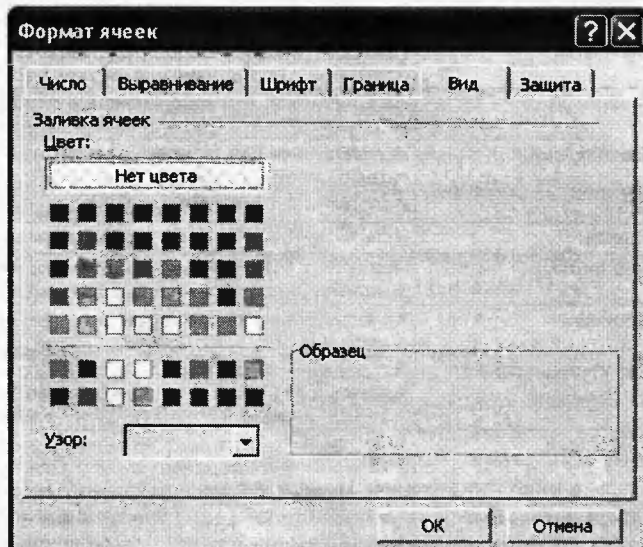
доставляется возможность сразу указать весь спектр параметров границ, начиная с вида и цвета обрамляющих линий и заканчивая видом самих границ. Отмена границ осуществляется аналогично их установке — либо щелчком мыши по кнопке **Нет** области **Все** на вкладке **Границы** (рис.21), либо выбором образца **Нет границы** в списке инструмента **Границы** на панели **Форматирование**.

Эффектно выглядит таблица, в которой используется заливка цветом для необходимых областей ячеек. Однако не следует чрезмерно перегружать таблицу обилием разнообразных не сочетающихся цветов и узоров. Тем более, что цветность интересна лишь в том случае, если распечатку вы планируете произвести на цветном принтере. Выбор цвета заливки выполняется с помощью инструмента **Цвет заливки**

- выберем тип линии — двойная, цвет — зеленый и установим внешние границы для диапазона ячеек A3:B9. Затем выберем тип линии — одинарная и установим внутренние границы для указанной области;

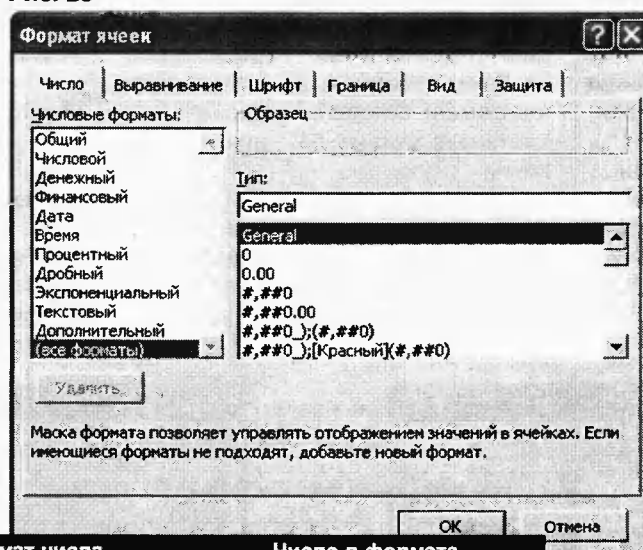
- установим бледно-зеленый цвет заливки для диапазона ячеек A4:B9 и травяной с узором 12,5%-й серый для ячеек A3, B3. Можно дополнительно произвести выбор цвета узора.

Рис. 22



Наша таблица почти готова, осталось определить формат данных, в первую очередь числовых, в ячейках рабочего листа. Как правило, если вы не прилагаете специальных усилий, информация вводится в ячейки в формате **Общий**, который используется для отображения как текстовых, так и числовых значений произвольного типа. Что касается текстовых данных, то с ними сложностей обычно не возникает, поскольку для их

Рис. 23



отображения обычно используется формат **Общий**. А вот для представления чисел у вас существует целый набор встроенных форматных кодов. Выполните команду **Формат/Ячейки...** и переключитесь на вкладку **Число**. При выборе строки (**все форматы**) в поле **Числовые форматы** в ваше распоряжение поступят все форматные коды, которые могут быть использованы для представления данных (рис.23). Примеры использования некоторых из них приведены в табл.8.

Если предлагаемых форматных кодов вам покажется недостаточно, создайте свой собственный. Его вы можете ввести в поле **Тип** (рис.23), например, **###0,00\$**. Числа 1200 и 34500,2 в данном формате отобразятся как 1 200,00\$ и 34 500,20\$ соответственно.

Теперь укажите щелчком мыши какой-нибудь из форматов данных (**Общий**, **Числовой**, **Денежный**, **Финансовый**, **Дата**, **Время**, **Процентный**, **Дробный**, **Экспоненциальный**, **Текстовый**, **Дополнительный**) в поле **Числовые форматы**. Практически в каждом случае, за исключением выбора **Общего** и **Текстового**, вы сможете выбрать определенный тип того или иного формата и изменить некоторые параметры, такие как число десятичных знаков, денежные единицы или вид представления отрицательных чисел.

Табл. 9

Табл. 8

Вводимое число	Формат числа (форматный код)	Число в формате
3455,46	Основной	3455,46
0,25444536455	Основной	0,254445265
1,324324	0	1
1,324324	0,00	1,32
2223433,237	##0	2 223 433
2223433,237	##0,00	2 223 433,24
456,6	##0р.;##0р.	457р.
456,6	##0р.;[Красный]##0р.	457р. (цвет шрифта — красный)
-621,72	##0р.;[Красный]##0р.	-621р. (цвет шрифта — красный)
0,23456	##0,00р.;##0,00р.	0,23р.
-0,23456	##0,00р.;[Красный]##0,00р.	-0,23р. (цвет шрифта — красный)
3	0%	300%
3	0,00%	300,00%
0,2323232	0,00E+00	2,32E-01
0,2323232	##0,0E+0	232,3E-3
0,5	#»»?/?	1/2
1/03/2005 или 1.3.05	ДД.ММ.ГГГГ	01.03.2005
1/03/99 или 1.3.99	ДД.МММ.ГГ	01.мар.05
1/03/99 или 1.3.99	ДД.МММ	01.мар
1/03/99 или 1.3.99	МММ.ГГ	мар.05
12:20	ч:мм:сс	12:20:00
11:20:30	ч:мм	11:20
11:20:30	мм:сс	20:30

Числовые форматы	Общая характеристика и назначение
Общий	Используется для отображения текстовых и числовых значений произвольного типа
Числовой	Наиболее общий способ представления чисел. Позволяет отобразить число как целое или в виде десятичной дроби с определенным числом десятичных знаков
Денежный	Используется для форматирования ячеек, в которые заносятся денежные значения. Позволяет отобразить числовое значение в любых денежных единицах (р., \$, ...) и с заданным числом десятичных знаков
Финансовый	Используется для отображения денежных величин. Задает их автоматическое выравнивание по разделителю целой и дробной части
Дата	Используется для отображения даты или даты и времени одновременно
Время	Используется для отображения времени или даты и времени одновременно
Процентный	При использовании данного формата значения ячеек умножаются на 100 и выводятся с символом %. Например, число 2 можно представить в виде 200%
Дробный	Используется для перевода десятичных дробей в простые. Например, число 0,25 можно отобразить в виде 1/4, 2/8, 4/16, 25/100...
Экспоненциальный	Используется для нормализованного представления в основном слишком больших или слишком малых числовых значений. Например, число 2450000000000 можно отобразить в виде 2,45E+11, что означает 2,45·10 ¹¹ , а число 0,000000001234 — в виде 1,234E-09, что означает 1,234·10 ⁻⁹
Текстовый	Текстовый формат дает возможность отобразить как текстовую, так и числовую информацию в том виде, в котором она введена в ячейку. Для числовых значений используется в том случае, если необходимо ввести числа как текст, например, номера телефонов
Дополнительный	Дополнительные форматы предназначены для работы с базами данных и списками адресов. Используются для форматирования ячеек, в которые вводятся почтовые индексы, номера телефонов и табельные номера



С.ГРИНЧУК,
E-mail: grinchuk@nihe.niks.by

Разработка web-сайтов в Microsoft Office FrontPage 2003

(Продолжение. Начало в №№8-12/2004, 1-4/2005)

Вставка и управление параметрами встроенных фреймов

Завершая разговор об использовании фреймов, рассмотрим возможности так называемых **встроенных** (или, как их еще называют, **плавающих**) фреймов. Встроенные фреймы весьма похожи на обычные фреймы, за тем лишь исключением, что их можно внедрять в обычные web-страницы. Используя встроенные фреймы, вы получаете возможность встраивать один документ внутри другого. При этом плавающий фрейм будет отображаться на web-странице в виде прокручиваемого текстового поля. Одним из преимуществ встроенных фреймов является то, что при их использовании не требуется создавать отдельную страницу фреймов. Кроме того, для встроенного фрейма при необходимости можно задать возможность обтекания текстом (как в случае с графическими изображениями).

В каких же случаях можно использовать плавающие фреймы? Представьте, что web-страница содержит перечень услуг или товаров, предоставляемых некоторой фирмой. На эту же страницу с помощью встроенного фрейма можно поместить договор или контракт для заполнения посетителями сайта. Плавающие фреймы можно также использовать для отображения примеров других web-страниц.

Рассмотрим вопросы вставки и форматирования встроенных фреймов на практике. Выполняем обычные стартовые действия

1. Запуск Microsoft Office FrontPage 2003 и

2. Открытие web-сайта myweb2, созданного на предыдущих занятиях.

3. Вставка и управление параметрами встроенных фреймов.

Как вы помните, наша старая до-

машняя страница **main.htm** содержит ссылки на все остальные документы сайта. В дополнение к гиперссылке на страницу с коллекцией интернет-адресов воспользуемся встроенным фреймом для внедрения этой коллекции (файл **favorite.htm**) прямо в документ **main.htm**.

Двойным щелчком мыши откройте в режиме редактирования документ **main.htm**. В представлении **Конструктор** вставьте пустой текстовый абзац перед абзацем "Карта сайта". Для добавления на страницу встроенного фрейма установите текстовый курсор в созданный пустой абзац и выполните команду **Вставка/Встроенная рамка**. Тут же в указанном месте документа появится встроенный фрейм. Как и в случае с обычными фреймами, после этого следует определить документ для отображения в созданном фрейме (с помощью кнопок **Задать начальную страницу...** и **Создать страницу**). Воспользуйтесь кнопкой **Задать начальную страницу...** и выберите в качестве начальной страницы фрейма документ **favorite.htm** (рис.39).

Для тестирования встроенного фрейма сохраните документ **main.htm** (инструмент панели **Стандартная**) и включите просмотр в браузере (инструмент). Обратите внимание, что плавающий фрейм действительно отображается на web-странице в виде прокручиваемого текстового поля. Заметьте, что т.к. все гиперссылки документа **favorite.htm** ранее мы настраивали для открытия в отдельном окне браузера, эти же ссылки, находясь внутри встроенного фрейма, тоже нормально открываются (обычно документ, открываемый по ссылке, загружается в тот же фрейм.

В таком случае нам пришлось бы производить дополнительную настройку гиперссылок, как мы это делали для страниц фреймов).

А теперь поговорим о возможностях форматирования встроенных фреймов. Прежде всего вы можете настроить размеры встроенного фрейма. Для этого вернитесь в Microsoft FrontPage и щелчком мыши по границе фрейма выделите его (встроенный фрейм будет выделен как рисунок). Затем, установив курсор мыши на один из маркеров выделения и нажав левую кнопку мыши, переместите границу фрейма в требуемом направлении.

Остальные параметры встроенного фрейма настраиваются с помощью диалогового окна **Свойства встроенной рамки**. Традиционно для вызова этого окна следует выделить встроенный фрейм (щелчком мыши по границе) и выполнить команду **Формат/Свойства** (любители использования контекстного меню, вызываемого щелчком правой кнопкой мыши, должны иметь в виду, что в данном случае щелчок необходимо выполнять по границе встроенного фрейма).

Как видите, практически все параметры диалогового окна **Свойства встроенной рамки** нам уже знакомы (рис.40). Как и в случае обычных фреймов, в поле **Имя** этого диалогового окна отображается имя текущего фрейма (I1). Поле **Начальная страница** предназначено для указания документа, отображаемого во встроенном фрейме. Как и прежде, для смены документа, загружаемого во фрейм, используется кнопка **Обзор...** В разделе **Размер рамки** можно задать ширину и высоту встроенного

Краткая информация об использовании тех или иных форматов данных содержится в табл.9.

Выбор формата для каждой ячейки можно произвести заранее, до ввода данных, или назначить уже после того как данные будут введены. Некоторые числовые форматы могут быть заданы с помощью соответствующих кнопок: **Денежный формат** — , **Процентный формат** — , **Формат с разделителем** — на панели **Форматирования**.

Рис. 24

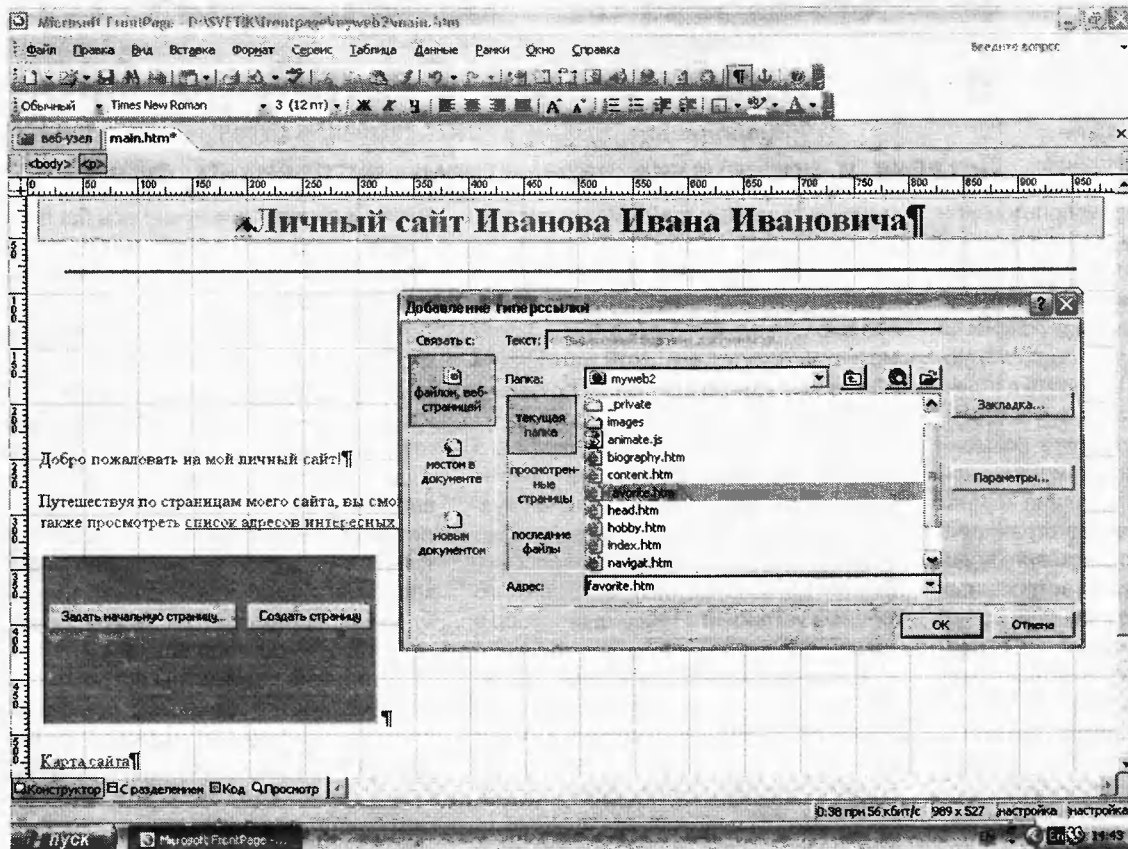
	А	В
1	Таблица 1. Почасовая оплата	
2	сотрудников.	
3	<i>Должность</i>	<i>Оплата за час</i>
4	Профессор	8 500,00р.
5	Доцент	7 000,00р.
6	Старший преподаватель	6 900,00р.
7	Преподаватель	6 800,00р.
8	Методист	6 000,00р.
9	Лаборант	5 200,00р.

матирование. Здесь же содержатся кнопки **Увеличить разрядность** — и **Уменьшить разрядность** — , позволяющие установить нужное число десятичных знаков при отображении числа.

Вернемся к нашей таблице и отформатируем числовые значения в соответствии с рис.24. Завершим урок процедурой сохранения файла.

(Продолжение следует)

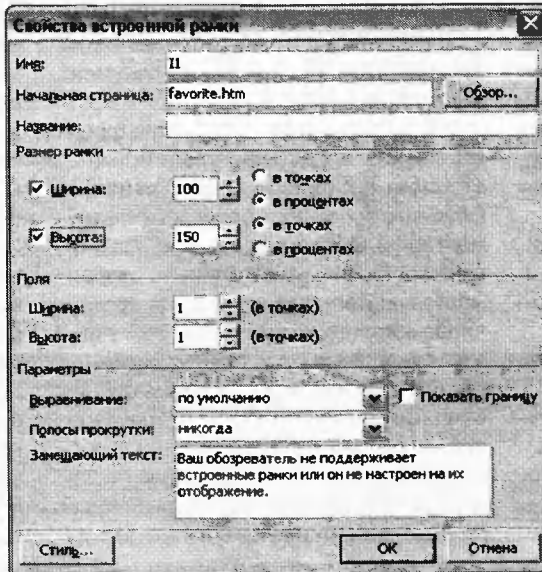
Рис. 39



фрейма как в пикселах, так и в процентах. Раздел **Поля** используется для определения в пикселах размеров пустых полей внутри фрейма — расстояний между границами фрейма и его содержимым. С помощью поля **Выравнивание** раздела **Параметры** вы можете определить выравнивание встроенного фрейма относительно текста web-страницы (используя варианты **по левому краю** и **по правому краю** можно задать возможность обтекания фрейма текстом). С помощью поля **Полосы прокрутки** вы можете управлять отображением полос прокрутки у встроенного фрейма, в с помощью флажка **Показать границу** можно управлять включением и отключением отображения границы фрейма. Последнее поле **Замещающий текст** используется для ввода текста, который должен отображаться в браузере в случае, если он не поддерживает фреймы.

Экспериментируя с параметрами встроенного фрейма, можно, так же как и со страницей фреймов, добиться иллюзии целостности просматриваемого документа. К примеру, определите для фрейма ширину 100%. Высоту фрейма настройте таким об-

Рис. 40



разом, чтобы документ **favorite.htm** отображался во фрейме целиком, без использования полос прокрутки. В качестве размеров отступов содержимого фрейма от его границ укажите в полях **Ширина** и **Высота** 1 пиксел. В поле **Полосы прокрутки** установите вариант **никогда** и снимите флажок **Показать границу**.

Сохраните документ **main.htm** и протестируйте в браузере. Согласи-

тесь, теперь достаточно трудно догадаться о том, что наша исходная страница содержит встроенный фрейм. Выглядит она как цельная обычная web-страница, хотя ее содержимое и формируется на основе нескольких документов. Подобный прием удобно использовать также в том случае, когда web-страница содержит часто меняющийся материал (например, список товаров и цен). Тогда этот часто обновляемый материал можно хранить в виде отдельного документа, который и будет редактироваться по мере необходимости, а в исходную web-страницу вставить просто встроенный фрейм.

Конечно, в нашем случае вряд ли целесообразно использовать встроенные фреймы. Поэтому после краткого знакомства с плавающими фреймами восстановим status quo. Чтобы удалить встроенный фрейм, вернитесь в Microsoft FrontPage, щелчком мыши по границе выделите фрейм и нажмите клавишу **Delete** на клавиатуре. Естественно, при этом документ **favorite.htm**, отображавшийся в удаленном фрейме, останется в пвпке сайта.

На этом наше сегодняшнее занятие мы заканчиваем. Не забудьте сохранить документ **main.htm** и выполнить

4. Завершение работы с Microsoft Office FrontPage 2003.

В следующий раз мы обсудим вопросы подготовки разработанного web-сайта к публикации в сети Интернет, а также возможности Microsoft FrontPage по самой процедуре публикации.

(Окончание следует)

СЛОВО О СЛОВАРЕ

В.КУЦ,

E-mail: victor_kootz@mail.ru

Уровень советской системы образования высоко котируется во всем мире — это непреложный факт. Однако и “на старуху бывает проруха” — наряду с прекрасно поставленной и проверенной многолетней практикой технологией преподавания точных и естественных наук, ее гуманитарная составляющая откровенно “хромала”. В частности, особо слабым было преподавание (и, естественно, усвоение учащимися) иностранных языков. Пресловутое владение английским на уровне “читаю и перевожу со словарем” уже стало просто “притчей во языцех”, хотя, с другими языками ситуация была гораздо хуже. Казалось бы, издержки старой системы безвозвратно ушли в прошлое, однако нет — ведь большая часть активного ныне населения не может похвастать образованием иным, чем то самое, советское, со всеми его достоинствами и недостатками. Да и новая школа большими успехами в деле подготовки широких масс подрастающего поколения, свободно владеющего как минимум одним иностранным языком, пока еще похвастать не в состоянии.

Так что волей-неволей приходится большинству из нас при работе с документами на языках, отличных от родного, прибегать к помощи верной палочки-выручалочки — словаря иностранных слов (без которого — ни “читаю”, ни “перевожу”...). Каким именно словарем обзаводиться, конечно, дело вкуса каждого. Сплошь и рядом встречаются люди, доверяющие лишь традиционным бумажным изданиям. Но серьезный словарь, имеющий солидные размеры и вес, крайне неудобно постоянно носить с собой, да и поиск нужного слова в нем занимает немалое время, а компактный — имеет слишком маленький “словарный запас”. Поэтому многие выбирают более практичный вариант, прибегая к помощи иных словарей — электронных. Эти словари, базирующиеся на самых современных компьютерных технологиях, пользуются в наши дни все большим спросом. Хотя некоторые, наименее подготовленные (или наиболее ленивые) пользователи предпочитают словарям так называемые “пакетные переводчики”, которые полностью переводят весь

текст. Но человек, имеющий пусть даже самые элементарные познания в иностранном языке, пользоваться такими переводчиками никогда не станет — качество их перевода давно стало одной из самых любимых тем для авторов анекдотов на компьютерную тематику. Хотя спорить не буду, пакетные переводчики занимают свою нишу и иногда бывают очень даже полезны. Но сегодня речь пойдет не о них.

Электронные словари бывают нескольких видов: они выпускаются как в виде программ, предназначенных для установки на персональный компьютер (настольный, портативный, либо карманный), так и в виде самостоятельных электронных устройств — “карманных переводчиков”, “записных книжек с функцией словаря” и тому подобных устройств. Миниатюрные карманные переводчики, как и записные книжки — вещь, конечно, очень удобная, особенно в дальних поездках, однако одним из важнейших параметров любого словаря (в том числе и электронного) остается объем словарной базы, в наибольшей степени влияющий на степень “полезности” такого словаря. А по этому параметру программно реализованные словари, располагающие практически неограниченным объемом памяти, без сомнения, вне конкуренции.

Сегодня мы поговорим об одном из лучших представителей программных словарей, и даже не просто словаре, а о настоящей совокупности электронных словарей — Lingvo, очередная, десятая версия которого не так давно была выпущена российской компанией ABBYY Software.

Первая, англо-русская версия данного словаря увидела свет еще в далеком 1990 г. С тех пор Lingvo постоянно развивается, завоевывая все большую популярность, в первую очередь, благодаря простому и чрезвычайно удобному интерфейсу, обширной словарной базе и хорошо организованному механизму поиска. Начиная с 8-й версии, словарь Lingvo превратился из англо-рус-

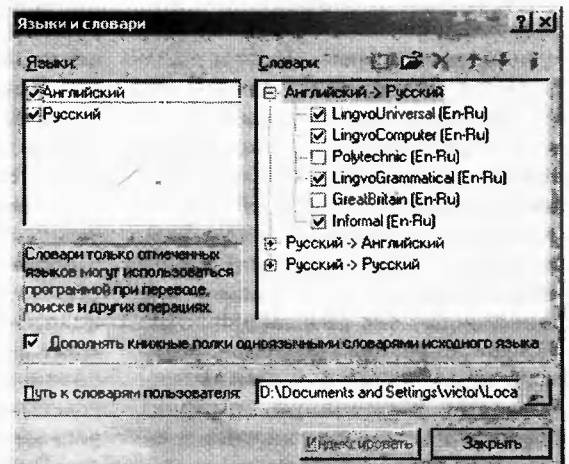
ского в многоязычный. Наш сегодняшний герой, Lingvo 10.0, включает в себя уже около 10 000 000 переводов. Для сравнения можно указать, что такой объем соответствует примерно 70 томам обычных, бумажных словарей.

Многоязычная версия Lingvo 10.0, объем которой достиг двух компакт-дисков (правда, основная программа по-прежнему умещается на одном диске, а второй содержит “облегченные”, учебные версии словарей и вспомогательные программы), включает 78 общих и специализированных словарей на 5 языках:

- англо-русские, русско-английские;
- немецко-русские, русско-немецкие;
- французско-русские, русско-французские;
- итальянско-русские, русско-итальянские;
- испанско-русский, русско-испанский.

В состав Lingvo, помимо словарей общей лексики, входят специализированные медицинские, экономические, политехнические, компьютерные и другие (рис.1). Каждая словарная статья включает в себя не только подробные переводы и всевозможные варианты толкования слова, но и различные комментарии по грамматике и фонетике, а также указание на допустимые формы слова и примеры их правильного употребления. Все словари разбиты на группы, называемые “Тематическими книжными полками”. При этом все словари на “книжной полке” Lingvo можно теперь группировать по тема-

Рис. 1



тикам при помощи новой "Функции настройки словарной полки". Работая с документами определенной тематики, можно быстро переключаться между соответствующими "полками" — перевод и поиск будет проводиться только в активных словарях выбранной группы.

Достаточно большой интерес представляют лингвострановедческие словари: по Великобритании (появился еще в Lingvo 9.0) и США — "Американа II" (в Lingvo 10.0). Оба этих словаря представляют собой справочники терминов, достопримечательностей, названий английских улиц, музеев, журналов, телепередач и т.д. Они дают не только переводы и толкования сугубо английских понятий, но и пояснения относительно различных явлений государственного устройства, образования, быта, национальных традиций и даже национальных видов спорта. Короче, они являются весьма полезным источником информации для всех, кто желает глубже познаться с культурой и бытом Великобритании и США. Из других нововведений десятой версии, пожалуй, наибольший интерес может представить совершенно новый толковый словарь компьютерных терминов Э.М.Пройдакова и Л.А.Теплицкого.

Строго говоря, электронный словарь Lingvo 10.0 существует в двух вариантах, заметно различающихся назначением, и, как следствие, возможностями. Помимо наиболее полной версии — многоязычной, также выпущена его англо-русская версия, содержащая примерно 6 млн. переводов в 36 словарях общей, разговорной и тематической лексики. Немаловажно, что все вышеуказанные программы могут быть установлены как на обычный настольный ПК либо ноутбук, так и на карманный компьютер (поддерживаются обе конкурирующие платформы: Pocket PC и Palm OS). Возможность использования словаря в карманном компьютере особенно интересна с точки зрения высокой мобильности его пользователя — установив на КПК многоязычный словарь Lingvo 10.0, он получает в свое распоряжение мощнейший "инструмент перевода", содержащий как разговорники (по 500 наиболее употребительных выражений для каждого из пяти языков, входящих в основной комплект Lingvo 10.0), так и полноценные электронные словари, снабженные удоб-

ной системой поиска. Словарная база такого словаря будет не меньше, чем в "настольной" версии, естественно, при условии, что ваш "наладонник" располагает достаточным объемом оперативной памяти. Впрочем, почти в любом современном КПК при необходимости объем памяти может быть увеличен путем использования дополнительных флэш-карт.

Еще одной полезной функцией словаря Lingvo является фонотека, составленная из 25 тыс. слов — 15 тыс. самых популярных в разговорной речи английских слов (входят в обе основные версии словаря) и 10 тыс. немецких (только в многоязычной версии). Аудиословари помогут всем тем, кто самостоятельно работает над улучшением своего произношения, ведь все эти слова озвучены носителями языка — профессиональными дикторами из Великобритании и Германии, а значит, в правильности их звучания сомневаться не приходится.

В "десятке" гораздо большее, чем в ее предшественниках, уделяется внимание тем, кто изучает иностранные языки. Трудно переоценить значение грамматического словаря английского языка, который присутствует как в английской, так и в многоязычной версии Lingvo. Этот словарь, содержащий около 500 тем (статей), является гипертекстовым справочником по грамматике английского языка и является довольно неплохим учебным пособием (или шпаргалкой — кому как нравится). Он содержит описания основных грамматических понятий, а также правила и примеры их использования. В карточках словаря можно найти не только информацию об использовании каждой отдельной грамматической конструкции, но и о разнице между близкими конструкциями. Кроме того, в словаре собрана информация о нормах речевого этикета, о различиях между британским и американским вариантами английского языка, в частности, о различиях в произношении некоторых слов и т.д.

Хорошим дополнением к грамматическому словарю может стать специальное приложение для изучающих языки — Lingvo Tutor, впервые появившееся в Lingvo 10, которое поможет эффективно пополнить словарный запас. В англо-русской версии Tutor включает 2 учебных слова-

ря (англо-русский и русско-английский), тогда как в многоязычной — 10 учебных словарей на всех 5 языках. С его помощью можно создавать собственные и подключать пользовательские учебные словари (в форматах LUD или XML). Урок в Lingvo Tutor может запускаться вручную или автоматически через заданный интервал времени в ходе работы на компьютере. В процессе изучения слов ведется статистика: количество изучаемых слов, количество выученных слов и т.д. Учебные карточки включают в себя переводы, транскрипцию и примеры, которые могут использоваться в качестве дополнительной информации к урокам. Карточки учебных словарей можно распечатать и повторять слова, даже когда под рукой нет компьютера.

Для всех поддерживаемых языков (английского, немецкого, французского, итальянского и испанского) имеются свои разговорники, содержащие примерно по пятьсот наиболее употребительных фраз на каждом из языков. Для большего удобства использования они разбиты на разделы: знакомство, хобби/интересы, встреча, погода, развлечения, путешествия, назначение встречи, общие фразы, здоровье/настроение, гостиница, на границе, ресторан/кафе, транспорт, официальные службы, магазины, спорт, числительные, надписи.

Как и в предыдущих версиях, Lingvo 10.0 хорошо интегрирован в Microsoft Word и может вызываться непосредственно из текстового процессора. При запуске Lingvo из Word происходит автоматический перевод слова, на котором в момент запуска стоял курсор, либо всех выделенных слов.

Рис. 2



Программа состоит из двух основных компонентов — запускаемого модуля Агент Lingvo (занимающего около 1200 Кб оперативной памяти) и подключаемых словарей. Агент Lingvo создает в трее иконку, щелкнув по которой можно запустить главное рабочее окно словаря (рис.2), содержащее список слов и органы управления программой:

Дизайн окна стилизован под Windows XP, т.е. полностью соответствует последней компьютерной моде. При минимизации рабочего окна оно пропадает с панели и из списка задач, но не закрывается — впоследствии словарь можно вызвать, кликнув на сидящую в трее иконку Агента. Это очень удобно, особенно, если открыто много приложений. Агент также занимается перехватом буфера. Язык при этом определяется автоматически. В этом окне

же имеется маленькая иконка запуска аудиословаря (рис.3).

Однако, постоянно открытое окно словаря на Рабочем столе требуется далеко не всегда. Тем, кто работает со словарем время от времени или даже постоянно, но именно для перевода отдельных неизвестных слов, лучше воспользоваться наиболее простым и быстрым способом работы — выделить на странице с иностранным текстом незнакомое слово (или словосочетание) и нажать сочетание клавиш Ctrl + C + C (или Ctrl + Ins + Ins). Буквально через мгновение на экране появляется основное окно программы, а за ним и словарные карточки, на которых отобразятся все возможные варианты перевода выбранных слов. Наряду с отдельными словами, Lingvo может переводить и устойчивые словосочетания.

Такой поиск (создатели называют его "поиском с учетом морфологии") позволяет довольно быстро и с удобством вспомнить, например, перевод английских неправильных глаголов или немецких сложносоставных слов-"композигов".

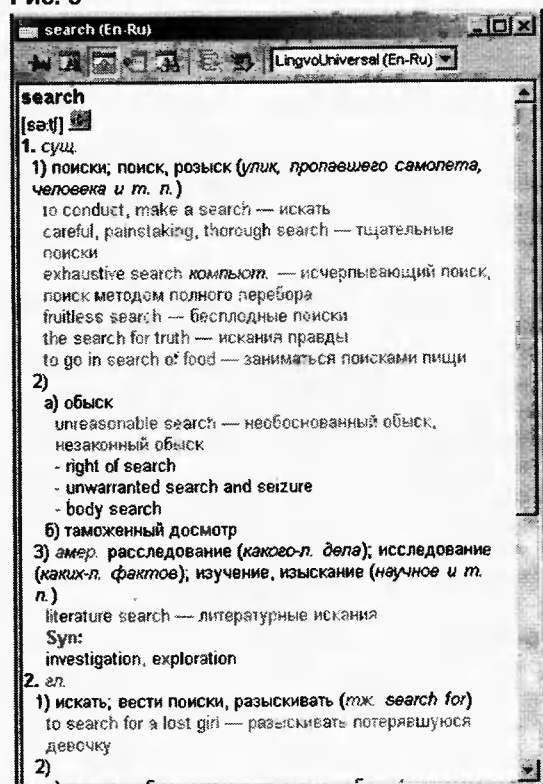
Благодаря входящему в официальную поставку Lingvo компилятору языка словарных описаний в формате DSL, каждый пользователь имеет возможность создать собственный словарь, который будет без всяких ограничений поддерживаться программой наряду с его "родными" словарными базами (при этом все возможности программной оболочки Lingvo останутся в вашем полном распоряжении). При этом карточки вновь созданных словарей могут быть выполнены максимально похожими по структуре на те, которые содержатся в базовых словарях Lingvo, а также

содержать все необходимые элементы оформления. Вновь созданные словари вы можете не только использовать для собственных нужд, но и поделиться плодами своих трудов с другими пользователями Lingvo, внося свой посильный вклад в развитие этого замечательного продукта. Разместить свои словари можно на сайте Ассоциации лексикографов Lingvo

(www.LingvoDA.ru), где уже находятся более 100 бесплатных словарей на самых разных языках (например, арабском, китайском, эсперанто) и самой всесторонней тематики (словарь аббревиатур, бухгалтерских терминов, словари идиом, словарь хакерской терминологии и др.), созданных другими пользователями Lingvo.

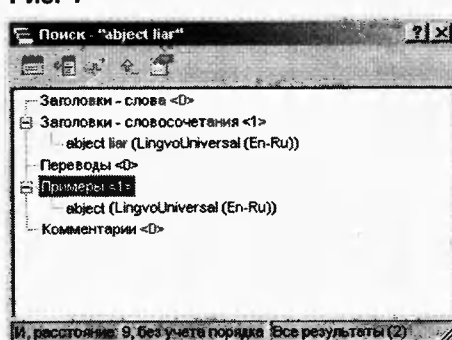
Одним словом, новинка у компании ABBYY получилась весьма интересной. Lingvo можно смело рекомендовать как переводчикам-полиглотам, так и обычным пользователям для работы в офисе и дома. Однако немаловажен и вопрос цены: вряд ли кого-нибудь заинтересует электронный словарь, стоящий дороже своего бумажного аналога. Впрочем, судя по информации от производителя, Lingvo 10.0 продается по вполне разумным, даже для нашей небогатой страны, ценам. Англо-русский словарь ABBYY Lingvo 10.0 стоит около 465 руб., а многоязычную версию можно приобрести за 955 руб.

Рис. 3



все установленные словари показаны отдельными пиктограммами, поэтому вы легко можете активировать или деактивировать любой из установленных словарей. Как только вы введете интересующее вас слово (или дважды "щелкните" по нему в списке слов), откроется соответствующая словарная карточка, в которой даны все варианты его перевода, толкования и примеры использования слов, показана транскрипция, а так-

Рис. 4



В программе довольно эффективно реализована функция поиска, что немаловажно при столь обширной словарной базе. Поиск нужного слова осуществляется во всех подключенных словарях (рис.4). Результаты поиска нужного слова или словоформы каждый раз появляются на экране в четко структурированном виде: сколько раз в каких словарях и в какой части каждой словарной статьи

было обнаружено искомое слово — в заголовке, примерах употребления, комментариях. Поиск термина по мере его набора ("инкрементный" поиск) упрощает просмотр словарных статей. Даже если во вводимом слове допущена ошибка, в постоянно обновляемом списке терминов легко можно отыскать правильное. Кроме того, программа способна перевести слово вне зависимости от того, в какой форме оно было задано пользо-

MATLAB. РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Б.КИСЕЛЕВ

г.Минск, БГАТУ,
каф.ВТ.

Дифференциальными уравнениями можно описать очень широкий круг задач, поэтому данная тема практически необъятна. Рассмотрим вводные понятия для численного решения обыкновенных дифференциальных уравнений (ДУ) и перечислим некоторые возможности MATLAB по решению такого рода задач. Покажем, что даже в простых задачах могут возникать ситуации, когда приходится серьезно задумываться, какую из имеющихся функций применять. Другими словами, наличие мощной математической системы не освобождает от необходимости владеть теорией и глубоко понимать задачу.

Краткие теоретические сведения

Рассмотрим простейшую задачу на примере ДУ первого порядка

$$y' = f(x, y), \quad (1)$$

где $f(x, y)$ — любая функция переменных x и y .

Решить ДУ (1) — значит найти такую функцию $y(x)$, которая удовлетворяла бы уравнению (1) при начальных условиях $x = x_0$ и $y = y_0 = y(x_0)$ (задача Коши).

Наиболее распространенная группа методов решения этой задачи использует разложение решения (1) в окрестности точки x_0 в ряд Тейлора [1]:

$$f(x) = f(x_0) + (x - x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \dots + \frac{(x - x_0)^n}{n!} f^{(n)}(x_0) + \dots \quad (2)$$

Применив (2) к решению $y(x)$, оставив первые два члена ряда (2) и заменив y' правой частью (1), получим:

$$y(x) = y(x_0) + (x - x_0)y'(x_0) = y(x_0) + (x - x_0)f(x_0, y_0),$$

или, используя сокращенную запись,

$$y = y_0 + (x - x_0)f_0. \quad (3)$$

Возьмем малое приращение Δx_0 и найдем решение в точке $x_1 = x_0 + \Delta x_0$:

$$y_1 = y(x_1) = y_0 + (x_0 + \Delta x_0 - x_0)f(x_0, y_0) = y_0 + \Delta x_0 f_0 = y_0 + \Delta y_0,$$

где $\Delta y = f(x, y)\Delta x$.

Аналогично найдем решение в следующей точке:

$$x_2 = x_1 + \Delta x_1; \quad y_2 = y_1 + \Delta x_1 f(x_1, y_1) = y_1 + \Delta x_1 f_1.$$

Продолжая таким образом, можно найти решение в любой точке n :

$$x_{n+1} = x_n + \Delta x_n; \quad y_{n+1} = y_n + \Delta x_n f_n. \quad (4)$$

Если шаг изменения $\Delta x = h = \text{const}$, то

$$x_{n+1} = x_n + h; \quad y_{n+1} = y_n + hf_n. \quad (5)$$

Выражения (4) и (5) являются простейшими вариантами формулы Эйлера для решения ДУ с заданными начальными условиями.

Например, решим уравнение $y' = 1 + x - y$ при $x_0 = 0$, $y_0 = 1$, $h = 0.1$. Найдем решение в точках: $x = 0.1, 0.2, 0.3, 0.4$:

$$y_1 = 1 + 0.1(1 + 0 - 1) = 1;$$

$$y_2 = 1 + 0.1(1 + 0.1 - 1) = 1.01;$$

$$y_3 = 1.01 + 0.1(1 + 0.2 - 1.01) = 1.029;$$

$$y_4 = 1.29 + 0.1(1 + 0.3 - 1.029) = 1.0561.$$

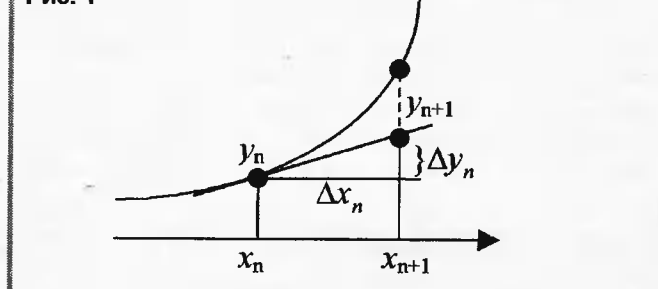
Точное решение в точке $x = 0.4$ будет $e^{-0.4} + 0.4 = 1.0703$.

Достоинством формулы Эйлера является ее простота,

кроме того, переход из предыдущей точки решения (n) в последующую ($n+1$) осуществляется за один шаг, т.е. знать более ранние точки не требуется. Такие формулы называются одношаговыми. Недостатком их является невысокая точность. Погрешность формулы Эйлера на одном шаге пропорциональна квадрату шага решения Δx^2 .

Геометрическая интерпретация формулы Эйлера проста, т.к. $y'(x)$ есть наклон касательной к решению $y(x)$ в точке x_n (рис.1). Из вывода (4) видно, что точка y_{n+1} лежит на этой касательной. Штриховой линией показана погрешность.

Рис. 1



Оставляя в (2) три члена, по аналогии с (5) получим:

$$y_{n+1} = y_n + h y'_n + \frac{h^2}{2!} y''_n.$$

Высшие производные можно вычислить через частные производные:

$$y'' = f'(x, y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} y'.$$

Используя (1) и промежуточную точку между n и $n+1$, можно представить члены последнего выражения через h , x_n , y_n и $f(x_n, y_n)$ [1]. Подобным образом можно добавлять следующие члены ряда (2), представлять производные более высоких порядков и получать различные формулы Рунге-Кутты. Наибольшее распространение получила формула Рунге-Кутты 4-го порядка. Она использует пять членов ряда (2), и ее простейший вариант имеет вид:

$$x_{n+1} = x_n + h \quad (6)$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

где k_i — коэффициенты Рунге-Кутты, которые рассчитываются по формулам:

$$k_1 = f(x_n, y_n)h;$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})h;$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})h;$$

$$k_4 = f(x_n + h, y_n + k_3)h.$$

Для иллюстрации рассчитаем один шаг приведенного выше примера с помощью (6):

$$k_1 = 0.1(1 + 0 - 1) = 0;$$

$$k_2 = 0.1(1 + 0.05 - 1) = 0.005;$$

$$k_3 = 0.1(1 + 0.05 - (1 + \frac{0.005}{2})) = 0.00475;$$

$$k_4 = 0.1(1 + 0.1 - (1 + 0.00475)) = 0.009525;$$

$$y_1 = 1 + \frac{1}{6}(0 + 2(0.005 + 0.00475) + 0.009525) = 1.0048375.$$

Точное решение: $e^{-0.1} + 0.1 = 1.004837418$. Метод Эйлера давал $y_1 = 1$.

Широкое распространение получили также многошаговые методы, которые построены на основе интерполяционных многочленов. Они проще и работают быстрее методов Рунге-Кутты при одинаковой точности, но для вычисления n -й точки требуют знания решения в нескольких предыдущих точках. Поэтому самые первые точки рассчитываются с помощью одношаговых методов. Кроме того, многошаговые методы вполне реализуют свои преимущества на относительно гладких функциях. Наиболее популярным среди многошаговых является метод Адамса.

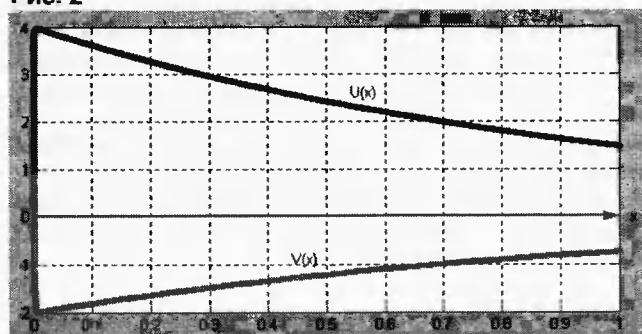
Обилие методов связано с обилием различных особенностей в решениях ДУ даже в "благополучных" на первый взгляд задачах. Например [2], система уравнений

$$\begin{aligned} u' &= 998u + 1998v \\ v' &= -999u - 1999v \end{aligned} \quad (7)$$

при $u(0)=v(0)=1$ имеет решение (рис.2):

$$\begin{aligned} u &= 4e^{-x} - 3e^{-1000x} \\ v &= -2e^{-x} + 3e^{-1000x} \end{aligned}$$

Рис. 2



После короткого начального промежутка решение близко к

$$\begin{aligned} u &= 4e^{-x} \\ v &= -2e^{-x} \end{aligned}$$

Если решать методом Эйлера с шагом 0.01, то получим:

$$\begin{aligned} u_1 &= 1 + 0.01(998 + 1998) = 30.96 & v_2 &= -239.8... \\ v_1 &= 1 + 0.01(-999 - 1999) = -28.98 & v_2 &= 241/04... \end{aligned}$$

Точное решение: $u_1 = 3.96006$, $v_1 = -1.97996$; $u_2 = 3.92079$, $v_2 = -1.96040$.

Происходящее можно пояснить рис.3, где представлены интегральные кривые переменной u , на которые попадают точки нашего решения. Быстрая переходная часть решения, которая, казалось бы, давно исчезла, на самом деле на каждом шаге уводит решение в сторону. Так, из точки u_i мы проводим касательную до абсциссы x_{i+1} . В силу

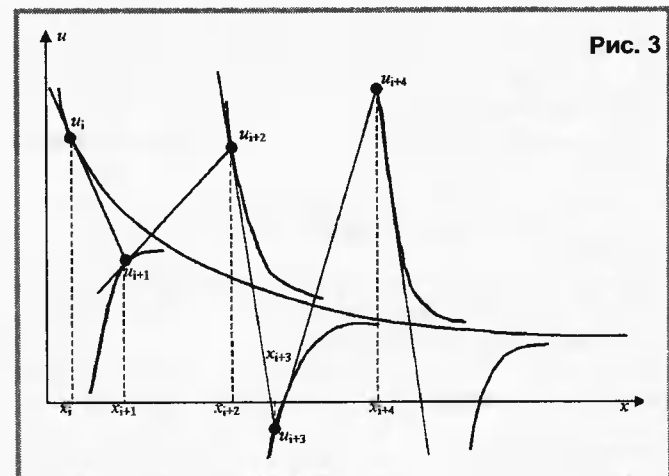


Рис. 3

того что мы отклонились от истинного решения, снова появляется быстрая составляющая, и т.д. Задачи такого типа называют жесткими. Шаг решения надо выбирать малым, чтобы за его величину быстрая составляющая изменялась незначительно, но тогда общее число шагов будет велико, и время вычислений — тоже. Это приводит к накоплению ошибок округления, поэтому для жестких задач используют другие алгоритмы. Простейшим примером является обратный метод Эйлера. Он использует неявную формулу:

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}). \quad (8)$$

На каждом шаге приходится решать уравнение относительно y_{n+1} . Физически это означает (рис.4), что касательную мы проводим не к точке (x_n, y_n) , а к следующей точке (x_{n+1}, y_{n+1}) .

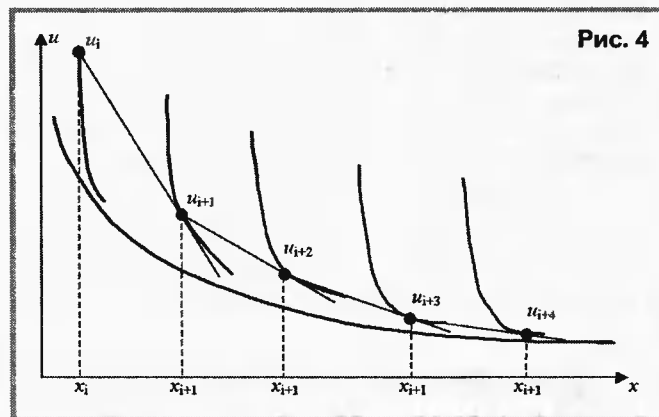


Рис. 4

Для рассмотренного примера первый шаг, рассчитанный по формуле (8), даст:

$$\begin{aligned} u_1 &= 1 + 0.1(998u_1 + 1999v_1) \\ v_1 &= 1 + 0.1(-999u_1 - 1999v_1) \end{aligned}$$

Решая эту систему линейных уравнений относительно u_1 и v_1 , получим: $u_1 = 4.3975$, $v_1 = -2.1806$.

В связи с обилием методов решения ДУ система MATLAB имеет целый набор "решателей" (solver) ДУ для разных задач [3]. Решатели обычно содержат в названии буквы **ode** (Ordinary Differential Equation) и номер, например, **ode45** — решатель методом Рунге-Кутты 4-го порядка. Схема решения состоит из следующих этапов:

1. Приведение дифференциального уравнения к системе дифференциальных уравнений 1-го порядка.
2. Написание специальной файл-функции (m-функции) для системы уравнений.
3. Вызов подходящего решателя.
4. Визуализация результата.

Пример применения решателя

Исследуем процесс, описываемый дифференциальным уравнением

$$a_2 \frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y = x(t) \quad (9)$$

при начальных условиях: $t = t_0$, $y = y_0$, $y' = y'_0$.

Такие уравнения описывают движение колебательных систем — груза, подвешенного на пружине; маятника; электрического RLC-контура и т.д. — под воздействием возмущения $x(t)$.

Пусть значения коэффициентов уравнения (9) следующие: $a_0 = 1$, $a_1 = 0.04$, $a_2 = 0.01$, начальные условия —



нулевые, а $x(t) = 1(t)$ (ступенчатое воздействие). Введя обозначения $y_1 = y$, $y_2 = y'$, уравнение (9) приведем к виду:

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \frac{1}{a_2}(x(t) - a_1 y_2 - a_0 y_1) \end{aligned} \quad \text{или} \quad \begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \frac{1}{0.01}(1 - 0.04 y_2 - y_1) \end{aligned} \quad (10)$$

$$\begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Напишем m-функцию, которая вычисляет правые части системы уравнений (10). Функция должна иметь не менее двух входных аргументов: переменную x , по которой производится дифференцирование, и вектор переменных y . Для нашего примера такая m-функция будет иметь вид:

```
function F=difur(x,y)
F=[y(2); 100*(1-0.04*y(2)-y(1))];
```

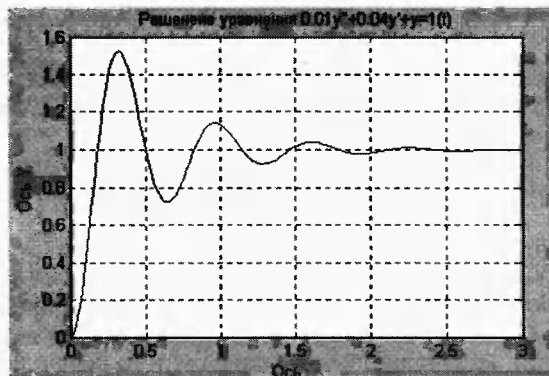
Эта функция будет вызываться решателем каждый раз, когда потребуется вычислить правые части уравнений в конкретной точке x .

Теперь для решения уравнений (10) можно вызвать функцию **ode45**, простейшая форма обращения к которой:

```
[X, Y] = ode45(@difur,[0, 3],[0, 0]);
```

Здесь **difur** — имя функции, вычисляющей правые части (10); символ **@** сообщает системе, что это имя функции, а не переменной; **[0, 3]** — интервал, на котором ищется решение; **[0,0]** — начальные условия.

Рис. 5



Отобразим решение $y = y_1$ на графике (рис.5):

```
plot(X, Y(:,1), 'k'); grid
title('Решение уравнения 0.01y''+0.04y'+y=1(t)')
xlabel('Ось X'); ylabel('Ось Y')
```

Ко всем решателям задачи Коши форма обращения одинакова. Самая простая форма обращения:

```
[T,Y] = solver(odefun,tspan,y0)
```

Более полная:

```
[T,Y] = solver(odefun,tspan,y0,options,p1,p2,...)
```

Существуют другие формы обращения (см. справочную систему Matlab).

Здесь **solver** — один из решателей: **ode45**, **ode23**, **ode113**, **ode15s**, **ode23s**, **ode23t**, или **ode23tb**.

Аргументы:

- **odefun** — функция, которая вычисляет правые части системы уравнений. Все решатели решают систему уравнений вида (1), и все, за исключением **ode23s** — вида $M(t,y)y' = f(x,y)$;

- **tspan** — вектор, задающий интервал интегрирования **[t0,tf]**. Чтобы получить решение в определенных точках (по возрастанию или убыванию), следует задать эти точки **tspan = [t0,t1,...,tf]**;

- **y0** — вектор начальных условий;

- **options** — интегральный аргумент, создаваемый функцией **odeset** (см. ниже: пример 2);

- **p1,p2...** — произвольные параметры, передаваемые в **odefun**;

- **T,Y** — матрица решений **Y**, где каждая строка соответствует времени, возвращенному в вектор-столбце **T**.

Запомним некоторые рекомендации по применению решателей [3].

Часто решатель **ode45** дает хорошие результаты, поэтому при отсутствии опыта рекомендуется начинать вычисления с него. Решатель **ode23** тоже использует формулу Рунге-Кутты, но более низкого порядка точности, соответственно, более быструю, и рекомендуется для слабо жестких задач. Для решения нежестких задач с высокой точностью хорошо решатель **ode113** (использует метод переменного порядка Адамса-Башфорта-Милтона). Он особенно выгоден, когда правые части уравнений вычисляются по сложным формулам.

Для решения жестких уравнений подходит решатель **ode15s**, основанный на многошаговом методе Гира. Если требуется решить жесткую задачу с невысокой точностью, то подойдет решатель **ode23s**, реализующий метод Розенброка второго порядка.

Все решатели пытаются найти решение с относительной точностью 10^{-3} .

За более подробной информацией обращайтесь к справочной системе и литературе.

Рассмотрим два примера.

Пример 1

Пример иллюстрирует влияние жесткости. Он имеется в справочной системе MATLAB и почти во всех общих книгах по этой системе.

Решим систему нелинейных ДУ Ван-дер-Поля, которая имеет вид:

$$\begin{aligned} \dot{y}_1 &= y_2; \\ \dot{y}_2 &= -y_1 + K(1 - y_1^2)y_2, \end{aligned}$$

здесь K — коэффициент нелинейности задачи, а сама нелинейность — в круглых скобках.

В качестве начальных условий возьмем $y_1 = 2$, $y_2 = 0$.

Создадим файл **vdp.m**, который вычисляет вектор правых частей:

```
function F=vdp(x,y,K)
F=[y(2); -y(1)+K*(1-y(1)^2)*y(2)];
```

Для решения уравнений используем функцию **ode45** — решатель, использующий классический метод Рунге-Кутты 4-го порядка:

```
K=2;
[X, Y] = ode45(@vdp,[0, 30],[2, 0],[],K);
plot(X, Y(:,1),X, Y(:,2))
```

Результат представлен на рис.6.

Если поменять значение коэффициента нелинейности на 200, т.е.

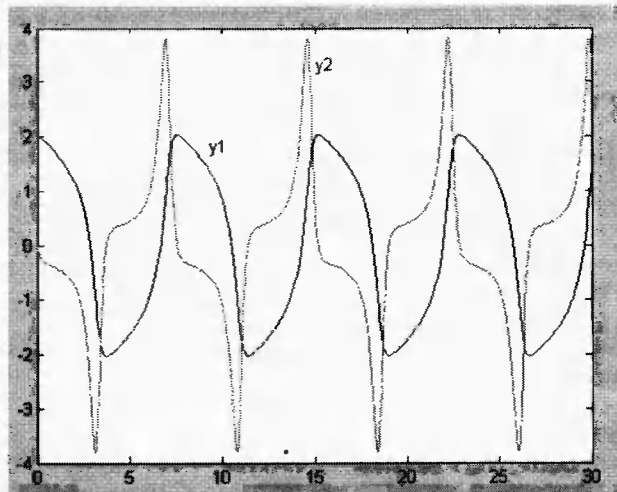
```
K=200;
[X, Y] = ode45(@vdp,[0, 1000],[2, 0],[],K);
plot(X, Y(:,1)),
```

то увидим, что метод Рунге-Кутты плохо справляется с этой задачей (при очень больших K вообще не справляется). Здесь увеличен интервал решения уравнений, т.к. сильно увеличивается период колебаний, и на график выведена только одна переменная y_1 , из-за большой разницы в амплитудах y_1 и y_2 .

Известно, что задача Ван-дер-Поля при больших K яв-



Рис. 6

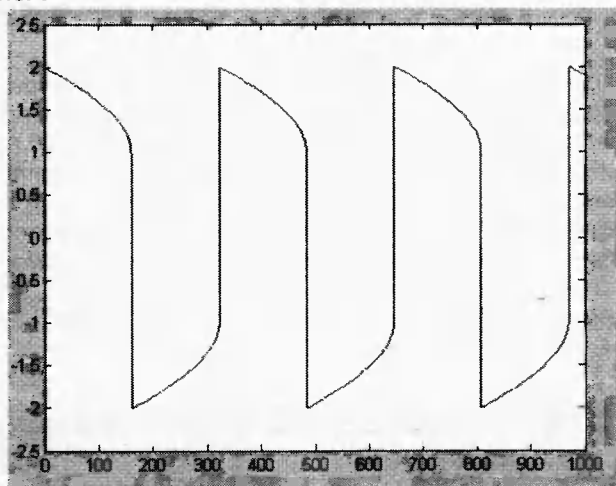


ляется жесткой, поэтому подберем другой решатель, например, **ode15s**, и введем:

```
K=200;
[X, Y] = ode15s(@difur,[0, 1000],[2, 0],[],K);
plot(X, Y(:,1))
```

Результат представлен на рис.7. При этом время решения практически не изменилось по сравнению со случаем $K=2$.

Рис. 7



Пример 2

Пример иллюстрирует влияние точности вычислений [3]. Выше упоминалось, что для управления работой решателей используется параметр **options**, который следует формировать функцией **odeset**. Общая форма записи: **options = odeset(..., 'имя опции', значение,...)**

Полный список имен опций и их значений есть в справочной системе. Мы будем использовать эту функцию для задания относительной точности решения (имя опции **'RelTol'**).

Решим систему уравнений:

$$\begin{cases} y_1' = y_2 \\ y_2' = -\frac{1}{t^2} \end{cases}$$

на отрезке $[a, 100]$ при начальных условиях $y_1(a) = \ln a$, $y_2(a) = 1/a$ и $a = 0.001$.

Точным решением этой системы является $y_1 = \ln t$, $y_2 = 1/t$.

Составим m-функцию для вычисления правой части системы:

```
function dydt=rsbad(t,y)
```

```
dydt=[y(2); -1/t^2];
```

Решим задачу для трех различных значений относительной точности и сравним результаты с точным решением. Надеемся, что из текста программы понятно использование функции **odeset** и параметра **options**.

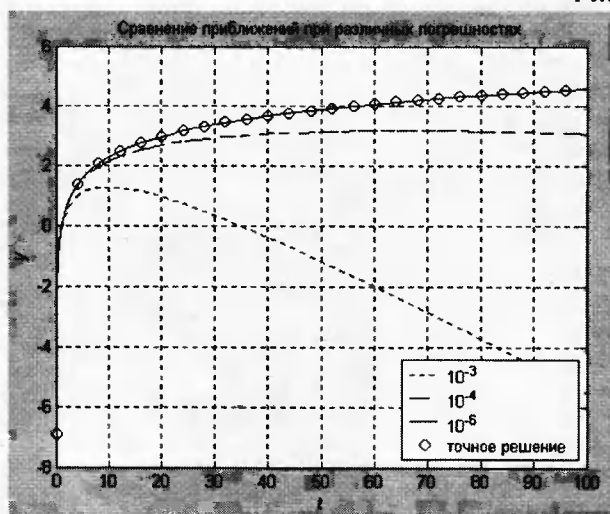
```
a = 0.001;
% задание начальных условий
Y0 = [log(a); 1/a];
% вызов решателя ode45 при различных погрешностях
% и вывод графиков приближенных решений
```

```
% точность 0.001 (по умолчанию)
[T, Y] = ode45(@rsbad, [a 100], Y0);
plot(T, Y(:,1), 'k-')
% точность 0.0001
options = odeset('RelTol', 1.0e-4);
[T, Y] = ode45(@rsbad, [a 100], Y0, options);
hold on
plot(T, Y(:,1), 'k-')
% точность 0.000001
options = odeset('RelTol', 1.0e-6);
[T, Y] = ode45(@rsbad, [a 100], Y0, options);
plot(T, Y(:,1), 'k-')
```

```
% вывод графика точного решения
t = [a:4:100];
y = log(t);
plot(t, y, 'ko')
% нанесение информации на график
xlabel('\itt')
ylabel('\ity')
title('Сравнение приближений при различных погрешностях')
legend('10^{-3}', '10^{-4}', '10^{-6}', 'точное решение', 4)
grid on
hold off
```

Результаты представлены на рис.8. Видна существенная зависимость результатов от точности решения. Только при относительной точности 10^{-6} численное решение практически совпадает с точным.

Рис. 8



Возможности MATLAB не исчерпываются решением задачи Коши. Функция **bvp4c** предназначена для численного решения граничных задач для обыкновенных ДУ произвольного порядка и систем. Функция **dsolve** позволяет найти аналитическое решение ДУ и систем.

Пакет ToolBox PDE (Partial Differential Equations) позво-



ляет решать стационарные и нестационарные задачи, описываемые ДУ в частных производных. Заметим, что эта тема достаточно активно обсуждается в форуме [4].

Упражнения

1. Решите систему уравнений (7) разными решателями.
2. Исследуйте модель Лотка-Вольтерра борьбы за существование [3]. Обозначим $y_1(t)$ — число жертв, $y_2(t)$ — число хищников. Число жертв и хищников в течение времени t изменяется по закону

$$\begin{cases} y_1' = Py_1 - py_1y_2 \\ y_2' = Ry_2 + ry_1y_2 \end{cases}$$

где P — увеличение числа жертв в отсутствие хищников; R — уменьшение числа хищников в отсутствие жертв. Вероятность поедания хищником жертвы пропорциональна их числу y_1y_2 , при этом слагаемое $-py_1y_2$ соответствует вымиранию жертв, а ry_1y_2 — появлению хищников. Для примера взять $P = 3$, $R = 2$, $p = r = 1$. Считать, что в начальный момент времени было три жертвы и четыре хищника. Решение отобразить на плоскости y_1y_2 (должна получиться замкнутая кривая). Решить систему

для $t = 100$. Попробуйте два решателя **ode45** и **ode 23s** при точности задаваемой по умолчанию, т.е. 10^{-3} . Сравните результаты.

Вопросы для самопроверки

1. Какие существуют методы решения ДУ?
2. Что значит решить ДУ?
3. В чем заключается сущность метода Эйлера?
4. В чем заключается сущность метода Рунге-Кутты?
5. Почему существует много формул для решения ДУ?
6. Чем отличаются одношаговые методы от многошаговых?
7. Когда ДУ являются жесткими?
8. Как решить ДУ с помощью решателей **ode** системы MATLAB?

Литература

1. Калиткин Н.Н. Численные методы. — М.: Наука, 1978.
2. Форсайт Д., Малькольм М., Моултер К. Машинные методы математических вычислений. — М.: Мир, 1980.
3. Ануфриев И.Е. Самоучитель Matlab 5.3/6.x. — СПб.: БХВ-Петербург, 2002.
4. <http://matlab.ru>

С.РЮМИК,
г.Чернигов.

Usus est optimus magister (nam.)
(Опыт — наилучший учитель)

Интернет-калейдоскоп, freeware #18

Что такое компьютерное эсперанто или интернет-эсперанто? Иногда таким словом обозначают универсальный метаязык, служащий для автоматического перевода сообщений с любого языка на любой. Представим, как удобно было бы вести электронную переписку, используя интернет-эсперанто. К примеру, составил письмо на родном языке, отправил его в Интернет. По дороге к адресату оно преобразуется в словоформы метаязыка, а затем транслируется в родной язык получателя с полным сохранением смысла.

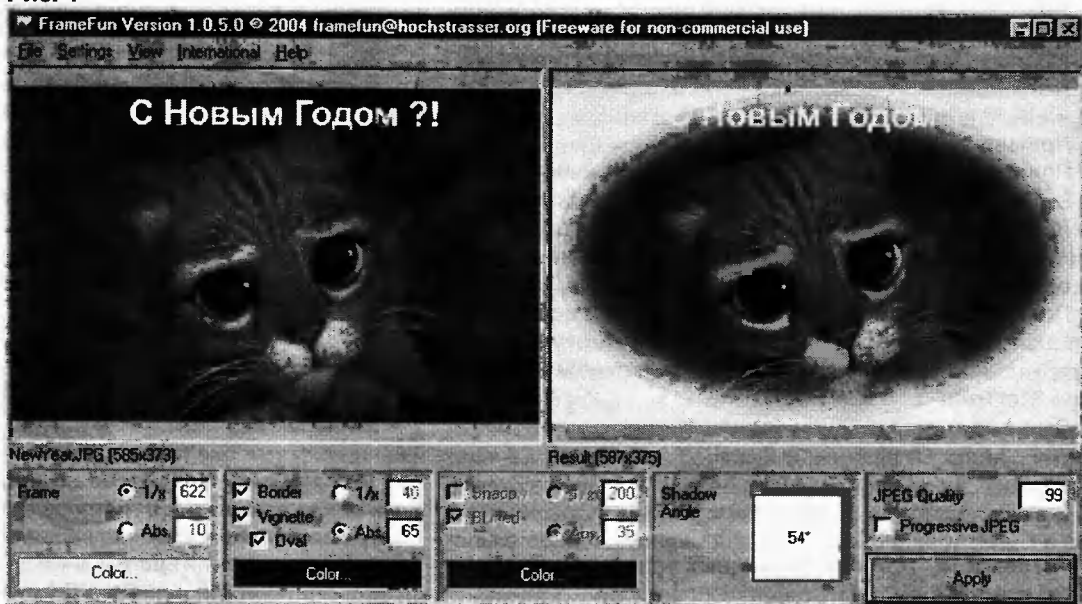
Появление подобных виртуальных переводчиков не за горами. В конце 90-х годов японские специалисты по искусственному интеллекту начали работу над компьютерным эсперанто в рамках проекта UNL (Universal

Networking Language). Сейчас этот проект стал международным, но оказалось, что для отладки всех его языковых модулей требуется больше времени, чем планировалось.

Тем, кто не хочет ждать, можно потренироваться на обычном эсперанто, тем более, что его легко изучить всего за 16 дней (<http://www.esperanto.mv.ru/Esp16/index.html>)! В Интернете представлены

популярные российские эсперанто-страницы <http://esperantio.narod.ru>, <http://evortaro.narod.ru>. Для интеллигентного человека к лицу знать хотя бы несколько слов на этом универсальном языке общения, например, "Magazino VIA KOMPUTERO", что означает "Журнал ВАШ КОМПЬЮТЕР".

Рис. 1



Личный календарь в объемной рамке

Календарь — это вещь полезная и не дефицитная. Но как сделать единственный в мире календарь для себя, любимого, для души? Программа CalendarPainter (автор — Waseo, <http://www.waseo.de/Freeware/Progs/CPainter/cpsetup.exe>, 1,8 Мб) предоставляет такую возможность. "Строительство" календаря происходит в 6 этапов, по многочисленным шаблонам и подсказкам. Естественно, красными днями в таком календаре станут не только государственные праздники, но и собственные знаменательные даты, вводимые вручную. Рисунки и фото к календарю импортируются в форматах *.bmp, *.jpg, поддерживается кириллица, поквартальное и годовое размещение месяцев на листе.

Чтобы календарь выглядел стильно, надо создать вокруг него объемную 3D-рамку. Программа FrameFun (автор — Benedikt Hochstrasser, <http://www.hochstrasser.org/wiki/files/FFSetup-1.0.5.0.exe>, 1,2 Мб) позволяет это сделать в интерактивном режиме с установкой произвольного угла тени, ее глубины, овальной или прямоугольной формы рамки, цвета фона, "дымчатого" эффекта. Подобным образом можно обрабатывать не только календари, но и любые рисунки, фотографии, слайды, придавая им изящные формы (рис.1).

Сам себе герольдмейстер

Что такое геральдика? Не спешите отвечать. В книгах можно найти сразу несколько объяснений. Название произошло от слова "herold" (польское "herb", чешское "erb"), которым называли знатока гербов, официально удостоверявшего правильность их составления.

Расцвет геральдики приходится на эпоху средневековья, когда на щите у каждого рыцаря красовалась неповторимая эмблема. Если бы в те времена существовали компьютеры, то рыцари могли бы конструировать себе гербы при помощи программы "Shields, Knights and Heraldry" (автор — Owl & Mouse Educational Software, <http://www.yourchildlearns.com/herinst.exe>, 1,3 Мб). Здесь выбираются размеры щита, тип и количе-

ство геральдических фигур. После распечатки на принтере, детали раскрашивают и склеивают вместе (рис.2).

Рис. 2



Следует знать, что в геральдике не существует произвольных цветовых оттенков, каждый цвет несет определенную смысловую нагрузку, например, символ надежды, радости, достоинства, силы. Различают цвета металлов (серебро, золото), финифтей (красный, голубой, зеленый, черный, коричневый, пурпурный, серый, телесный), мехов (соболь, куница, горностай). Главное правило — чтобы металл не накладывался на металл, финишь на финишь, мех на мех [1].

Можно ли в наше время разработать свой собственный герб и легитимно передавать его по наследству от отца к детям? Такая практика существует, причем в международном масштабе (<http://sovets.geraldika.ru/article/10088>), но называется она не пожалование герба, а самостоятельное принятие герба. Разумеется, если программист захочет изобразить на гербе компьютер, то рисовать его надо максимально стилизованным (детализировка не поощряется). Кроме того, предстоит придумать лаконичный девиз, который располагается на ленте обрамления. В качестве идеи на будущее — сделать русскоязычный аналог программы "Shields, Knights and Heraldry" с воз-

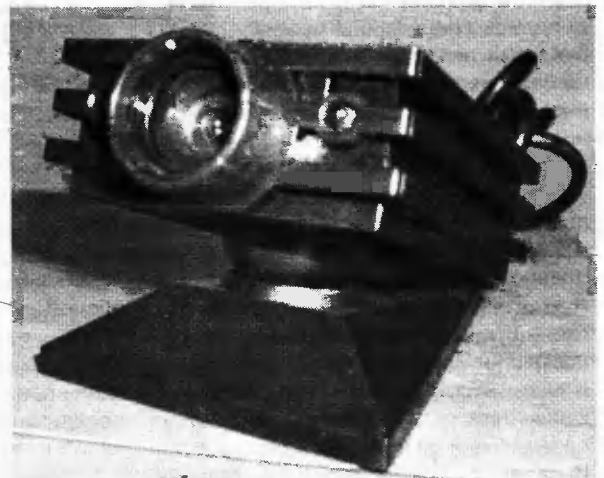
можностью компьютерной раскраски и распечатки гербов, выбора французской, итальянской или польской формы щита, количества разделителей и т.д.

Web-камера от игровой приставки

Домашняя видеокамера является желанным, но очень дорогостоящим элементом быта. Однако японская фирма Sony для игровой приставки "PlayStation 2" разработала видеокамеру "EyeToy" (рис.3), которая продается по цене 35...45 USD. Умельцы сразу сообразили, как подключить ее через порт USB к персональному компьютеру и привели подробную методику инсталляции для Windows 98/XP на странице <http://www.ionstorm.co.uk/files/sairuk>.

Из полезных сфер применения "webcam" — системы охранной сигнализации, видеодомофон, монитор удаленного наблюдения через Интернет за происходящими в квартире событиями. Можно разместить на своем сайте специальное окошко с web-камерой по аналогии с многочисленными виртуальными интернет-"биноклями", списки которых приведены на сайтах <http://webcam.ru>, <http://webcam-online.narod.ru>, <http://webcam.mnc.ru> (Москва, вид с МГУ).

Рис. 3



Ключевые слова для поиска: компьютерное эсперанто, Calendar Painter, геральдика, EyeToy.

Литература

1. Винклер П.П. Гербы городов, губерний, областей и посадов Российской империи. — СПб, 1899 г. (веб-версия — http://www.antiquebooks.ru/history_of_russia).



А.КАШКАРОВ,
г.С.-Петербург.

Разговаривая с оппонентом — всегда улыбайся...

*Половину жизни ты работаешь на свое имя,
а другую половину — имя работает на тебя.
Философ, имени которого я не помню.*

Откуда что пошло

С самых первых шагов развития сети в народе появилось понятие "флеймз" — оскорбления, имеющие место в электронном общении. Я бы не сказал, что это новая проблема, но она остра и, к сожалению, не замечать ее не удастся. Корни проблемы уходят к "воспитанию с младых ногтей" и последующему опыту конкретных индивидуумов, поэтому ее базис скорее психологический, чем искусственный, приобретенный, и типичный определенному возрасту, географическому контингенту или лицам с невысоким образовательным уровнем. Человек может быть высокообразован, иметь заслуги и награды, но остаться невоспитанным. Тут уж, наверное, ничего не поделаешь. Чтобы не попасть в положение оскорбляющего или оскорбленного (что в обоих случаях, на мой взгляд, неприемлемо) существуют несложные правила-заповеди, защищающие пользователей в сетях. Как они появились?

Пока компьютеризация всей страны не стала масштабной, основными пользователями сети Интернет были работники официальных учреждений и научных институтов. Порядок и способы общения в сети определялись специальными инструкциями, не выполнить которые не позволяло то самое воспитание, кроме того, такое поведение считалось дурным тоном, поэтому этикет основывался на устоявшихся в научных кругах правилах обмена информацией, и без координатных изменений плавно перешел в норму современного общения в сети — от нее сегодня отдает симбиозом интеллектности и нигилизма. Приобретенный нигилизм — скорее, норма времени, такая же, как тусовка (тусня), со своими правилами и обществом, а общество в Интернете нельзя назвать поголовно пожилым, а скорее, наоборот.

Многие используют Интернет именно в тех целях, для которых он создавался — для поиска информации. Для других Интернет стал местом удовлетворения личных амбиций. Для третьих Интернет из информационной "паутины" превратился в развлечение, оставаясь при этом источником информации и связи. Всевозможные чаты (конференции), интернет-пейджеры, голо-

совые сессии в реальном времени и электронная почта — все атрибуты сетевого общения предполагают некие заповеди, нарушение которых, очевидно, приведет к хаосу и вседозволенности в сети, что вряд ли понравится уважающему себя пользователю.

Сегодня ситуация такова, что если определенный хостинг обслуживает модератор или администратор сайта, контролирующий поток информации и являющийся цензором, не пропускающим оскорбления, горячие эмоции или некорректные формы критики, то сообщения по e-mail практически лишены контроля эмоций и всплесков ненормативной лексики.

Между тем нормы этикета важны как для пользователей, так и для тех, кто их обслуживает. В электронных конференциях пользователей в процессе обсуждения появились новая норма поведения — нетикет (netiquette, от английского net — "сеть" и французского



etiquette — "этикет"). Ее обсуждение открыто и актуально по сей день, хотя и появились некоторые основополагающие правила. В зависимости от вида использования Интернета набор этих правил может быть разным.

Этикет в локальных компьютерных сетях

Правила поведения пользователя в локальных сетях (Интернет) уважающих себя обществ (организаций) с любой формой собственности устанавливаются в виде должностных инструкций. Но вот ведь где "собака порылась". Существует немало организаций, в которых каких-либо правил и инструкций не существует, да и не любая инструк-

ция может охватить все вопросы этикета. Тут уж все дело в воспитании.

Вот некоторые общие правила (хотя применение того или иного правила зависит от технического оснащения сети):

- не передавайте никому ваше имя и пароль для входа в сеть: любые действия, совершенные в сети под вашим именем, потом могут быть соотнесены непосредственно с вами;

- если вы оставляете компьютер без присмотра более чем на 10 минут, перед уходом прекратите выполнение всех программ с сетевой поддержкой (или связанных с обменом данными по сети) и закройте их (если это невозможно сделать в силу выполняемой задачи, то предупредите об этом факте сисадмина);

- старайтесь без необходимости не запускать несколько программ с сетевой поддержкой;

- прежде чем начать перемещение в сети большого объема данных с компа на комп, оцените необходимость этого действия, возможность архивации и разбиения данных на отдельные, меньшие по объему пакеты. Только в случае невозможности рекомендуемого варианта прибегайте к перемещению всех данных;

- при наличии у вашего ПК жесткого диска отдавайте предпочтение сохранению данных именно на нем, а не на дисках общего пользования (если таковым не является диск вашего компьютера, и если это правило не противоречит решаемым задачам);

- пользуясь общим (системным) почтовым ящиком, старайтесь избегать помещать туда большие сообщения (объемом более 1 Мб — зависит от конкретного ПК);

- перед установкой на ПК нового программного обеспечения с сетевой поддержкой или с возможным коллективным использованием проконсультируйтесь с сетевым администратором и проверьте программное обеспечение на лицензионную чистоту и чистоту от вирусов;

- следите за тем, чтобы работающие программы не наносили вред каким-либо общим (сетевым) ресурсам и ресурсам других пользователей сети.

Применение коллективного принтера накладывает на пользователей локальной сети дополнительные правила:

- следите, чтобы не распечатывались лишние копии отправленного вами задания;

- старайтесь не распечатывать что-

либо (документ и т.п.) сразу после завершения каждого незначительного изменения — многие программы позволяют просмотреть образец возможной распечатки на экране;

- следите за тем, чтобы ваши распечатки не скапливались у принтера — забирайте их, по возможности, сразу после окончания печати.

При появлении вопросов по эксплуатации сети или программ, использующих сетевые ресурсы, обратитесь к сетевому администратору (сисадмину) или воспользуйтесь соответствующей документацией.

Правила интернет-этикета для детей (которые возможно окажутся полезными и опытным взрослым):

- не сообщайте частной информации о себе (фамилию, номер телефона, адрес, номер школы) без разрешения родителей;

- встреча в реальной жизни со знакомыми по интернет-общению — не всегда хорошая идея, поскольку очевидна разница в электронном общении и при реальной встрече. Если вы все же хотите встретиться, сообщите об этом родителям, чтобы они знали, где вас искать;

- не открывайте письма электронной почты, файлы или Web-страницы, полученные от людей, которых вы реально не знаете. Так можно приютить вирус;

- никому не давайте свой пароль;
- будьте вежливыми в электронной переписке, и ваши корреспонденты будут вежливыми с вами;

- в электронных письмах не применяйте текст, набранный в ВЕРХНЕМ РЕГИСТРЕ — это воспринимается в сети как крик и может расстроить вашего собеседника. Вместо этого можно воспользоваться более корректным выражением эмоций — смайликами :), :(и т.д.;

- не присылайте в письме информацию большого объема (картинки, фотографии и т.п.) без предварительной договоренности с вашим собеседником;

- не рассылайте писем с какой-либо информацией незнакомым людям без их просьбы — это воспринимается как "спам" и досаждают пользователям;

- ведите себя в сети так, как бы вы хотели, чтобы вели себя с вами!

Десять основных заповедей этикета Интернета

1. Помните о человеке!

Даже через посредство "мертвой" сети и напичканного электроникой компьютера вы общаетесь с живым человеком, а часто — со многими людьми одновременно... Не позволяйте одурманить себя атмосферой анонимности и вседозволенности — помните, что на другом конце провода такой же человек, как и вы...

Сочиняя электронное послание, представляйте, что все это говорите человеку прямо в лицо и старайтесь, чтобы при этом вам не было стыдно за свои слова. Отсюда второе правило:

2. Следуйте в сети тем же правилам, которым следуете в реальной жизни!

Нарушение законов человеческого общения, моральных норм в сети, возможно, и пройдет для вас относительно безнаказанным... Но будет ли при этом чиста ваша совесть? Осадок, оставшийся у оппонента, может повлиять и на его поведение в сети, а это уже цепная реакция, отголоски которой, если ее не контролировать, обязательно ощутите и вы сами.

3. Вы находитесь в киберпространстве!

Его границы куда шире, чем границы привычного человеческого общества, и в разных его частях могут действовать свои законы. Поэтому, сталкиваясь с новым для вас видом общения в сети, изучайте его законы и признавайте их приоритет. Скажем, в любой группе новостей, форуме или даже канале IRC существуют собственные, локальные правила (rules) — ознакомьтесь с ними, прежде чем отправлять свое первое сообщение! А главное — помните о неписаных правилах: например, правиле четвертом:

4. Бережно относитесь ко времени и мнению других людей!

Обращайтесь за помощью только тогда, когда это действительно необходимо — тогда вы всегда можете рассчитывать на помощь и поддержку. Однако не дергайте других пользователей по пустякам, иначе с вами просто перестанут общаться — ведь в общении заложен принцип добровольности. Сетевое время не только ограничено, но и для многих весьма дорого! И помимо ваших проблем, у собеседников могут быть еще и собственные... Однако этот принцип имеет и обратную сторону, зафиксированную в правиле пятом:

5. Старайтесь выглядеть достойно в глазах собеседников!

Не экономьте время на "условностях" хорошего тона и правил грамматики и орфографии. Даже комплименты теряют в убедительности, будучи воплощенными в форме:

"Прзэт чувак я тащусь от тебя и твaих книг круто пиши ищo"

Хотите ли вы сами получить такое письмо?

Не каждый любит также получать письмо, в котором вначале показан текст его исходного сообщения. Такой диалог допустим только в случаях деловой переписки, например с редакциями журналов или с заинтересованными лицами.

Из этого правила вытекает правило шестое:

6. Не пренебрегайте советами знатоков и делитесь своими знаниями с друзьями!

Будьте благодарны тем, кто тратит свое время, отвечая на ваши вопросы. Но и сами, получив письмо с вопросом от другого пользователя, не спешите отправлять это послание в мусорную корзину, каким бы нелепым и наивным оно ни казалось. Отсюда правило седьмое:

7. Сдерживайте страсти.

Вступать в дискуссии никакой этикет не запрещает, однако не опускайтесь до брани и ругательств — пусть даже ваш визави сознательно провоцирует на это.

8. Относитесь с уважением не только к своей, но и к чужой приватности!

Если по каким-то причинам вы хотите сохранить анонимность в сети, признавайте эти права и за собеседником. Более того — он имеет право на анонимность и приватность, даже если вы выступаете "с открытым забралом". Побочное следствие из этого правила: не публикуйте информацию из ваших частных писем без согласия их отправителей, не копайтесь в чужих почтовых ящиках и, тем более, в чужих компьютерах! Господа хакеры, это относится к вам... Так же, как и следующее правило:

9. Не злоупотребляйте своей властью и влиянием в сети!

Завоевать доверие трудно, а потерять весьма легко!

10. Будьте терпимы к недостаткам окружающих людей!

Не смотрите на то, соблюдают или нет собеседники правила сетевого этикета, соблюдайте их сами! В конце концов, предельно вежливо порекомендуйте собеседнику ознакомиться с этими правилами...

А для тех, кто не может или не хочет выучить эти заповеди, наконец, самое важное правило:

"Разговаривая со свидетелем, всегда улыбайся".

Этот совет Глеба Жеглова актуален в интернет-общении не меньше, чем в реальной жизни. Искренняя улыбка, даже если ее никто не видит, не может генерировать отрицательных эмоций, а поэтому подсознательно защищает пользователя и его собеседника от нежелательных последствий сессии.

Наверное, слишком громким заявлением было бы то, что эти несложные правила явятся панацеей от всевозможных бед электронного общения, но если их соблюдать, то вам точно не придется делать себе, с сожалением, виртуальную "сепуку" — обряд харакри у самураев, а вашим корреспондентам — купировать полученные от вас отрицательные эмоции.



ЛИСТАЯ СТРАНИЦЫ

Как можно защитить авторские права на выставленные в Интернете материалы, в частности изображения, рассказывает С.Шляхтина в статье "Водяной знак для фотографии" (КомпьютерПресс, №1/2005, С.177...181)

Водяные знаки для защиты денежных купюр от подделки начали использовать уже очень давно, сегодня все более широкое распространение получают программные продукты для защиты авторских прав, позволяющие добавлять водяные знаки к изображениям. Технологии подобной защиты также разработаны для аудио- и видеофайлов, черно-белых текстовых документов.

Водяные знаки могут быть как видимыми (или печатными), так и скрытыми (или цифровыми). Печатный водяной знак представляет собой нечто видимое (подпись или логотип автора). Такие водяные знаки показывают, что данное изображение принадлежит конкретному автору или компании. Однако опытный компьютерный график при желании сможет удалить печатный водяной знак или замкнуть его на другой.

Цифровой водяной знак — это цифровой код (как правило, он представляет собой идентификационные данные — ID, URL, адрес электронной почты, логотип и пр.), вставленный в цифровое изображение, чтобы идентифицировать информацию об авторских правах. В отличие от печатного водяного знака, он невидим, при

этом биты, представляющие водяной знак, разбросаны внутри графического файла, поэтому они не могут быть идентифицированы или изменены. Цифровой водяной знак выдерживает такие изменения файла как масштабирование, вращение, компрессия с потерями и др.

Порой это единственный способ доказать в дальнейшем свои авторские права на созданные рисунки, фотографии и элементы дизайна.

Компьютерные технологии позволяют изменить любое изображение до полной неузнаваемости, и при необходимости опытные специалисты-фальсификаторы могут сделать монтаж так, что обнаружить подлог будет практически невозможно. В то же время, в некоторых случаях очень важно знать, была ли совершенна подделка полученного цифрового изображения или нет. Речь идет об отпечатках пальцев, фотографиях с места преступления, результатах разного рода экспертиз, фотографических доказательствах исследований экспериментов и т.д. Без маркирования цифровыми водяными знаками здесь просто не обойтись. При желании с помощью цифровых водяных знаков можно защитить не только изображения, распространяемые в Интернете, но и вообще любые изображения, в том числе такие официальные документы как водительские права, паспорт и т.п.

Программное обеспечение, предназначенное для внедрения печатных водяных знаков, достаточно дешево и разнообразно. Среди имеющихся на рынке приложе-

ний можно найти и профессиональные пакеты с широкими возможностями по редактированию создаваемых водяных знаков, и бесплатные программы — просты и с минимумом вариантов обработки вставляемых меток (см. таблицу).

Цифровой водяной знак должен быть устойчив к разного рода трансформациям и преобразованиям изображения. Соответствовать этому требованию достаточно сложно, тем не менее, разработаны конкурирующие технологии и предлагаются специальные программы для внедрения цифровых водяных знаков. Как правило, в основе таких программ лежит использование для создания водяного знака шума — случайных данных, которые имеются в большинстве цифровых файлов. В итоге увидеть цифровой водяной знак без специальной программы, которая по его наличию в состоянии идентифицировать подлинность изображения, невозможно. По данному классу ориентировано в большей степени на крупные компании и стоит достаточно дорого. С основными функциями двух его образцов можно ознакомиться на основе демо-версий. Пакет MyPictureMarc (разработчик — Digimarc Corporation, <http://www.digimarc.com>; Mac PowerPC и Windows 95/98/NT/2000/Me/XP; демо-версия — <http://downloads.digimarc.com/readmarc/DigimarcPlugins1.7Win.exe>, 366 Кб). Пакет EIKONAmark 4.5.1 (разработчик — Alphatec Ltd; <http://www.alphatecltd.com>, 1,3 Мб; демо-версия — http://www.alphatecltd.com/download/bikonamark_setup.exe).

*Программа	Photo WaterMark 5.6.2.2	Media Resizer Pro Thumbnail Creator 2.35	MBizGroup Batch PhotoEditor 1.5	Robo-Enhancer1.0	Watermark 2.0.2	1-More WaterMarker 1.11	Image Brander 1.2.109.143
Разработчик	UniDream Marketing Technologies, Inc.	Media Resizer	MBizGroup, LLC	ACD Systems	Script Software	1-More.com	92Hicks.com
Сайт программы	http://www.photo-watermark.com	http://www.media-resizer.com	http://www.mbiz-group.com	http://www.acdsystems.com	http://www.script-software.com	http://1-more.com	http://www.92-Hicks.com
Размер дистрибутива, Мб	1,8	3,32	2	2,8	2,17	1,32	5,2
Способ распространения	Shareware (демо-версия с ограничениями)	Shareware (демо-версия с ограничениями)	Shareware (демо-версия с ограничениями)	Shareware (демо-версия с ограничениями)	Shareware (демо-версия с ограничениями)	Shareware (демо-версия с ограничениями)	Freeware
Работа под управлением	Windows 95/98/Me/NT/2000/XP и Mac OS для Virtual PC	Windows 95/98/Me/NT/2000/XP/2003 Server	Windows 95/98/Me/NT/2000/XP/2003 Server	Windows 95/98/Me/NT4.0 Service Pack 6/2000/XP	Windows 98 и выше и OS X Jaguar	Windows 95/98/Me/NT4/2000/XP	Windows 98/Me/NT/2000/XP
Цена, USD	40	79	10	69,95	20	19	



Как запустить Windows на Mac, знают практически все — для этого можно использовать Virtual PC, которая когда-то разрабатывалась Compaq, а затем была приобретена Microsoft. А вот как запустить Mac OS X под Windows — этого не знает практически никто. Завесу над тайной приподнимает О.Шарабанов в статье "Яблоко в окне" (CHIP, №1/2005, С.128...130).

Наиболее продвинутые пользователи знают или слышали о Basilisk II. Однако этот эмулятор был выпущен уже пять

лет назад и эмулировал лишь достаточно медленный по сегодняшним меркам процессор 68040 (предшественник PowerPC). Соответственно, максимум, что получится установить с помощью Basilisk II, — это Mac OS 8.1. Как вы понимаете, сейчас это уже неактуально. Какое-то время назад группа энтузиастов предложила создать новую программу-эмулятор, которая бы распространялась с открытым исходным текстом и позволяла компьютерам на базе процессоров Intel и AMD эмулировать архитек-

туру PowerPC (хотя бы серии G3). То, что получилось, разработчики назвали PearPC. Ее создание заняло 18 месяцев, а суммарный объем кода составил около 70 000 строк. Программа работает с различными версиями как Linux-, так и Windows-платформ; время установки Mac OS X 10.2 на компьютер с процессором Athlon XP 1,3 МГц колеблется в промежутке от двух до трех часов.

Для инсталляции потребуются: PearPC (pearpc.sourceforge.net), образ жесткого диска (pearpc.sourceforge.net),



образ диска с драйверами 3Com (pearpc.sourceforge.net), утилиты MacDrive (www.mediafour.com), UltraISO (www.ezbsystems.com) и TransMac (www.asy.com), а также дистрибутив Mac OS X 10.2, ОС Windows 98/Me/2000/XP и не менее 256 Мб памяти.

Первое, что потребуется сделать — установить утилиту MacDrive, которая обеспечит возможность прочесть установочные CD с Mac OS X. Без нее Windows их просто не увидит. Затем с помощью про-

граммы UltraISO снимают образы первых двух дисков, назвав их macoscd1.iso и macoscd2.iso соответственно. Далее распаковывается PearPC, например, в директорию C:\PearPC (без подкаталогов). В эту же директорию распаковывается виртуальный жесткий диск выбранного объема. В завершение в нее переносятся созданные образы установочных дисков. На этом самая простая часть инсталляции завершена.

Далее производится настройка эмуля-

тора — в любом текстовом редакторе пишется конфигурационный файл для PearPC.

После запуска инсталлятора производится выбор и форматирование раздела установки, настройка интерфейса и сети. Если потребуется перенести какой-либо файл или папку из Windows на виртуальный диск установленной Mac OS X, помочь в этом сможет утилита TransMac, скачать которую всегда можно с сайта компании-разработчика www.asy.com.

Несмотря на давние обещания разработчиков, любая операционная система по-прежнему далека от идеального состояния "установил и забыл". Время, потраченное на инсталляцию и настройку, трафик, съеденный системными патчами и обновлениями драйверов, постоянная опасность случайного сбоя, после которого машину придется восстанавливать вручную, — все это знакомо почти любому продвинутому пользователю. А возможно ли вообще создать такую операционную систему, которая не требовала бы длительной инсталляции (или вообще обходилась бы без нее), была бы максимально устойчива и поддерживала бы при этом самую разнообразную аппаратуру, присущую современному компьютеру? **Об ОС, не требующих установки и наладки**, рассказывает Ю.Анищенко в статье "Живые диски": система, о которой вы мечтали! (Подводная лодка, №1/2005, С.140...143).

Речь идет о так называемых LiveCD — специальным образом подготовленных операционных системах, которые загружаются с компакт-дисков без инсталляции на жесткий диск. Практически для всех LiveCD за основу взяты дистрибутивы существующих полновесных ОС, которые были существенно переработаны. Как правило, от оригинала остаются ядро и загрузчик, основные подсистемы (графическая, сетевая, дисковая и т.п.), интерфейс пользователя и набор драйверов. Помимо этого, в состав LiveCD входит прикладное программное обеспечение. После запуска происходит распаковка на виртуальный диск, создаваемый в оперативной памяти. Иногда система предлагает использовать заранее созданный swar-раздел на жестком диске.

Очевидное преимущество LiveCD — неизменность системных настроек. Поскольку файлы конфигурации "намертво" прошиты на компакт-диске, они будут загружаться каждый раз заново, вне зависимости от того, что происходило во время прошлой рабочей сессии. Можно даже смело выключать машину, не завершив работу ОС. Проблем с сохранением результатов работы не возникает. LiveCD поддерживают практически всю периферию, в том числе дисководы и flash-накопители. USB-брелка на 256 Мб должно хватить для большинства случаев. Результаты своего труда можно послать по электронной почте, выложить на FTP или

в сетевую папку. Некоторые разработки позволяют сохранять данные на локальные жесткие диски.

Knoprix (www.knoppr.net/knoprix) — одна из первых LiveCD-систем, наиболее известная широким массам пользователей PC. За основу системы взят популярный дистрибутив Debian. Это "полноценная" LiveCD, занимающая целый компакт-диск и заряженная огромным количеством системных и прикладных программ. На фоне прочих ее принято считать эталоном универсальности. Рядовой пользователь практически не замечает разницу между ней и полноценной инсталлированной на жесткий диск операционной системой. В набор предустановленных приложений входят деловые пакеты, неплохой набор мультимедийных приложений, средства для создания и отладки скриптов Tel и Python и игры.

Damn Small Linux (www.damnsmalllinux.org) выполнена в формате Business Card CD — крохотного диска, вмещающего не более 50 Мб информации. Тем не менее, DSL — полноценная система с графическим интерфейсом пользователя, набором программ делового и развлекательного характера. В состав дистрибутива входит пара текстовых редакторов, браузер изображений и графический редактор класса Paint, менеджер электронных таблиц и адресная книга, полный набор средств для работы в режиме онлайн (браузер, клиент электронной почты, интернет-пейджер).

LiveCD Auditor (www.remote-exploit.org) также создан на базе Knoprix, но, в отличие от оригинала, сделавшего ставку на универсальное применение, или сверхкомпактного DSL, эта система решает более узкий круг задач и представляет интерес в первую очередь для системных администраторов. Auditor — специальный дистрибутив, в котором собраны разнообразные средства для проверки и обеспечения безопасности компьютерных систем. На диске пользователь найдет несколько сотен отборных программ из арсенала матевого хака, которые позволяют обнаружить любое слабое место в сети. Система поддерживает приличную функциональность — можно распечатать свои документы, послушать музыку или выйти в Интернет. Недостаток Auditor — необходимость достаточно часто обновлять его версию. Из-

за того что в его состав входит несколько сотен постоянно совершенствующихся программ, периодически весь дистрибутив приходится собирать заново.

FreeSBIE (www.freesbie.org) — за основу этого LiveCD взята FreeBSD — прекрасно зарекомендовавшая себя UNIX-система, отличающаяся высокой стабильностью работы и считающаяся идеальной платформой для хостинга веб-сервера. Контент сайтов можно хранить и на другом носителе информации (скажем, на специальном разделе жесткого диска, flash-драйве и т.п.), зато надежность и устойчивость к попыткам взлома возрастает многократно. Да и восстановить работу сервера после аппаратного сбоя будет намного проще — достаточно перезапустить компьютер. Не обладая всеми удобствами для обычного пользователя, которые может предложить Knoprix, FreeSBIE, тем не менее, вполне подойдет и на роль системы для домашнего компьютера или ноутбука — к ней прилагаются офисные средства (тот же OpenOffice), мультимедийные проигрыватели, программы для работы в Интернете и т.п.

Windows LiveCD — оказывается, LiveCD можно сотворить даже из такой тяжеловесной операционной системы как Windows, которая, казалось бы, жить не может без огромных дисковых пространств, намертво привязывается к аппаратному обеспечению и т.д. При некотором старании вполне возможно заставить работать с компакт-диска даже Windows XP. Ничто не мешает пользователю сделать LiveCD на основе легально приобретенного дистрибутива системы и пользоваться им вместо самой системы. Сборку можно сделать специальными утилитами, например PE-Builder (www.pu2.ru). Для созданного с ее помощью образа диска существуют целые каталоги модулей расширений (например, www.bootcd.us/BartPE_Plugins_Repository.php), которые позволяют настроить функциональность будущего LiveCD по своему вкусу.

В Интернете можно найти немало LiveCD, которые загружат ваш компьютер альтернативными операционными системами. Например, "живой диск" собрали разработчики перспективного проекта ReactOS (www.reactos.org), который посвящен созданию системы Open Source, полностью совместимой на двоичном



уровне с Microsoft Windows. Правда, LiveCD с ReactOS предлагает только минимальный набор полезного в хозяйстве софта и служит, скорее, для знакомства с возможностями системы, но при необ-

ходимости состав загрузочного диска можно изменить своими руками, добавив программы, которыми вы обычно пользуетесь в Windows (структура системного реестра аналогична той, что применя-

лась в Windows NT, так что "симулировать" установку программы не составит никакого труда). На скриншоте с сайта разработчиков даже показано, как под ReactOS работает Quake II.

В тестовой лаборатории журнала CHIP проверена возможность разгона бюджетных видеокарт GeForce 6800LE и 6200 (CHIP, №1/2005, С.14...15).

Оказывается, с помощью бесплатной утилиты RivaTuner (начиная с версии 2.0 RC 15.2) можно совершить гениальный трюк — превратить недорогие видеокарты GeForce (модели 6800 и 6800LE) в вы-

сокопроизводительные варианты GeForce 6800GT. Весь фокус заключается в том, что утилита RivaTuner умеет включать заблокированные вершинные и пиксельные конвейеры — и все это без какого-либо вмешательства в аппаратную часть плат. Программа переписывает некоторые внутренние регистры и таким образом активирует все шесть вершинных и шестнадцать пик-

сельных конвейеров.

Испытания, проведенные в тест-центре журнала, на 80% оказались успешными: четыре из пяти видеокарт GeForce RivaTuner сумела превратить из "золушек" в "принцесс". Наилучшие результаты продемонстрировала модель AOpen Aeolus GeForce 6800LE DV128. Включение конвейеров не вызвало абсолютно никаких артефактов графики. Попутно производительность обновленной видеокарты выросла вдвое. Результаты разогнанных GeForce 6800 LE от разных производителей представлены в таблице.

Разумеется, гарантировать безошибочную работу деактивированных в заводских условиях конвейеров невозможно, но вместе с тем, вероятность приобретения вполне "разгоняемой" видеокарты, способной к модификации, очень высока. Если же эксперимент окажется не совсем удачным, всегда можно вернуться на исходные позиции.

Новая методика также была опробована на эталонной видеокарте NVIDIA GeForce 6200. Результат оказался положительным — она превратилась в GeForce 6600 благодаря включению четырех пиксельных конвейеров.

Производитель	AOpen	MSI	ASUS
Модель	Aeolus GeForce 6800LE	NX6800-TD128 DV128	V9999/TDGame
Цена, USD	280	350	450
Модификация чипа	LE	Стандартный	Стандартный
Память, МБ	128	128	256
Номинальная тактовая частота чипа/памяти, МГц	300/700	325/700	350/1000
Номинальное количество вершинных/пиксельных шейдеров	4/8	5/12	5/12
Максимальная тактовая частота чипа/памяти, МГц	410/900	390/910	410/1120
Максимальное количество вершинных/пиксельных шейдеров	6/16	6/16	6/16
Количество баллов 3DMark05 (до/после модификации)	2125/4315	2334/4109	3699/5216
Прирост производительности, %	+103	+76	+41

Что это такое, компьютеры формата SFF (у нас более известные под именем "barebone"), рассказывает М.Десяреев в статье "Анатомия Barebone" (Подводная лодка, №1/2005, С.92...97).

Сегодня уже многие знают, что под этим названием скрывается комплект из малогабаритного корпуса (как правило, его габариты — 20х20х25 см), блока питания и материнской платы, в которую интегрировано все, что только можно. Иногда в комплекте поставляются клавиатура и мышь. Преимущество подобных комплектов очевидно — они позволяют максимально просто и быстро собрать готовые компьютеры — достаточно добавить лишь процессор, память и жесткий диск. Таким образом, barebone — это костяк, платформа, на которой строится компьютер, и пользователь сам по своим средствам выбирает, какую память и в каких объемах устанавливать, какой использовать процессор и какой жесткий диск. Возможность дальнейшего расширения компьютера определяется конструкцией той или иной платформы.

В сегодняшнем понимании слово "barebone" изменило свой первоначальный смысл: теперь под этим определением понимаются любые компьютеры компактного формата SFF (Small Form Factor). Родоначальником формата SFF явилась компания Shuttle, которая первой представила не просто новый корпус сравнительно малых размеров, но целый класс компьютеров, которые, несмотря на свои габари-

ты, почти не уступали по функциональности большим компьютерам, собранным в корпусах формата ATX. При этом была сохранена совместимость с форматом ATX по разъемам плат расширения, памяти, периферии и (в основном) блокам питания, правда, размер самой системной платы и способ ее крепления пришлось изменить. Сегодня barebone-платформы формата SFF выпускает каждый уважающий себя производитель материнских плат. Самые распространенные платформы довольно ярко выделяются свойственными только им функциональными или дизайнерскими особенностями.

Большинство современных платформ формата SFF имеют один 5,25"-отсек для оптического привода, существенная их часть имеет и один внешний 3,5"-отсек. Флоппи-дискетод для подобных компьютеров — ужасный архаизм, поэтому многие производители barebone-платформ предпочитают устанавливать в 3,5"-отсек картридер для флэш-карточек. Необходимость обеспечения простого и удобного подключения внешних устройств, как временных, так и стационарных, привела к появлению портов как на задней панели корпуса (для стационарных устройств), так и на передней. С обратной стороны корпусов устанавливаются основные порты и разъемы материнской платы. Здесь же расположены, как правило, три вентиляционные решетки и два отсека для плат расширения. Жесткие диски, оптический привод и дис-

ковод устанавливаются на специальную корзину, причем установка этих комплектующих может производиться "голыми руками", без использования отвертки. Приводы крепятся на корзину с помощью специальных ползунков, а сама корзина устанавливается в корпус на защелках. Жесткие диски устанавливаются в самом верху корзины, чтобы не занимать лишнее место в корпусе и не подогревать его внутреннюю атмосферу. Бытует ошибочное мнение, что в корпусах формата SFF сложно осуществить вентиляцию, т.к. чем меньше объем корпуса, тем проще его вентилировать. Но при этом надо обращать особое внимание на расположение основных нагревательных элементов, аэродинамику и направление воздушных потоков. Вот почему кабели здесь укладываются в шасси, а в боковых стенках проделываются отверстия, так что горячему воздуху просто негде скапливаться.

Именно во время сборки компьютера начинаешь понимать преимущество barebone-платформы перед китайским корпусом и материнской платой в коробочке. Материнская плата идеально подобрана под корпус. Все разъемы легко доступны, их расположение продумано и ничего здесь не установлено "случайно" или "просто так". Все кабели уже проложены по компьютеру, закреплены на станине корпуса и подключены к системной плате. Осталось поставить процессор, память, оптический дискетод и винчестер.



Borland C++ Builder 6

для начинающих

О.ВАЛЬПА,
E-mail: sandh@narod.ru

(Продолжение. Начало в №№7-12/2004, 1-4/2005)

В этой статье рассказывается о том, как просто можно создать в среде разработки Borland C++ Builder 6 программу текстового редактора.

Текстовый редактор

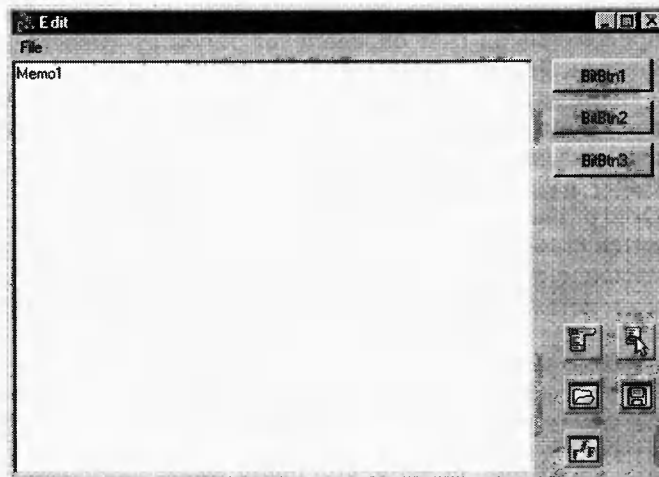
В предыдущий раз мы рассмотрели свойства компонентов. Теперь на примере создания программы текстового редактора рассмотрим применение этих свойств на практике.

Запустите Borland C++ Builder 6 и создайте новый проект с именем **edit**. Поскольку текстовый редактор предназначен для работы с текстовыми строками, нам понадобится компонент **Memo** из закладки **Standard**. Поместите его в центре формы проекта. Теперь поместите на форму компонент **MainMenu** из той же закладки. Наконец, добавьте на форму три стандартные кнопки **BitButton** из закладки **Additional**. Они понадобятся для формирования в программе органов управления, для открытия и сохранения файлов и для других функций. Поскольку текстовый редактор должен при открытии и сохранении файлов обращаться к дисковым накопителям, необходимо добавить на форму диалоги **Open Dialog** и **Save Dialog** из закладки **Dialogs**. Кроме того, добавьте на форму диалог **Font Dialog** из этой же закладки, он облегчит нам задачу настройки шрифтов из программы. Эти компоненты при работе программы являются невидимыми, поэтому место их размещения на форме не имеет значения. Но все же лучше располагать такие компоненты без наложения, для того чтобы они не скрывали друг друга. Последний компонент, который необходимо поместить на форму, — **PopupMenu** из закладки **Standard**. Этот компонент позволит создать в программе контекстное меню, появляющееся при нажатии правой кнопки мыши.

Разместим выбранные компоненты на форме и настроим их свойства. Начнем с главного компонента — формы приложения. Выберите в инспекторе объектов в поле селектора объектов **Form1**. Измените свойство **Caption** этого объекта на название программы редактора **Edit**. Размеры формы установите с помощью мыши, захватив любой угол или сторону формы и перемещая их в любом направлении. При необходимости размер формы всегда можно будет изменить на любом этапе разработки программы. В свойстве **Position** установите значение **poDesigned**. Это обеспечит центровку нашего приложения при запуске по центру экрана. Щелкните по компоненту **MainMenu** левой кнопкой мыши. При этом откроется окно дизайнера меню. В свойстве **Caption** инспектора объектов введите название **&File** и нажмите клавишу **Enter**. На форме приложения при этом появится строка главного меню. Таким образом, на форме можно будет видеть поле главного меню и учитывать его размеры. Теперь можно временно закрыть окно дизайнера меню и приступить к размещению остальных компонентов на форме. Вначале разместите на форме компонент **Memo** так, чтобы он занимал всю ее свободную область, за исключением правой полосы для кнопок. Кнопки расположите друг под другом, не изменяя их размеры. В остав-

шейся свободной части формы можно расположить оставшиеся невидимые компоненты приложения. Не забывайте, что перемещать компоненты можно группами, предварительно выделив эти компоненты с помощью мыши. В результате должна получиться форма, подобная приведенной на **рис.1**.

Рис. 1



Теперь приступим к изменению свойств остальных компонентов. Начнем с компонента **Memo**, выбрав его в инспекторе объектов. Вначале удалим надпись **Memo1** с компонента, чтобы она не появлялась при запуске приложения. Для этого изменим свойство **Lines**, щелкнув по кнопке с тремя точками правее надписи **Tstrings** этого свойства. При этом откроется окно редактора строки **String List Editor**, в котором необходимо удалить строку **Memo1** и нажать кнопку **OK**. Теперь добавим для этого компонента линейку прокрутки для обеспечения возможности просмотра и редактирования больших областей текста с помощью мыши. Для этого в свойстве **ScrollBars** необходимо выбрать значение **ssBoth**. Чтобы работало контекстное меню, необходимо в свойстве **PopupMenu** выбрать значение **PopupMenu1**. Это значение стало доступным после помещения на форму проекта диалога **PopupMenu**. Настройку данного меню произведем позже.

Настроим свойства кнопок в соответствии с выполняемыми ими функциями. Для этого изменим поочередно свойство **Caption** для всех трех кнопок на **Open**, **Save** и **Font** соответственно. Для того чтобы при наведении на эти кнопки изменялся тип курсора, изменим еще одно групповое свойство кнопок. Для этого выделите все три кнопки с помощью мыши, и в появившемся объекте **3 items selected** в инспекторе объектов измените свойство **Cursor** на значение **crHandPoint**.

Настройку главного меню произведем при помощи дизайнера меню, вызываемого двойным щелчком левой кнопки мыши по компоненту **MainMenu**. Ранее мы уже ввели один пункт меню под названием **&File**. Символ "амперсанта" перед **File** служит для выделения первой буквы названия пункта в меню быстрого доступа путем ее подчеркивания. Следующий по горизонтали пункт меню назовем **&Edit**.



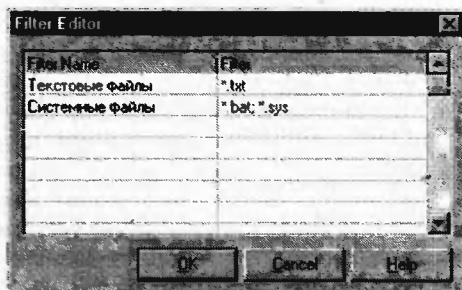
Теперь заполним пункты меню по вертикали. Щелкните левой кнопкой мыши по пункту меню **File**, а затем ниже него по пустому полю. Свойство **Caption** этого пункта меню измените на **Open** и нажмите клавишу **Enter**. Следующий пункт назовите **Save**. Аналогично расширим по вертикали пункты меню **Edit**, добавив пункты **Font** и **Clear**.

На этом создание главного меню программы редактора завершено. Все введенные пункты меню теперь будут присутствовать в приложении, но пока не будут выполняться, поскольку не созданы функции для выполнения соответствующих команд. Создадим эти функции.

Для этого воспользуемся компонентами диалогов, помещенных ранее на форму приложения. С помощью диалогов работы с файлами можно задать направленность приложения за счет определения типов открываемых файлов. Это делается с помощью свойства **Filter**.

Выберите в инспекторе объектов компонент **OpenDialog1** и нажмите кнопку с тремя точками правее свойства **Filter**. При этом появится диалоговое окно **Filter Editor**, в котором производится настройка типов файлов. Окно разбито на два поля. В левом поле **Filter Name** вводятся строки пояснения. В правом поле **Filter** вводятся расширения файлов. Для указания нескольких типов файлов необходимо записать их через точку с запятой. Если задать несколько строк с пояснениями и типами файлов, то при работе приложения их можно будет выбирать с помощью выпадающего списка. Заполните оба поля диалогового окна **Filter Editor** в соответствии с рис.2 и нажмите кнопку **OK**.

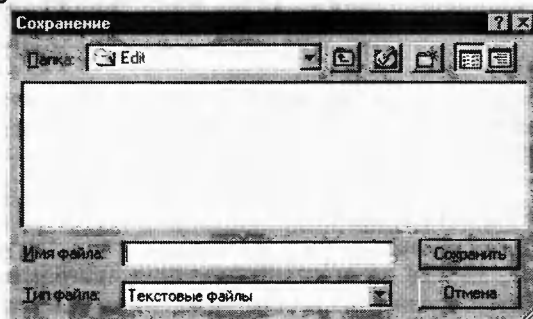
Рис. 2



Аналогично выполните настройку компонента **SaveDialog1**. Для корректной работы приложения оба эти диалога должны иметь одинаковую настройку свойства **Filter**. Чтобы облегчить эту работу и избежать ошибок, можно использовать два способа. Первый способ уже был рассмотрен ранее, когда несколько компонентов кнопок выделялись в группу и соответствующее свойство задавалось одновременно для всех выделенных компонентов. Второй способ заключается в том, что, задав свойство для одного из компонентов, можно скопировать и вставить его для других компонентов с помощью комбинации клавиш **Ctrl+C** и **Ctrl+V** соответственно. Так, после задания свойства **Filter** компонента **OpenDialog1** с помощью **Filter Editor**, в поле этого свойства появится запись: **"Текстовые файлы|.txt|Системные файлы|.bat;*.sys"**. Выделив эту строку с помощью клавиш **Home** и **Shift+End**, скопируйте ее в буфер с помощью комбинации клавиш **Ctrl+C**. Теперь можно вставить эту строку в свойство **Filter** для компонента **SaveDialog1**. Примечательно, что работу этих компонентов можно сразу же проверить, еще до выполнения программы. Дело в том, что после задания свойства **Filter** этих компонентов двойной щелчок по компонентам приводит к открытию диалога на основе заданных свойств.

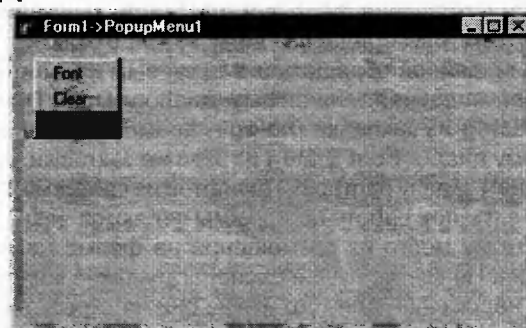
Например, после двойного щелчка по компоненту **SaveDialog1** откроется окно, изображенное на рис.3.

Рис. 3



Теперь настроим компонент **FontDialog1**. Основным свойством данного компонента является **Font**, с помощью которого можно выбрать тип шрифта, его цвет, размер, стиль и т.п. Эти установки будут использоваться в программе текстового редактора по умолчанию. С помощью свойства **Options** можно настроить вид и поведение диалога. Например, можно добавить в него кнопку справки, установив для подсвойства **fdShowHelp** значение **true**.

Рис. 4



Нам осталось произвести настройку контекстного меню **PopupMenu**. Данная настройка производится аналогично настройке главного меню **MainMenu**. Добавим в это меню пункты с названиями **Font** и **Clear** через свойство **Caption**, как это показано на рис.4.

Кроме того, добавьте название **PopupMenu1** в свойство **PopupMenu** главной формы приложения.

На этом настройка компонентов завершена, и полученная визуальная оболочка программы редактора. Однако без программных строк это приложение не будет работать. При запуске приложения будут открываться пункты меню, вводиться и редактироваться текст, но только потому, что эти функции заложены в сами компоненты. Нажатие на кнопки и вызов команд из главного меню не будут приводить к каким-либо действиям программы, поскольку не заданы функции обработки событий. «Вдохнем» в приложение жизнь. В программе редактора имеется четыре типа операций. Это открытие и сохранение файлов, настройка шрифтов и очистка окна редактора. Все эти операции можно вызвать разными способами. Например, с помощью главного меню, кнопок или контекстного меню. Можно написать функции для каждого способа, но это усложнит программу и внесет трудности при ее создании и отладке. Правильнее будет создать такую функцию один раз, а затем задать ее выполнение другим компонентам с помощью инспектора объектов. Создадим функцию для команды главного меню **Open**.



Для этого откройте на главной форме пункт меню **File** и щелкните левой кнопкой мыши по команде **Open**. При этом активизируется редактор кода с заготовкой функции обработки. Введите между фигурными скобками следующий текст программы:

```
if (OpenDialog1->Execute())  
    Memo1->Lines->LoadFromFile (OpenDialog1->FileName);
```

Обратите внимание на то, что среда разработки сама предлагает выбрать функцию из множества разрешенных функций для компонента, после введения символа стрелки. Эти программные строки позволят при запуске диалога открытия файла загрузить содержимое открываемого файла в свойство **Lines** компонента **Memo1** для отображения на экране. Теперь задайте выполнение этой функции кнопке команды **Open** с помощью инспектора объектов. Для этого щелкните левой кнопкой мыши по кнопке **Open** и выберите в инспекторе объектов закладку **Events** (события). Раскройте список в событии **OnClick** и выберите в нем строку **Open1Click**. Таким образом, будет задано, что при нажатии кнопки в программе редактора запустится та же функция открытия файла, что и для команды **Open** из пункта главного меню **File**. Запустите программу с помощью команды **Run** и убедитесь, что команды открытия файла работают и выбирают заданные нами типы файлов.

Аналогично создадим функции для оставшихся трех типов команд. Начнем с команды сохранения файлов **Save**. Откройте на главной форме пункт меню **File** и щелкните левой кнопкой мыши по команде **Save**. При этом активизируется редактор кода с заготовкой функции обработки. Введите между фигурными скобками следующий текст программы:

```
if (SaveDialog1->Execute())  
    Memo1->Lines->SaveToFile (SaveDialog1->FileName);
```

Эти программные строки позволят при запуске диалога сохранения файла загрузить содержимое свойства **Lines** компонента **Memo1** в сохраняемый файл. Назначьте выполнение этой функции кнопке с командой **Save** с помощью инспектора объектов. Для этого щелкните левой кнопкой мыши по кнопке **Save** и выберите в инспекторе объектов закладку **Events**. Раскройте список в событии **OnClick** и выберите в нем строку **Save1Click**. Таким образом будет задано, что при нажатии кнопки **Save** в программе редактора запустится та же функция сохранения файла, что и для команды **Save** из пункта главного меню **File**.

Для команды **Font** из пункта **Edit** главного меню введите с помощью инспектора кодов строки:

```
if (FontDialog1->Execute())  
    Memo1->Font=FontDialog1->Font;
```

Это позволит изменять шрифт текста компонента **Memo1** при открытии диалога **Font Dialog1**. Задайте выполнение этой функции кнопке с командой **Font** с помощью инспектора объектов, для чего раскройте список в событии **OnClick** и выберите в нем строку **Font1Click**.

Теперь добавим вызов этой функции в контекстное меню. Выделите на форме компонент **PopupMenu1** и в инспекторе объектов щелкните по кнопке с тремя точками в строке свойства **Items**. При этом откроется окно

Form1→PopupMenu. В этом окне щелкните на поле **Font** и в инспекторе объектов на закладке **Events** выберите для события **OnClick** строку **Font1Click**.

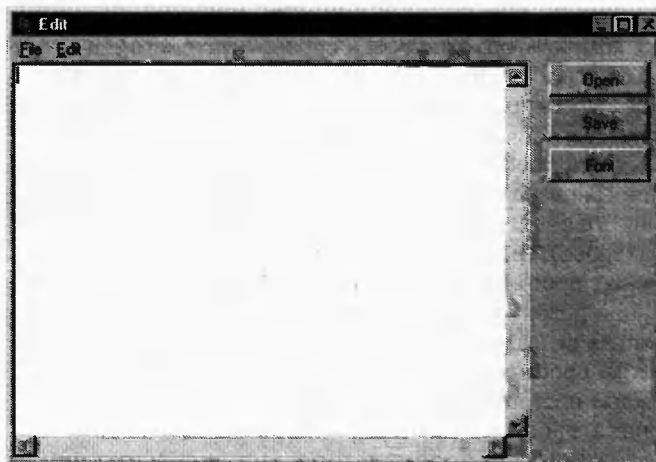
Наконец, создадим функцию обработки очистки редактора. Для команды **Clear** из пункта **Edit** главного меню введите с помощью инспектора кодов одну-единственную строку:

```
Memo1->Clear();
```

Эта программная строка будет производить очистку содержимого компонента **Memo1**. Аналогично предыдущему описанию назначьте выполнение данной функции через контекстное меню, выбрав строку **Clear1Click** для события **OnClick**.

На этом создание программы редактора заканчивается, и можно приступить к его проверке. Предварительно сохраните весь проект в отдельном каталоге, например, с именем **Edit**, а затем запустите его на выполнение. На экране должно появиться окно, аналогичное окну, изображенному на рис.5.

Рис. 5



Нажмите кнопку **Open** и выберите какой-либо файл на диске. После этого загрузите его в редактор. Измените с помощью клавиатуры текст файла и сохраните его с новым именем, нажав кнопку **Save**. Нажмите кнопку **Font** и измените шрифт текста редактора. Убедитесь в том, что текст изменился в соответствии с вашими установками. Из главного меню выполните команду **Clear** и убедитесь, что поле редактирования очистилось. Проверьте работу остальных пунктов главного меню. Далее проверьте работу контекстного меню, нажав правую кнопку мыши на поле редактора и выбрав одну из двух команд. Обратите внимание на то, что при наведении курсора на кнопки он должен менять свою форму.

Редактор готов к работе. Создайте для него иконку, аналогично тому, как мы это делали раньше, и используйте его как отдельное законченное приложение. Со временем вы сможете добавлять в него новые функции и возможности, повышая тем самым его функциональность и свое мастерство.

Исходные файлы проекта, а также саму программу текстового редактора можно скачать с сайта редакции журнала <http://radio-mir.com>.

(Продолжение следует)



Интегрированная среда VISUAL C++ 6.0 и работа с консольными приложениями

А.КОРБИТ

г.Минск, БГУИР,

ФИТУ, каф.ВМиП

E-mail: a_korbit@mail.ru

При изучении языка программирования C/C++ работа в средах программирования под DOS (например, в Borland C++ 3.1) сложностей не вызывает. Это объясняется простой однофайловой схемой получения исполняемого модуля:

исходный модуль → объектный модуль → исполняемый модуль.

Современные ПК как правило оснащены мощными системами программирования под windows, такими, например, как Bilder C++, Visual C++, VatCom и т.д. И в таких современных средах программирования механизмы создания проектов, не требующих Window-интерфейса, несколько другие. Рассмотрим, как реализован такой механизм в Visual C++.

Что такое Visual C++ (Developer studio)

Интегрированная среда разработки (Integrated Development Environment, или, сокращенно — IDE) — это программный продукт, объединяющий текстовый редактор, компилятор, отладчик и справочную систему.

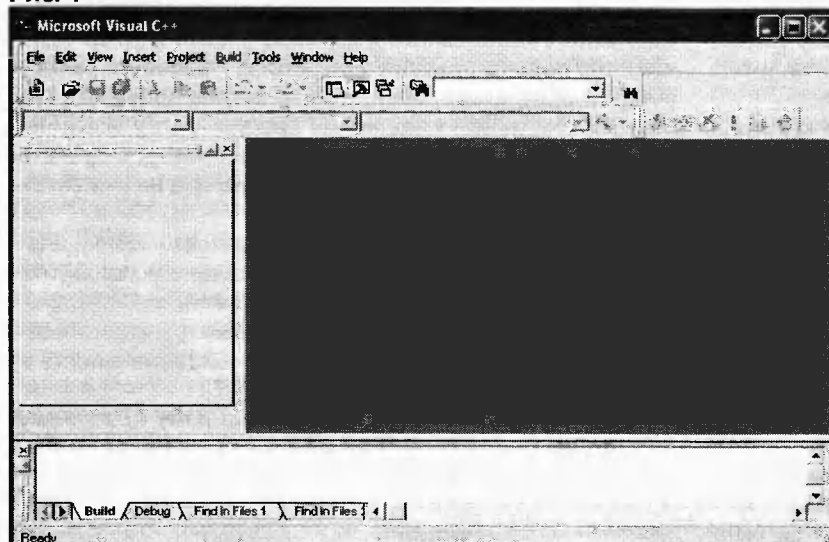
Любая программа в среде Visual C++ всегда создается в виде отдельного проекта. Проект (project) — это набор взаимосвязанных исходных файлов и, возможно, включаемых заголовочных файлов, компиляция и компоновка которых позволяют создать исполняемую программу. Основу Visual C++ составляет рабочая область (project workspace). Она может содержать любое количество различных проектов, сгруппированных вместе для согласованной разработки — от отдельного приложения до библиотеки функций или целого программного пакета. Решение же простых (учебных) задач сводится к оформлению каждой программы в виде одного проекта, т.е. рабочая область проекта будет содержать ровно один проект.

Вид рабочего стола Visual C++.

Консольное приложение

После запуска Visual C++ появляется главное окно программы, вид которого приведен на рис.1 (в зависимости от настроек Visual C++, его вид может быть несколько иным).

Рис. 1



Экран Visual C++ разделен на четыре основные зоны. Сверху расположены меню и панели инструментов. Кроме них, рабочий стол Visual C++ включает в себя три окна.

Окно **Project Workspace** (окно рабочей области) расположено в левой части экрана. Первоначально окно закрыто, но после создания нового проекта или загрузки существующего проекта это окно будет содержать несколько вкладок.

Справа расположено окно **Editor** (окно редактирования). Его используют для ввода и проверки и редактирования исходного кода программы.

Окно **Output** (окно вывода) служит для вывода сообщений о ходе компиляции, сборки и выполнения программы и сообщений о возникающих ошибках.

Для кнопок панелей инструментов предусмотрена удобная контекстная помощь — если навести курсор мыши на кнопку и задержать на пару секунд, то всплывет подсказка с назначением данной кнопки.

Developer Studio позволяет создавать проекты различных типов, которые ориентированы на различные сферы применения. Большинство типов проектов являются оконными Windows-приложениями с соответствующим графическим интерфейсом. Но также предусмотрена работа и с консольными приложениями. При запуске консольного приложения операционная система создает консольное окно, через которое идет весь ввод-вывод данных программы. Такая работа и представляет имитацию работы в операционной системе MS-DOS или других операционных системах в режиме командной строки. Этот тип приложений больше всего подходит для изучения языка C/C++, так как не требует создания Windows-кода для пользовательского интерфейса. Рассмотрим приемы работы с консольными приложениями более подробно (рекомендуется для размещения проектов создать специальную рабочую папку).

Создание нового проекта

Для создания нового проекта типа "консольное приложение" выполните следующие действия.

1. Выберите в строке меню главного окна команду **File/New...**
2. В открывшемся диалоговом окне New выберите вкладку **Projects**:
 - выберите тип проекта: **Win32 Console Application**;
 - введите имя проекта в текстовом поле **Project Name**, например **Ir1**;
 - в текстовом поле **Location** введите имя каталога (полный путь к нему) для размещения будущих файлов проекта (если указанный вами каталог отсутствует, то он будет создан автоматически). Путь к будущему проекту можно выбрать, щелкнув на кнопке, расположенной справа от текстового поля Location;
 - щелкните левой кнопкой мыши на кнопке **OK**.
3. Щелчок запустит встроенный мастер приложений Application Wizard, который откроет



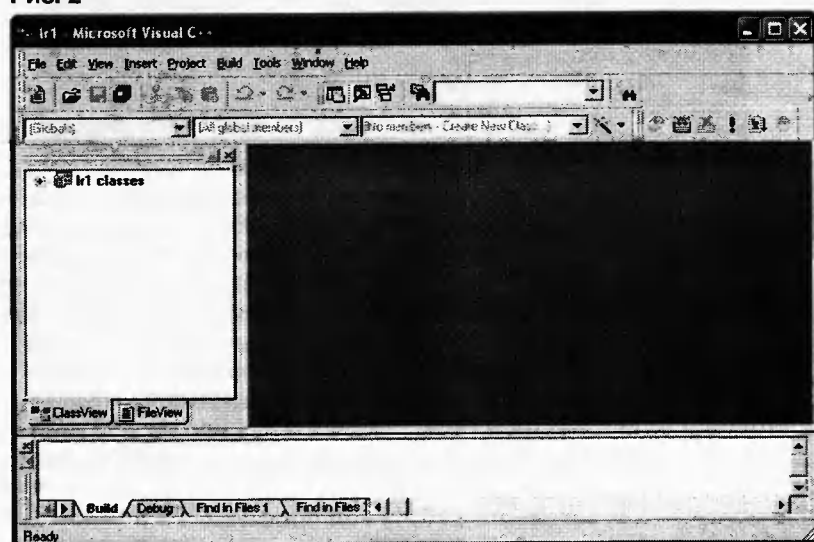
диалоговое окно **Win32 Console Application — Step1 of 1** с предложением определить, какой подтип консольного приложения желаете создать. Выберите тип **An empty project** (пустой проект) и щелкните на кнопке **Finish**.

4. После щелчка появится окно **New Project Information** (информация о новом проекте) со спецификациями проекта и информацией о каталоге, в котором будет размещен создаваемый проект.

5. Щелкните на кнопке **OK**.

После выполненных шагов рабочий стол примет вид, показанный на **рис.2**, а в папке **lr1**, созданной мастером приложений, будут размещены файлы **First.dsw**, **First.dsp**, **First.opt**, **First.ncb** и папка **Debug** (или **Release** — в зависимости от конфигурации проекта).

Рис. 2



Краткое описание каждого из файлов:

- **First.dsw** — файл рабочей области проекта, используемый внутри интегрированной среды разработки. Он объединяет всю информацию о проектах, входящих в данную рабочую область;

- **First.dsp** — проектный файл, используемый для построения (building) отдельного проекта или подпроекта;

- **First.opt** — файл, содержащий опции рабочей области проекта. Благодаря этому файлу при каждом открытии рабочей области проекта все параметры Developer Studio, выбранные во время последнего сеанса работы с данной рабочей областью, будут восстановлены;

- **First.ncb** — служебный файл. Он создается компилятором и содержит информацию, которая используется в инструменте интегрированной среды под названием ClassView. Панель ClassView находится в окне Project Workspace и показывает все классы C++, для которых доступны определения в рамках данного проекта, а также все элементы этих классов. В задачах данного практикума данная панель будет пустой, так как классы C++ в них не используются;

- **Debug** — папка, в которую будут помещаться файлы, формируемые компилятором и сборщиком. Из этих файлов нас будет интересовать, в общем-то, только один — исполняемый файл, имеющий расширение .exe.

Как видно из **рис.2**, окно Project Workspace теперь открыто. В нем отображены две вкладки — **Class View** и **File View**. Вкладка **Class View**, как легко убедиться, пустая. Щелчком мыши переключаемся на вкладку **File View**.

Она предназначена для просмотра списка файлов проекта. Щелкнув мышью на значке "+", откроем список **First files**. Появится дерево списка файлов, содержащее пиктограммы трех папок: **Source Files** (исходные коды), **Header Files** (заголовочные файлы), **Resource Files** (файлы ресурсов). Так как в консольных приложениях файлы ресурсов не используются, последняя папка всегда будет пустой. Первоначально и первые две папки пустые, так как в качестве подтипа консольного приложения был выбран пустой проект (опция **An empty project**).

Добавление к проекту файлов с исходным кодом

При создании консольного приложения можно или добавить уже существующий файл с исходным кодом (**lr1.cpp**), т.е. чтобы исключить путаницу, желательно, чтобы имя файла с исходным кодом совпадало с именем проекта), который был создан при помощи других оболочек, или создать новый файл во встроенном текстовом редакторе среды программирования Visual C++.

Добавление существующего файла

В этом случае необходимо выполнить следующие действия:

1. Скопируйте исходный файл (**lr1.cpp**) в папку рабочей области проекта (в данном случае — **lr1**).

2. Вернитесь к списку **lr1 files** в окне **Project Workspace** проекта и щелкните правой кнопкой мыши на папке **Source Files**;

3. В появившемся контекстном меню щелчком мыши выберите команду добавления файлов **Add Files to Folder...**;

4. В открывшемся диалоговом окне **Insert Files...** выберите нужный файл (**lr1.cpp**) и щелкните на кнопке **OK**.

Создание и добавление нового файла

В этом случае необходимо выполнить следующие действия:

1. Выберите в строке меню главного окна команду **File/New....** В результате откроется диалоговое окно **New**.

2. На вкладке **Files**:

- выберите тип файла (в данном случае — **C++ Source File**);

- в текстовом поле **File Name** введите нужное имя файла (в данном случае — **lr1.cpp**). Флажок **Add to project** должен быть включен;

- щелкните на кнопке **OK**.

В обоих случаях после добавления файла, во-первых, в окне **Project Workspace** раскроется папка **Source Files** списка файлов проекта, и в нее будет помещен файл **lr1.cpp**. Во-вторых, окно редактора **Editor** очистится (станет белым), а в его левом верхнем углу будет мерцать черный текстовый курсор, предлагая ввести какой-нибудь текст.

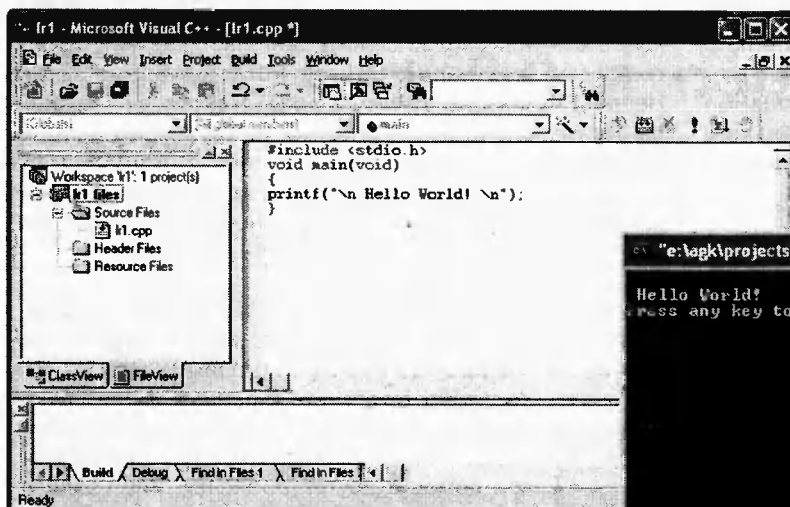
Введем, например, такой текст:

```
#include <stdio.h>
void main(void)
{
    printf("\n Hello World! \n");
}
```

В случае необходимости переключитесь на вкладку **File View**. Открываем список **lr1 files**, папку **Source Files** и убеждаемся, что в проекте создан файл **lr1.cpp** с только что набранным кодом.

Рис.3 иллюстрирует вид главного окна Visual C++ после проделанной работы.

Рис. 3



Компиляция, компоновка и выполнение проекта

Эти операции можно выполнить или через подменю Build главного окна, или при помощи кнопок панели инструментов, или при помощи комбинации горячих клавиш.

Данное подменю объединяет команды для компиляции, сборки и отладки программ.

Основные команды меню **Build**:

- **Compile** — компиляция выбранного файла. Результаты компиляции выводятся в окно Output;

- **Build** — компоновка проекта. Компилируются все файлы, в которых произошли изменения с момента последней компоновки. После компиляции происходит сборка (link) всех объектных модулей, включая библиотечные, в результирующий исполняемый файл. Сообщения об ошибках компоновки выводятся в окно Output. Если обе фазы компоновки завершились без ошибок, среда программирования создаст исполняемый файл с расширением .exe (для данного примера — `lr1.exe`), который можно запустить на выполнение;

- **Rebuild All** — команда, аналогичная Build, но компилируются все файлы проекта, независимо от того, были ли в них произведены изменения, или нет;

- **Execute** — выполнение исполняемого файла, созданного в результате компоновки проекта. Если же в исходный текст были внесены изменения, то производится перекомпилирование, перекомпоновка и выполнение файла.

Операциям Compile, Build и Execute соответствуют первая, вторая и четвертая кнопки панели инструментов Build MiniBar, которая расположена на рабочем столе (рис.3) справа сверху, рядом с системными кнопками. Перечислим их слева направо с указанием комбинаций горячих клавиш:

1-я — **Compile** — **Ctrl+F7**; 2-я — **Build** — **F7**; 4-я — **Execute Program** — **Ctrl+F5**.

Откомпилируйте проект, например щелчком на кнопке Build. В процессе компиляции в окне вывода Output отображаются диагностические сообщения компилятора и сборщика. Если все в порядке, в окне вывода получим следующую последнюю строку:

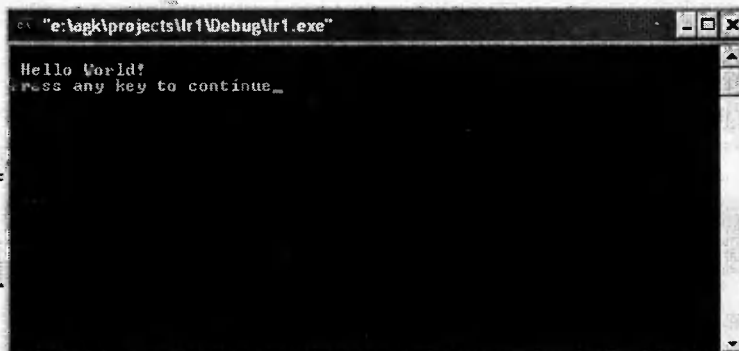
```
lr1.exe - 0 error(s), 0 warning(s).
```

Теперь запускаем приложение на выполнение щелчком, например, на кнопке Execute Program (**Ctrl+F5**). Появится окно приложения `r1`, изображенное на рис.4.

Конфигурация проекта

Visual C++ позволяет строить проект либо в отладочной конфигурации (**Win32 Debug**), либо в итоговой конфигурации (**Win32 Release**). Начинать работу нужно в отладочной конфигурации, которая обычно установлена по умолчанию. Для того чтобы проверить, какая текущая конфигурация в про-

Рис. 4



екте, нужно выбрать в подменю **Project** пункт **Settings....** Откроется диалоговое окно **Project Settings**. Смотрим, какое значение установлено в окне комбинированного списка **Settings For:...** Если это не **Win32 Debug**, то переключитесь на нужное значение через команду меню **Build/Set Active Configuration...** Но в отладочной конфигурации даже минимальный проект имеет очень большие размеры. Так, только что созданный проект `r1` занимает на диске более 1 Мб. Поэтому после получения финальной версии проект нужно перекомпилировать в итоговой конфигурации (**Win32 Release**).

Как закончить работу над проектом

Можно выбрать меню **File**, пункт **Close Workspace**. А можно просто закрыть приложение Visual C++.

Как открыть существующий проект

Для того чтобы открыть существующий проект, нужно проделать следующие действия:

- запустить на выполнение среду программирования Visual C++;

- выбрать в подменю **File** пункт **Open Workspace...**;

- в открывшемся диалоговом окне найти папку с нужным проектом, в ней — файл `ProjectName.dsw`, и открыть найденный файл, щелкнув по нему мышью.

В случае, если нужный проект был в работе сравнительно недавно, можно:

- запустить на выполнение Visual C++;

- выбрать подменю **File**, навести курсор мыши на пункт **Recent Workspaces**;

- если в появившемся меню со списком последних файлов, с которыми шла работа, есть файл `ProjectName.dsw`, щелкнуть по нему мышью.

Еще один вариант:

- не вызывая Visual C++, найти папку с нужным проектом, в ней — файл `ProjectName.dsw`;

- щелкнуть мышью на файле `ProjectName.dsw`, и ОС запустит на выполнение среду Visual C++, открыв при этом нужный для работы проект.

Литература

1. Т.Павловская, Ю.Щупак. *С/С++. Программирование на языке высокого уровня. Структурное программирование (Практикум)*. — СПб: Питер, 2003.



В.КАРЦЕВ,
Минск, БГУИР.

Исключения C++ и структурные исключения C в C++ Builder

Какие бывают исключения в C++ Builder

Известно, что обработка исключительных ситуаций в программе позволяет программисту создать приложение, "гибко" реагирующее на неизбежно возникающие при работе программы сбойные ситуации. Такая обработка позволяет компактно сосредотачивать обработку ошибок, которые могут возникнуть в разных частях функции или даже в разных функциях, в одном месте кода программы, что хорошо сказывается на размере исходного кода программы. Приложение, содержащее обработку исключений, легче отлаживать. Использование средств обработки исключительных ситуаций позволяет повысить надежность работы приложения. Таким образом, обработка исключений является важным средством в творческом арсенале любого программиста.

В данной статье рассмотрим обработку исключительных ситуаций в приложениях, созданных с помощью C++ Builder 6, где система управления исключительными ситуациями включает в себя: управление исключительными ситуациями, характерное для языка C++; управление исключительными ситуациями, характерное для языка C; управление исключениями, генерация которых связана с использованием библиотеки визуальных компонентов VCL.

Любой из упомянутых видов управления исключительными ситуациями и все сразу вместе могут быть использованы в одном и том же приложении. Далее будем в основном затрагивать вопросы, связанные с первыми двумя видами исключений.

Следует сразу сказать, что особенностей применения рассматриваемых типов исключений очень много. Остановиться на всех в рамках данной статьи невозможно. Поэтому будет рассмотрена лишь часть таких особенностей.

Управление исключениями, характерное для языка C++

Для реализации управления исключительными ситуациями, присущими языку C++, в общем случае используется та же конструкция, что и для исключений VCL [1]:

```
try
{операторы, при выполнении которых может возникнуть
исключительная ситуация}
catch (<описание исключительной ситуации E1>)
{обработчик исключительной ситуации E1>}
catch (<описание исключительной ситуации E2>)
{обработчик исключительной ситуации E2>}
...
catch (<описание исключительной ситуации En>)
{обработчик исключительной ситуации En>}
catch (...)
{обработчик непредвиденной заранее исключительной ситуации}
```

В скобках <> здесь и ниже указывается описание параметра. Сами скобки <>, разумеется, при наборе текста программы не нужны.

Описание исключительной ситуации имеет вид:

<тип исключительной ситуации> E_i ,

где E_i — необязательная переменная. В качестве типа исключительной ситуации языка C++ можно указать int, char, const char*, void*, ссылку на класс и т.п.

Если при обработке исключительных ситуаций VCL и исключительных ситуаций языка C++ используется одна и та же конструкция, то может возникнуть вопрос: "А есть ли между ними какие-нибудь различия?". Да, есть. В том, что эти виды исключений отличаются друг от друга, убеждает такой пример:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int i;
    My my;
    try
    {i=StrToInt(Edit1->Text);
     if(i==5) throw my;
     if (i==7) throw «7»;
     if (i==9) throw EIntError("Введено число 9!");}
    catch(const Exception& ex)
    {ShowMessage("Исключение VCL!" + ex.Message);}
    catch(char*)
    {ShowMessage("Исключение C++! Обработчик блока catch(char*).*");}
    catch(My&)
    {ShowMessage("Исключение C++!");}
}
```

После ввода в Edit1 числа 5 сработает catch (My&), а не catch(const Exception& ex). Здесь My является классом, описанным вне тела функции void __fastcall TForm1::Button1Click(TObject *Sender);. В примере считается, что он не имеет никакого отношения к классу TObject. Если бы различий между исключениями C++ и исключениями VCL не было, то обработчик, стоящий после catch(const Exception& ex), стал бы единственным обработчиком, реально работающим в этой функции. Но класс Exception не имеет никакого отношения к классу My. Именно поэтому в данном примере, после ввода числа 5 в Edit1, на экране появится окно с сообщением "Исключение C++!". Похожая ситуация складывается и при вводе числа 7. В этом случае сработает обработчик блока catch(char*), а не блока catch(const Exception& ex).

Как видно из примера, различия между исключениями VCL и C++ не мешают им вместе "сосуществовать" в этом примере. Так, при вводе числа 9, на экране в результате обработки исключения VCL появится сообщение "Исключение VCL! Введено число 9!". Мне не известны какие-либо ограничения для подобного рода соединения в одной конструкции try...catch этих двух видов исключений.

Примером другого отличия (и не последнего) исключений VCL от исключений C++ является то, что C++-исключения могут перехватываться не только по ссылке и по указателю, но и по значению.

Обработка исключительной ситуации C++ осуществляется следующим образом. Если при выполнении операторов, стоящих после try (но до первого catch), выброшена (термин C++ [2]) исключительная ситуация, то управление вычислительным процессом сразу же переходит:

- 1) либо к обработчику одного из catch-блоков, который содержит "подходящий" формальный параметр;
- 2) либо к обработчику catch-блока, у которого в качестве формального параметра указано многоточие (...);
- 3) либо к обработчику исключения, выполняемому по умолчанию.

В качестве "подходящего" формального параметра могут выступать:



- 1) ссылка на класс, соответствующий данной исключительной ситуации;
- 2) ссылка на класс, являющийся для рассматриваемого класса исключения базовым;
- 3) тип, совпадающий с типом исключительной ситуации, и т.п.

Если подходящих обработчиков несколько, то будет выполнен тот, который окажется первым в указанной выше синтаксической конструкции.

Следует сказать, что у рассмотренного подхода к формированию обработки исключительных ситуаций, характерных для C++, имеется ряд особых "вариантов" применения. Один из них будет рассмотрен в данной статье более подробно чуть позже. Я имею в виду специальные классы, связанные с обработкой исключений, входящие в состав библиотеки Standard C++ Library, содержащей в себе стандартную библиотеку шаблонов STL.

Управление исключениями, характерное для языка C

Некоторые из исключений, характерных для языка C, и наиболее вероятные причины их возбуждения описаны в таблице.

Тип структурного исключения	Причина возникновения исключения
EXCEPTION_ACCESS_VIOLATION	Попытка обращения к недоступной области памяти
EXCEPTION_BREAKPOINT	При выполнении приложения была обнаружена точка останова (точка прерывания)
EXCEPTION_FLT_DIVIDE_BY_ZERO	При выполнении операции над вещественными числами обнаружилось деление на ноль
EXCEPTION_FLT_OVERFLOW	Слишком большое значение показателя при вычислении степени вещественного числа
EXCEPTION_INT_DIVIDE_BY_ZERO	При выполнении операции над целыми числами обнаружилось деление на ноль
EXCEPTION_PRIV_INSTRUCTION	Попытка выполнения команды, недопустимой в данном режиме работы компьютера
EXCEPTION_NONCONTINUABLE_EXCEPTION	Программа пытается продолжить вычисления после возбуждения невозобновимого исключения

Достоинством исключительных ситуаций языка C является то, что они могут быть обработаны и в случае компонентов, которые скомпилированы не в данной C++-среде.

Эти исключения называются структурными, структурированными (structured exception) или исключениями Win32. Такое управление исключительными ситуациями, характерное для языка C, в модулях, написанных на C++, обеспечивается с помощью одной из двух синтаксических структур (перед словами `except` и `finally` в приведенных ниже конструкциях следует ставить два символа подчеркивания, а не один).

Две конструкции управления структурными исключительными ситуациями в C++-модулях:

```
try
{<операторы>}
except(<фильтрующее выражение >)
{< операторы обработчика >}
и
try
{< операторы >}
finally
{<операторы, обеспечивающие завершение блока try>}
```

Первая конструкция (с ключевым словом `__except`) получила название кадрированного управления, а вторая — завершающего [2]. Операторы, записанные в фигурных скобках непосредственно после ключевого слова `try`, носят название тела защищенного кода (guarded body of

code). Данные синтаксические конструкции могут быть вложены одна в другую. Из этих двух конструкций, вообще-то говоря, лишь первая осуществляет действительно обработку исключительных ситуаций. Конструкция `try...__finally` служит для обеспечения выполнения тех действий в программе, которые необходимо осуществить независимо от того, возникла в теле защищенного кода исключительная ситуация или нет. Эти действия и описаны в фигурных скобках, стоящих после слова `__finally`.

В модулях, написанных на C, а не на C++, используется ключевое слово `_try` вместо `try`.

Фильтрующее выражение (filter expression) может принимать одно из трех возможных значений:

- EXCEPTION_EXECUTE_HANDLER;
- EXCEPTION_CONTINUE_EXECUTION;
- EXCEPTION_CONTINUE_SEARCH.

Обработка исключительной ситуации в случае первой конструкции выполняется в зависимости от значения фильтрующего выражения. Пусть при выполнении операторов тела защищенного кода заявлена (термин языка C [2]) исключительная ситуация. Если значение фильтрующего выражения равно `EXCEPTION_EXECUTE_HANDLER`, то

управление вычислительным процессом сразу же передается обработчику блока `__except`. Если фильтрующее выражение равно `EXCEPTION_CONTINUE_EXECUTION`, то не приняв специальных мер, можно запросто "подвесить" систему, так как, в конечном счете, вычислительный процесс будет постоянно возвращаться к тому месту кода программы, в котором возникло исключение.

При значении фильтрующего выражения, равном `EXCEPTION_CONTINUE_SEARCH`, будет предпринята попытка поиска другого обработчика исключения, а обработчик, стоящий после `__except`, будет фактически проигнорирован.

При использовании кадрированного управления исключительными ситуациями (возможно, что такое управление следует именовать фреймовым, исходя из буквального перевода frame-based exception) следует иметь в виду, что получить необходимую информацию об исключении можно с помощью функций `GetExceptionInformation()` и `GetExceptionCode()`.

Работа с указанными выше функциями имеет ряд особенностей. Действительно, фильтрующее выражение может быть довольно сложным. В частности, из него можно осуществлять вызов так называемой фильтрующей функции или функции фильтра (filter function). Эту функцию может написать сам пользователь C++ Builder. Для работы функции фильтра, как правило, требуется информация, которую можно получить с помощью функции `GetExceptionInformation()`. Но функция фильтра не имеет права вызывать функцию `GetExceptionInformation()`, позволяющую получить самую подробную информацию о возникшем исключении, хотя само фильтрующее выражение может содержать вызов функции `GetExceptionInformation()`. Поэтому на практике внутри фильтрующего выражения вызывают функцию `GetExceptionInformation()` и присваивают ее значе-



ние указателю на структуру EXCEPTION_POINTERS, а вот значение этого указателя и передают функции фильтра. При этом может быть использован один из двух вариантов формирования фильтрующего выражения. В первом случае фильтрующее выражение состоит из двух частей, соединенных операцией запятой. Первая часть имеет вид `pointer = GetExceptionInformation()`, вторая — вызов функции фильтра `filterfunction(pointer)`. Во втором случае запись фильтрующего выражения приобретает вид

```
filterfunction(pointer = GetExceptionInformation()).
```

Здесь `pointer` — указатель на структуру EXCEPTION_POINTERS.

В результате работы функции `GetExceptionInformation()` можно получить указатель на структуру EXCEPTION_POINTERS, которая, в свою очередь, содержит два указателя на структуры. Первый из указателей имеет тип PEXCEPTION_RECORD (указатель на структуру EXCEPTION_RECORD), второй — PCONTEXT (указатель на структуру CONTEXT). Структура EXCEPTION_RECORD содержит информацию о коде исключения, о возобновимости или невозобновимости исключения, об адресе места возникновения исключения и т.д. Из структуры CONTEXT можно извлечь информацию о состоянии регистров процессора в момент возникновения исключительной ситуации. Такая информация может понадобиться, например, при обработке исключения, когда фильтрующее выражение оказывается равным EXCEPTION_CONTINUE_EXECUTION.

В отличие от функции `GetExceptionInformation()`, которую следует вызывать только из фильтрующего выражения, функцию `GetExceptionCode()` можно вызывать либо из фильтрующего выражения, либо из блока обработчика `__except`. Функция `GetExceptionCode()` позволяет узнать код исключения

(типа DWORD). Она, как и функция `GetExceptionInformation()`, не имеет формальных параметров.

Как выбросить или заявить исключение

В C++ Builder предусмотрены средства для выбрасывания и заявок исключений. В случае C++-исключений это осуществляется посредством ключевого слова **throw** с последующим за ним необязательным операндом. Например,

```
throw "Ошибка данных";
```

Для заявки исключений языка C используется функция

```
RaiseException (DWORD expcode, DWORD expflag, DWORD numarg,
const DWORD* pointarg);
```

Где:

- `expcode` — код исключительной ситуации. Для исключений, генерируемых пользователем, необходимо 29-й бит установить в 1; биты с 0 по 27 определяются пользователем для формирования индивидуально выбранного кода исключения; бит 28-й — резервный; 30-й и 31-й биты определяют характер исключительной ситуации (ошибка, предупреждение и т.д.).

- `expflag` — флаг исключения, определяющий возобновимое (EXCEPTION_CONTINUE) или невозобновимое (EXCEPTION_NONCONTINUABLE) исключение;

- `numarg` — количество аргументов в массиве, на который указывает `pointarg`;

- `pointarg` — указатель на первый элемент массива аргументов. Значение аргументов определяется в зависимости от назначения генерируемой исключительной ситуации.

(Продолжение следует)

ИССЛЕДОВАНИЕ ПРОГРАММ

CyberManiac /HI-TECH,
www: <http://wasm.ru/>

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КРЭКИНГА

(Продолжение. Начало в №№8-12/2003, 1-12/2004, 1-2,4/2005)

Патчинг и перехват системных вызовов — отнюдь не прерогатива одних лишь крэкеров. Разработчики защит, хотя и с изрядным опозданием, тоже взяли на вооружение идею перенаправления системных вызовов на собственный код. Как я уже упоминал, сравнительно недавно в большом количестве появились навесные защиты, которые позволяют упрячивать любую программу и все необходимые для ее работы файлы в упакованный и зашифрованный виде внутрь одного-единственного EXE. При этом сама программа может обращаться к своим файлам как средствами Win32 API, так и при помощи высокоуровневых функций (которые, по сути, являлись «обертками» для все тех же системных вызовов). О том, каким образом работают такие защиты, вы уже наверняка догадались — в исполняемый файл дописывается секция, где хранится код, обрабатывающий вызовы системных функций для работы с файлами. Необходимые для работы программы файлы упаковываются и цепляются в «хвост» программы, а таблица импорта дорабатывается таким образом, чтобы вызовы функций работы с файлами перенаправлялись на обработчики, находящиеся во внедренной секции. Когда защищенная про-

грамма попытается обратиться к файлу, задача обработчиков заключается в том, чтобы проверить, к какому именно файлу происходит обращение, и либо передать это обращение операционной системе в неизменном виде (если программа обращается к файлу вне «хранилища»), либо имитировать работу системного вызова, но в действительности считывание данных осуществлять из упакованного «хранилища». Однако ирония судьбы заключается в том, что метод взлома полностью аналогичен методу защиты. Действительно, если разработчик «отклонил» вызовы функций работы с файлами на собственный код, то почему бы и крэкеру не проделать ту же самую операцию? То есть вклинить между программой и обработчиком свой собственный «обработчик обработчика», который будет сбрасывать все «спрятанные» файлы в надежное место. Основную проблему составляет поиск входных и выходных точек этих обработчиков, но здесь могут помочь «особые приметы», которые способны выдать чужеродный код:

1. Т.к. вклинить «левый» код в середину практически невозможно, остаются варианты с расширением секций (тогда нужный код окажется перед началом или в самом



конце программы), созданием новой секции (что иногда заметно по "странным" адресам, заметно отличающимся от адресов основной программы) или динамическим выделением куска памяти и размещением там обработчика (самый сложный для реализации способ — а потому самый маловероятный).

2. Если защищенная программа читает упакованные файлы "по требованию", каждый вызов функций чтения файлов внутри программы будет сопровождаться обращениями программы к "хвосту" своего EXE-файла. Если программа читает все упакованные файлы в память сразу, такое обращение будет выполнено во время запуска программы. В обоих случаях эти вызовы вполне "уязвимы" для точек останова.

3. Наиболее уязвимы те вызовы, которые передаются в систему в неизменном виде (т.е. обращения к файлам, не находящимся внутри упакованного "хранилища") — установка брейкпойнтов на функции работы с файлами позволяет найти их без особого труда, после чего можно добраться и до точек входа и выхода в защитные процедуры-обработчики системных вызовов.

4. В памяти почти навряд ли будут застревать куски спрятанных файлов, и если вы знаете, какая информация в этих файлах может оказаться, вы можете попытаться найти эти куски памяти и попытаться выяснить, каким образом эти куски там появляются (например, при помощи поиска ссылок на начала таких кусков или установкой брейкпойнтов на запись в память).

Все эти внедрения в чужой процесс, поиски переходников и прочее вполне способны повергнуть в шок начинающего. Это не страшно — даже если вы никогда не напишете собственный API-шпион, знания о том, как они работают, вполне могут вам пригодиться в дальнейшем. Как вы уже догадались, написание приложений, перехватывающих системные вызовы — занятие далеко не самое простое и требующее определенных знаний и навыков. С другой стороны, большинство API-шпионов ограничиваются лишь ведением журнала системных вызовов и не позволяют активно вмешиваться в работу программы. Для начинающего крэкера, у которого есть желание перехватить какой-нибудь системный вызов, но нет опыта в системном программировании, это звучит как приговор. Но ведь так хочется иногда не только посмотреть, откуда взялись те или иные параметры, но еще и поменять их "на лету", если они вам чем-то не понравились...

Я уже говорил, что из любой безвыходной ситуации существует как минимум два выхода. Этот афоризм верен и в нашем случае, однако от вас все же потребуются определенные навыки в программировании. Итак, наша задача — вклинить свой код между системной DLL и программой, которая ее вызывает. Одна из первых идей, которые приходят в голову, заключается в том, чтобы вместо "родной" DLL подsunуть свою собственную, которая содержала бы функции с такими же именами, что и "настоящая". Эта библиотека должна, помимо вызова "родных" функций из оригинальной библиотеки, выполнять еще и те операции, которые вы в нее заложите. А уж в собственной DLL вы вольны запрограммировать все, что угодно — от сбора статистики вызовов до анализа и подмены параметров функций. Такой подход к перехвату вызовов из DLL, основанный на подмене оригинальных библиотек, называется DLL

wrapping'ом. Чисто технически создание подменной DLL выполняется следующим образом:

1. Необходимо получить список всех функций (в том числе и тех, которые не имеют имен, а только ординалы), экспортируемых той DLL, которую вы планируете подменить. Если вы уверены, что программа не импортирует функции этой DLL при помощи GetProcAddress или каким-либо более изощренным способом, вы можете обойтись и списком функций, импортируемых подопытной программой из DLL. Получить такой список можно, к примеру, при помощи утилиты DUMPBIN.

2. Изготовить на основе этого списка болванку будущей подменной DLL. В простейшем случае эта болванка будет выглядеть как набор одинаковых кусков. Поскольку болванка имеет регулярную структуру, ее можно сгенерировать автоматически. Для простейшего случая, когда нужно перехватить одну лишь функцию MessageBoxA из user32.dll, наша подменная библиотека, написанная на MASM, будет выглядеть примерно так:

```
PUBLIC MessageBoxA
```

```
.data:
IsLoaded dd FALSE
aMessageBoxA dd 0
huser32 dd 0
u32 db "e:\Windows\System32\user32.dll",0
nMessageBoxA db "MessageBoxA",0
```

```
.code:
```

```
DllEntry proc hInstance:HINSTANCE, reason:DWORD,
reserved1:DWORD
```

```
mov eax,TRUE
ret
```

```
DllEntry Endp
```

```
CheckAndImport proc
```

```
.if !IsLoaded
mov IsLoaded,TRUE
invoke LoadLibrary,ADDR u32
mov huser32,eax
invoke GetProcAddress,huser32,ADDR nMessageBoxA
mov aMessageBoxA,eax
endif
ret
```

```
CheckAndImport endp
```

```
MessageBoxA::
```

```
; Адрес метки MessageBoxA и прочих аналогичных нужно включить
; в таблицу экспорта
; при помощи директив
; LIBRARY user32
; EXPORTS MessageBoxA,
; помещенных в def-файл
```

```
invoke CheckAndImport
```

```
; Здесь вы можете разместить код, исполняемый перед вызовом
; перехватываемой функции
; Например, поменять местами заголовки и текст сообщения,
; как это сделано ниже
```

```
mov eax,[esp+8]
xchg [esp+12],eax
xchg [esp+8],eax
```

```
mov eax, MessageBoxA_
jmp eax
```



Процедура `CheckAndImport` выполняет важную функцию — во время первого вызова любой из функций она подгружает настоящую `user32.dll`, получает адрес функции `MessageBoxA` и помещает этот адрес в переменную `aMessageBoxA`. Если вам известно, какая из функций DLL будет вызвана первой, можно сократить код, разместив вызов функции `CheckAndImport` только в этой функции. Вообще, внешне этот код выглядит довольно тяжеловесно — экспорт меток, специфический способ обращения к параметрам вызова через смещение относительно значения ESP, динамическая загрузка библиотек — и тут же отсутствие возможности выполнить собственный код после вызова `MessageBox`. Причина тяжеловесности проста — этот код представляет собой максимально упрощенную адаптацию реально использовавшейся мною подменной DLL. Что же касается мнимой невозможности выполнить собственный код после вызова `MessageBoxA`, эту проблему проще всего решить при помощи подмены лежащего на стеке адреса возврата — нужно подправить его таким образом, чтобы после выполнения команды `jmp <имя_функции>` возврат выполнялся не в программу, а на следующую за `jmp` команду. Разумеется, старый адрес возврата тоже надо где-то сохранять — он вам понадобится, чтобы вернуть управление программе. Если вам известно количество параметров, помещаемых на стек при вызове, и их размерность, никаких сложностей с подменой адреса возврата не возникнет. Но вот если в процедуру передается неизвестное заранее количество параметров (такое, в частности, возможно в программах, написанных на C), общего решения этой проблемы не существует, так что вам придется действовать по обстоятельствам и изобретать метод определения количества параметров на стеке самостоятельно.

Но почему я вдруг отклонился от темы этой главы и уделил столько времени технике подмены DLL? Причина в том, что DLL wrapping нередко используется совместно с патчингом исполняемого файла. Внимательно посмотрев на текст подменной DLL, вы увидите, что загрузка `user32.dll` производится с явным указанием расположения этой библиотеки. Однако даже если скомпилировать эту DLL с именем "`user32.dll`" и положить рядом с подопытной программой, единственным результатом выполнения которой является вывод `MessageBox`'а с неким сообщением, вы все равно не добьетесь желаемого результата — программа вызовет эту функцию напрямую из системной библиотеки, проигнорировав вашу приманку. Что же делать?

Самый простой и доступный выход заключается в том, чтобы переименовать нашу подменную библиотеку в `user33.dll`, а потом забраться в исполняемый файл шестнадцатеричным редактором и поменять там имя импортируемой библиотеки аналогичным же образом. Теперь наша программа будет вместо стандартной библиотеки Windows подгружать нашу `user33.dll` и вызывать `MessageBoxA` именно из нее. Теоретически можно было поступить и несколько иначе — переименовать системную библиотеку `user32.dll` в `user33.dll`, а на место `user32.dll` положить нашу подменную библиотеку (разумеется, исправив путь к "настоящей" динамически загружаемой библиотеке), но на практике проделывать фокусы с заменой системных библиотек крайне нежелательно. Хотя если речь идет не о системной библио-

теке, то действительно можно обойтись одним лишь переименованием "настоящей" DLL и помещением на ее место "поддельной".

Изучая перехват системных вызовов, мы вторглись в высокие сферы патчинга процессов, и теперь настало время всерьез углубиться в эту тему. Хотя "в общем пользовании" всевозможные `launcher`'ы, правящие файлы прямо в памяти, попадают сравнительно редко, для домашнего использования техники патчинга процессов они более чем актуальны. Если у вас есть сжатая упаковщиком EXE-программа, но нет желания заниматься ее распаковкой, либо вы просто хотите поэкспериментировать над программой, проверяющей контрольную сумму собственного файла, патчинг кода "на лету" — это весьма эффективный способ добиться желаемого с минимальными усилиями.

Модификация кода программы в памяти может выполняться двумя путями: записью данных в адресное пространство процесса извне или же внедрением в адресное пространство процесса собственного кода, который уже будет работать внутри подопытного процесса и выполнять необходимые действия по патчингу. В подавляющем большинстве случаев вторжение в чужое адресное пространство извне осуществляется при помощи последовательности вызовов `VirtualProtectEx-WriteProcessMemory`, хотя возможны и более сложные варианты с выходом в нулевое кольцо (правда, после того как линейка Windows 9x начала утрачивать актуальность, простые в осуществлении способы выхода в Ring0 остались не у дел). Для того чтобы добраться до процесса, вам почти наверняка потребуется его дескриптор (он же "хэндл" — "handle"). Если запуск программы и патчинг ее процесса выполняется одной и той же программой, особых сложностей не возникает — функция `CreateProcess[Ex]` возвращает хэндл порождаемого ею процесса. Но это не единственный подход к добыче желанного дескриптора — до него также можно добраться через получение идентификатора процесса, породившего окно (при помощи функции `GetWindowThreadProcessId`), либо через анализ "снимка" всех процессов, полученных при помощи функций `CreateToolhelp32Snapshot`, `Process32First` и `Process32Next`.

Правка кода путем предварительного внедрения в адресное пространство подопытного процесса применяется значительно реже, однако эти знания тоже могут вам понадобиться. Наиболее распространенными способами попасть в чужое адресное пространство являются: использование хуков, запуск программы в режиме отладки и создание потока при помощи функции `CreateRemoteThread` (этот прием применим только под Windows линейки NT). Однако ничто не ограничивает ваше творчество — вы вольны комбинировать различные приемы для достижения наилучшего эффекта. Например, можно пропатчить программу в районе точки входа так, чтобы происходила загрузка некой DLL и вызов функции, которая восстановит испорченные патчингом байты, создаст дополнительный поток и затем вернет управление основной программе.

(Окончание следует)



Е.ЛЕБЕДЕВ,
г.Череповец.

Знакомство с WinAsm Studio

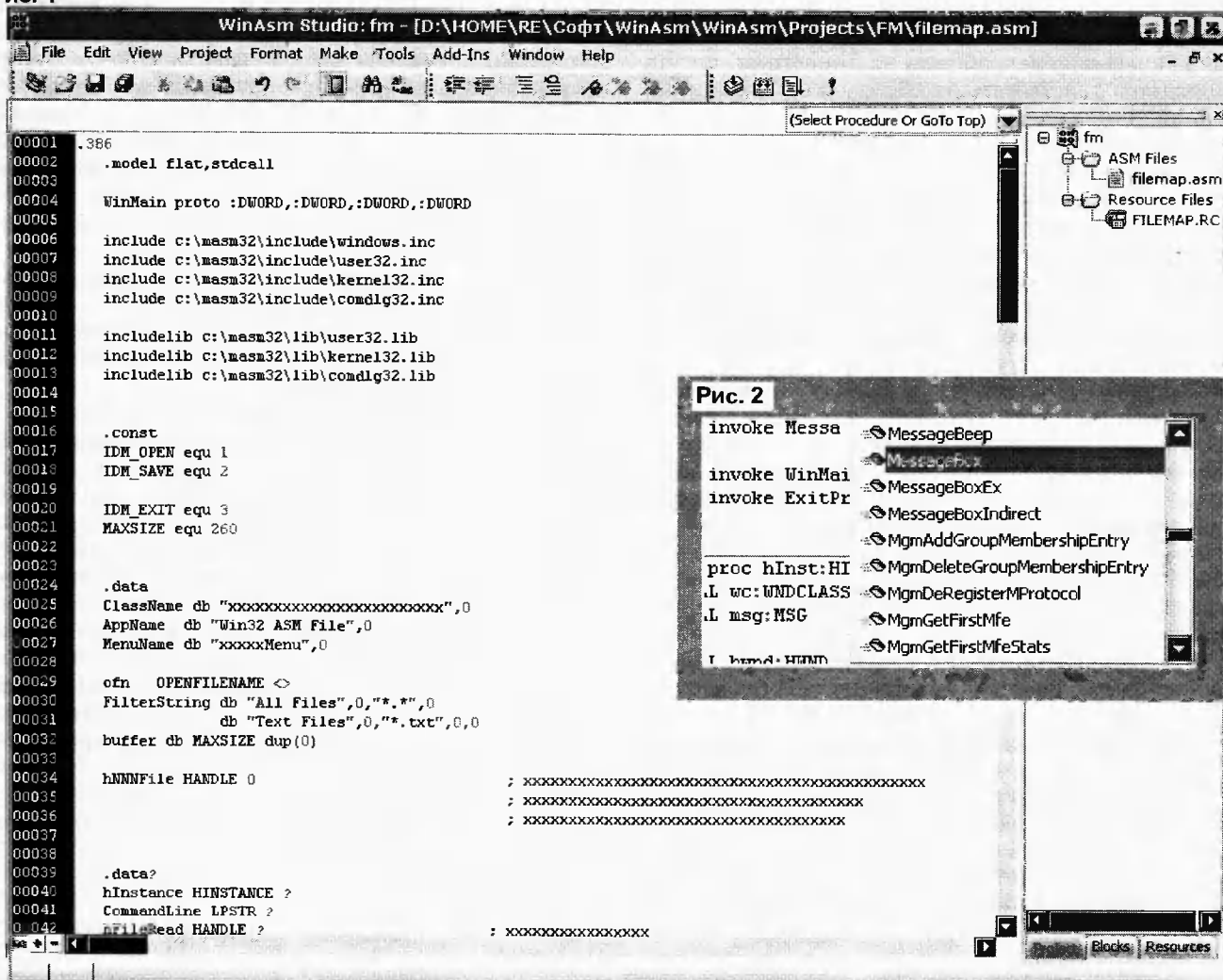
Во Всемирной паутине существует огромное количество сайтов, посвященных различным редакторам. Лидерство среди них, пожалуй, занимают текстовые редакторы с расширенной функциональностью. За "лидером" данной серии продуктов тянутся редакторы языка HTML, 3D-редакторы, звуковые редакторы и др. В каждой из приведенных в качестве примера подгрупп есть свои лидеры и свои "аутсайдеры" (outsider).

Знакомство

Я хотел бы остановиться на обзоре одного "чуждого" (в хорошем смысле этого слова :) программного продукта — WinAsm Studio. Автор программы (Antonis Kyrianiou) пишет: "...WinAsm Studio — это интегрированная среда программирования для создания 32-разрядных приложений на языке Ассемблер...". Общий вид рабочего окна программы показан на рис.1.

После запуска WinAsm Studio перед нами открывается среда разработки с очень приятным и, главное, дружелюбным интерфейсом. Вся оболочка разделена на три основных "зоны": панель инструментов, окно кода и окно с фай-

Рис. 1



лами проекта. Это действительно удобно и гармонично.

Панель инструментов включает в себя основные, самые необходимые функции:

- "New project" — создать новый проект. В этом случае пользователю предоставляется неплохой выбор. Можно создать проекты следующих типов: "Стандартный EXE", "Стандартная DLL", "Консольное приложение", "Static library", "DOS проект", "Другой EXE", "Другой не-EXE". Помимо пустых шаблонов, нам даются шаблоны стандартных диалоговых окон, стандартных программ типа exe и com (для DOS), шаблоны приложений SDI и MDI;
- "Open project" — открыть проект;
- "Save file" — сохранить активный файл проекта;
- "Save All" — сохранить все файлы проекта;
- кнопки для работы с текстом — в этот раздел входят кнопки Вырезать ("Cut"), Копировать ("Copy") и Вставить ("Paste");
- "Undo" и "Redo" — операции отмены/повтора последнего действия;
- "Show/Hide Project Explorer" — показать/скрыть проводник файлов проекта;
- "Find" — стандартный диалог поиска в тексте;
- "Replace" — диалог замены;
- "Increase Indent" и "Decrease Indent" — кнопки настройки отступов в ассемблерном коде;
- "Comment block" — закомментировать строку. Пользователь ставит курсор в нужную строку или даже выделяет определенный кусок текста, и тот становится комментарием;
- "Uncomment block" — раскомментировать текст;

• "Toggle bookmark" — поставить "закладку". Программист ставит на нужные строки кода закладки и потом может свободно перемещаться именно по этим закладкам. Это очень удобно, и позволяет сэкономить массу времени, особенно при работе с большими листингами;

- "Remove bookmark" — удалить "закладку";
- "Assemble" — ассемблировать код;
- "Link" — создать исполняемый файл или библиотеку (dll);
- "Go All" — сделать "Assemble" и "Link", а потом, в случае успеха — "Execute";
- "Execute" — запустить программу.

Окно кода включает в себя ряд приятных особенностей, среди которых:

- подсветка синтаксиса;
- распознавание стандартных оконных (и других!) функций;
- обширная система помощи;
- нумерация строк кода;
- функциональное контекстное меню.

Работая со средними и большими по размерам листинга программами, в первую очередь сталкиваешься с проблемой удобного чтения кода. Когда сегодня ты работаешь над проектом, потом на некоторое время "запускаешь" его и через определенное количество дней возвращаешься к нему — код очень тяжело воспринимать, да еще если он отображается одним цветом. В WinAsm есть функция подсветки синтаксиса, которая делает код приятным для восприятия. Также имеется система помощи, как в популярных продуктах от фирмы Borland. Пример такой помощи (подсказки) показан на рис.2. В частности, на рисунке показан пример всплывающей подсказки. Пользователь начал вводить имя функции "MessageBox", и тут же появилась подсказка с возможностью выбора огромного количества функций. Далее, после того как функция выбрана, следует вторая подсказка (рис.3) — програм-

Рис. 3

```
invoke GetCommandLine
mc
invoke MessageBox,
```

ма любезно предоставляет нам подсказку, какие параметры принимает данная функция. В нашем примере видно, что в функцию передаются:

- hWnd — оконный дескриптор (от какого окошка будет вызываться функция);
- lpText — текст сообщения;
- lpCaption — текст заголовка окна;
- wType — тип MessageBox-окна.

Рассмотренные подсказки очень полезны при работе с большим количеством функций. После довольно непродолжительного общения с WinAsm'ом меня приятно удивили функциональные возможности программы, относящиеся к работе с ресурсами.

Ресурсы

Все исполняемые файлы имеют свои ресурсы. Если в дорогих программных средах типа Borland Delphi или Borland C++ Builder ресурсы автоматически встраиваются в файл, то на языке Ассемблер программист сам указывает, что и куда поместить, благодаря этому сокращается код приложений, созданных на Ассемблере.

В WinAsm Studio включен специальный редактор ресурсов, который позволяет быстро создавать окна и их компоненты. Рассмотрим пример создания ресурсов.

Сначала создадим новый проект (File → New project). Выберем тип проекта — "Standart EXE".

Перед нами откроется чистое окно кода. В нашем случае мы ничего не будем набирать, а сразу же откроем окошко с

ресурсами (Project → Add new rc). Перед нами появится окно редактора ресурсов, показанное на рис.4. Как мы видим, по сути, нам предоставлены два окна: RC Options и Toolbox. Первое окошко отвечает за параметры создаваемого окна (формы): "ShowHide Grid" — показывать/скрывать сетку при редактировании; SnapToGrid — привязывать курсор к ячейкам сетки и т.д. Второе окошко — это некий "конструктор", содержащий в себе набор компонентов. В первоначальном состоянии активны только компонент создания новой формы и режим курсора. Воспользуемся первым компонентом и получим заготовку формы, как показано на рис.5.

После создания формы на панели с компонентами активизируются все остальные пункты. Теперь мы смело можем создавать то оконное приложение, которое нам нужно. Для примера создадим простейшую форму с одним полем ввода (EditBox) и кнопкой (Button), как показано на рис.6.

Форма готова, теперь мы можем посмотреть, какой код получится в результате таких совершенно нехитрых манипуляций. Для этого в меню выключаем Project → Visual Mode и получаем код сгенерированного файла ресурсов (рис.7).

Рассмотрим редактирование параметров ресурсов. Перекроем редактор в визуальный режим (Project → Visual Mode) и выберем на форме компонент Edit. Сразу же справа на панельке, в табличке отобразятся его параметры (рис.8):

- Name — имя компонента в ресурсах файла;
- ID — уникальный идентификатор в ресурсах;
- Left — смещение от левого края формы;
- Top — смещение от верха формы;

Рис. 4



Рис. 5

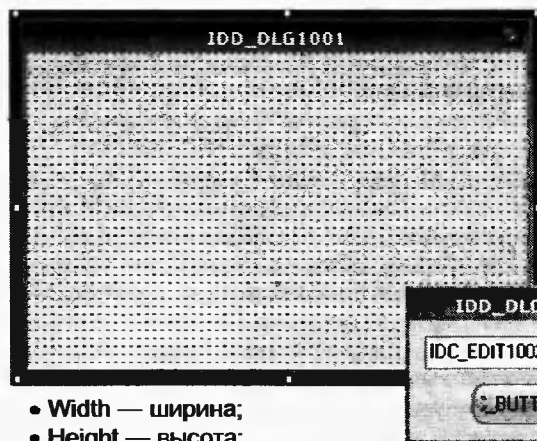


Рис. 6

- Width — ширина;
 - Height — высота;
 - Style — стиль;
 - ExStyle — дополнительный стиль;
 - Visible — показывать при старте формы или нет;
 - Text — текст в компоненте.
- Если мы выберем параметр "Style" или "ExStyle", то пе-



Рис. 7

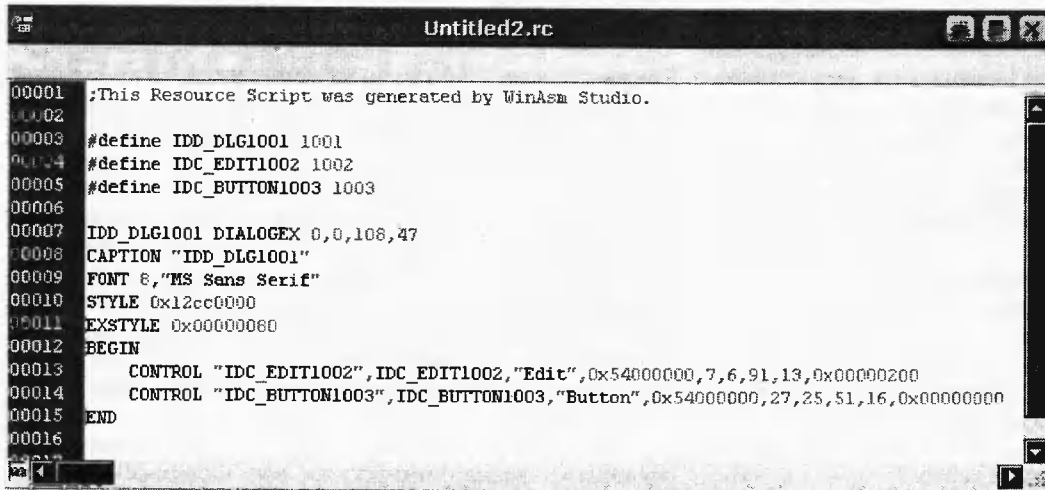


Рис. 8

Name	IDC_EDIT1002
ID	1002
Left	7
Top	6
Width	91
Height	13
Style	0x54000000
ExStyle	0x00000200
Visible	TRUE
Text	IDC_EDIT1002

ред ним появится небольшой кружок, при "клике" по которому откроется дополнительное окошко. Пример такого окна стилей показан на рис.9.

Окно "Window Styles" содержит в себе все доступные для данного компонента стили.

Рис. 10



Например, если изменить значение "No" на "Yes" в стиле "WS_DISABLED", то компонент будет неактивен.

Теперь рассмотрим еще два интересных момента использования WinAsm Studio — создание оконного меню и заполнение структуры "Version".

Создание нового меню производится в специальном окошке. Для его активации на панельке (рис.10) следует выбрать пункт "New Menu".

После нажатия кнопки появится окошко с диалогом создания меню, представленное на рис.11.

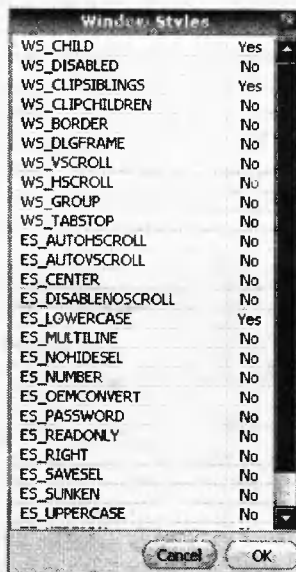
В окне "New Menu" нам предоставляется возможность создать меню любой сложности. Сначала вводим имя для меню в пункте "Name". Это имя должно быть уникальным и содержать приставку "IDM_". Как и у компонентов типа Edit и Button, у каждого меню есть свой уникальный (среди меню) идентификатор.

В разделе "Menu Item" мы работаем с конкретным пунктом меню. Для примера создадим пункт меню "Текст_пункта_меню". Для этого заполним поле "Caption", которое будет содержать название пункта меню; поле "Name" — имя этого пункта в меню; ItemID — уникальный номер этого пункта.

Также мы можем указать состояние меню: MF_CHECKED — отмеченное "галочкой"; MF_GRAYED — недоступное; MF_DISABLED — запрещенное.

В группе "Type" можно выбрать тип пункта меню: MF_STRING — обычная строка; MF_BITMAP — с рисун-

Рис. 9



WinAsm присутствует аналогичная функция. Для этого нажимаем на кнопку "Add Version Info" на панели ресурсов (рис.10). Появится окошко, как показано на рис.12.

В структуре "Version" мы можем заполнить стандартные поля типа: ProductName — имя продукта; ProductVersion — версия продукта; LegalCopyright — права на продукт; FileVer — версия файла; OS — операционная система; Language — язык; FileType — тип файла; Charset — кодировка.

Рис. 11

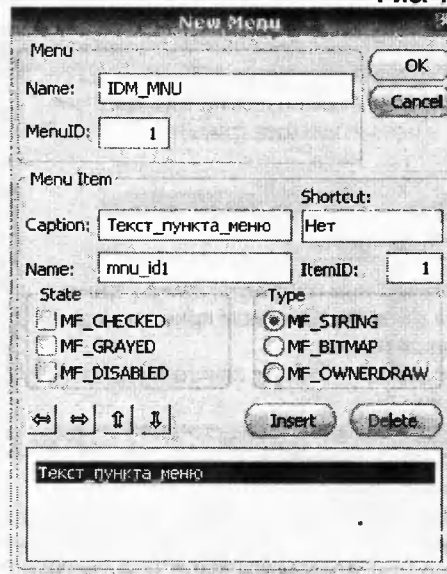
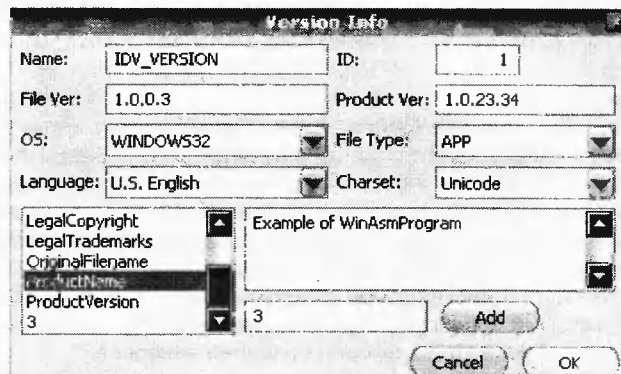


Рис. 12



ляет рассматриваемый программный продукт хорошей средой для разработки программного обеспечения любого уровня сложности.



Р.КАРПАЧ,

E-mail: Commander-Norton@tut.by
www.fdd5-25.net

Шаманские пляски вокруг CD-ROM TEAC CD-55a

Вот раньше, года этак в 60-е, все было классно — sex, drugs & rock'n'roll. А сейчас? Suxx, bugs & plug'n'play...

Не пугайся, мой дорогой читатель, на дворе по-прежнему XXI век. А мы сидим на навороченных Pentium III и IV. Но то, о чем я сегодня хочу написать, может заинтересовать каждого мало-мальски любознательного читателя, имеющего отношение к компьютеру.

Рождение компакт-диска

Изобретение цифрового компакт-диска традиционно приписывается двум компаниям: либо голландцы из Philips его придумали сами, либо им помогали японцы из Sony. Произошло это в самом начале 1980-х. Но есть другая версия — CD изобрел американский физик еще в 1960-х. Авторство вышеназванных фирм подтверждают многие источники, к примеру, популярная энциклопедия Wikipedia. Если верить ей, Philips и Sony совместно разработали цифровой компакт-диск в 1980-м, а спустя два года в городе Лангенхаген (Langenhagen), близ Ганновера, началось его массовое производство. Потом подключились Microsoft и Apple, чьими стараниями CD превратился в CD-ROM, который в 1987 г. совершил революцию в мире персональных компьютеров. Вот, стало быть, и вся история появления компакт-диска.

Теперь "альтернативная" точка зрения. В 1931 г. в Бремертоне (Bremerton), штат Вашингтон, родился Джеймс Расселл (James T. Russell). Свое первое изобретение он совершил в шесть лет — построил корабль с дистанционным управлением, в трюме которого по волнам ходил его завтрак. В 1953 г. Расселл закончил колледж в Портленде и стал бакалавром физики. В качестве физика он и устроился на работу в лабораторию компании General Electric, где затеял ряд экспериментальных проектов. Считается, что Джеймс Расселл был одним из первых, кто стал пользоваться цветным телевизионным экраном и клавиатурой в качестве человеко-машинного интерфейса. Он же первым спроектировал и построил агрегат для сварки пучком электронов. В 1965 г., когда базирующийся в штате Огайо институт Battelle Memorial открыл в Ричлэнде Тихоокеанскую северо-западную лабораторию (Pacific Northwest Laboratory), Расселл стал ее старшим

научным сотрудником. Тогда он уже знал, в каком направлении будет работать. Дело в том, что физик был страстным любителем классической музыки. И, как многие меломаны того времени, часто расстраивался из-за ухудшающегося со временем качества записи на виниловых пластинках. Пытаясь внести усовершенствования, ученый даже пытался использовать в качестве звуко-снимателя иголки кактуса. Однажды в субботу днем Расселл решил набросать схему лучшей, по его мнению, цифровой системы записи и воспроизведения звука. В итоге он "родил" действительно революционную идею — придумал устройство, в котором отсутствовал физический контакт между компонентами процесса проигрывания записи. На тот момент Расселл был знаком с цифровой записью данных на перфокартах и магнитной ленте, но понял, что лучший способ — использовать свет. 0 и 1, темнота и свет — раздвигал физик — если двоичный код достаточно хорошо уплотнить, в нем можно хранить не просто мелодии, а целые энциклопедии. В институте ученому, хотя и не сразу, но пошли навстречу, разрешив работать над персональным проектом по переводу аналогового сигнала в цифровой вид. И через пару лет Расселл изобрел первую оптико-цифровую систему записи и воспроизведения, которую запатентовал в 1970 г. Он нашел способ записи на жесткий фоточувствительный диск данных в виде крошечных "битов", светлых и темных, каждый — микрон в диаметре. Лазерный луч считывал двоичный код, а компьютер преобразовывал данные в электронный сигнал, который тогда было сравнительно просто преобразовать в слышимую или видимую "трансляцию". Вот это и было первым цифровым компакт-диском. В 1970-е изобретатель продолжил совершенствовать свое детище, пытаясь приспособить его к любой форме данных. Как и множество разработок, опередивших свое время, CD поначалу не слишком заинтересовал инвесторов. Но в 1971 г. рискованый бизнесмен Эли Джекобс (Eli Jacobs) основал корпорацию Optical Recording и пригласил Расселла в команду, которая должна была придумать видеодиск. Идея была такая — распространять телевизионные программы на небольших пластмассовых носителях, рассылая их по почте, что-

бы люди могли в любой момент посмотреть любимую передачу. По сути, речь шла о том, что сейчас называется видеоманитонами и кассетами для них. В 1974 г. на выставке в Чикаго компания представила оптико-цифровую телевизионную машину для записи и воспроизведения, первое устройство, которое переводило цветное изображение в цифру, но мир не перевернулся, инвесторы не отреагировали. Через год, летом 1975-го, в лабораторию Расселла наведальсь представители Philips и невысоко оценили его работу: "Они сказали, что это очень хорошо для хранения данных, но вы не можете приспособить это для видео или аудио", — вспоминал физик. Нужно сказать, что за несколько лет до посещения лаборатории голландская компания выпустила свой лазерный диск для аналогового оптического видеоплеера. В Нидерландах были убеждены, что "аналог" — единственно возможный вариант. Два месяца спустя после экскурсии по лаборатории изобретателя, Philips представила компакт-диск — практически точно такой же. В конечном счете, не только Philips, но и Sony, и другие компании вплотную занялись продвижением технологии Расселла, не упоминая его имени. Сам Расселл, впрочем, не монополизировал права на технологию: "Трудно сказать, сделали ли эти люди все сами, независимо от меня. Ведь в том, что у двух или более человек, находящихся в разных местах, может родиться одна и та же идея, ничего необычного нет. Вполне возможно, мы работали параллельно. Но позже они заплатили за это". Действительно, Sony и Philips выплатили лицензионные платежи от продаж проигрывателя компакт-дисков. Деньги получили институт Battelle Memorial, корпорация Optical Recording и ее хозяин Джекобс. В 1992 г. Time Warner и другие производители дисков подали на Optical Recording в суд, заплатив, в конечном счете, 30 миллионов долларов за нарушение патентов, так как суд решил — у корпорации эксклюзивные права на технологию CD. Тем не менее, из всех этих денег Расселл ни разу не получил ни цента, так как 26 патентов на "компакт" принадлежали его работодателю, то бишь — Optical Recording. Однако и это изобретателя не остановило. Он продолжил работать над оптическими системами хранения



данных и придумал нового конкурента накопителям на жестких дисках — оптическую память произвольного доступа (Optical random-access memory — ORAM). В этой системе нет вращающегося диска и вообще — ни одной движущейся части, данные считываются светом. В 1991 г. Расселл вместе с партнером Полом Наем (Paul Nye) создал компанию Ioptics — специально для разработки ORAM. Но даже несмотря на многомиллионные инвестиции от Microsoft, дело прогорело, система оказалась невостребованной. Чем сейчас занимается изобретатель, “породивший” за свою жизнь более 50 патентов, сказать сложно. Последние упоминания о нем в прессе датированы 2000 г., когда 53-летний Расселл был удостоен премии Vollum Award за выдающийся вклад в развитие науки и техники. Разумеется, во всей этой истории можно усмотреть стремление американцев приписать себе изобретение всех жизненно важных вещей. Но даже если это так, Джеймс Расселл пионером в своем деле быть не перестанет. Поэтому, справедливости ради, пусть, наряду с новаторством Philips и Sony, живет и эта версия появления CD.

Подключение CD-ROM

Как-то ко мне в руки попал старый 486/33 МГц, короче говоря, ничем не выдающийся компьютер. На нем был установлен дисковод, жесткий диск, но не было CD-ROM. Можно было купить новый привод. Но зачем? Платить цену, равную цене самого компьютера, за какой-то CD-ROM? Нет, это нонсенс! Проще уж “порыться” по друзьям и знакомым, на худой конец, заглянуть на компьютерную свалку в какой-нибудь фирме. Уж там точно что-то да найдется! И действительно нашлось. Один мой хороший знакомый пожертвовал мне CD-ROM TEAC странного внешнего вида — сам привод отличался маленькими габаритами, и хотя он был размером в стандартных 5 дюймов, корпус был наполовину меньше, чем у стандартного IDE CD-ROM. Во всяком случае, вещь оказалась на самом деле довольно любопытная. Вначале я не понял ее отличия от банального “резака”. Но это только поначалу. Загвоздка выявилась, когда я попытался установить этот самый TEAC в компьютер! Каково же было мое удивление, когда вместо нового диска E: в системе не появилось ровным счетом ничего. Нет, ну бывает же? Мало того, машина перестала грузиться. Выключив компьютер, я направился на поиски ответа.

Каким-то лихим ветром меня занесло в редакцию журнала, где еще более непонятным образом оказался мануал для этого вида CD-ROM. Выяснилось, что для работы столь “зверского” привода нужен адаптер или звуковая карта с соответствующим выходом. Но больше всего меня поразил тот факт, что адаптеры для TEAC бывают 8- и 16-битными, что, теоретически, опускает планку работы приводов данного класса до машин на базе процессоров 8088. Представьте себе такое зрелище! Но, лиха беда начало. После всех злоключений я решил обратиться к глобальному разуму, то есть к Интернету, чтобы узнать все о TEAC CD-55a, именно так назывался привод. Оказалось, что в природе существуют три не IDE-интерфейса CD-ROM: Sony, Panasonic, Mitsumi. Чем они отличаются, я так и не понял. Но знаю, что у Sony самое малое количество штырьков на выходе, зато для Panasonic и Mitsumi подходит обычный IDE-шлейф.

Сейчас же история о том, как пришлось выбирать соответствующую звуковую карту, чтобы подключить CD-ROM. Кстати, мой TEAC оказался с интерфейсом Panasonic, что намного облегчило поиски. А чтобы вы долго не искали драйвера к таким приводам, вот вам ссылка — <http://www.fdd5-25.net/drivers/cdromas.zip>.

Честно говоря, я сразу решил убить двух зайцев — получить звуковую карту к 486-му компьютеру с возможностью подключения CD-ROM. После долгих скитаний по одному из минских рынков я обнаружил агрегат — ESS 688f. Он был весь утыкан перемычками, словно компьютерный ежик (см. рисунок). Почти такой же, как был изображен в мануале к CD-ROM. “Оно!” — радостно подумал я. За три самых настоящих американских рубля звуковая карта ESS 688f была приобретена в частное пользование.

Не скажу, что мне особо верилось в работоспособность всех комплектующих, но все же я приступил к экспериментам. Первым и самым важным вопросом, вставшим на повестке дня, был: “Куда и что втыкать?”. Глупо?

Может быть, но когда перед тобой лежит израненная паяльником материнская плата и такая же звуковая карта, невольно задумываешься: “А не сгорит ли все это?” Не сгорело. Но проблемы только начинали прибавляться. Разработчики ESS 688f не удосужились отметить, как правильно втыкать шлейф. Поэтому стоило ожидать того, что первый “блин” окажется комом. Лампочка на CD-ROM загорелась и не хотела гаснуть. Пришлось выключать компьютер. Когда же дело дошло до установки драйверов, все заклинило окончательно. Вот, допустим, при подключении обычного CD-ROM в DOS мы пишем что-нибудь в таком духе:

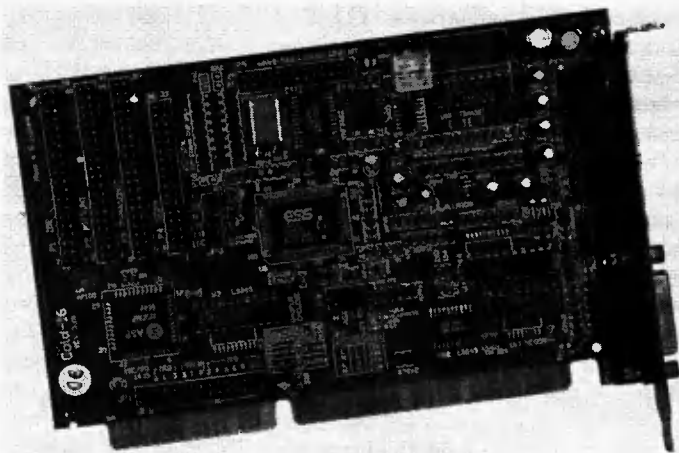
Autoexec.bat:

C:\DOS\MSCDDEX.EXE /S /d:mscd001

Config.sys:

DEVICE=C:\DOS\qscdrom.sys /d:mscd001

У многих и с этим возникали проблемы. Но когда нужно еще указать тип адаптера и номер его порта в памяти, здесь даже самый опытный пользователь без документации застрянет надолго! Пришлось воспользоваться тем, что было найдено в бескрайних просторах Интернет. Все обнаруженное сводилось к двум банальным файлам-архивам с драйверами. К счастью, в одном из этих архивов был обнаружен html-документ с прибли-



тельным описанием установки TEAC CD-55a. “Странные создатели этого описания указали, что работает даже на 386-х компьютерах, а документ в гипертекст засунули. Буржуи!”, — подумал я. Вот что я почерпнул из этого текста:

- 1) TEAC CD-55A 8-BIT I/F card.
- 2) TEAC CD-55A 16-BIT I/F card.
- 3) Any TRUE SoundBlaster cards with Panasonic CD-ROM interface.
- 4) SoundBlaster compatible card with a Panasonic interface.



Кому непонятно, поясню, что этот CD-ROM может подключаться через упомянутые устройства. Это вновь вселило в меня некоторую надежду на его работоспособность. Тем более, что в тексте упоминался некий "SoundBlaster compatible", чем, по сути, и являлась ESS 688f. Я потерпел и начал читать дальше. Но то, что мне пришлось увидеть, повергло бы любого в шок. Пример:

The TEAC 16-bit card (It's default address is 02C0h)

The recommended addresses and card jumper-settings are as follow:

ADDRESS *SWITCH-SETTINGS*

2C0 Set all 3 jumpers to the OFF position

2E0 Set only jumper 3 to the ON Position

300 Set only jumper 2 " "

320 Set jumpers 2 & 3 " "

340 Set only jumper 1 " "

360 Set jumpers 2 & 3 " "

"Ну, приехали! А ведь так хорошо все начиналось!", — невольно пронеслась мысль. Написанное далее еще больше меня взбудоражило. На русском это звучит примерно так:

Пожалуйста, определите, какой адрес для интерфейса CD-ROM находится в звуковой карте, а также, какой порт и диапазон адреса I/O.

Это сейчас мне понятно, о чем шла речь. Ведь далее текст звучал еще более устрашающе:

Note that you'll be using the CD-ROM I/O port address and not the SoundBlaster compatibility address of 220h. Please reference the documentation provided by the soundcard's manufacturer as TEAC doesn't keep SoundBlaster compatible cross-reference documentation.

То есть мне следовало где-то, неизвестно где, установить адрес звуковой карты на 220h и затем еще выяснить, какой адрес она использует для CD-ROM, и собственно прописку самого привода. Положение напоминало классический детектив. Чем дальше шло расследование, тем больше появлялось загадок и неясностей. Я попытался мыслить логично — методом взаимноисключений. И первое, что было сделано — звонок тому самому знакомому, который отдал этот CD-ROM. Его ответ поверг меня в страшное смятение. Он сказал, что адрес CD-ROM — 170h. То есть получалось: 220h, 170h. Ничего не понимая, я начал дальше вчитываться в незнакомый английский язык документации. А дальше предлагалось модернизировать файл config.sys, записав в него параметры драйвера того самого CD-ROM.

Для восьмибитной карты адаптера строка была такой:

DEVICE=C:\TEAC\TEAC_CDA.SYS /D:TEAC-CDA /P:230 /T:1;

Для шестнадцатибитной:

DVICE=C:\TEAC\TEAC_CDA.SYS /D:TEAC-CDA /P:2C0 /T:1;

Для оригинального SB:

DEVICE=C:\TEAC\TEAC_CDA.SYS /D:TEAC-CDA /P:220 /T:0;

Для аналога SB — строка:

DEVICE=C:\TEAC\TEAC_CDA.SYS /D:TEAC-CDA /P:340 /T:2.

"Бред!", — подумал я. Ведь документация противоречила сама себе. Как можно в аналоге SB установить адрес 220h, и самое главное, каким образом? В этот момент я окончательно запутался и начал судорожно вспоминать утилиты, которые могли бы помочь в решении этого вопроса. Чисто ради эксперимента были испробованы все варианты строк инициализации — ни одна не сработала. Единственное, чего удалось достичь — возможности поиска установленного CD-ROM при загрузке ДОС. Что чуть-чуть обнадеживало. Уныло нажимая кнопку перезагрузки, я вчитывался в текст документации. В конце ее приводились всевозможные причины неработоспособности привода. Но все, как и следовало ожидать, сводилось к рекомендации — проверьте кабель. Со всем отчаявшись, я уж решил искать драйвера к звуковой карте ESS 688f. Однако у них у всех оказалась привычка запускаться только из-под Windows. "Может, DOS не тот", — пронеслась в голове мысль. Хотя MS-DOS версии 5.0 и является полным анахронизмом, но mscdex.exe в нем присутствовал, а значит, CD-ROM должен работать без проблем. Остался один-единственный вариант — перемычки. Как ими пользоваться, знает каждый. Но вот как их правильно установить, чтобы все заработало? Здесь с первого раза никто не разберется! При помощи все того же Интернет я узнал, что ESS 688f использует адрес 320h для подключения CD-ROM. Значит, каким-то образом следовало выставить его на звуковой карте. Для этого пришлось экспериментировать с группой перемычек JP2. Расположив их нужным образом, далее следовало в JP1 верно выставить I/O. Здесь пришлось действовать широко распространенным методом научного тыка. В итоге CD-ROM наконец-то был обнаружен старым добрым DOS при таких параметрах в config.sys:

device=C:\teac_cda.sys /t:2 /p:320

Это соответствует значениям аналога звуковой карты SB с адресом 320h. К слову сказать, сам TEAC CD-55a оказался довольно неплохим девайсом. Он великолепно читает диски, недоступные для чтения 12-скоростным IDE CD-ROM. Правда, привыкнуть к его стилю работы сложно. Если обычный IDE-привод все время опрашивается системой, то PANASONIC может увидеть компакт-диск только в момент его чтения. То есть вставили компакт-диск и сразу же обращаетесь к нему. Иначе потом вы диском воспользоваться не сумеете. Вот такую проблему мне пришлось решить на досуге.

Онлайн-распаковка архивов

Хочу со страниц журнала рассказать об одной идее — вдруг кто-нибудь заинтересуется и реализует ее.

Сейчас в Интернете довольно часто встречаются RAR-архивы. И у меня иногда не получается их распаковать из-за нехватки памяти. Дело в том, что у меня не самый современный компьютер — 486 DX2/66, 8 МБ RAM. Но предполагаю, что эта проблема касается не только владельцев старых компьютеров. Есть и современные устройства с возможностью работы в Интернете, но обладающие значительным объемом памяти (например, некоторые модели мобильных телефонов). А так как для распаковки RAR-архивов может потребоваться много памяти, то могут возникнуть проблемы с распаковкой.

Что я предлагаю? Можно создать специальный сайт для онлайн-распаковки RAR-архивов (а может, и еще каких-нибудь). Посетитель закачивает архив на этот сайт (или, может быть, просто указывает адрес этого архива в Интернете), а обратно получает уже распакованное содержимое этого архива (или, может быть, получает этот архив, конвертированным в формат ZIP, или получает не все содержимое архива, а только нужные ему файлы).

И. РОЦИН,
г. Москва

ZXNet: 500:95/462.53

E-mail: bestview@mtu-net.ru

WWW: <http://www.ivr.da.ru>



Возвращаясь к напечатанному

("РЛ. ВК" 12/2000, С.36; "РМ. ВК" 9/2001, С.37)

И.РОЩИН,

г.Москва,

ZXNet: 500:95/462.53

E-mail: bestview@mtu-net.ru

WWW: http://www.ivr.da.ru

Расширения файлов TR-DOS

(Окончание. Начало в №№2-4/2005)

Расширение	Комментарий	Происхождение расширения	Информация о формате или ссылка на нее
U			
U	Двухэкранная картинка, упакованная с распаковщиком	Возможно, от "doUble" ("двойной")	
V			
v	Векторное изображение, созданное в графическом редакторе Burial Gfx Editor	От "vector" ("векторный")	Где найти описание формата, я не знаю, но вы можете попробовать изучить исходные тексты BGE 3.05, доступные на сайте Open Source ZX (http://opensourcezx.narod.ru), и самостоятельно разобраться в формате
vi1	Дамп микросхемы CMOS. Создается программой CMOS Commander	От названия советского аналога микросхемы CMOS (512ВН1)	В файле содержатся значения всех 64 регистров CMOS, каждое значение занимает один байт
VSP	Главный файл проекта для программы Video Studio	От "Video Studio Project"	http://zxdocs.fatal.ru , раздел "Formats" (архивы "VSP.ZIP" и "VSP7.zip")
W			
W	Текстовый файл, созданный в редакторе ZX Word или Modern Word	От названия редактора	См. расширение "TXT"
wav	Звуковой файл в формате WAV. Программа для обработки — WAV Player	От "wave" ("волна")	1) http://forum.alpe.ru , раздел "Форматы файлов", тема "WAVE File Format". 2) статья "Описание форматов звуковых файлов (выборка сэмплов)" — электронный журнал Inferno #1. 3) http://www.codenet.ru/progr/formt/rawsam.php
we0, we1, ...	Архив с почтовыми пакетами, сформированный в среду	Первые два символа — от "Wednesday" ("среда"), третий символ — номер файла (начиная с 0)	Формат архива зависит от использованного архиватора. Формат почтовых пакетов (pkt-файлов) — см. расширение "pkt"
WRD	Текстовый файл	От "word"	См. расширение "TXT"
X			
X	Графический файл в формате PCX. Программы для обработки — см. расширение "psx"	Видимо, от буквы "X" в названии формата	См. расширение "psx"
x	Сжатый графический файл в формате PCX. Для просмотра используйте PCX viewer 2.04 XM (by S.F.D.)	Видимо, от буквы "X" в названии формата	
XAS, xAS	Исходный текст ассемблерной программы в формате ассемблера XAS	От названия ассемблера	См. документацию к ассемблеру XAS. В дополнение к этой документации привожу список токенов (они кодируются байтами #80...#F6): LDIR, LDDR, LDI, LDD, CPIR, CPDR, CPI, CPD, INIR, INDR, INI, IND, OUTI, OTIR, OUTD, OTDR, RETI, RETN, NEG, RLD, RRD, PUSH, POP, ADD, SUB, ADC, SBC, AND, OR, XOR, CP, INC, DEC, BIT, RES, SET, RLC, RRC, RL, RR, SLA, SRA, SLI, SRL, LD, EX, IN, OUT, IM, RST, DJNZ, JP, JR, CALL, RET, EXX, CPL, DAA, RLCA, RRCA, RLA, RRA, NOP, HALT, DI, EI, SCF, CCF, ORG, ENT, EQU, WORK, DB, DW, DM, DS, !ASSM, !CONT, LTEXT, LCODE, BC, DE, HL, IX, IY, SP, AF, (C), B, C, D, E, H, L, (HL), A, (BC), (DE), HX, LX, HY, LY, I, R, NZ, Z, NC, PO, PE, P, M, !ON, !OFF, (SP), AF', USEL, IFNZ, IFZ, MAKE
XaS, xaS	Макросы для ассемблера XAS	От названия ассемблера	
Y			
Y	1) музыкальный модуль, созданный в редакторе Fast Tracker 2) RGB-экран, упакованный с распаковщиком		См. расширение "ftc"



Z

z80	Snapshot (файл, в котором запоминается состояние эмулируемого "Спектрума" — содержимое памяти, значения регистров процессора и некоторая другая информация). Программа для обработки — RUN .Z80 0.1	От названия эмулятора "Z80", в котором используется этот формат	1) документация к эмулятору "Z80". 2) файл "DOCS\Z80.HDR" из комплекта поставки программы ZX Spectrum Navigator для PC (найти ее можно на сайте http://sn.nnov.ru)
zas	Исходный текст ассемблерной программы в формате ассемблера ZX ASM 10	От названия ассемблера	Формат похож на текстовый. Отличие в том, что байт 6, за которым следует байт x, обозначает x-#80 пробелов, а один из байтов 2, 3, 4, 5, за которым следует байт x, обозначает токен с номером x-#20 (если нумеровать токены с 0). Причем от значения байта перед x зависит способ вывода этого токена: если байт равен 2 или 4 (т.е. 0-й бит сброшен), токен выводится строчными буквами, а если 3 или 5 (т.е. 0-й бит установлен) — прописными; если байт равен 4 или 5 (т.е. 1-й бит сброшен), токен выводится с пробелом после него, а если 2 или 3 (т.е. 1-й бит установлен) — без пробела. Список токенов: LD, EX, IM, RST, RET, ADD, ADC, SUB, SBC, AND, XOR, OR, CP, PUSH, POP, INC, DEC, IN, OUT, JP, CALL, JR, DJNZ, RLC, RRC, RL, RR, SLA, SRA, SLI, SRL, BIT, RES, SET, NOP, HALT, DI, EI, RLCA, RLA, RRCA, RRA, EXX, DAA, CPL, CCF, SCF, LDI, LDIR, LDD, LDDR, CPI, CPIR, CPD, CPDR, NEG, INF, INI, INIR, IND, INDR, OUTI, OTIR, OUTD, OTDR, RETI, RETN, RLD, RRD, ORG, EQU, DB, DW, DS, DEFB, DEFW, DEFS, INSERT, INCLUDE, IF, IFDEF, IFNDEF, IFUSED, IFNUSED, ELSE, ENDIF, MAKE, B, C, D, E, H, I, (HL), A, XH, XL, YH, YL, (IX, (IY, (BC), (DE), I, R, AF, BC, DE, HL, IX, IY, SP, (SP), AF', (C), NZ, Z, NC, C, PO, PE, P, M, PHASE, UNPHASE, DC, ENT, REPEAT, ENDR, LOADTAB, MACRO, ENDM, CREATE, MAKELAB, SAVEOBJ, EXITM, IFP, EXA, RETZ, RETNZ, RETC, RETNC, RETM, RETP, RETPO, RETPE, JPZ, JPNZ, JPC, JPMC, JPM, JPP, JPPO, JPPE, CALLZ, CALLC, CALLM, CALLPE, CALLNZ, CALLNC, CALLP, CALLPO, JRZ, JRNZ, JRC, JRNC
ZIP	Архив ZXZIP	От названия архиватора	Файл "DOCS\ZXZIP.HDR" из комплекта поставки программы ZX Spectrum Navigator для PC (найти ее можно на сайте http://sn.nnov.ru). Файл "DOCS\ZXZIP.HDR" из комплекта поставки программы ZX Spectrum Navigator для PC (найти ее можно на сайте http://sn.nnov.ru). Также указанный файл доступен на сайте http://zxdocs.fatal.ru , в разделе "Formats" (архив "ZXZIP.ZIP")
zip	Архив PKZIP. Программа для обработки: pkUNZIP 1.41 Fast. Если нужно только посмотреть оглавление архива, то это можно сделать с помощью BestView. Примечание: может встретиться и архив ZXZIP с таким расширением	От названия архиватора	Документация в формате HTML — http://www.pkware.com/products/enterprise/white_papers/appnote.html . Документация в формате TXT — http://www.pkware.com/products/enterprise/white_papers/appnote.txt
zxz	Архив ZXZIP (в MS-DOS и IS-DOS эти архивы часто хранятся именно с таким расширением). Перед раскрытием архива на ZX Spectrum надо изменить его расширение на "ZIP", а если архива длиннее 255 секторов, т.е. есть файлы-продолжения, то их надо переименовать в "*****.ZIP"	От названия архиватора	См. расширение "ZIP"



TimeZero — online-игра нового поколения

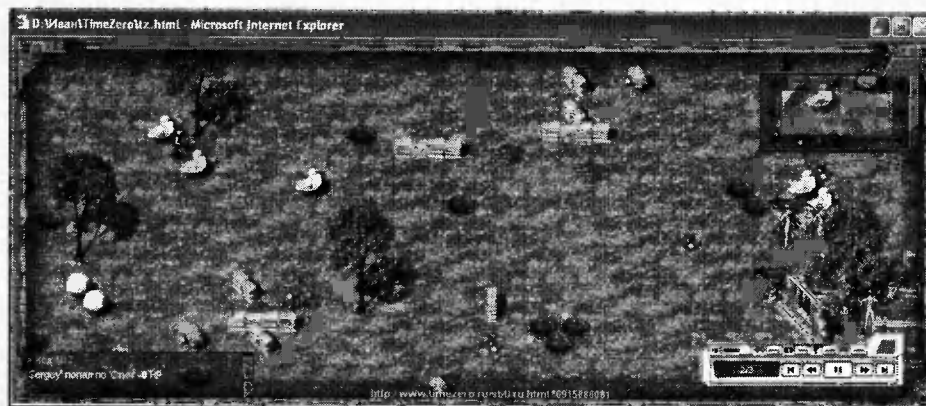
И.ШИРКО,
г.Минск, БГУИР, ФИТУ.
E-mail: FDC@tut.by

Сейчас Интернет стал обычной вещью, вполне доступной рядовому обывателю. Поэтому появляются online-игры, часто требующие постоянного соединения с Интернетом. Одной из таких игр является TimeZero — Точка Отсчета. Эта игра создана в Macromedia Flash 7 и, должен отметить, сделана очень профессионально.

можно победить игрока более высокого уровня. На первых двух уровнях выходить из города нельзя, поэтому драться можно только в специальном здании — Арене. Драться там можно не только один на один, но и "стенка на стенку". Первые три уровня игрок вполне может жить и развиваться за счет государства, используя бесплат-

роду шахтам, в которых обитают разнообразные монстры (в основном крысы-мутанты). Но, кроме монстров, там находятся и ресурсы, которые потом можно продать за деньги. А за деньги можно купить разнообразные вещи. Все бы хорошо, да вот некоторые игроки не хотят сами добывать ресурсы и тем зарабатывать деньги, а занимаются разбоем (в игре их называют мародерами). Так что нагруженный добытыми ресурсами счастливый игрок вполне может оказаться жертвой нападения и лишиться своей добычи. Но есть в игре шахты, которые находятся под защитой полиции (да-да, там еще есть и полиция). По этим шахтам можно бродить, не опасаясь мародеров. А если на вас все-таки напали, то вызывайте полицейского и тяните время, дожидаясь его прибытия. Раз уж затронули полицейских, то давайте рассмотрим и остальные профессии.

Старатель — самая распространенная и доступная профессия. Чтобы ее получить, нужно в университете купить квест за 30 монет, который необходимо выполнить — найти в шахтах редкий минерал. Когда минерал найден, его нужно принести в университет, заплатить 100 монет за сертификацию — и вы уже старатель. Эта профессия дает следующие преимуще-



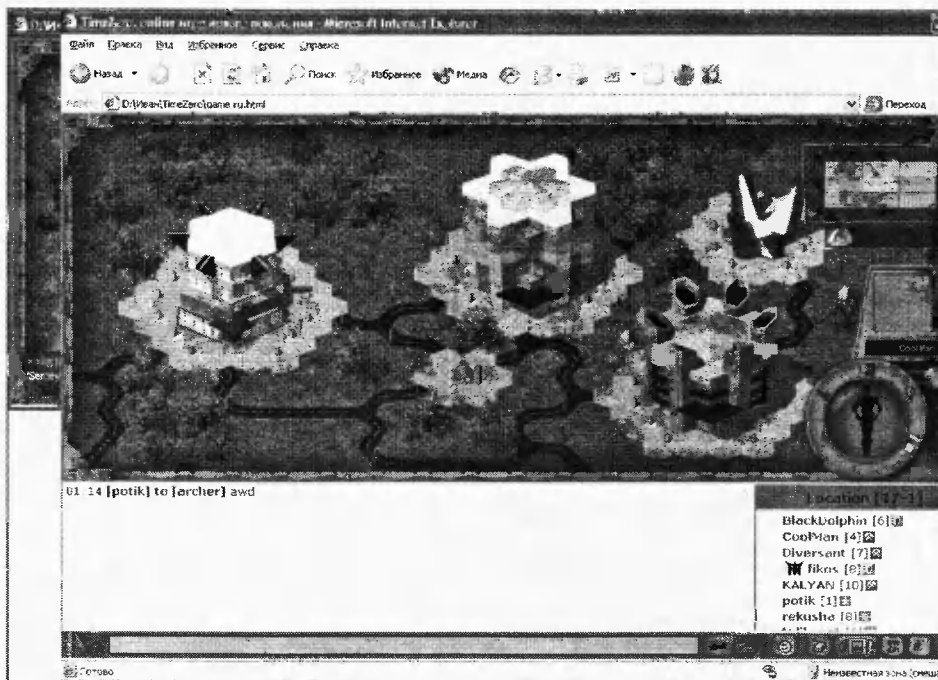
Чтобы начать игру, нужно зарегистрироваться на сайте TimeZero.ru. В принципе, после этого можно уже сразу играть, но я рекомендую скачать клиент игры, который позволит уменьшить сетевой трафик до нескольких десятков килобайт в час.

Вот теперь можно приступать к игре. Введите свой логин и пароль и попадете в город New Moscow. Появитесь вы там в стиле Терминатора — в чем мать родила и без оружия. Но это нетрудно исправить, т.к. среди видимых зданий есть заброшенный арсенал, в котором можно найти все необходимое для начала своей виртуальной жизни в TimeZero. Вы обитаете в большом игровом мире, поделенном на клетки — локации, по которым персонаж может перемещаться. Игрок имеет ряд параметров: Сила, Ловкость, Интуиция, Выносливость, Интеллект, Меткость. Эти параметры увеличиваются при переходе персонажа на следующий уровень, переход на который осуществляется при достижении необходимого опыта. Опыт набирается в сражениях с другими персонажами либо с монстрами. Бои проходят в пошаговом режиме — игроки ходят по очереди. В бою очень большое значение имеет тактика. При определенном стечении обстоятельств вполне

можно победить игрока более высокого уровня. На первых двух уровнях выходить из города нельзя, поэтому драться можно только в специальном здании — Арене. Драться там можно не только один на один, но и "стенка на стенку". Первые три уровня игрок вполне может жить и развиваться за счет государства, используя бесплат-

ные оружие и одежду и "на халяву" лечась в местном госпитале. А вот потом (при переходе на более высокие уровни) лечение станет платным, да и без новой амуниции обходиться будет трудновато.

Когда первые два уровня преодолены, для персонажа начинается новая жизнь — он может выходить из города. На невысоких уровнях игроки обычно ходят только по близким к го-





ства: сильно повышается навык владения холодным и легким оружием и увеличивается вместительность рюкзака (т.е. можно таскать больше ресурсов). Но немного уменьшается навык владения тяжелым оружием. Еще один важный бонус — если старателя «убили», то у него есть шанс сохра-

без торговцев? Только им разрешено расхваливать свой товар в общем чате в пределах города. Товар они могут выставлять на своей витрине, также у торговцев повышается вместительность рюкзака.

Инженер — довольно редкая профессия. Это ученые, которые могут

пить офис клана.

TimeZero поражает своей реальностью — мало того, что вы играете с людьми, но и игровой мир максимально приближен к реальности. Здесь можно учиться в университете, получать профессию, работать на заводе, покупать и продавать. Есть даже полицейские, которые могут отправить игрока на каторгу. Можно подарить девушке цветы. Здесь выбирают мэра и даже уже собираются устроить нынешнему мэру импичмент... Единственное, здесь нельзя умереть, но попасть в больницу с большим количеством серьезных травм любому игроку придется не раз и не два...

Еще хочется рассказать, как в TimeZero Новый год отмечали. Еще в старом 2004 г. во все магазины города завезли фейерверки, так что время от вре-

мени в разных локациях гремели залпы, сопровождаемые яркими вспышками. Ну а непосредственно на новогодние праздники в свободный доступ выложили необычные мешочки, которые позволяли мгновенно восстанавливать здоровье как во время боя, так и вне него. Теперь представьте себе, какие там дуэли происходили, когда все постоянно подлечивались, и победа зависела лишь от количества чудесных мешочков у игрока...

P.S. Предупреждаю всех любителей Орега — чат в ней не работает, так что лучше выбрать другой браузер.



нить при себе часть ресурсов.

Сталкер — эта профессия повышает способности к путешествию по поверхности, подобно тому как у Старателя — к хождению по шахтам.

Наемник — игроки этой профессии очень сильны в бою, только у них существенно увеличен навык владения тяжелым оружием. Чтобы стать наемником, нужно принести голову белого стича (это вид монстров, похожих на человека-зомби) — знак доблести и отваги.

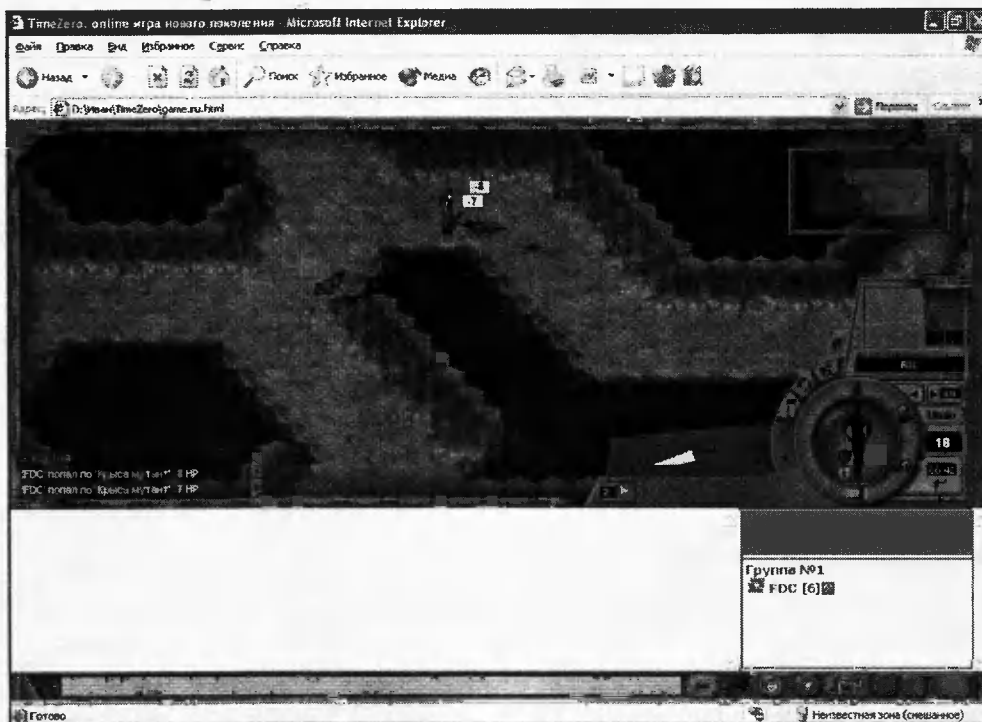
Корсар — раз есть полицейские, должны быть и бандиты. Эта профессия для людей, которые не хотят жить по законам общества. Для того чтобы стать корсаром, нужно убить 30 игроков и нескольких полицейских более высокого уровня. Ну и, конечно, заплатить определенную сумму денег. Преимущества профессии: сильно повышается навык владения средним оружием, появляется возможность свободного прохода по шахтам (т.е. невозможно попасть в засаду монстров, но нельзя искать ресурсы), увеличенная скорость передвижения вне города, при убийстве другого игрока корсару достается 9% денег жертвы. Но в городе корсар становится довольно беззащитным — все его навыки падают до нуля, и замедляется передвижение.

Торговец — как же обойтись

производить новое оборудование и оружие.

Дилер — это тоже своего рода торговец, но только покупать у него можно монеты за реальные деньги. Курс на момент написания статьи — 100 медных монет за один WMZ (доллар в WebMoney).

В игре существуют кланы, каждый из которых проводит свою политику. Время от времени случаются клановые войны, или наоборот, заключаются союзы. Любой игрок может организовать собственный клан, но для этого нужно сделать клановый сайт и ку-





КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

Для публикации бесплатных объявлений некоммерческого характера о покупке и продаже компьютерных комплектующих и программ, их текст можно присылать в письме по адресам:



119454, г. Москва, а/я 37 или
220095, г. Минск-95, а/я 199,
передавать по тел. в Минске (017) 249-41-47,
либо через E-mail:

rl@radiopage.by
rm@radio-mir.com
WWW: http://radio-mir.com

Начинающий радиолюбитель для радиолюбительских расчетов за символическую цену купит компьютер.
Тел. в Минске 253-33-44. Павел.

Продам книгу Гомера Л. Давидсона "Поиск неисправностей и ремонт электронной аппаратуры без схем", М., 2004. 142400, г. Ногинск, Московской обл., а/я 75, В.С. Яланский.

Продаю дешево исправную, б/у, ISA-плату Winradio 3100i-DSP (без антенны).
E-mail: winradio3100@mail.ru

Меняю срочно диски с "Office XP" (лицензионный), ASPLinux 7.3, "Blue eye DOS", "C osmo", "Open office", "UNA", "PersonPro" на б/у ноутбук, системный блок или "Dreamcast".
E-mail: 486dx33@mail.ru



Радиомир. Лучшие конструкции. Выпуск 1.

ООО "Радиомир Пресс", 2005 — 112 с.

Книга представляет собой сборник статей, опубликованных в разные годы в журнале "Радиомир" и заново отредактированных для данного издания.

В выпуске приведены схемы и описания устройств, используемые радиолюбителями в различных областях.

По поводу приобретения книг можно обращаться в редакцию по E-mail: rm@radio-mir.com, а также во все организации, имеющие журнал "Радиомир" в розничной продаже. Информация о них приведена рядом.

Выберите себе вариант подписки!

Подписка через почтовые отделения

Радиомир

- для жителей России и стран СНГ (кроме Беларуси): 48996 — подписка по каталогу Агентства "Роспечать" (72370 — годовая), 25785 — подписка по Объединенному каталогу "Пресса России" с адресной рассылкой (электронный адрес подписки в INTERNET — www.presscafe.ru);

- для жителей Беларуси: 48996, (489962 — для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

Радиомир. КВ и УКВ

- для жителей России и стран СНГ (кроме Беларуси): 48924 — подписка по каталогу Агентства "Роспечать", (71545 — годовая);

- для жителей Беларуси: 48924, (489242 — для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

Радиомир. Ваш компьютер

- для жителей России и стран СНГ (кроме Беларуси): 48925 — подписка по каталогу Агентства "Роспечать" (45995 — годовая);

- для жителей Беларуси: 48925, (489252 — для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

Кроме того, предприятия регионов России, а также ближнего и дальнего зарубежья могут оформить подписку на журналы "Радиомир", "Радиомир. КВ и УКВ", "Радиомир. Ваш компьютер" через ООО "Корпоративная Почта" по телефонам: (095) 953-92-62, 953-92-02, 953-93-20.

Внимание! Адресная подписка через редакцию. Подписаться на имеющиеся в наличии отдельные номера журналов, а также на любой период, начиная со следующего после оплаты месяца, можно через редакцию. Для этого нужно оплатить необходимую сумму через Сбербанк России или оформить почтовый перевод на наш расчетный счет. Текущие цены приведены в таблице. В цену включена доставка журналов в отдельном конверте по адресу подписчика, указанному в графе "Назначение платежа" при оплате через Сбербанк России или в графе "Для письма" при оплате почтовым переводом. Наложным платежом журналы не высылаются.

Наши платежные реквизиты для жителей России и стран СНГ — получатель: ООО "Радиомир Пресс", ИНН 7729404741, КПП 772901001, р/с 40702810900010000084 в ООО КБ "Агропромкредит", ф-л Центральный, г. Москва, корр. счет 30101810500000000109, БИК 044525109

Адрес банка: 125315, г. Москва, Ленинградский пр-т, дом 76, корп. 4;
При оплате через Сбербанк в графе "Назначение платежа" необходимо написать свой почтовый индекс, полный адрес, фамилию, имя и отчество полностью и точно перечислить, какие конкретно номера какого из журналов Вы заказываете. При оплате почтовым переводом все сведения о заказе необходимо указать в графе "Для письма".

Для ускорения процесса получения журналов заказ можно продублировать по E-mail: rm-sales@radio-mir.com. Вся информация — там же или по тел. в г. Минске (017) 221-01-10, (017) 505-13-65.

Год	Можно заказать следующие номера журналов			Расценки на 1 экз. (с учетом пересылки)		
	Радиолюбитель	Радиолюбитель. Ваш компьютер	Радиолюбитель. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2000	3 — 6, 8 — 12	3 — 6, 8, 10 — 12	6, 7, 9, 11, 12	20	800	6
2001	4 — 5	2, 5, 6	2, 3, 5, 6	25	900	6
	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2001	7, 9, 10, 12	7, 9 — 12	7, 8, 10, 11	25	1000	7
2002	2 — 12	1 — 12	1 — 3, 5 — 12	30	1500	7
2003	1 — 12	1 — 12	1, 5 — 8, 10 — 12	35	1800	8
2004	1 — 6, 8 — 12	1 — 6, 8 — 12	1 — 10, 12	40	2100	9
2005	1 — 12	1 — 12	1 — 12	45	3000	11

Приобретение отдельных номеров журналов

В РОССИИ:

В ЗАО "Интерпресс — Экспресс": тел. в Москве (095) 208-85-50, 208-67-55, e-mail: inter@aif.ru, http://www.interpress-express.com.

В магазинах радиодеталей "ЧИП и ДИП" (единая справочная — тел. (095) 780-95-09):

- г. Москва, ул. Гиляровского, д. 39 (ст. метро "Проспект Мира" — радиальная);
- г. Москва, ул. Ивана Франко, д. 40, к. 1, стр. 2 (платф. Рабочий поселок, 15 мин. от Белорусского вокзала);
- г. Москва, ул. Беговая, д. 2;
- г. Москва, ул. Земляной вал, д. 34.

На радиорынках в Москве: Митинском (места R4, S8, K52, E50, G56), Царицынском (место 121).

В ООО "Альянс-Книга КТК": 115533, г. Москва, ул. Наргитинская, д. 6, эт. 1, оф. VII (ст. метро "Наргитинская"), тел. (095) 258-91-94, 258-91-95, 258-91-96, 111-05-98, 111-63-28. E-mail: abook@mail.ru, abook@inbox.ru

На радиорынках: "Царицыно" (место 13/А), "Митино" (ряд 1, контейнер 17 и место Т-8).

На УКРАИНЕ:

В Киеве в фирме "Торм", тел. (044) 227-72-73.

В КАЗАХСТАНЕ:

В фирме ТОО СП "Аргументы и факты. Казахстан": тел. в Алма-Ате (3272) 50-22-60, 50-21-64.

В БЕЛАРУСИ:

В Минске в магазинах "Книга XXI век", пр. Ф. Скорины, д. 92, тел. (017) 264-27-97 (ст. метро "Московская") и "Глобус", ул. Володарского, д. 16, тел. (017) 227-30-67 (ст. метро "Площадь Независимости").



Коды цветов для HTML

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

Подписка-2005!

радиомир

E-mail: rm@radio-mir.com

WWW: <http://radio-mir.com>

— для жителей России и стран СНГ (кроме Беларуси): 48996 — подписка по каталогу Агентства "Роспечать" (72370 — годовая), 25785 — подписка по Объединенному каталогу "Пресса России" с адресной рассылкой (электронный адрес подписки в INTERNET — www.presscafe.ru);

— для жителей Беларуси: 48996, 489962 (для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

- В мире оживших звуков
- Рядом с телефоном
- Танцуем от питания
- Автоматика всегда поможет
- Сам себе лекарь
- Вокруг автомобиля
- Первым делом технология
- Видеотехника
- Измерения
- Не только новичку
- Личная радиосвязь
- Справочный материал
- Радиолюбительская ярмарка

радиомир

КВ и УКВ

— для жителей России и стран СНГ (кроме Беларуси): 48924 — подписка по каталогу Агентства "Роспечать", (71545 — годовая);

— для жителей Беларуси: 48924, 489242 (для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

- | | |
|-----------------------|-----------------------------|
| ➤ Новости | ➤ Компьютер на радиостанции |
| ➤ QUA | ➤ УКВ |
| ➤ Дипломы | ➤ Усилители |
| ➤ DX-info | ➤ Трансиверы |
| ➤ Соревнования | ➤ Антенны |
| ➤ Эфирная робинзонада | ➤ Доска объявлений |

радиомир

Ваш компьютер

— для жителей России и стран СНГ (кроме Беларуси): 48925 — подписка по каталогу Агентства "Роспечать" (45995 — годовая);

— для жителей Беларуси: 48925, 489252 (для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

- | | |
|--------------------------|------------------------|
| ➤ Мультимедиа | ➤ Диалог программистов |
| ➤ Не только новичку | ➤ Рецепты |
| ➤ Коммуникации | ➤ Мир 8 бит |
| ➤ Уроки программирования | ➤ Игротека |
| | ➤ Доска объявлений |

Кроме того, предприятия регионов России, а также ближнего и дальнего зарубежья могут оформить подписку на журналы "Радиомир", "Радиомир. КВ и УКВ", "Радиомир. Ваш компьютер" через ООО "Корпоративная Почта" по телефонам: (095) 953-92-62, 953-92-02, 953-93-20.

В Украине журналы "Радиомир", "Радиомир. КВ и УКВ", "Радиомир. Ваш компьютер" можно приобрести в фирме "Торм", тел. (044) 227-72-73.