

радиомир

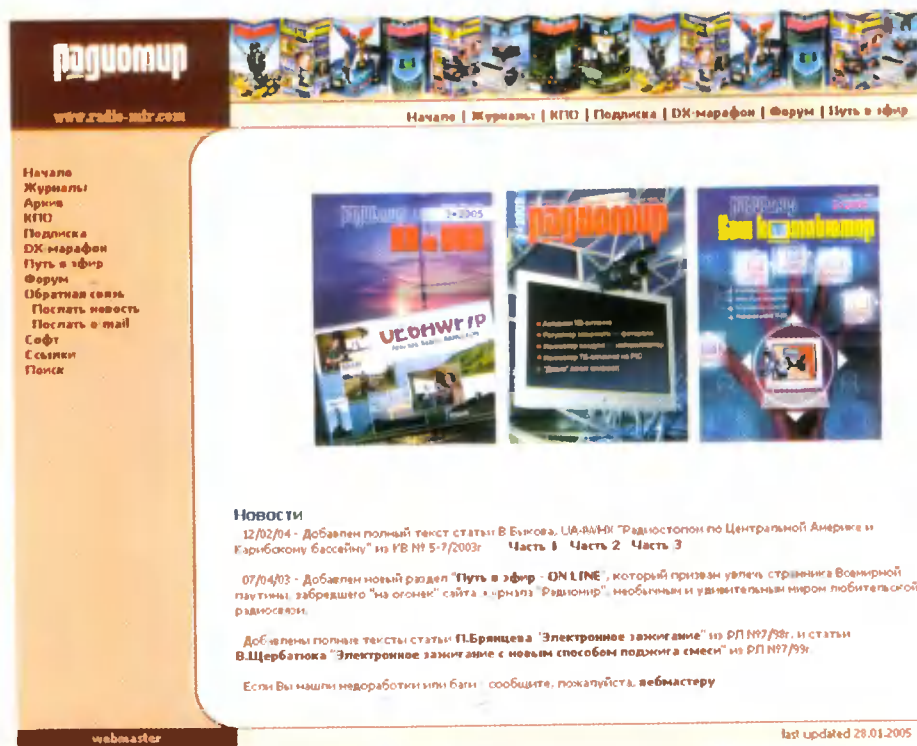
6•2005

Ваш компьютер

- => Скинь "маму" с платформы Socket A
- => Организация памяти и структура данных
- => "Рабовладение" в iS-DOS на Sinclair
- => Как преобразить свой компьютер
- => Адаптер для ПК



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 СТРУЖАНКОВА

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 30101810500000000109
 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. Коштыянца, 6 — 233, тел. (095) 105-99-89

Подписано к печати 26.04.2005 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 № 1177.

радиомир Ваш компьютер

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

№6/июнь/2005

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

МУЛЬТИМЕДИА

В.КУЦ. Цифровое фото. Современное состояние и перспективы 2

НЕ ТОЛЬКО НОВИЧКУ

А.ГРИНЧУК. Windows XP для пользователя
 Обслуживание дисков 4

Е.ЗАЙЦЕВА. Эффективная работа с Microsoft Word XP
 Вставка и управление таблицами в документе 7

Е.ШАКЕЛЬ. Эффективная работа с табличным процессором
 Microsoft Excel XP
 Продолжаем оформлять 12

В.КУЦ. ABBYY PDF Transformer 1.0 15

А.ГОРЯЧКИН. Скинь "маму" с платформы Socket A 17

С.РЮМИК. Интернет-калейдоскоп, freeware #19 22

ДАЙДЖЕСТ 24

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

А.КОРБИТ, Т.КРИВОНОСОВА. Организация памяти
 и структура данных 28

ДИАЛОГ ПРОГРАММИСТОВ

CyberManiac /HI-TECH. Теоретические основы крэкинга 31

В.КАРЦЕВ. Исключения C++ и структурные исключения C
 в C++ Builder 32

И.РОЩИН. Программа для форматирования текстов 34

РЕЦЕПТЫ

А.КАШКАРОВ. Адаптер для ПК 37

А.АНИСКОВИЧ. Как преобразить свой компьютер 37

МИР 8 БИТ

Е.ИЛАСОВ. "Рабовладение" в iS-DOS на Sinclair 39

А.ПЕХИМЕНКО. Обзор электронной прессы о Spectrum 42

ИГРОТЕКА

Р.КАРПАЧ. Из истории компьютерных игр,
 или почему же IBM, а не ЕС? 44

А.ВЕНДИЛОВСКИЙ. Lord of the Rings. Battle of Middle-Earth 46

ДОСКА ОБЪЯВЛЕНИЙ

Куплю, продам, обменяю 48

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



Цифровое фото.

Современное состояние и перспективы

В.КУЦ,

E-mail: victor_kootz@mail.ru

Цифровая фотография окончательно и бесповоротно завоевала сердца фотолюбителей всего мира. Изобилие самых разнообразных моделей цифровых фотоаппаратов (ЦФА), представленных на рынке, просто поражает. Сегодня выпускаются и крошечные, но вполне функциональные аппараты размером с кредитную карточку, и "монстродальнейшие" зеркальные камеры со сменной оптикой и безумным количеством ручных настроек, способные удовлетворить запросы самых взыскательных профессионалов, и примитивнейшие "мыльницы", годящиеся разве что на роль детской игрушки. Более того, использование ЦФА уже не сводится только к получению исключительно статических изображений, многие из них способны снимать целые видеоролики (включая звуковое сопровождение), которые затем можно просмотреть на собственном ЖК-дисплее фотоаппарата или на бытовом телевизоре. Уже сейчас некоторые камеры оснащаются беспроводными интерфейсами различных типов (IrDA, Bluetooth) для оперативной пересылки отснятых фотоматериалов на принтеры, сотовые телефоны или беспроводные сети. Среди недорогих аппаратов большую популярность получили комбинированные устройства, включающие в себя, помимо собственно ЦФА, еще и диктофон, MP3-проигрыватель и даже простейшую видеокамеру. Да, неискушенному фотолюбителю достаточно трудно разобраться во всем многообразии моделей современных ЦФК, диапазон цен на которые сегодня простирается от 50 — 60 USD за модели начального уровня и вплоть до 1500 — 2000 USD за полупрофессиональные "зеркалки" со сменными объективами. А ведь между двумя этими полюсами — множество промежуточных вариантов. Для того чтобы пользователь сам смог сориентироваться в возможностях, предоставляемых той или иной приглянувшейся ему цифровой камерой, вкратце ознакомимся с некоторыми их особенностями, которые необходимо учитывать при выборе конкретного устройства.

Одним из важнейших параметров, характеризующих качество съемки ЦФА, является разрешение, определяемое количеством пикселей, содержащихся в оптическом сенсоре (Image Sensor) фотоаппарата (чем их больше, тем более детализированной может быть фотография). В качестве оптического сенсора в камерах любительского уровня обычно используются приборы с зарядовой связью ПЗС (CCD) или чипы КМОП (CMOS), состоящие из множества светочувствительных элементов (фотодиодов), сгруппированных в матрицу. Одним из наиболее существенных отличий между КМОП- и ПЗС-сенсорами является то, что в КМОП-сенсоре АЦП (аналого-цифровой преобразователь) является составной частью чипа, тогда как в ПЗС-сенсоре для оцифровки аналогового сигнала с матрицы приходится использовать дополнительную микросхему АЦП. По этой причине классический КМОП-сенсор имеет несколько более высокий уровень шумов, хотя и дешевле в производстве, чем ПЗС. Как неоспоримое достоинство КМОП-матриц следует отметить то, что они более экономичны и компактны, чем их конкуренты, а также менее требовательны к сопутствующей электронике, поэтому они являются идеальным решением для использования в КПК и камерофонах (мобильных телефонах со встроенной камерой).

Однако в последнее время достигнут впечатляющий прогресс в развитии технологии КМОП-матриц. Например, созданная компанией Kodak КМОП-матрица, основанная на фирменной технологии High Performance Low Noise, содержит 14 млн. пикселей и имеет улучшенные схемы аналоговой и цифровой обработки сигнала. Ее размеры обеспечивают полное покрытие стандартной площади кадра 35-миллиметровой пленки, а диапазон чувствительностей ISO простирается от 6 до 1600.

Что же касается конкретных значений разрешения, обеспечиваемых современными ЦФА, то мегапиксельные "мыльницы", достаточно популярные в 2003 г., в 2004 г. посте-

пенно отошли в область преданий. В нише наиболее дешевых решений их сменили 2- и даже 3-мегапиксельные камеры. Гораздо больше возможностей для пролучения более или менее качественных снимков предоставляют фотоаппараты с разрешениями от 3 до 5 мегапикселей, относящиеся к любительскому классу. А вот с 5 мегапикселей начинается "царство" серьезных полупрофессиональных моделей. Разумеется, такое деление ЦФА по классам в зависимости от разрешения очень условно — встречаются и достаточно солидные 3-мегапиксельные камеры (например, Canon PowerShot S1 IS) которые даже профессионалу не будет зазорно взять в руки, и 5-мегапиксельные "игрушки" (по большей части выпускающиеся под гордым именем "ноунейм"). Ведь давно уже ни для кого не секрет, что наращивание "мегапикселей", сопровождаемое в последнее время мощной рекламной шумихой, абсолютно не гарантирует адекватное увеличение качества фотоснимков. Скорее, даже наоборот — рост количества пикселей на матрице (при сохранении, а то и уменьшении ее физических размеров) чаще всего приводит к повышению уровня шумов (со всеми вытекающими из этого негативными последствиями). Так что физические размеры матрицы (независимо от ее типа) гораздо важнее, чем количество содержащихся в ней пикселей. Тем не менее, многие производители с гордостью размещают информацию о миллионах пикселей матрицы на самых видных местах своих фотоаппаратов, тогда как указать ее физические размеры стесняются даже в спецификациях. Интересно, почему?

На качество полученных фотоснимков влияют не только параметры сенсора изображения, но и качество оптики, хотя наиболее дешевые цифровые "мыльницы" умудряются обходиться вообще без объектива, используя для фокусировки изображения на матрице обычную пластмассовую линзочку. Не очень сильно от такого, с позволения сказать, "объектива" отличаются и ши-



рокоугольные объективы с фиксированным фокусным расстоянием. Диафрагма в них обычно фиксирована, а выдержка и другие важнейшие параметры съемки регулируются автоматикой, которая, как известно, ошибается ничуть не реже, чем человек (а в фотографии — гораздо чаще). И только гораздо более сложные объективы с переменным фокусным расстоянием (вариообъективы) способны обеспечить достаточную для получения высококачественных снимков глубину резкости изображения объектов, находящихся на различном расстоянии от фотографа. Оптическое переменное фокусное расстояние таких объективов не следует путать с цифровым "zoom"-ом, осуществляемым процессором фотокамеры и влияющим на качество фотоснимка далеко не в лучшую сторону.

В дорогих ЦФА вариообъектив обычно имеет достаточно высокую степень кратности (порядка 10X) с большим диаметром входной линзы, а отличительная особенность настоящих профессиональных камер — использование съемных объективов, унифицированных с объективами для пленочных фотоаппаратов. В аппаратах высокого класса обычно предусматривается возможность изменять фокусировку, а также параметры, влияющие на экспозицию (выдержку, диафрагму и пр.), ручную, хотя имеется и полный комплект автоматических режимов съемки.

Отснятые кадры сохраняются в виде графических файлов в форматах JPEG, TIFF или RAW. Обычно к ним прикрепляются и метаданные фотографии (значение диафрагмы, скорость затвора, баланс белого, коррекция экспозиции, включение вспышки, время/дата и т.д.). В формате JPEG, используемом в самых дешевых камерах, изображение сжимается в соответствии с заранее выбранным коэффициентом сжатия (обычно задается высокое, среднее или низкое сжатие). Формат TIFF позволяет сохранять изображение как в сжатом, так и в несжатом виде. До недавнего времени он был прерогативой дорогих ЦФА, претендующих на полупрофессиональный статус. Однако сейчас в этом качестве TIFF вытесняется форматом RAW, являющимся, по большому счету, просто оцифрованным сигналом яркости с

матрицы и прошедшим лишь предварительную обработку в процессоре камеры. Использование формата RAW способно гарантированно обеспечить сохранение изображения без потери качества, тогда как степень потери мелких деталей изображения при сохранении в TIFF и JPEG (в режиме максимального качества) на подавляющем большинстве любительских снимков примерно одинакова.

Графические файлы записываются либо во встроенную память (как правило, в недорогих моделях), либо на съемную флэш-карту (в подавляющем большинстве камер). В качестве внешней памяти в ЦФА чаще всего используются флэш-карты типов CompactFlash или SecureDigital. Карты памяти CompactFlash к сегодняшнему дню достигли наибольшей емкости 8 Гб и предельной скорости записи до 12 Мб/с, однако из-за слишком больших размеров они в настоящее время постепенно уступают "поле битвы" своим, пусть и не таким емким и быстрым, но гораздо более компактным конкурентам. Даже компании Canon и Nikon, являющиеся самыми старыми и последовательными поклонниками стандарта CompactFlash, в своих последних моделях компактных ЦФА перешли на миниатюрные карты памяти SecureDigital.

Другие же вендоры предпочитают свои собственные форматы флэш-карт. Так, все фотокамеры производства Sony используют различные варианты MemoryStick, а Olympus и FujiFilm — xD-Picture. Самыми дешевыми (в расчете на 1 Мб памяти) являются карточки CompactFlash, хотя в последнее время их догоняют SecureDigital. Карты Memory Stick и xD-Picture стоят примерно вдвое дороже.

Рынок

Давно поняв перспективность и прибыльность массового рынка цифрового фото, большинство производителей ЦФА включились в гонку по завоеванию "теплого места под солнцем". Как следствие этой гонки, на рынок было выброшено огромное количество "очень-многомегапиксельных" моделей по весьма заманчивой цене. Уже сейчас на рынке можно встретить как 3-Мп цифровую камеру за 500 USD, так и 5-Мп аппарат за 150 USD, что создает полную неразбериху в умах но-

вичков, усугубляемую шумной рекламой "гонки мегапикселей".

С другой стороны, не менее отчетливо проявляется другая, не менее интригующая тенденция на рынке цифрового фото, выразившаяся в преодолении к началу текущего года полупрофессиональными зеркальными камерами со сменной оптикой психологически важного для потребителя 1000-долларового рубежа. Такое снижение цен на технику, ранее являющуюся уделом исключительно профессиональной фотозлиты, значительно расширяет круг ее потребителей, в число которых теперь входят и "продвинутые" любители-энтузиасты.

В настоящее время большая часть продаж ЦФА приходится на трех- и четырехмегапиксельные камеры любительского класса. Это связано с тем, что, с одной стороны, цены на них упали до приемлемого для массового покупателя уровня, а с другой — их возможностей уже вполне достаточно для бытовой фотосъемки. Поэтому такие устройства успешно могут заменить традиционные компактные пленочные фотоаппараты. Эта же категория устройств останется наиболее массовой и в ближайшее время, хотя диапазон предлагаемых моделей здесь расширится. Базовыми характеристиками фотоаппаратов данного класса в ближайшее время станут: матрица размером от 4 до 6 мегапикселей, поддержка режима записи видеоклипов (вместе со звуком), максимально упрощенная для пользователя автоматическая настройка всех режимов съемки и работы камеры, а также наличие объектива с не менее чем 3-кратным оптическим увеличением.

Впрочем, производителям цифровых камер, особенно в секторе недорогих бюджетных решений, сегодня некогда особо почитать на лаврах — стремительно растет популярность КПК, оснащенных встроенными фотокамерами, и особенно камерофонов, пик роста продаж которых нас ожидает в ближайшие годы. Так, согласно опросу покупателей, проведенному аналитической компанией Big Research еще в январе 2004 г., касательно электронных продуктов, которые они планировали бы купить и обладание которыми существенно повысит имидж владельца, камерофоны (80% голосов опрошенных) за-

няли второе место, уступив лишь плазменным панелям (84%).

Снижение цен — общая тенденция для мирового рынка цифровых фотоаппаратов. Сегодня стоимость большинства трехмегапиксельных устройств гораздо ниже, чем цены на двухмегапиксельные модели годичной давности. Наиболее “предлагаемыми” остаются модели массового уровня по цене от 200 до 600 USD, и, ввиду постепенного удешевления моделей класса “продвинутых любителей”, скорее всего, мы будем еще долго наблюдать именно такое распределение по ценовым категориям. Быстрее всего снижается стоимость на старшие модели ЦФА, однако обольщаться перспективами глобального удешевления некогда достаточно дорогих моделей все-таки не стоит — производители очень зорко следят за рынком и “держат руку на пульсе”, позволяя ценам уменьшаться лишь до определенного, выгодного им предела. Общая тенденция снижения цен связана с приходом на рынок цифрового фото ряда новых, весьма агрессивных (как мы уже убедились) производителей, а также ростом амбиций ряда “старожилов” рынка, таких как, например, корейский компьютерный гигант Samsung.

Итоги

Заканчивая текущий обзор, хотелось бы отметить, что в целом, несмотря на обилие новинок и различных течений в развитии цифровых камер, в большей степени сегодняшний рынок можно назвать “успокоившимся” и, в некоторой степени, определившимся с основными тенденциями последующего развития. С другой стороны, в действиях некоторых компаний последнее время четко прослеживается тенденция усиления их маркетинговой активности. В частности, только этим можно объяснить повальное увлечение подавляющего большинства компаний, присутствующих на рынке, выпуском цифровых “мыльниц” с маленькими и достаточно шумными, но зато 4- и 5-мегапиксельными матрицами — явная подстройка под массового, но не всегда достаточно грамотного покупателя. Однако, как бы там ни было, современный рынок цифровых камер можно с уверенностью назвать по-настоящему массовым и весьма динамично развивающимся.

Windows XP для пользователя

Обслуживание дисков

В критических ситуациях самое главное — не терять голову. Мария-Антуанетта, за час до казни на гильотине

Любая информация хранится на вполне конкретном носителе — оптическом компакт-диске, флоппи-диске (дискете), на жестком диске (винчестере). И хотя в Windows XP сделано все для того, чтобы основные операции с файлами выглядели одинаково, иногда, особенно при сбоях системы, приходится разбираться с происходящим на заднем плане. Пользователю в критических ситуациях предлагается целый набор вспомогательных программ. Правильное их применение предполагает знание основных особенностей механизма записи и хранения информации.

До недавнего времени основным средством переноса небольших порций информации служила дискета размером в 3,5 дюйма (89 мм). В Проводнике вы можете увидеть надпись **Диск 3,5 (A)**. Дискета представляет собой пластиковый диск с магнитным покрытием, размещенный в пластиковом защитном корпусе. Внутри системного блока вашего компьютера работает, и пока не собирается сдавать свои позиции, жесткий магнитный диск — винчестер. По сути, это тот же набор дисков на общей оси, только основа у них алюминиевая, магнитный слой более качественный, да и размещен винчестер в наглухо запаянном металлическом кожухе. Запись и считывание осуществляется с помощью магнитных головок, по принципу действия похожих на те, что используются в аудио- и видеомagnetофонах. Но у дисков и кассет есть одно принципиальное различие. На дисках требуется большая плотность записи информации. Для этого диски разбиваются на строго определенное количество секторов и дорожек, самой мелкой структурной единицей является кластер. Счи-

тывающие головки получают информацию о местонахождении следующей порции файла и попадают в точно назначенное место. Служебная информация о формате диска и о начальном кластере каждого файла хранится в так называемом оглавлении диска.

Сам процесс разбиения диска на кластеры получил название “форматирование”. Как правило, в продажу поступают уже отформатированные дискеты и винчестеры. При работе компьютера возможны ошибки двух типов. Могут ошибаться программы и (особенно при неправильном выключении компьютера) “не по правилам” записываться данные на диск. Также со временем покрытие дисков может потерять свои магнитные свойства. Для анализа ситуации стоит начать с **проверки диска**. Для этого открываем Проводник и правой клавишей мыши щелкаем по значку выбранного диска (для примера у нас это дискета — **Диск 3,5 (A)**). В появившемся меню выбираем команду **Свойства**. В открывшемся диалоговом окне переключаемся щелчком мыши на вкладку **Сервис** (рис. 1).

Рис. 1

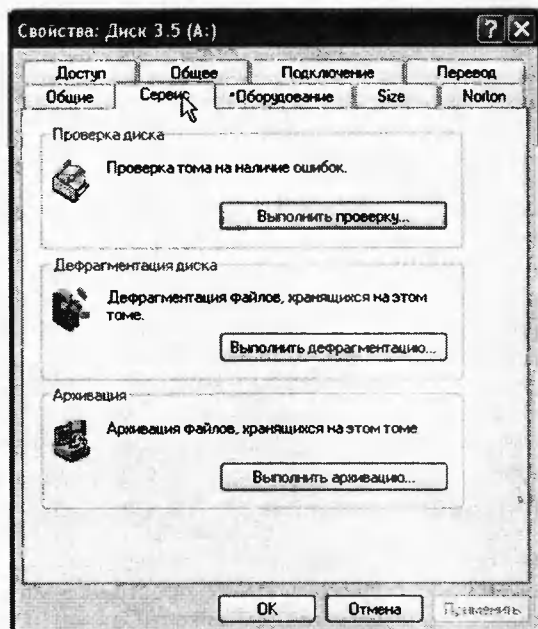
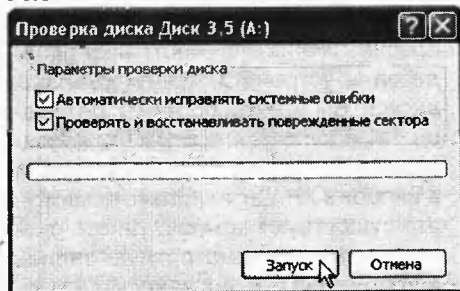


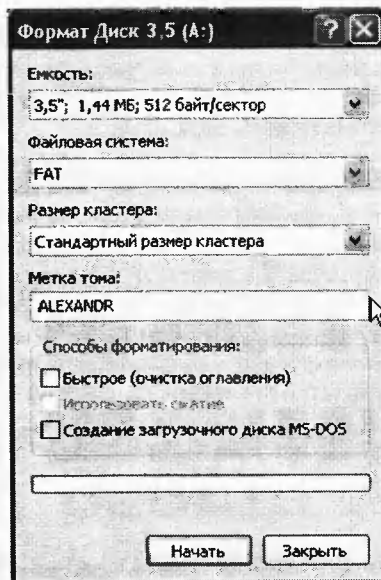
Рис. 2



Команда **Выполнить проверку** идет первой в списке. После нажатия соответствующей кнопки вам предложат (рис.2) автоматически исправлять системные ошибки (ошибки записи на диск) и попробовать восстановить сбойные сектора (это больше относится к магнитным свойствам поверхности диска). При работе с дискетой можно задействовать оба варианта, при обслуживании жестких магнитных дисков это в десятки раз замедляет работу и может потребовать перезагрузки компьютера. Если дискета в норме — диалоговое окошко закроется, в противном случае вы увидите отчет об ошибках.

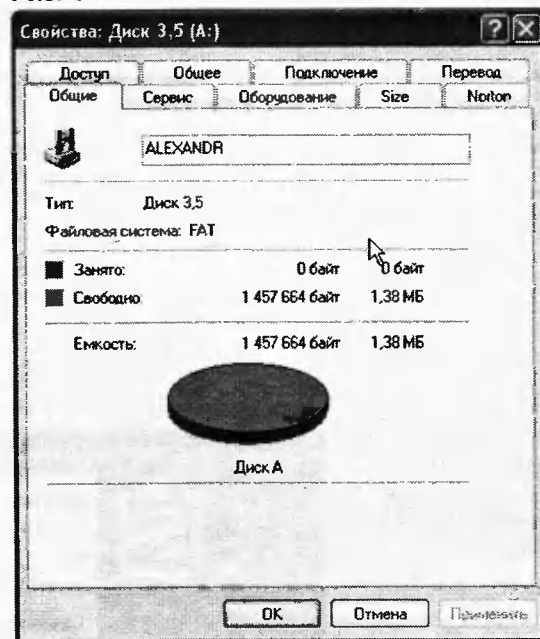
Иногда, особенно после «работы» компьютерного вируса, в качестве единственного выхода остается **форматирование диска**. При этом заново производится разбиение диска на сектора и дорожки, и вся информация, естественно, удаляется. Форматирование винчестера — процесс достаточно тонкий, даже сама стандартная программа форматирования находится на винчестере, и Windows не позволит просто так отформатировать диск, на котором установлена. Вдобавок, винчестер часто разбивается на несколько независимых логических дисков (другие названия — тома, или разделы). К форматированию винчестера обычно прибегают только при полной переустановке операционной системы. Зато старую дискету форматирование порой способно вернуть к жизни. Первым шагом у нас идет щелчок правой клавишей мыши по значку диска в Проводнике и выбор команды **Форматировать**. Windows XP вам укажет объем диска (1,44 МБ), тип файловой системы (FAT), сообщит об использовании стандартного размера кластера и предложит создание электронной подписи диска (автор выбрал Alexandr — рис.3). Часто эта

Рис. 3



метка не хуже подписи на дискете помогает разобраться с тем, чья это дискета, и что на ней записано. Просмотреть метку можно, вызвав с помощью правой клавиши мыши окошко свойств диска (рис.4). Помимо этого, предлагаются (рис.3)

Рис. 4



различные виды форматирования. Прежде всего — очистка только оглавления диска, тогда система просто не находит на диске информацию о файлах и воспринимает его как чистый диск. Лучше потерять полторы минуты, обой-

тись без этой опции и проделать все как положено. Также лучше отказаться от сжатия информации, так как не на каждой машине устанавливают программу для работы со сжатыми дисками, а экономится на сжатии не так и много. Дополнительно предлагается создать загрузочную дискету для аварийного входа в систему. Скорее всего, вам понадобится только одна такая дискета, остальные будем форматировать без этой «галочки».

Если диски после всех манипуляций заработали нормально, это еще не значит, что файлы на них размещены оптимальным образом. Не только работа с файлами, но даже элементарное перемещение окна программы приводит к записи-считыванию информации. Что же происходит при этом на диске? Во-первых, при удалении файла освобождается место, и это отмечается в оглавлении диска. При записи нового файла данные поступают на освободившиеся кластеры. Если файл целиком не помещается в отведенной области, находится следующая свободная

область, и оставшаяся часть поступает туда. В результате логически единый файл оказывается разбитым на несколько фрагментов (рис.5) — происходит фрагментация файла. Обратный процесс получил название **дефрагментация** и запускается после щелчка правой клавишей мыши по значку диска, выбора команды **Свойства**, затем вкладки **Сервис** и кнопки **Выполнить дефрагментацию** (рис.6). В зависимости от объема диска и производительности процессора дефрагментация занимает от 15 минут до, примерно, одного часа. Зато не только файлы, но и свободное пространство на диске

Рис. 5

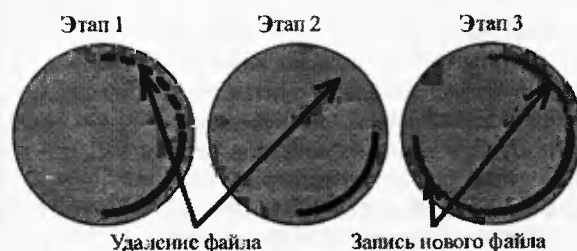


Рис. 6

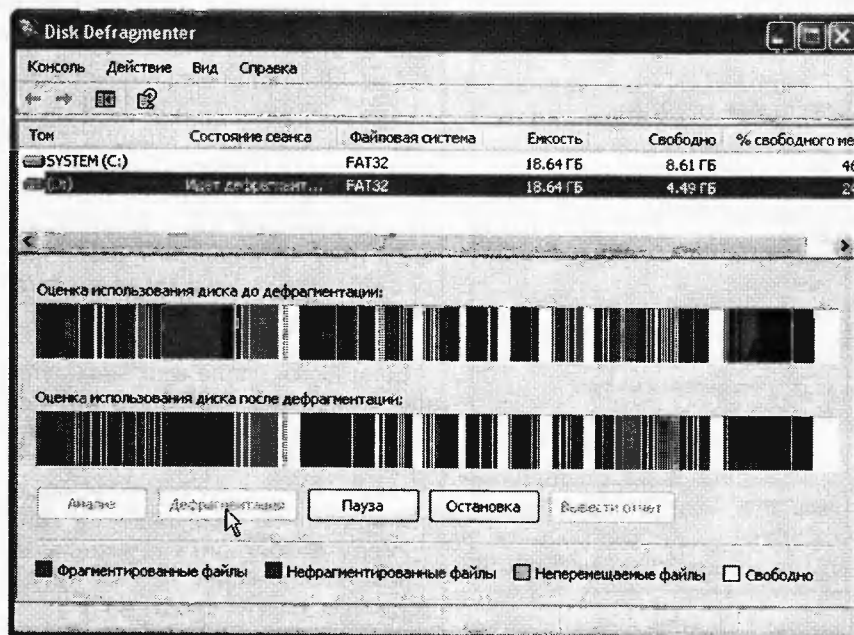
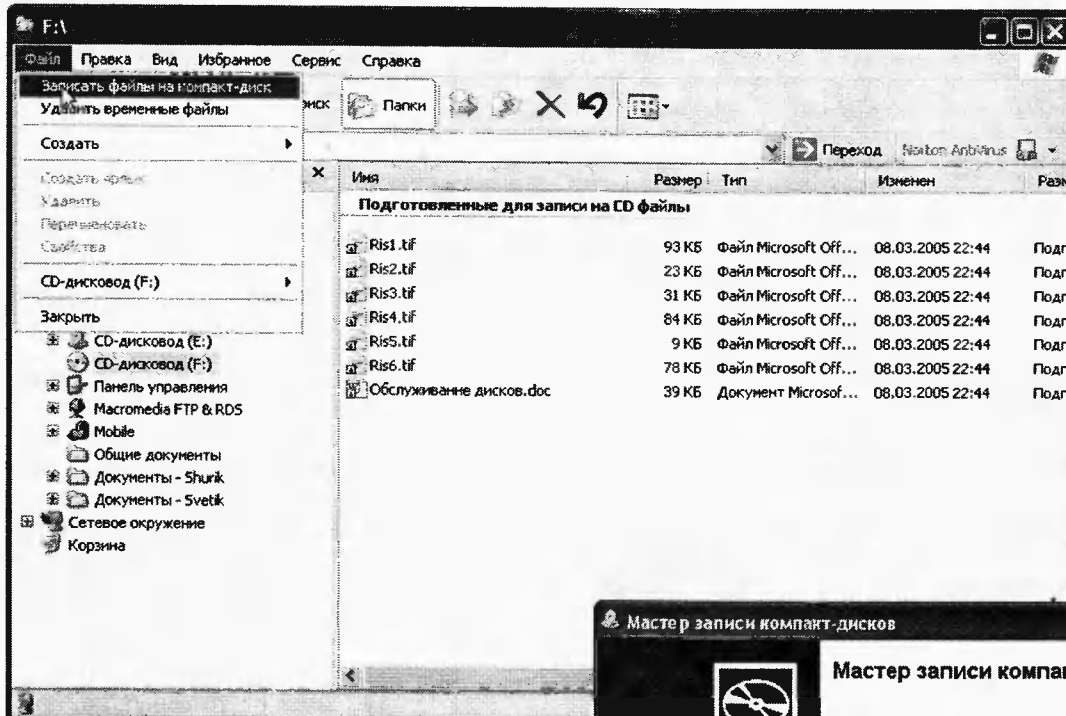


Рис. 7



оказывается собранным воедино. Теперь для считывания файла системе не приходится постоянно перепрыгивать с фрагмента на фрагмент, что повышает скорость работы.

Отдельного разговора заслуживает работа с оптическими компакт-дисками. Их запись и считывание производится с помощью небольшого полупроводникового лазера. Есть два типа устройств для работы с компакт-дисками: CD-ROM-приводы (осуществляют только считывание)

и CD-RW-приводы (запись и считывание). Следующая информация предназначена для счастливых обладателей устройств записи компакт-дисков. Вы можете обойтись без специализированных программ — поддержка записи на компакт-диск встроена в Windows XP. Стоит только помнить, что существуют компакт-диски, рассчитанные на однократную запись информации (на них наносится маркировка CD-R), и диски, на которых можно записывать и удалять информацию многократно (маркировка — CD-RW). Запись файлов на CD отличается от обычной записи только одним шагом. Начало обычное: копируем файлы (выделить, **Правка/Копировать**) и вставляем на компакт-диск (открыть в **Проводнике** диск, **Правка/Вставить**). **Внимание**, запись не происходит! Вы можете дополнительно скопировать папки и файлы на компакт. Только после под-

готовки всех файлов следует выбрать команду **Файл/Записать файлы на компакт-диск** (рис.7). Система предложит ввести имя или изменить имя диска, только после этого начнется запись (рис.8). Каждый сеанс записи сопровождается размещением крупной порции служебной информации на CD. Поэтому рекомендуется работать "крупным оптом", копируя сразу большое количество файлов и папок.

Рис. 8

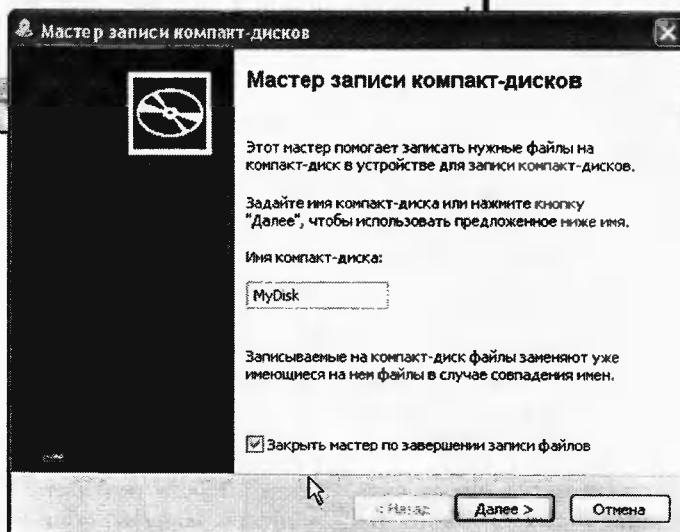
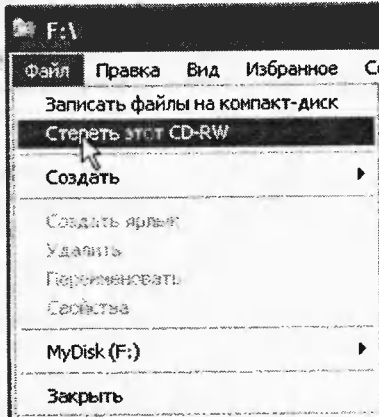


Рис. 9



Увы, возможностей Windows XP не хватает для удаления только части файлов с CD-RW-диска. Приходится записывать диск, а потом полностью его очищать: **Файл/Стереть этот CD-RW** — рис.9. Это естественная плата за многофункциональность Windows XP — операционная система может многое, но не все делает оптимальным образом. Например, в Windows XP встроена программа архивации файлов, но любая программа-архиватор предложит вам несколько дополнительных операций. Также есть специа-

лизированные программы для дефрагментации и проверки дисков. Но работать с ними сложнее, да и не все из них нормально функционируют под управлением Windows XP. К этой теме мы обязательно вернемся позднее, а в следующей статье мы, пользователи, проверившие и оптимизировавшие диски, обратимся к настройке самой операционной системы.

(Продолжение следует)

Эффективная работа с Microsoft Word XP

(Продолжение. Начало в №№1-5/2005)

Вставка и управление таблицами в документе

Наш шестой урок открывает важный и интересный блок, посвященный использованию таблиц в документах. Word предоставляет солидный арсенал средств подачи информации в наиболее компактной и наглядной табличной форме. Мы рассмотрим основные приемы создания, редактирования и форматирования таблиц, включая вычисления и применение автоформатов и мастеров, а также графическую поддержку таблиц в виде графиков и диаграмм, построенных при помощи надстройки Microsoft Graph. Начнем с создания небольшой таблички, которую вставим в уже готовый документ **Urok_5.doc**. Для этого выполним

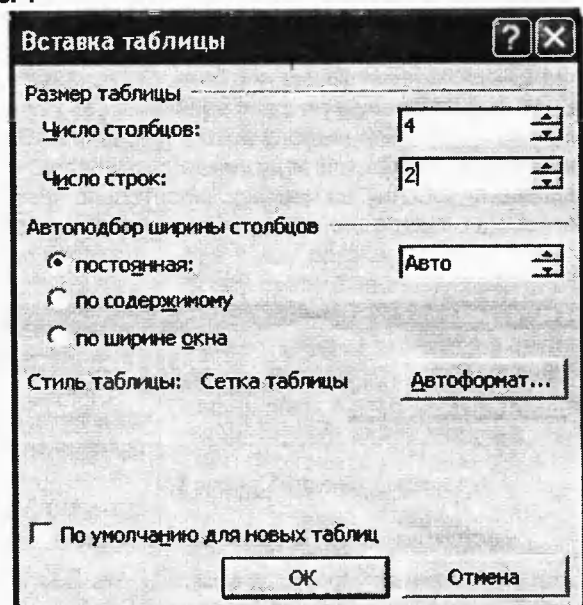
1. **Запуск Word**, после чего необходима

2. **Загрузка файла Urok_5.doc**. Выберем для размещения таблицы свободный абзац перед заключительной строкой "Приятного отдыха!". Щелкнем там мышкой, чтобы указать место для вставки таблицы. Вообще говоря, при создании таблиц можно действовать двумя способами: либо вставить пустой шаблон таблицы, а затем заполнять его информацией, либо преобразовать уже введенный текст в таблицу. Мы предпочтем первый из указанных приемов, поскольку он более упорядочен и не создает лишних проблем, например, с разделителями текста при преобразовании. Итак,

3. **Создание, редактирование и форматирование таблиц**. Перед вставкой таблицы нужно продумать ее формат, хотя бы необходимое количество столбцов, поскольку создание новых строк, как вы увидите ниже, не представляет проблемы. Со столбцами дело обстоит сложнее, так как при вставке столбца, как правило, приходится подгонять ширину таблицы и/или отдельных столбцов. В нашем примере мы хотим разместить таблицу, отражающую продолжительность пребывания — 1-й столбец; стоимость проживания — 2-й столбец, стоимость авиабилетов — 3-й столбец; и, наконец, итоговый — 4-й столбец. После указания места расположения таблицы выполняем команду **Вставить/Таблица** меню **Таблица**. В открывшемся диалоге (рис.1) укажем в соответствующих полях 4 столбца и 2 строки. В разделе **Автоподбор**

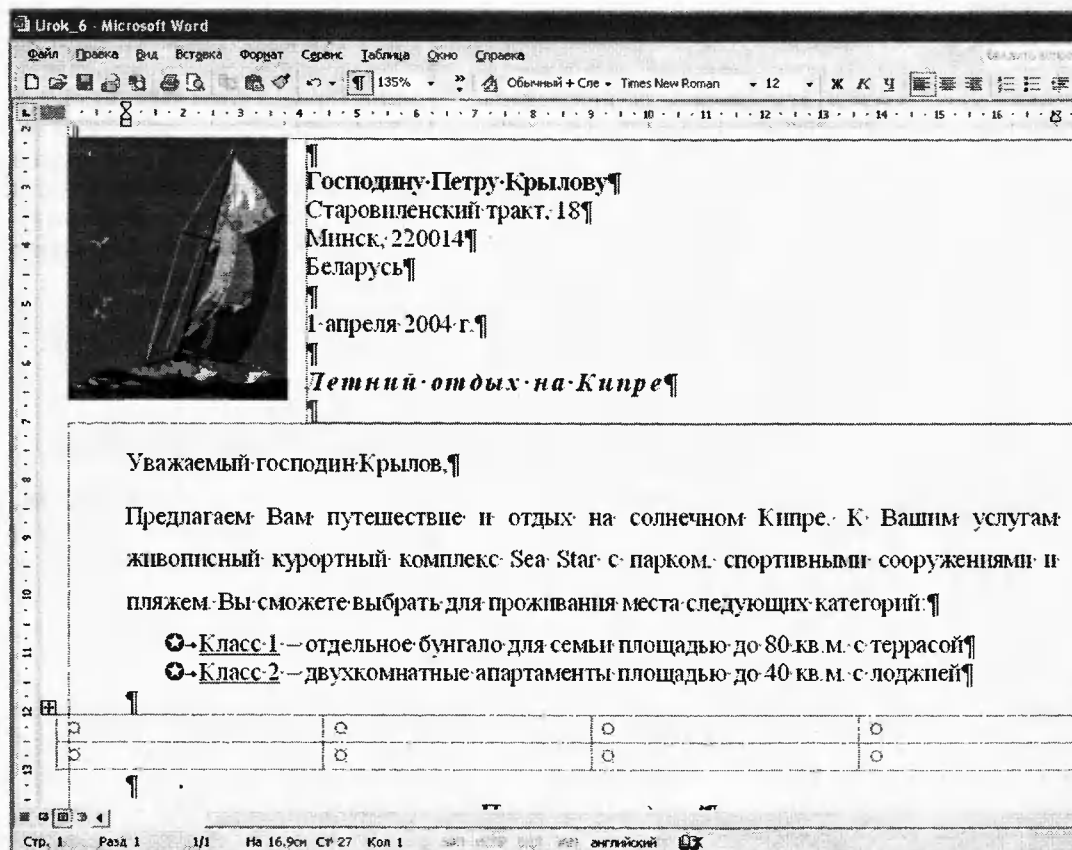
ширины столбцов можно сразу задать соответствующую величину, но если вы оставите значение **Авто**, таблица займет всю ширину листа. Если переключатель установлен в позиции **постоянная**, то все столбцы будут одина-

Рис. 1



ковой ширины. После щелчка **ОК** увидим результат, как показано на рис.2 — сетку таблицы 4x2. Линии, обрамляющие ячейки, по умолчанию имеют толщину 0,5 пт. Однако иногда возможна вставка таблицы вообще без обрамления. Тогда сетка таблицы видна только на экране, ее можно выключать и включать посредством команды **Скрыть сетку** или **Отображать сетку** меню **Таблица**. Если вы захотите расчертить таблицу при распечатке как-то особо, вам придется воспользоваться командой **Границы и заливка** меню **Формат**, но об этом позже. Не забудьте включить отображение непечатаемых символов,

Рис. 2

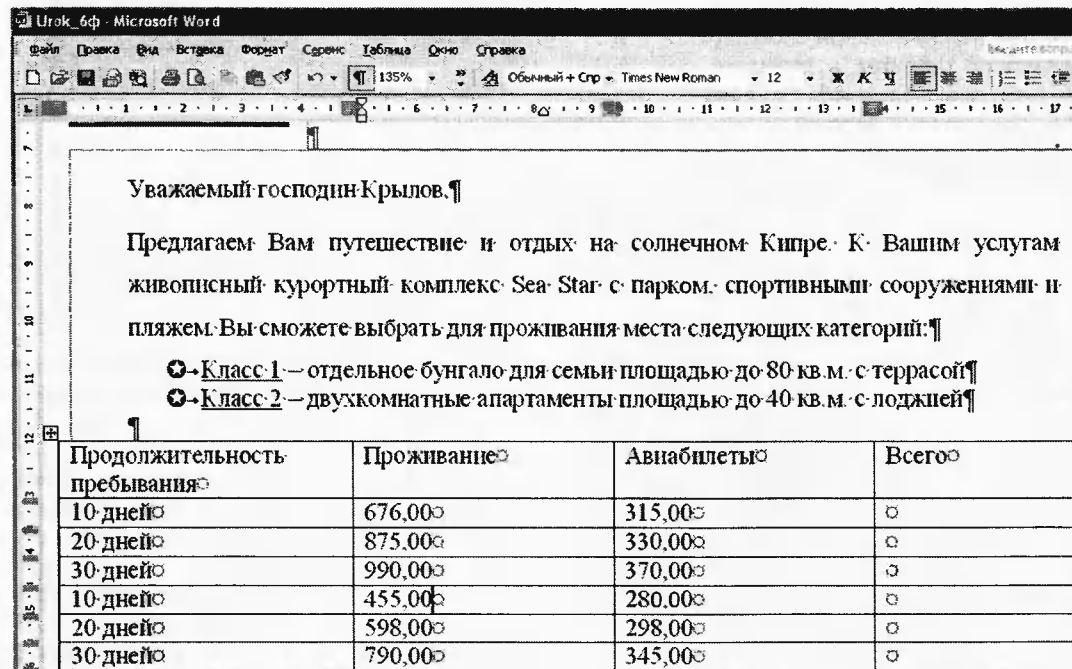


тогда вы увидите в каждой ячейке маркер — значок, похожий на чертика с четырьмя рогами. Он покажет вам, какой тип выравнивания текста в ячейке имеет место. У нас при вставке таблицы получилось выравнивание по левой границе. Чтобы таблица лучше смотрелась, можно выровнять таблицу по центру относительно вертикалей страницы. Можно это сделать сразу же, до заполне-

форму сдвоенной линии со стрелками в противоположные стороны, и на глаз изменить ширину или высоту методом "протащить и бросить". Для точного задания размеров ячеек в сантиметрах используется команда **Таблица/Свойства таблицы** после выделения соответствующего фрагмента.

Теперь заполним таблицу текстом. Щелчком мышкой в

Рис. 3



ния таблицы текстом. Для этого выделим всю таблицу следующим образом. Щелкните мышкой в одной из ячеек, тем самым установив там курсор, а затем нажмите комбинацию клавиш **Alt+Num5** (Num5 — это цифра 5 на дополнительной цифровой клавиатуре справа). После выделения таблицы щелкните по инструменту выравнивания по центру. Обратите внимание, что наши действия никак не повлияли на выравнивание текста в ячейках — маркеры ячеек по-прежнему расположены слева. Если вас не устраивает ширина столбца или высота строки, можно подвести мышку к соответствующим границам, где указатель принимает

форму сдвоенной линии со стрелками в противоположные стороны, и на глаз изменить ширину или высоту методом "протащить и бросить". Для точного задания размеров ячеек в сантиметрах используется команда **Таблица/Свойства таблицы** после выделения соответствующего фрагмента. Теперь заполним таблицу текстом. Щелчком мышкой в первой ячейке первой строки и наберем "Продолжительность пребывания" (кавычки не печатайте, я их использую, чтобы выделить вводимый текст!). Нажав клавишу **Tab**, перейдем ко второй ячейке первой строки и введем текст "Проживание", далее, в третьей ячейке — "Авиабилеты", в четвертой — "Всего". Указанный выше и другие способы перемещения по таблице отражены в таблице "Краткое содержание Урока 6" (табл.1). Нажатие клавиши **Tab** после введенного в четвертом столбце первой строки слова

Табл. 1

Краткое содержание Урока 6	
Действие	Реализация
1. Запуск Word	Щелчок по значку 
2. Загрузка файла Urok_5.doc	Файл/Открыть, указать маршрут и двойной щелчок по имени файла
3. Создание, редактирование и форматирование таблиц	
Создание таблиц	
Создание каркаса таблицы	Таблица/Вставить/Таблица и указать число столбцов, строк
Преобразование текста в таблицу	Ввести текст с разделителями: Tab между столбцами и Enter между строками, выделить и Таблица/Преобразовать/Текст в таблицу
Включение (выключение) линий сетки при отсутствии границ	Таблица/Отображать(Скрыть) сетку
Редактирование таблиц	
Перемещение в следующую/предыдущую ячейку таблицы	Tab /Shift+Tab
Выделение строки	Щелчок слева от левой границы строки (указатель ) или установить курсор и Таблица/Выделить/Строка
Выделение столбца	Щелчок сверху верхней границы столбца (указатель ) или установить курсор и Таблица/Выделить/Столбец
Выделение одной ячейки	Щелчок в левом нижнем углу ячейки (указатель )
Выделение нескольких ячеек	Выделить угловую ячейку и, удерживая кнопку мыши, протянуть требуемую область
Вставка новой строки	Установить курсор или выделить строку, над (под) которой вставляется новая, и Таблица/Вставить/Строки выше(ниже)
Вставка нового столбца	Установить курсор или выделить столбец, слева (справа) от которого вставляется новый, и Таблица/Вставить/Столбцы слева (справа)
Объединение или разбиение ячеек	Выделить и Таблица/Объединить или Разбить ячейки
Удаление строк, столбцов или ячеек	Выделить и Таблица/Удалить строки, Удалить столбцы или Удалить ячейки
Удаление содержимого строк, столбцов или ячеек	Выделить и Del
Установка размеров ячеек	Выделить и Таблица/Свойства таблицы
Подгонка границ ячеек	Подвести указатель мышки к границе и перетащить разделительную линию
Перемещение/копирование содержимого ячеек	Выделить и 1. Инструмент или контекстное меню Вырезать/Копировать, затем щелкнуть мышкой в место назначения и инструмент или контекстное меню Вставить. 2. Перетащить мышкой+Ctrl в место назначения
Вставка формулы в таблицу	Установить курсор в ячейку и Таблица/Формула
Переключение между режимами отображения кодов и значений формул в ячейках	Сервис/Параметры/Вид установить/сбросить флажок коды полей
Обновление значений формул по другим данным	Выделить ячейки и F9
Форматирование таблиц	
Прорисовка линий таблицы	Выделить и Формат/Границы и заливка, на вкладке Граница выбрать форму границ, тип прорисовки и указать ширину и цвет линий
Тонировка таблицы	Выделить и Формат/Границы и заливка, на вкладке Заливка выбрать цветовую гамму и тип тонировки
4. Сохранение файла под новым именем	Файл/Сохранить как, впечатать имя файла Urok_6.doc и Enter
5. Завершение работы Word	Файл/Выход

“Всего” приведет к перемещению во вторую строку. Здесь введем последовательно, как показано выше, в трех столбцах: “10 дней”, “676,00”, “315,00”, четвертый оставим пустым. Итак, курсор мигает в четвертой ячейке второй стро-

ки, и здесь наша таблица пока кончается. Чтобы продолжить ее, просто нажмем клавишу Tab, при этом создается новая строка. Другие способы создания новой строки показаны в таблице “Краткое содержание Урока 6”. В третьей строке заполним три столбца следующим содержанием: “20 дней”, “875,00”, “330,00”, четвертый — пропускаем. Создадим четвертую строку с содержанием: “30 дней”, “990,00”, “370,00”. Не забывайте — четвертый столбец оставляем свободным! Введенная в таблицу до этих пор информация относится к категории отдыха “Класс 1”. Теперь сделаем то же самое в отношении второго класса. Создадим и заполним еще три строки со следующим содержанием: “10 дней”, “455,00”, “280,00”; “20 дней”, “598,00”, “298,00”; “30 дней”, “790,00”, “345,00”.

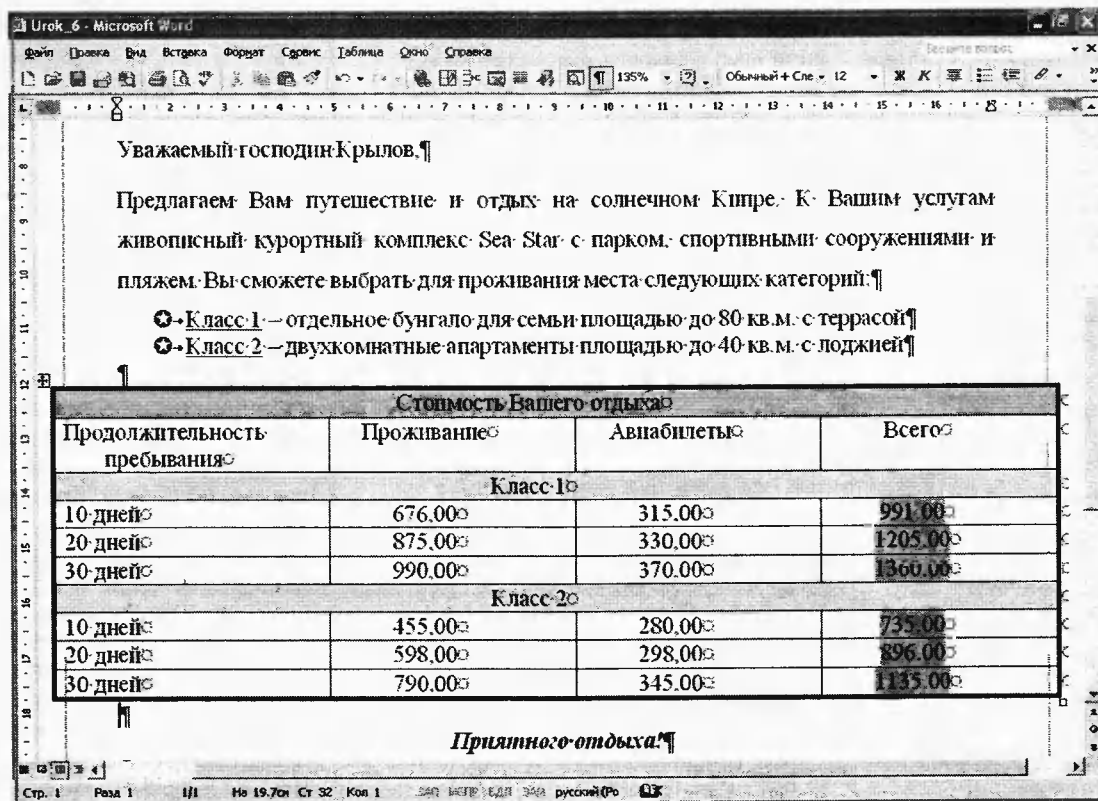
Конечно, в таком виде (рис.3) таблица нечитабельна. Нужно внести в таблицу поясняющие элементы, научиться приемам выделения и вставки фрагментов таблицы, которые отражены в таблице “Краткое содержание Урока 6” как процедуры редактирования таблиц. Вначале сделаем шапку таблицы. Щелчком мышки слева от левой границы первой строки выделим эту строку и выполним команду Таблица/Вставить/Строки выше. В результате появится новая, пока незаполненная первая строка. В ней предстоит ввести текст: “Стоимость Вашего отдыха” и расположить его по центру. После вставки новой строки она у нас осталась выделенной, поэтому сразу выполним команду Таблица/Объединить ячейки. Тогда вместо четырех столбцов в выделенной строке останется один объединенный, где мы и напечатаем указанный выше текст, а затем включим инструмент По центру. Надеюсь, что вы справились. Теперь такую же процедуру проделайте над строками, в которых содержится информация об отдыхе продолжительностью 10 дней по первому и второму классу. Введите в новых строках записи “Класс 1” и “Класс 2” и выровняйте их по центру. Сравните результат с рис.4, где показан окончательный вариант таблицы. После того, что мы уже проделали, осталось заполнить четвертый столбец, выполнив в его ячейках вычисление сумм чисел по строкам, а также от-

форматировать таблицу — привести к предложенному на рисунке виду. Выделим щелчком мыши слева от левой границы вторую строку с заголовками столбцов и воспользуемся инструментом По центру, чтобы расположить

текст как на образце. Теперь выделим ячейки с цифрами затрат (заодно и пока пустые ячейки с будущими расчетами). Как это сделать? Начинать лучше с верхней левой ячейки требуемого диапазона. Поместим курсор мыши в левый нижний внутренний угол ячейки с числом 676,00 (вторая ячейка третьей строки) до появления указателя в виде малой наклонной черной стрелки, нажмем кнопку мыши, и, удерживая ее, протянем диапазон ячеек вначале вправо до конца строки, а затем вниз, до конца табли-

(рис.5). В поле **Формула** после обязательного ввода знака равенства набирается расчетная формула. При этом в качестве операндов можно использовать как числа, так и ссылки или адреса ячеек таблицы. При адресации ячеек строка обозначается латинской буквой, столбец — арабской цифрой (табл.2). Адрес отдельной ячейки записывается буквой столбца и цифрой строки, на пересечении которых ячейка находится. При указании адреса фрагмента таблицы, состоящего из несколь-

Рис. 4



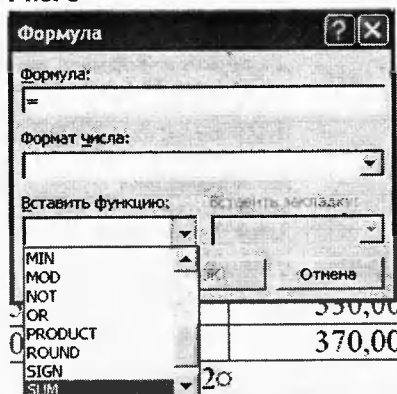
цы. Ничего, что выделилась строка "Класс 2" — для нее требуется то же выравнивание. Отпустим кнопку мыши и щелчком инструмент **По центру**.

А теперь мы приступаем к весьма серьезной операции — вычислениям в таблице Word. Текстовый процессор реализует вычисления благодаря вставке специального поля — процедуры внедрения определенной функции Word, вызывающей выполнение некоторой инструкции, например, вычисления суммы чисел в строках. Любое поле может быть отображено на экране в двух видах — кодом или значением. Для вычислений код обычно представляет запись формулы, а значение — результат расчета этой формулы. Переключение между этими двумя видами отображения полей осуществляется установкой или сбросом флажка **коды полей** диалогового окна **Сервис/Параметры/Вид**. Кроме того, Word позволяет обновлять результаты вычислений по инструкциям полей при замене исходных значений. Для этого необходимо выделить обновляемое поле (например, ячейку с формулой), а затем нажать функциональную клавишу обновления полей **F9** (верхний ряд клавиатуры) — и, например, сумма будет пересчитана.

А теперь о правилах игры при вставке формул в таблицу. Процедура запускается после указания места расположения формулы командой **Таблица/Формула**

ких ячеек, записывают через двоеточие сначала ссылку на левую верхнюю, затем на правую нижнюю ячейку диапазона (например, A1:B2). В формулах можно использовать знаки математических операторов (+, -, *, /, ^, %) и операторов сравнения (=, <, >, <=, >=, <>). Значительно облегчают задачу встроенные функции из поля списка **Вставить функцию** (рис.5), для которых можно задавать указанные выше операнды через запятую, а также использовать английские слова **left** или **above** в вычислениях для ячеек соответственно левее или выше ячейки с формулой.

Рис. 5



В диалоге **Формат числа** можно заказать способ представления результирующего числа.

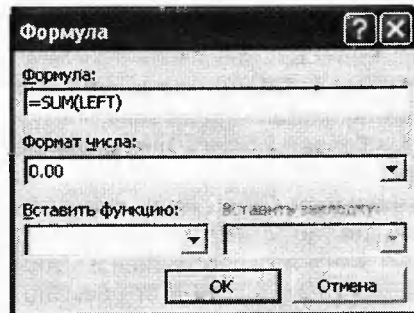
Ну что ж, попробуем вставить первую формулу в крайнюю правую ячейку четвертой строки, предварительно щелкнув там мышкой для установки верной пози-

Табл. 2

	A	B	C	D
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4

ции курсора. После выполнения команды **Таблица/Формула**, Word, проанализировав положение курсора, предложит в диалоге **Формула** (рис.6) как раз ту функцию, какую нам нужно: **=SUM(LEFT)**, то есть сумму чисел левее от указанной ячейки. Нам остается только выбрать формат числа.

Рис. 6

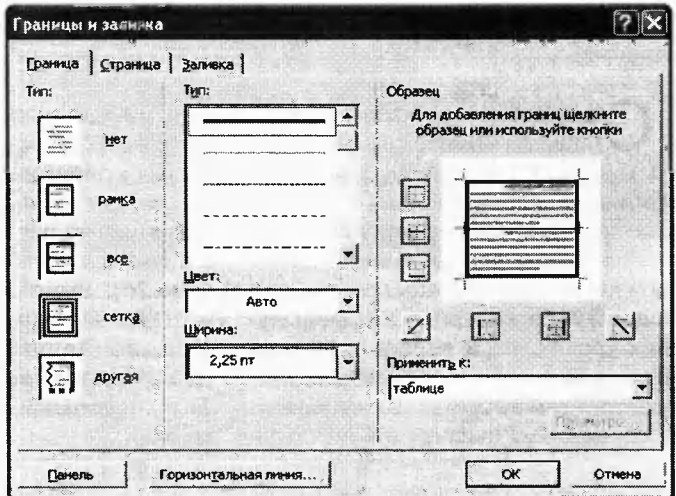


гает уже другую формулу — **=SUM(ABOVE)**. Ее можно подкорректировать прямо в строке диалога, выделив мышкой слово **ABOVE** и напечатав поверх выделения **LEFT**. В завершение щелкаем **OK** и получаем вторую формулу. Остальные можно сделать тем же манером, но можно и скопировать формулу в оставшиеся ячейки четвертого столбца, пользуясь приемами редактирования. Выделим ячейку с готовой формулой и скопируем выделение в буфер, используя инструмент **Копировать** или соответствующую команду контекстного меню (правая кнопка мыши). После этого можно выполнить вставку содержимого буфера в нужные ячейки (инструмент или контекстное меню **Вставить**). Пока получается не совсем правильно — ведь мы работаем в режиме отображения значения поля, а скопированные значения соответствуют числам одной и той же пятой строки. Однако код поля формулы верный — в этом можно убедиться, установив флажок **коды полей** в диалоге **Сервис/Параметры/Вид**. Сделав это второй раз, снова вернемся к значениям полей. Их придется пересчитать, исходя из чисел, стоящих слева в шестой, восьмой, девятой и десятой строках. Выделяем в этих строках последние ячейки с формулой (можно все сразу или по отдельности) и нажимаем **F9**. В результате получим правильные значения сумм чисел по строкам, представляющие расходы на отдых при соответствующих условиях. Обратите внимание — на рис.4 значения сумм затемнены серым цветом. Такую установку можно заказать с помощью диалога **Сервис/Параметры/Вид**, выбрав в списке **затенение полей** значение **Всегда**. Это визуально напоминает вам, что данный результат получен при помощи расчетных полей, а не просто набран с клавиатуры. При печати это затенение не передается.

Другие типы формул мы опробуем в следующий раз, а сейчас займемся внешним видом таблицы. Исходя из примера, таблица должна быть расчерчена сеткой, причем внешняя граница — более жирной линией. Кроме того,

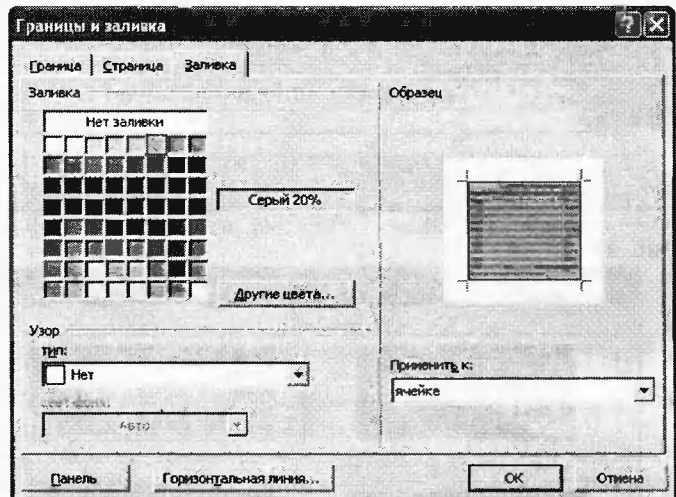
первая, третья и седьмая строки должны быть оформлены заливкой. Эта работа выполняется при помощи диалогового окна **Формат/Границы и заливка**. Выделим таблицу целиком и выполним соответствующую команду. На вкладке **Граница** (рис.7) выберем в разделе **Тип** об-

Рис. 7



разец **сетка** и установим в списке **Ширина** толщину внешней линии 2,25 пт. Внутренние линии остаются толщиной по умолчанию — 0,05 пт. Кнопкой **OK** закроем диалог. Затем выделим первую строку и снова выполним **Формат/Границы и заливка**. На вкладке **Заливка** (рис.8) выберем серую заливку 20%. Для третьей и седьмой строки сделаем серую заливку 10%. Теперь таблица украшает документ и хорошо читается.

Рис. 8



4. Сохранение файла под новым именем. Сохраним версию нашего документа с табличкой под новым именем, выполнив команду **Файл/Сохранить как**. В поле **Имя файла** поверх старого имени введем **Urok_6.doc** и **Enter**.

5. Завершим работу Word, выполнив команду **Выход** меню **Файл**.

На этом уроке мы только копнули верхушку горы под названием "работа с таблицами". Настоящие чудеса ждут нас впереди, и мы постараемся не пропустить ни одно из них.

(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 №№1-5/2005)

Продолжаем оформлять

Оформление электронной таблицы вы можете осуществлять не только путем непосредственного форматирования, выбирая необходимые параметры через панель инструментов **Форматирование** или команду меню **Формат/Ячейки...**, что было подробно рассмотрено на предыдущих уроках, но и воспользоваться функцией автоформатирования. Это позволит вам выбрать оформление из предложенных шаблонов и в считан-

ные минуты оформить выделенный блок ячеек в соответствии с предложенным образцом. Откроем рабочий документ **Учебный_файл.xls**. Переклочимся на лист 2 и введем информацию в ячейки рабочего листа (рис.25). Выделим диапазон ячеек A1:D8 и выполним команду меню **Формат/Автоформат...** В окне диалога **Автоформат** (рис.26) выберем необходимый образец оформления.

Снять оформление с таблицы проще всего при помощи команды меню **Правка/Очистить/Форматы**, предварительно выделив нужный диапазон ячеек.

Учимся работать с вычислениями

Что ж, настало время заняться вещами более важными, чем создание и форматирование таблиц средствами Microsoft Excel. Ведь основная функция табличного процессора состоит вовсе не в построении и оформлении таблиц, хотя и это немало важно. Научиться проводить обработку данных: просматривать, анализировать, обобщать, производить расчеты на основе имеющихся данных и строить прогнозы — вот наша задача. Наш сегодняшний урок введет вас в мир вычислений.

Для расчетов в Microsoft Excel используются формулы и функции. **Формула** — это выражение, начинающееся со знака равенства (=), знака плюс (+) или знака минус (-) и включающее числовые константы, ссылки на ячейки, имена ячеек, функции, набор стандартных операторов и специальных символов. Формулы вам предстоит научиться создавать самим. Вы можете использовать их для самых разнообразных вычислений. Напишите самую сложную формулу, на которую способно ваше воображение, и можете быть уверены, что ее можно вычислить средствами Microsoft Excel. Но помните, не всегда следует изобретать формулу для тех или иных расчетов. Microsoft Excel позволяет оперировать целым набором так называемых встроенных функций, что значительно упрощает вашу задачу. Встроенные функции Microsoft Excel XP называются **функциями рабочего листа** и представляют собой предопределенные формулы, предназначенные для вычислений, носящие специфический характер, соответствующие различным областям применения. Вам потребуется только выбрать ту, которая необходима для расчета требуемых

Рис. 25

	A	B	C	D
	Наименование товара	Количество единиц	Цена единицы товара (у.е.)	Общая стоимость
1				
2	Клавиатура PS/2 GENIUS EasyTouch	10	14	
3	Монитор LCD, ЖК 17" BENQ FP71E+	2	450	
4	Сканер планшетный HP ScanJet 2400c Q3841A	3	84	
5	Принтер лазерный HP LaserJet 1010 Q2460A	2	171	
6	Мышь PS/2 A4TECH/ SWOP-3 Optical	5	8,5	
7	Дискеты 3.5" 1.44 Mb VERBATIM	55	0,4	
8	Итого:			
9				

Рис. 26

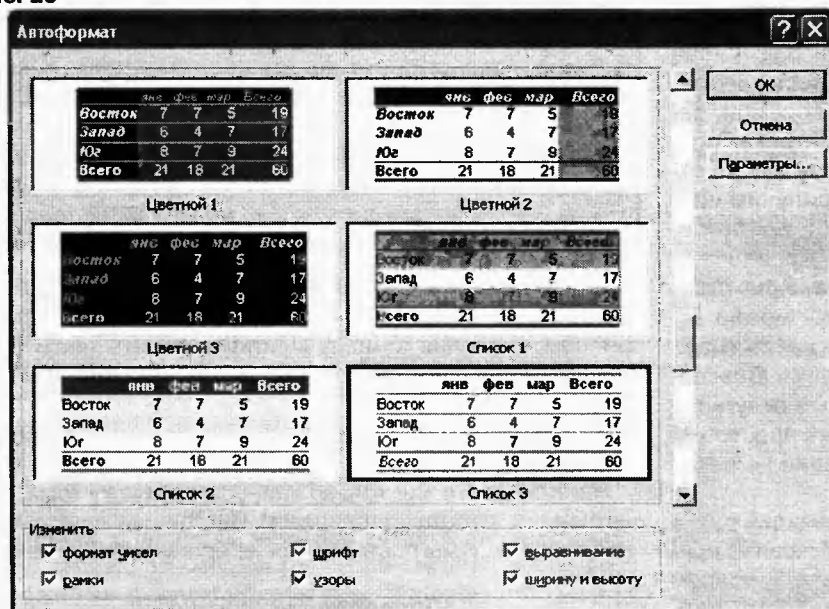


Табл. 10. Общая характеристика операторов

Обозначение	Действие	Примеры использования операторов	Результат вычисления
Арифметические операторы			
+ (сложение)	Складываются два арифметических выражения	=15+34 =A1+B7	49 Сумма арифметических выражений ячеек A1 и B7
- (вычитание)	Из первого арифметического выражения вычитается второе	=107-45 =C3-D1	62 Разность арифметических выражений ячеек C3 и D1
/ (деление)	Первое арифметическое выражение делится на второе	=64/4 =F12/11	16 Частное от деления арифметического выражения ячейки F12 и числа 11
* (умножение)	Перемножаются два арифметических выражения	=15*13 =G2*G3	195 Произведение арифметических выражений ячеек G2 и G3
% (процент)	Вычисляется процент от арифметического выражения	=1% =A1%	0,01 Процент от арифметического выражения ячейки A1
^ (возведение в степень)	Возводит первое арифметическое выражение в степень, заданную вторым арифметическим выражением	=5^8 =D1^2	390625 Арифметическое выражение ячейки D1 в степени 2
Операторы сравнения			
=	Равно	=F1=F2 =12=12	Проверяется равенство значений ячеек F1 и F2 ИСТИНА
>	Больше, чем	=F1>F2 =34>57	Проверяется, значение ячейки F1 больше, чем значение ячейки F2? ЛОЖЬ
<	Меньше, чем	=F1<F2 =7<7	Проверяется, значение ячейки F1 меньше, чем значение ячейки F2? ЛОЖЬ
>=	Больше или равно	=F1>=F2 =234>=124	Проверяется, значение ячейки F1 больше или равно значению ячейки F2? ИСТИНА
<=	Меньше или равно	=F1<=F2 =567<=1789	Проверяется, значение ячейки F1 меньше или равно значению ячейки F2? ИСТИНА
<>	Не равно	=F1<>F2 =15<>13	Проверяется неравенство значений ячеек F1 и F2 ИСТИНА
Текстовые операторы			
& (соединение текстовых строк)	Присоединяет вторую текстовую строку к концу первой текстовой строки	=A2&B2 ="табличный процессор"&"MS Excel XP" =12&56	Создается текстовая строка как результат присоединения содержимого ячейки B2 к концу содержимого ячейки A2 Табличный процессор MS Excel XP 1256
Операторы связи			
: (диапазон)	Ссылается на ячейки диапазона между указанными начальной и конечной ячейкой	=СУММ(A1:A10)	Суммируются арифметические выражения ячеек от A1 до A10
: (объединение)	Ссылается на объединение указанных ячеек (диапазонов ячеек)	=СУММ(B1:B2; C1:C2) =СУММ(A1:A5; C1:C3)	Суммируются арифметические выражения ячеек B1, B2, C1, C2 Суммируются арифметические выражения диапазонов ячеек A1:A5 и C1:C3
Пробел (пересечение)	Ссылается на пересечение указанных диапазонов ячеек	=СУММ(A1:B10 B1:C10)	Суммируются арифметические выражения ячеек, общих для диапазонов A1:B10, B1:C10, т.е. B1:B10

величин и показателей. Обычно, чтобы не усложнять себе жизнь, формулы рекомендуется использовать лишь в том случае, если нет подходящей функции для тех же расчетов.

Особенность работы с формулами в электронных таблицах заключается в том, что расчет по заданным формулам выполняется автоматически. Изменение содержимого какой-либо ячейки приводит к пересчету значений всех ячеек, которые с ней связаны формульными отношениями, и тем самым, к обновлению всей

таблицы в соответствии с изменившимися данными.

Синтаксис формулы и порядок выполнения операторов

А сейчас приступим к главному моменту. Постараемся постигнуть основные принципы создания формул. Как уже было упомянуто выше, формулу при желании можно поместить в любую ячейку рабочего листа. И тут же в ячейке вы увидите результат вычисления. Сама же формула отобразится в строке формул. Это наи-

более привычный режим работы. Однако при желании можно в ячейке вместо значения отобразить саму формулу.

Приступая к созданию формулы, вам всегда следует помнить следующее:

- формуле всегда должен предшествовать знак "=" или знаки "+" и "-",
- в формуле нельзя использовать символы русского алфавита;
- необходимо учитывать последовательность выполнения математических операций.

В формуле могут использоваться арифметические и текстовые операторы, операторы сравнения и связи. Полный список операторов, их краткая характеристика и примеры использования приведены в **табл.10**. Обратите внимание, что арифметические операции можно осуществлять только над числами, и результатом всегда является число. Операторы сравнения используются для сравнения как числовых, так и текстовых значений, а результатом операций сравнения всегда являются логические значения ИСТИНА или ЛОЖЬ. Текстовый оператор объединяет числовые или текстовые значения в одну строку символов. Независимо от того, текстовые или числовые значения вы объединяете, результатом всегда будет текстовая строка (табл.10). Адресные операторы предназначены для того, чтобы сделать ссылку на нужный диапазон ячеек.

Если в формуле используются несколько операторов различного типа, необходимо учитывать, что при обработке данных Microsoft Excel исполь-

$(F1+F3)^3-D1$ сначала выполняется операция сложения.

Создание и редактирование формул

Если вы разобрались в тонкостях использования операторов, то следующим шагом нашего путешествия по стране формул будет освоение приемов их ввода и редактирования. В Microsoft Excel XP формулы можно создавать несколькими способами: непосредственным вводом, указанием при помощи мыши и указанием при помощи клавиатуры. В случае непосредственного ввода вам придется напечатать формулу с клавиатуры в нужную ячейку рабочего листа и подтвердить ввод нажатием клавиши **Enter** или щелчком мыши по кнопке **Enter** (✓) перед строкой формул. При этом вы, конечно, не забыли, что в формуле не должны встречаться буквы русского алфавита, и заблаговременно установили латинский режим клавиатуры. Но вы можете упростить свою задачу и ссылки на адреса ячеек в формулах делать при помощи мыши

или перемещением курсора в соответствующую ячейку рабочего листа. На **рис.27** приведена таблица с данными о количестве и ценах на товары. Для вычисления общей стоимости товара каждого вида необходимо воспользоваться формулой:

Количество единиц товара * Цена
единиц товара = Общая стоимость

В случае непосредственного ввода формулы:

1. Установите табличный курсор в ячейку D2, в которую необходимо поместить формулу.
 2. Наберите с клавиатуры формулу $=B2*C2$.
 3. Подтвердите ввод формулы нажатием клавиши **Enter** или щелчком мыши по кнопке **Enter** (✓) перед строкой формул.
- В случае ввода формулы путем указания при помощи мыши:
1. Установите табличный курсор в ячейку D2, в которую необходимо поместить формулу.
 2. Наберите с клавиатуры знак равенства (=).
 3. Щелкните мышью по ячейке B2.
 4. Наберите с клавиатуры знак умножения (*).
 5. Щелкните мышью по ячейке C2.
 6. Подтвердите ввод формулы нажатием клавиши **Enter** или щелчком мыши по кнопке **Enter** (✓) перед строкой формул.
- В случае ввода формулы путем указания при помощи клавиатуры:
1. Установите табличный курсор в ячейку D2, в которую необходимо поместить формулу.
 2. Наберите с клавиатуры знак равенства (=).
 3. Переместите табличный курсор

Табл.11. Порядок выполнения операторов

Оператор	Обозначение	Приоритет
Оператор диапазона	:	1
Операторы объединения и пересечения	;	2
	пробел	
Оператор процентов	%	3
Оператор возведения в степень	^	4
Операторы умножения и деления	* /	5
Операторы сложения и вычитания	+ -	6
Текстовый оператор	&	7
Операторы сравнения	= > < >= <= <>	8

зует их в определенной последовательности. Например, в формуле $=F1+F5/15$ приоритет имеет операция деления, в формуле $=D3-D2*7+D13^2$ — возведение в степень, а в формуле $=A1+A2-A3+A4$ предполагается последовательное выполнение операций. Более подробно иерархия операторов отражена в **табл.11**. При необходимости нарушить общепринятую иерархию операций воспользуйтесь скобками, поскольку операторы, заключенные в скобки, имеют преимущество перед остальными, т.е. выполняются первыми. Например, в формуле

Microsoft Excel - Учебный файл

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка Видеотека

Аrial Cyr 10 Ж К Ц

D2 =B2*C2

	A	B	C	D
	Наименование	Количество единиц	Цена единицы	Общая стоимость
	товара	товара	товара (y.e.)	
1	Клавиатура PS/2 GENIUS EasyTouch	10	14	140
2	Монитор LCD, ЖК 17" BENQ FP71E+	2	450	
3	Сканер планшетный HP ScanJet 2400c Q3841A	3	84	
4	Принтер лазерный HP LaserJet 1010 Q2460A	2	171	
5	Мышь PS/2 A4TECH/ SWOP-3 Optical	5	8,5	
6	Дискеты 3.5" 1.44Mb VERBATIM	55	0,4	
7	Итого:			
8				
9				
10				

Готово

Рис. 27

в ячейку B2 при помощи клавиш управления курсором.

4. Наберите с клавиатуры знак умножения (*).

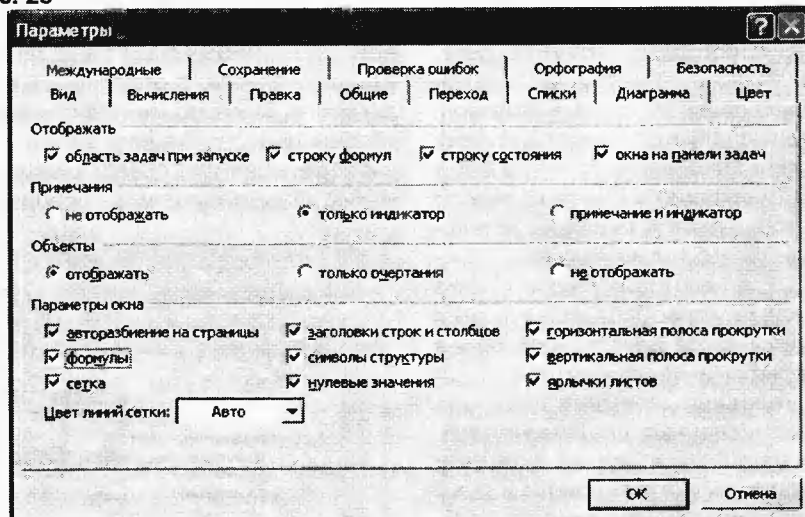
5. Переместите табличный курсор в ячейку C2 при помощи клавиш управления курсором.

6. Подтвердите ввод формулы нажатием клавиши **Enter** или щелчком мыши по кнопке **Enter** (✓) перед строкой формул.

После ввода формулы в ячейке отображается результат, а сама формула — в строке формул (рис.27). Это наиболее привычный режим работы с вычислениями. Однако при желании можно в ячейке вместо результата показать саму формулу, что немало важно в процессе поиска ошибок. Для этого выполните команду **Параметры...** меню **Сервис** и установите флажок **Формулы** на вкладке **Вид** (рис.28). Обратная процедура снятия флажка **Формулы** в окне **Параметры** (рис.28) вернет вас к обычному режиму отображения формул.

Теперь представьте себе ситуацию — вы допустили ошибку при вводе формулы. Во-первых, вы сразу же увидите сообщение об ошибке (например, #ЗНАЧИ, #ИМЯ?, #ССЫЛКА!, #ЧИСЛО!, #ДЕЛ/0 и др.). Исправить ситуацию

Рис. 28



можно путем редактирования введенной формулы. Редактировать можно непосредственно в самой ячейке, выполнив по ней двойной щелчок мыши. При этом в ячейке вместо результата вновь отобразится формула, причем ячейки, ссылки на которые встречаются в формуле, выделяются на рабочем листе цветными рамками, а сами ссылки отображаются в формуле тем же цветом. Все это облегчает и процесс редактирования, и проверку правильности формулы. С другой стороны,

можно установить табличный курсор на ячейку с формулой, щелкнуть мышью в строке формул — там отображается формула, и внести правки.

Разговор о создании более сложных формул, приемах их копирования и перемещения, а также о работе со встроенными функциями ожидает нас на дальнейших уроках. А в завершение сегодняшнего урока осуществим процедуру сохранения рабочей книги **Учебный_файл.xls**.

(Продолжение следует)

В.КУЦ,

E-mail: victor_kootz@mail.ru

ABBYY PDF Transformer 1.0

Различные производители ПО борются с компьютерным "пиратством" по-разному — некоторые делают главную ставку на репрессивные меры карательных органов, защищающих их интеллектуальную собственность от всяческих посягательств. Другие же, оценивая нынешнюю ситуацию более реалистично, пытаются противостоять этому злу (пока еще неизбежному) с помощью экономических рычагов, максимально удешевляя свои продукты начального уровня. Именно к таким вендорам можно отнести российскую компанию ABBYY Software, которая в последнее время все более активно пытается "отвоевать" у "пиратов" низший ценовой сегмент пользователей своего ПО. Для этого компания уже выпустила на рынок недорогие программы OCR FineReader Home Edition, электронный словарь Lingvo "Первый шаг" и конвертор PDF-файлов PDF Transformer. Два первых продукта

являются "усеченными" версиями "серьезных" пакетов FineReader Professional 7.0 и Lingvo 10.0 соответственно. А последний, достаточно оригинальный продукт ABBYY PDF Transformer 1.0, предназначенный для преобразования PDF-документов в редактируемые форматы, мы рассмотрим подробно.

Прежде всего, пару слов о самом формате PDF (Portable Document Format), который уже достаточно продолжительное время пользуется большой популярностью у самого широкого круга компьютерных пользователей. И популярность эта обусловлена тем, что документы в виде PDF-файлов обладают, помимо достаточно компактных размеров, таким важнейшим свойством как мультиплатформенность, то есть они одинаково успешно могут открываться практически на любом современном компьютере, какая бы ОС на нем ни стояла. Поэтому с ним работают

как дома, так и в офисе — в PDF хранят и пересылают электронные книги, статьи, брошюры, руководства пользователя, отчеты, договоры и листовки, в общем, самые разные документы. Для просмотра PDF-файлов компания Adobe, разработавшая этот формат, предлагает достаточно удобную программу Acrobat Reader. Но вот создавать (или редактировать) PDF гораздо проблематичнее, так как предназначенный для этих целей редактор Adobe Acrobat, рассчитанный исключительно на профессиональное применение, чересчур объемный и требовательный к ресурсам компьютера, да и стоит слишком дорого для подавляющего большинства частных пользователей.

Кроме того, формат PDF и сам по себе не лишен определенных недостатков, порой, довольно неприятных. В частности, при использовании кириллических шрифтов часто некорректно работают не только функции

поиска, но и обычные операции копирования/вставки. Еще одна не самая приятная особенность PDF-документа заключается в том, что если он защищен паролем, то пользователь, даже зная этот пароль, не имеет возможности ничего из этого документа скопировать. Поэтому, вместо того чтобы долго и, порой, безуспешно бороться с такими "особенностями" PDF, в некоторых случаях было бы полезно преобразовывать этот формат в какой-либо другой, более дружелюбный к пользователю.

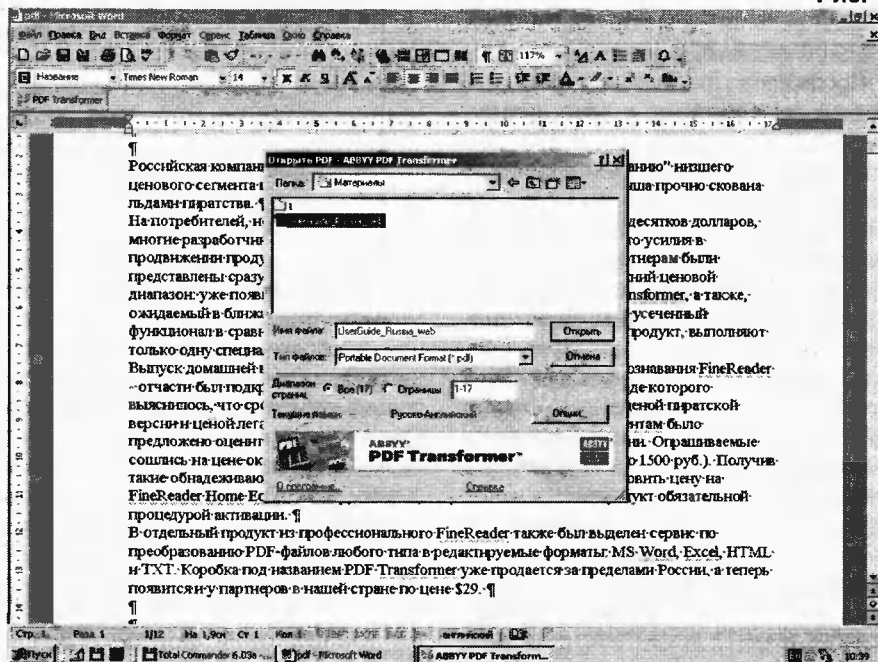
До недавнего времени довольно немногочисленные средства преобразования PDF в другие форматы были достаточно примитивны и далеко не всегда обеспечивали адекватную конвертацию текста (даже на латинице, с кириллическими шрифтами — вообще полный мрак). А уж о сохранении всех тонкостей форматирования исходного документа и говорить не приходится. Ситуация изменилась к лучшему только после выхода OCR-пакета ABBYY FineReader 7.0, средства работы с PDF которого были в состоянии решить задачу практически любой сложности. Но сам FineReader 7.0, впрочем, как и любой другой программный продукт, рассчитанный на профессионалов, достаточно объемный и дорогой. А упрощенная версия FineReader Home Edition, ориентированная на домашнего пользователя, инструментов для работы с PDF лишена.

Но теперь, благодаря новой программе PDF Transformer, все проблемы с конвертацией PDF-документов позади. Эта программа, базируясь на тех же технологиях, что и знаменитый FineReader, решает конкретную повседневную задачу — преобразование PDF-документов любых типов (поддерживаются форматы Adobe PDF версий 1.0 — 1.5) в наиболее популярные форматы DOC (RTF), XLS, TXT и HTML, редактирование которых не представляет никакой проблемы. При этом сохраняется полное оформление исходного документа: таблицы, картинки, колонки и заголовки. Как и положено истинно российской программе, корректно обрабатываются документы на русском языке (и еще на 36 других языках), поддерживаются даже нестандартные кодировки. Благодаря использованию уникальной технологии распознавания ABBYY FineReader, PDF Transformer способен преоб-

разовывать в редактируемый вид даже такие PDF-файлы, где вместо "текстового слоя" присутствует только его изображение (как в отсканированных документах). Не менее важна и такая особенность программы как ее способность корректно преобразовывать PDF-файлы, защищенные паролем.

корректно, с сохранением исходного вида, включая графику и сложные таблицы. После преобразования можно продолжать работу с PDF-файлом в нужном приложении. Полученные по электронной почте PDF-файлы можно преобразовать, вызвав программу прямо из Microsoft Outlook.

Рис. 1

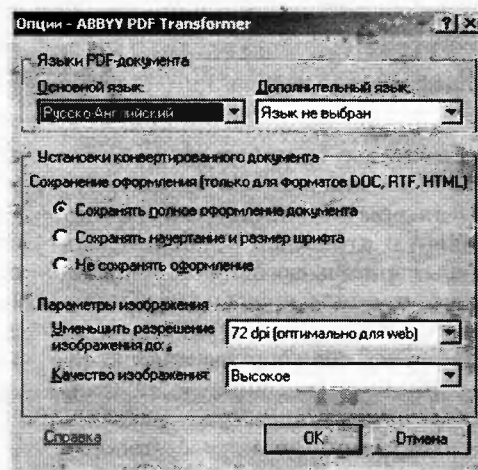


Несмотря на то что дистрибутив программы PDF Transformer не отличается особой компактностью и занимает чуть меньше сорока мегабайтов, сами разработчики именуют ее не иначе как "утилитой", тем самым намекая на предельную простоту ее использования. И действительно, она выполнена в виде мастера, который может вызываться из контекстного меню Windows Explorer или из меню и панелей инструментов Word или Excel (рис. 1).

Выбрав интересующий нас документ (кстати, PDF Transformer позволяет конвертировать не только весь документ целиком, но и выбрать отдельные его страницы), остается настроить при необходимости немногочисленные опции преобразования (рис. 2).

Здесь из списка можно выбрать два языка распознавания (основной и дополнительный), соответствующие языкам документа в формате PDF, а также определить с оформлением итогового документа и его качеством. Не густо, конечно, но в подавляющем большинстве случаев хватает и этого — преобразование осуществляется достаточно

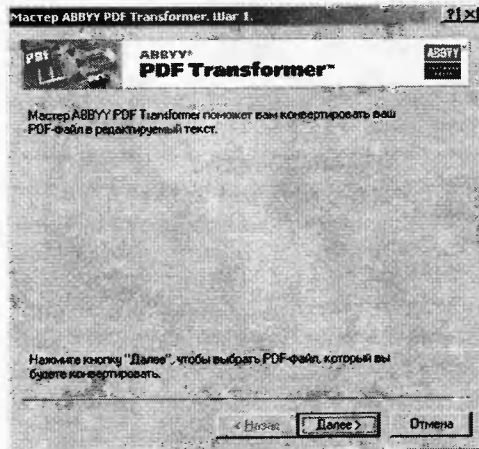
Рис. 2



Кроме того, PDF Transformer можно вызвать при помощи иконки на рабочем столе. При этом будет запущена специальная программа — "мастер", которая поможет выбрать и конвертировать нужный файл (рис. 3).

Из недостатков утилиты следует отметить невозможность сохранения гиперссылок, а также отсутствие инструментов для группового конвертирования нескольких файлов. Что касается собственно преобразования, то наибольшие проблемы PDF

Рис. 3



Transformer, видимо, испытывает с мелким, замысловато расположенным текстом, а также с некоторыми специфическими элементами оформления. Так, некоторые рекламные баннеры (содержащие большое коли-

чество текста) преобразуются в текст и т.п. В силу этого, естественно, не всегда в полной мере сохраняется исходное форматирование документа. Но в данном случае речь идет именно об отдельных казусах, вполне естественных для заданий высокого уровня сложности и полностью автоматического режима функционирования программы. Большинство предназначенных для распространения документов — научные статьи, пресс-релизы, техническая документация — как правило, лишены каких-либо дизайнерских изысков, и проблем с ними наверняка не возникнет. Одновременно нужно учитывать и то, что полученные в результате файлы подлежат дальнейшему редактированию, и большинство огрехов удастся быстро поправить. В этом, собствен-

но, и заключается весь смысл применения PDF Transformer.

Коробочный вариант программы ABBYY PDF Transformer 1.0 уже продается в нашей стране по цене 29 USD, что гораздо ниже, чем за рубежом. Испытательная версия продукта доступна на сайте www.abbyy.ru. Естественно, ее функциональность по сравнению с полной версией несколько ограничена. Она позволяет преобразовать только 15 PDF-документов, причем в многостраничных документах будут обработаны только первые 3 страницы, тогда как полнофункциональная версия программы конвертирует многостраничные документы объемом до 9999 страниц. После того как лимит документов будет исчерпан, а программа не будет зарегистрирована — испытательная версия перестанет работать.

Скинъ “маму” с платформы Socket A

А.ГОРЯЧКИН,

г.Кыштым Челябинской области.

E-mail: anatoliy_goryach@mail.ru

Платформа Socket A (Socket 462) обеспечивает работу систем на базе семейства процессоров седьмого поколения AMD — Duron, Athlon и Athlon XP. Толпа фанатов Intel может в гневе топтать ногами и кричать, но никто не сможет поспорить с тем, что Athlon XP — достойный конкурент Pentium 4 и обладает высокой производительностью при умеренной стоимости. Для платформы Socket A выпускаются чипсеты различных производителей, в том числе VIA, nVIDIA, AMD, SIS, ALi и ATI. На базе этих чипсетов на рынке имеется огромное множество материнских плат от производителей ABIT, Albatron, ASUS, Chaintech, DFI, EPoX, Gigabyte, Leadtek, Microstar, Soltek, EliteGroup, Biostar, ASRock, PCPartner и др. В этой статье сделан краткий обзор чипсетов для платформы Socket A и описано несколько материнских плат, построенных на этих чипсетах.

Чипсеты для платформы Socket A

На рынке чипсетов для платформы Socket A доминируют два производителя — тайваньская фирма VIA Technologies и американская фирма nVIDIA. Между ними ни на минуту не прекращается жесткая конкурентная

борьба. До недавних пор лидерство принадлежало фирме VIA — самому крупному поставщику наборов системной логики. VIA добилась популярности, прежде всего, производя быстрые и дешевые чипсеты для платформы Socket A. Но в последнее время ее позиции были серьезно потеснены nVIDIA.

Кроме VIA и nVIDIA, для платформы Socket A выпускают чипсеты фирмы SIS, ALi и ATI. Но чипсеты тайваньского производителя SIS не особенно популярны в сегменте Socket A, а фирма ALi, хотя и выпускает несколько чипсетов, но ее продукция не пользуется успехом у производителей материнских плат. Попытки канадской фирмы ATI предложить на рынке чипсетов свои разработки не дали желаемых результатов. По этой причине я не буду останавливаться на чипсетах SIS, ALi и ATI.

nVIDIA nForce2

Фирма nVIDIA удерживает прочные позиции не только на рынке чипсетов графических карт. Выпустив свой первый чипсет для платформы Socket A — nForce, она уверенно вошла на пьедестал, обойдя популярные чипсеты VIA, не говоря уж о прочих SIS, ALi и ATI. Чипсет nForce2 унаследовал все

лучшее от своего предшественника — набора системной логики nVIDIA nForce. В новом чипсете включена поддержка многих современных технологий, таких как AGP 8x, USB 2.0, ATA-133, Serial ATA, IEEE 1394, LAN, шестиканальный аудиокодек AC'97.

Чипсет nForce2 является самым производительным чипсетом для платформы Socket A. Какие же отличительные особенности позволили чипсету nForce2 уйти в отрыв от своих конкурентов?

Прежде всего, это высокая пропускная способность шины HyperTransport (800 Мб/с), связывающей северный и южный мосты. Надо особо отметить, что шина HyperTransport обладает большим потенциалом, именно ее все фирмы-производители применили в разработках своих чипсетов для процессоров восьмого поколения AMD Hammer. Шине HyperTransport, которая является собственной разработкой AMD, свойственны низкая латентность и гибкие характеристики. С использованием DDR-технологии эффективная тактовая частота может быть в пределах от 400 до 1600 МГц.

Во-вторых, технология Dual Channel Memory Architecture, являющаяся по своей сути эволюционным развитием технологии TwinBank и обеспечиваю-

щая двухканальный 128-битный доступ к шине памяти. Это позволяет увеличить теоретическую пропускную способность шины памяти до 6,4 Гб/с при использовании памяти PC3200, что, в свою очередь, существенно повышает производительность компьютера.

В третьих, этот чипсет унаследовал от своего предшественника запатентованную и улучшенную интеллектуальную систему предсказания и предвыборки данных, запрашиваемых центральным процессором — DASP (Dynamic Adaptive Speculative Pre-Processor).

Рис. 1



В четвертых, включена поддержка 400-мегагерцевой системной шины для процессоров Barton.

В роли северного моста чипсета nForce2 "выступает" чип nForce2 Ultra 400 (старое название северного моста — nForce2 SPP-128), а в роли южного — nForce2 MCP (Media and Communications Processor). Существуют модификации северного моста с одноканальным контроллером памяти — nForce2 400, с интегрированным графическим ядром nForce2 IGP, а также модификация южного моста с поддержкой встроенного сетевого контроллера и звукового процессора APU (Audio Processing Unit) — nForce2 MCP-T.

VIA KT600

На смену KT400, который вполне заслуженно пользуется очень широким спросом среди поклонников платформы Socket A, пришел очередной чипсет от VIA — KT600 (рис.1). На сегодняшний день KT600 — это прогрессивный и продвинутый чипсет для платформы Socket A. Как известно, KT400 поддерживает память DDR400 и процессоры Athlon XP с системной шиной 333 МГц. Новый чипсет обеспечивает поддержку 400-мегагерцевой системной шины и 400-мегагерцевой шины памяти. Это должно способствовать продвижению на рынок новейших процессоров AMD — Barton. Перевод на 400-мегагерцевую шину позволяет применять модули памяти PC3200 в

синхронном режиме, что является оптимальным с точки зрения быстродействия. В чипсете KT600 применен продвинутый контроллер памяти, поддерживающий технологию FastStream64. За счет использования расширенного массива буферов упреждающей выборки, в контроллере памяти уменьшается время ожидания выдачи данных.

Новый чипсет обеспечивает возможность использовать в качестве центрального процессора любые модели процессоров Socket A компании AMD, работающие на частоте системной шины 200, 266, 333 или 400 МГц. Как и в предыдущем чипсете, в KT600 взаимодействие между микросхемами северного и южного мостов осуществляется по 533-мегабайтной шине V-Link 8x. Кстати, напомним, что шина

V-Link Bus является собственной разработкой фирмы VIA. Южный мост чипсета реализован на микросхеме VT8237. На рис.2 показана структурная блок-схема чипсета KT600.

Системные платы для платформы Socket A

В моей домашней лаборатории были протестированы три материнские платы платформы Socket A — Gigabyte GA-7VT600 на чипсете VIA KT600, EPoX EP-8RDA3 и EP-8RDAE. Последние две были построены на базе чипсета nForce2. Внимание уделялось производительности, надежности, качеству изготовления, функциональным возможностям, установке драйверов и операционных систем и др.

Системная плата Gigabyte GA-7VT600

Материнская плата GA-7VT600 — очередная новинка от фирмы Gigabyte из серии K7 Triton (рис.3).

Рис. 2

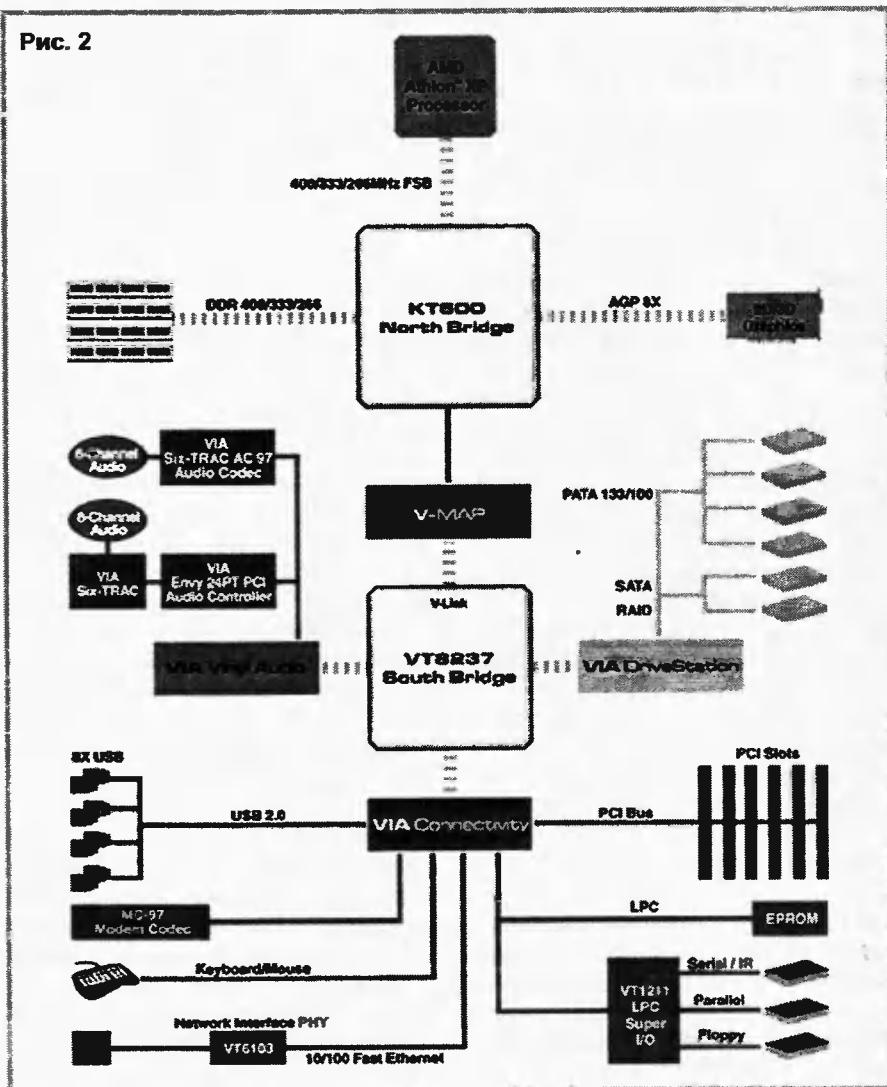
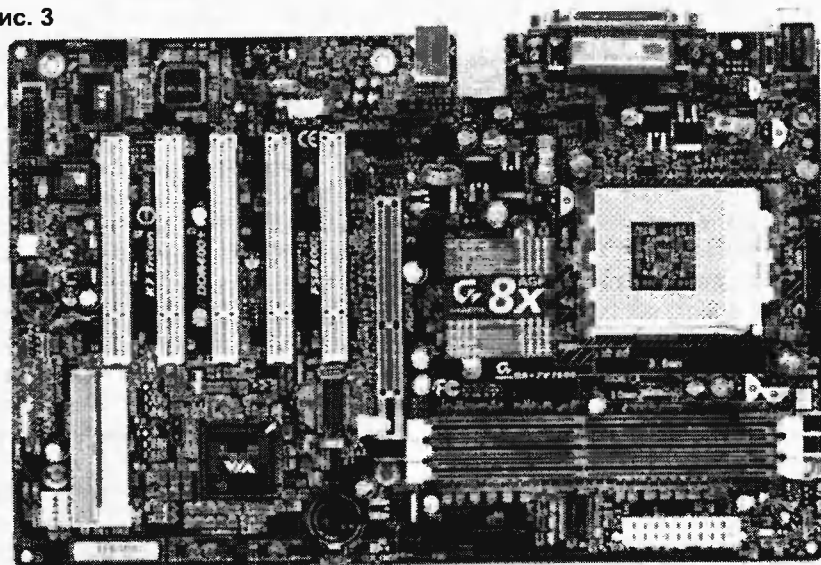


Рис. 3



Она предназначена для построения компьютерных систем на базе процессоров семейства AMD Athlon (Athlon Thunderbird/Duron/Athlon XP), выполненных в корпусе OPGA 462 Pin и устанавливаемых в процессорный разъем Socket A. Эта модель системной платы выполнена в форм-факторе ATX и имеет размеры 305x200 мм. В качестве базового набора системной логики при построении этой платы послужил чипсет VIA KT600.

Для установки модулей оперативной памяти, в качестве которой предусмотрено использование DIMM-модулей спецификации PC2100 (DDR266), PC2700 (DDR333) или PC3200 (DDR400), на плате оборудовано три DIMM-слота. Максимальный объем поддерживаемой оперативной памяти составляет 3 Гб. В случае использования модулей памяти PC3200, при работе шины памяти на частоте 200 МГц для обеспечения стабильной работы системы следует использовать модули памяти, рекомендуемые производителем материнской платы (эту информацию можно найти на официальном сайте компании Gigabyte Technology).

Микросхема северного моста чипсета охлаждается съемным металлическим радиатором. Контроллер графического порта северного моста, соответствующий требованиям спецификации AGP 3.0, позволяет использовать 1,5-вольтовые графические карты с интерфейсом AGP 4x/8x. Для их установки на плате имеется AGP-слот, который оборудован оригинальной защелкой, фиксирующей установленную в слот видеокарту. Для работы с IDE-устройствами используются возможно-

сти интегрированного в микросхеме южного моста двухканального IDE-контроллера. Таким образом, материнская плата Gigabyte GA-7VT600 позволяет подключать до четырех IDE-устройств с интерфейсом ATA 33/66/100/133 или ATAPI. Интегрированный в микросхеме южного моста USB-контроллер позволяет поддерживать работу шести портов USB 2.0, двум из которых нашлось место на выходной панели платы, а для подключения еще четырех портов на плате оборудованы два штырьковых разъема F_USB1 и F_USB2. Производитель не ограничился только поддержкой USB-интерфейса. В качестве микросхемы контроллера ввода-вывода использован чип ITE IT8705F, поддерживающий работу основных интерфейсов ввода-вывода и позволяющий осуществлять основные функции аппаратного мониторинга. Наличие шестиканального звукового AC'97-кодека Realtek ALC655 делает возможным воспроизведение звука формата 5.1. Для установки плат расширения, помимо уже упомянутого слота AGP, оборудовано пять 32-битных слотов PCI v2.2.

Меню установок Setup BIOS базовой системы ввода-вывода Award/Phoenix BIOS позволяет производить стандартные настройки работы компьютерных подсистем, а также осуществлять мониторинг напряжения питания в контрольных точках и температурного режима процессора. Есть возможность активизировать опцию звукового предупреждения в случае сбоя в работе процессорного и системного вентиляторов. Предусмотрено изменение частоты FSB от 100 до 250 МГц, при этом базовую частоту можно установить с

помощью dip-переключателя, имеющегося на плате. Для стабильной работы системы при разгоне в Setup BIOS имеется возможность изменять напряжение питания процессора (номинал, +5% или +10%), AGP- и DIMM-слотов (номинал, +0,1 В, +0,2 В или +0,3 В). Кроме того, в BIOS этой системной платы, как, впрочем, и во все ныне выпускаемые материнские платы компании Gigabyte, включена утилита Q-Flash, позволяющая в случае необходимости обновить BIOS, не загружая операционную систему. Здесь же будет уместным напомнить, что для изменения настроек работы с памятью в меню Setup BIOS материнских плат компании Gigabyte существует дополнительный пункт меню, вызов которого осуществляется комбинацией клавиш Ctrl+F1.

В комплект поставки системной платы Gigabyte GA-7VT600 входят: один 80-жильный IDE-шлейф, один 40-жильный IDE-шлейф, шлейф для подключения флоппи-дисков, планка расширения для подключения двух USB-портов, съемная металлическая панель для выходов платы (I/O Shield). Кроме того, к материнской плате прилагается компакт-диск со всеми необходимыми для работы драйверами и целым рядом полезных утилит. В их числе — фирменная утилита для разгона системы EasyTune4, утилита @BIOS, предоставляющая простой и удобный способ обновления версии BIOS из Windows, и утилита SIV (System Information Viewer), позволяющая отслеживать температуру процессора и материнской платы, а также системные напряжения.

К достоинствам платы можно отнести ее высокую производительность. В этом плане эта материнская плата заметно "пошумнее" плат Gigabyte на чипсетах VIA KT400 и KT333. Тестирование проводилось с помощью популярной тестово-диагностической программы SiSoftware Sandra 2002 Standard.

К сожалению, материнская плата Gigabyte GA-7VT600 не лишена некоторых недостатков. К ним можно отнести отсутствие фирменной технологии Gigabyte — Dual BIOS, позволяющей в случае повреждения основной микросхемы BIOS загрузить компьютер с помощью дублирующего чипа. Кроме того, микросхема BIOS впаяна своими выводами в плату, что делает ее замену в случае повреждения крайне затруднительной. Разъе-



мы IDE-контроллера можно было перенести в более удобное место, что позволило бы использовать полноразмерную плату расширения в пятом, последнем слоте шины PCI. Нет поддержки двухканального контроллера памяти и интерфейса Serial ATA. Не лишней была бы установка вентилятора на радиаторе северного моста. Поэтому установка дополнительного вентилятора в корпусе системного блока просто необходима для оптимальной работы системы охлаждения. Утилита SIV оставляет желать лучшего. Работает она не совсем корректно, а ее функциональные возможности ограничены. Создается впечатление неполноценности этой утилиты. Скачать полноценную версию SIV с сайта производителя не представляется возможным.

Пожалуй, самым существенным недостатком материнской платы GA-7VT600 является отсутствие правильной калибровки температурного датчика CPU. Это выражается в том, что уже при включении компьютера аппаратный мониторинг показывает температуру процессора от 50 до 80°C. В этом можно убедиться, контролируя в Setup BIOS в меню PC Health Status показания Hardware Monitoring. В реальности же перегрев процессора не происходит. Выходит, что температура процессора завышена градусов на 25...40 по сравнению с реальной. Подобный дефект имеет место и в материнских платах GA-7VT600-L. Посетив форум на русскоязычном сайте gigabyte.ru, можно заметить, что этот дефект проявляется у довольно широкого круга пользователей, приобретших именно эту плату. Можно, конечно, построить на базе этой материнки систему, но, не контролируя реальную температуру, есть шанс спалить процессор. Поэтому при покупке материнки надо обязательно проверять ее на предмет температуры нагрева процессора. Подобные проколы непростительны для Gigabyte и подрывают доверие пользователей к этой фирме. Становится непонятно, что это — временный спад или начало заката фирмы.

Учитывая все вышеперечисленные недостатки, приобретение этой платы целесообразно лишь для тех, кто любит сам создавать себе проблемы и портить свои нервы. Фанатам Gigabyte и VIA лучше остановить свой выбор на платах, построенных на проверенных временем чипсетах VIA

KT 400, KT 400A или, в крайнем случае, на "старичке" KT333.

Системная плата EPoX EP-8RDA3

Эта материнская плата построена на чипсете nForce2 (nForce2 Ultra 400 + MCP) от небезызвестной американской фирмы nVIDIA, которая великолепно дебютировала на новом для себя рынке чипсетов (рис.4). Контроллер графического порта чипа nForce2 SPP полностью соответствует требованиям спецификации AGP 3.0, что обеспечивает возможность поддержки интерфейса AGP 8x. Как и на всех платах, выполненных на базе чипсета nForce2 Ultra 400, ключевой "фишкой" является поддержка двухканального контроллера оперативной памяти, что в случае использования двух

описании к этой плате говорится о том, что мощность блока питания должна составлять не менее 250 Вт, а с учетом подключения дополнительных карт расширения и периферии — не менее 300 Вт. Интересной особенностью этой платы является наличие дополнительного 4-пинового разъема питания, размещенного рядом с основным 20-пиновым. К числу дополнительных особенностей материнской платы можно отнести световую индикацию кодов ошибок, выполненную в виде двух 7-сегментных светодиодов. Микросхема BIOS не впаяна выводами в плату, а установлена, как говорят специалисты, в "кровать". Это означает, что чип BIOS в случае необходимости замены может быть извлечен без проблем. Дополнительно к

Рис. 4



или трех модулей DIMM позволяет существенно увеличить производительность системы. Двухканальный контроллер памяти поддерживает три 184-пиновых модуля памяти DDR200/266/333/400 с общим объемом до 3 Гб.

Материнская плата EPoX EP-8RDA3 имеет полноразмерные габариты форм-фактора ATX — 305x245 мм. В

этом, в руководстве для пользователя наглядно показано, с помощью какого инструмента должна быть извлечена микросхема BIOS.

По своей сути материнская плата EP-8RDA3 является базовой моделью для целой серии плат, отличающихся между собой наличием встроенных дополнительных контролле-

Табл. 1

Модель	Debug LED	USB2.0	P-ATA IDE	S-ATA IDE	RAID	IEEE1394	LAN
8RDA3	+	6	2	-	-	-	-
8RDA3I	+	6	2	-	-	-	100MbE
8RDA3G	+	6	2	-	-	+	Dual 100MbE
8RDA3+	+	6	2	2	S-ATA	+	Dual 100MbE

ров. Технические характеристики материнских плат серии EP-8RDA3 приведены в табл. 1.

Материнская плата EP-8RDA3 — настоящая находка для любителей overclocking'a. В Setup BIOS на вкладке Advanced Chipset Features возможно изменять множитель частоты CPU, частоты системной шины, оперативной памяти и шины AGP, причем независимо друг от друга. Нет только возможности регулировать частоту шины PCI. Для стабильной работы "разогнанной" системы понадобится регулировка питающих напряжений CPU, AGP, DIMM и чипсета (VDD). Все эти опции, на радость оверклокерам, присутствуют в BIOS на вкладке Power BIOS Features.

Традиционно материнская плата комплектуется компакт-диском с набором драйверов и утилит. Из числа последних весьма полезной, на мой взгляд, является утилита USDM (Unified System Diagnostic Manager). Этот системный менеджер обладает более широкими функциональными возможностями в сравнении с вышеупомянутой утилитой SIV (System Information Viewer) от Gigabyte. Например, функциональные возможности утилиты позволяют вести журнал в текстовом формате. В него записываются показания датчиков температуры, величины системных напряжений и скорость вращения вентиляторов. Предусмотрена подача звукового сигнала тревоги в случае отклонения заданных параметров от нормы. Но и это еще не все. При достижении критической температуры процессора, например в случае остановки вентилятора, менеджер выполнит выключение компьютера и тем самым спасет CPU от перегрева и сгорания (рис. 5).

Несколько слов о комплектации платы. Форма ее поставки — retail. В красочной коробке, кроме системной платы, есть все необходимое — два шлейфа ATA133, кабель для FDD, планка для Game- и Com-портов, щиток, закрывающий разъемы платы (I/O Shield). Коробка оформлена стильно — на ней изображена фигура какого-то хищного

Табл. 2

Модель	Debug LED	AC3 encoder	USB2.0	LAN
8RDAE	+	+	6	-
8RDAEN	+	+	6	Dual

зверя в прыжке. Скелет этого зверя отчетливо виден, как под рентгеном. В этом же стиле оформлено и руководство для пользователя. Наверно, таким образом фирма EPoX решила запугать своих конкурентов.

Системная плата EPoX EP-8RDAE

Как и EP-8RDA3, материнская плата EP-8RDAE также является базовой моделью для серии плат, отличающихся между собой наличием встроенных дополнительных контроллеров. Технические характеристики материнских плат серии EP-8RDAE приведены в табл. 2.

Ключевое отличие системной платы EPoX EP-8RDAE от предыдущей заключается в том, что она построена на чипсете nVIDIA nForce2 400. Как говорилось выше, чипсет nForce2 400 имеет один 64-битный канал для взаимодействия с шиной памяти. Это, конечно, отразилось на ее стоимости. По сравнению с EP-8RDA3, она стоит дешевле на 400 — 500 руб. Учитывая сравнительно невысокую ее цену, установка платы позволит создать современную, высокопроиз-

водительную и надежную систему, не заботясь о скором ее апгрейде. Эту плату можно порекомендовать тем

пользователям, которые желают приобрести плату с наилучшим соотношением цена — производительность.

Общие рекомендации

Во всех материнских платах интегрирован шестиканальный аудиоконтроллер AC'97. В отличие от Gigabyte, которая отдает предпочтение кодекам ALC, производства фирмы Realtek, в платах EPoX установлены чипы C-Media CMI 9739A, оснащенные неплохим табличным синтезатором. По субъективным оценкам, озвучивание системных событий кодеком ALC655 происходит с некоторой долей задержки, что в какой-то степени умаляет достоинства этого кодака.

В плане надежности на материнской плате следует обращать внимание на микросхему BIOS. Если она установлена в панельке, а не припаяна выводами к плате, выбирайте именно эту плату.

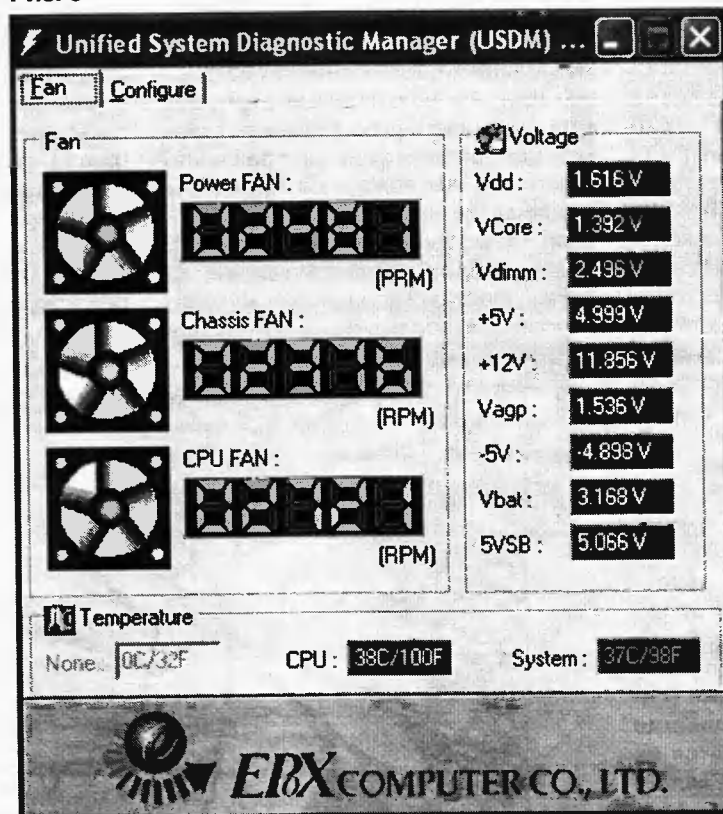
При покупке системной платы нелишним будет обратить внимание на ревизию материнской платы. В более поздних ревизиях, как правило, устраняются старые ошибки и недоработки плат.

При выборе материнской платы для платформы Socket A приобретать лучше ту, которая поддерживает современные технологии: DDR400 PC3200; FSB 400 МГц; AGP 8x; USB 2.0; ATA-133. Это позволяет избежать морального устаревания платы в течение продолжительного времени и отдалает срок неизбежного в дальнейшем апгрейда.

Закключение

Материнские платы, построенные на базе чипсетов с двухканальным контроллером памяти, предпочтительнее с точки зрения производительности. Однако, стоимость подобных плат существенно выше. Компьютерным энтузиастам, которым нужна максимальная производительность и которые готовы

Рис. 5





платить за нее, можно рекомендовать материнскую плату EPoX EP-8RDA3 или более "навороченную" модель — EPoX EP-8RDA3+.

В плане надежности материнские платы EPoX занимают более высокую ступень по сравнению с платами Gigabyte. Это мнение основывается на личном опыте, и мою точку зрения разделяют специалисты фирм, специализирующихся на ремонте компьютерной техники.

Технические характеристики материнских плат приведены в табл.3.

И в заключение можно сказать следующее. Все вышеописанные платы можно отнести к платам среднего класса. Как правило, такие платы оснащены большим количеством разъемов и слотов, достаточно укомплектованы дополнительными контроллерами. BIOS'ы этих плат имеют гибкую и оптимальную систему настроек. Возможности этих плат позволяют построить на их базе мощный игровой, мультимедийный или домашний компьютер, по "плечу" которому современные игры, трех-

Модель	GA-7VT600	EP-8RDA3	EP-8RDAE
Производитель	Gigabyte	EPoX	EPoX
Чипсет	KT600	nForce2	nForce2
Северный мост	KT600	nForce2 Ultra 400	nForce2 400
Южный мост	VT8237	nForce2 MCP	nForce2 MCP
Связь между мостами	V-Link 8X 533 Mb/sec	HyperTransport 800 Mb/sec	HyperTransport 800 Mb/sec
FSB	266/333/400 MHz	266/333/400 MHz	266/333/400 MHz
RAM	DDR266/333/400	DDR266/333/400	DDR266/333/400
BIOS	AWARD	AWARD	AWARD
AGP	4x/8x	4x/8x	4x/8x
PCI	5	5	5
USB 2.0	6	6	6
P-ATA	2 ATA-133	2 ATA-133	2 ATA-133
RAID	нет	нет	нет
Serial ATA	нет	нет	нет
Fire Wire IEEE 1394	нет	нет	нет
Audio	AC'97 Realtek ALC655	AC'97 CM19739A	AC'97 CM19739A
LAN	нет	нет	нет
Form Factor	305x200 mm	305x245 mm	305x245 mm
Комплектация	OEM	Retail	Retail

мерное моделирование, обработка видео и звука. Отсутствие на этих платах дополнительных интегрированных контроллеров FireWire IEEE-1394, Serial

ATA, RAID, LAN, конечно, снижает их функциональность, но, по большому счету, они вряд ли пригодятся рядовому пользователю.

С.РЮМИК,

г.Чернигов.

Интернет-калейдоскоп, freeware #19

Поверил я алгеброй гармонию.
А.С.Пушкин, "Моцарт и Сальери"

Мы живем в мире цифр, и любое физическое событие воспринимаем через количественные показатели. Виртуальные явления, такие как Интернет, тоже "оцифровываются". Достаточно посетить на портале Rambler страницу "Руметрика" (<http://rumetrika.rambler.ru/proj/about.html>), чтобы ощутить на цифрах и графиках пульс жизни в русскоязычной части Сети. Инструменты для измерения активности Рунета — это поисковые машины и счетчики-рейтинги. Конечно, абсолютные цифры, которые они предоставляют, дают лишь приблизительную оценку, но по ним можно проследить динамику роста Рунета, оценить активность пользователей и их интересы.

"Вселенная фракталов"

Именно так называется сайт Новосибирского государственного университета (НГУ), посвященный удивительным математическим конструкциям — трехмерным самоподобным фракталам (<http://fractals.nsu.ru/fractals.chat.ru/>). Фракталы легче всего представить как геометрические объекты, в которых содержатся их уменьшенные изображения. Сколько фрактал ни увеличивай, из любой его

части на вас будет смотреть маленькая копия, почти как в голографии.

Термин "фрактал" (лат. "fractus" — дробный) изобрел 30 лет назад Бенуа Мандельброт, математик из фирмы IBM. Именно компьютер помог ему рассчитать и по точкам построить эти графические изображения. Многие из них похожи на то, что мы встречаем в природе: снежинки, морской конек, деревья, цветы (рис.1), разряд молнии, горные массивы.

Различают геометрические, алгебраические, стохастические фракталы. Для

построения первых из них используют программы, например IFS Builder 3d (<http://fractals.nsu.ru/bin/ifs3d-1.6-pre1-win.zip>, 355 Кб, НГУ), Fractal Builder (http://fractals.nsu.ru/fractals.chat.ru/fb_v09alpha.zip, 228 Кб, НГУ). Фракталы в них приобретают объемность (рис.2), стереопанораму и даже мультипликацию (<http://fractals.nsu.ru/films/>).

Сейчас по всему миру стало модным проводить выставки фрактальной живописи. В Интернете размещены галереи "цифрового искусства" (<http://www.geocities.com/ffgnl>, <http://www.fractalus>.

Рис. 1

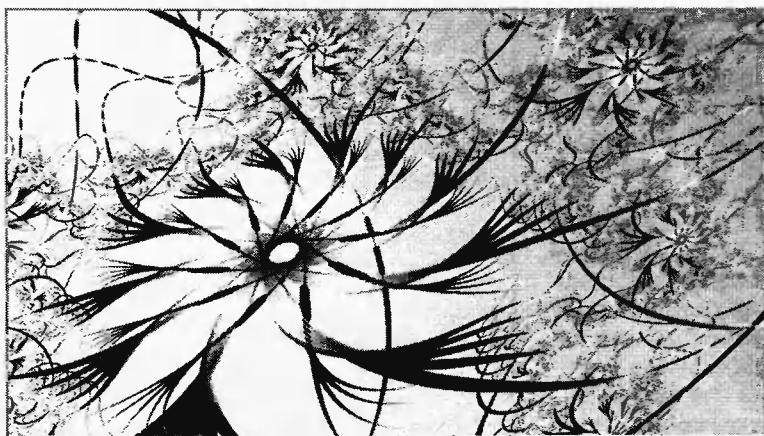


Рис. 2



com/sylvie, <http://www.donarcher.com>). Фрактальные модели применяются в компьютерных играх для создания обстановки, которую трудно отличить от реальной. И даже в экономических прогнозах используют модели нейронных сетей со структурными графами из самоподобных фракталов.

Macintosh vs. IBM PC

Если расшифровать стандартное английское сокращение "vs." как "versus" (против), то станет понятным, что на "ринг" вызываются два крупнейших бренда компьютерной техники. Так сложилось, что в странах СНГ компьютеры "Apple Macintosh" (от англ. McIntosh — сорт яблок) в численном отношении значительно отстают от IBM PC. Но те из счастливиц, которые поработали на графических MAC-станциях, долгое время не перестают восхищаться дружелюбностью интерфейса и скоростью прорисовки изображений.

Почувствовать себя хоть на минуту в роскошном кресле у престижного "Макинтоша" поможет программа iTunes. Это примерный аналог проигрывателя звуковых файлов WinAmp. Перечень его функций на русском языке приведен на странице <http://www.apple.ru/itunes/jukebox.html>. Фирма Apple на сайте <http://www.apple.com/itunes/download/> предусмотрительно разместила для бесплатного скачивания две версии iTunes — под MacOS и под Windows 2000/XP (21,3 Мб, рис.3).

Налицо хорошая возможность сравнения методологии проектирования интерфейсов программ для различных компьютерных систем. Мир Apple исповедует свою оригинальную философию, в ней есть что позаимствовать и чему поучиться.

Нумерология, цифрология, гороскопы, биоритмы

Можно ли по расположению небесных светил или по цифрам даты рождения предсказать наперед судьбу человека?

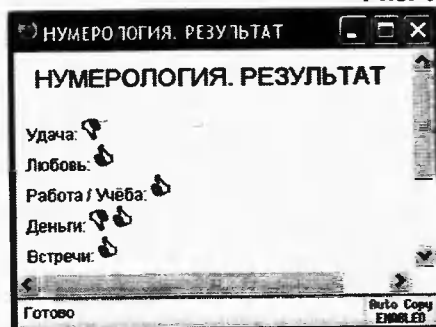
Рис. 3



Тех, кто твердо верит в астрологию, в обратном не переубедить. Однако лучший советник — это практика. Распаковав текстовый скрипт http://download.ware.ru:8080/win/5421_numerology-free.zip (1 Кб), переименовав его в numerology.html и заполнив цифры дат, можно прямо из Интернета получить нумерологическое послание из "будущего" (рис.4).

Если эксперимент проводить по-настоящему чисто и проверять совпадение только за прошедшее время, не зная прогноза наперед, то подтвердится универсальная формула теории вероятности "50% на 50%". Другое дело — прогноз на будущее, когда испытуемому сообщается (точнее, исподволь внушается) информация о предстоя-

Рис. 4



щих плюсах и минусах жизни. Психика человека мобилизуется и подсознательно настраивается на избирательное восприятие действительности. Процент совпадения прогнозов за длительный промежуток времени в этом случае будет выше. Специалисты говорят: "Внушение, чтобы быть действенным, должно сопровождаться самовну-

шением при наличии внешних раздражающих факторов".

Примерно такие же сеансы информационной терапии происходят при чтении гороскопов (<http://valsistem.nm.ru/dl/researcher.exe>, 280 Кб, автор — В.Дожиков, <http://www.team9a.web.ur.ru/Horoscope.rar>, 48 Кб, автор — Е.Банников) и при изучении графиков биоритмов (http://andrei512.narod.ru/programs/Bioritm_09.2004.zip, 301 Кб, автор — Andrei, рис.5).

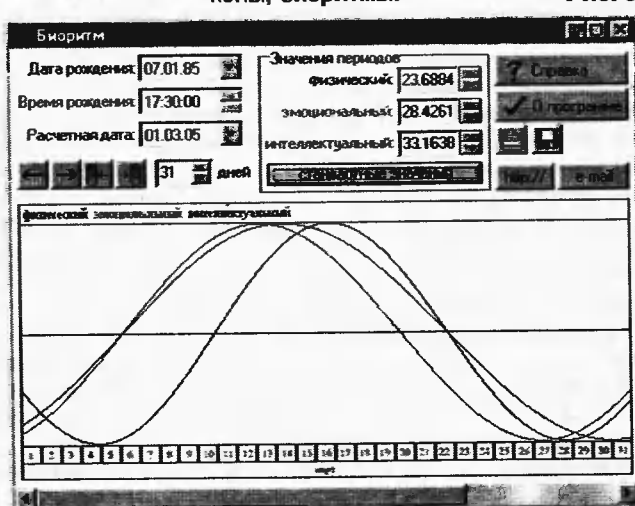
Это не значит, что гороскопы надо перелистывать, не читая. Напротив, в них содержится отличный инструктаж по безопасному способу жизни (<http://horo.mail.ru>, <http://horoscopes.rambler.ru>). Гороскопы для всех знаков Зодиака учат только хорошему, в частности, будь осмотрительным, не торопись с выводами, экономь деньги, не поддавайся на авантюры, налаживай отношения с партнерами, не ссорься с близкими.

По части биоритмов — никто не отрицает их существование. Проблема в том, что при сильных стрессах (например, болезни) циклы могут растягиваться или сжиматься во времени. С накоплением "задержек" изменяется начальная фаза биоритмов, превращаясь из предсказуемой величины в случайную. Чтобы выявить реальное состояние дел, надо пару месяцев ежедневно вести дневник наблюдений и по точкам "самочувствия" нарисовать графики физического, эмоционального и интеллектуального уровня. Только после этого можно прогнозировать ближайшее будущее.

Программисты могли бы автоматизировать этот процесс и в программы расчетов биоритмов добавить функцию ручного ввода чисел "самооценки" с дальнейшей экстраполяцией графиков и привязкой начальной фазы к реальному состоянию здоровья.

Ключевые слова для поиска: RUметрия, фракталы, iTunes, гороскопы, биоритмы.

Рис. 5





ЛИСТАЯ СТРАНИЦЫ

Оценке расстановки сил в стане браузеров посвящена статья А. Евдокимова "Вторая браузерная война" (Подводная лодка, №2/2005, С.94...97).

В середине 90-х годов MS Internet Explorer и Netscape Navigator (два лучших на тот момент браузеров) столь отчаянно боролись за первенство, что позднее с чьей-то легкой руки это состязание назвали "браузерной" войной. Тогда благодаря грамотной маркетинговой политике Microsoft Internet Explorer на долгие годы стал ведущим браузером, каковым он остается и сейчас. Однако радость победы испортила "третья" сила — браузер с музыкальным названием Opera. Долгое время именно эта программа считалась главным конкурентом IE, но настоящий вызов микрософтовскому гиганту бросило, положив начало второй браузерной войне, open source-приложение Mozilla Firefox.

Требования, которые пользователи обычно предъявляют к браузерам, достаточно просты. Они обязаны быстро работать, одновременно быть удобными и надежными в экс-

плуатации. То есть должны уметь открывать с максимальной скоростью даже самые сложные сайты и не занимать при этом слишком много системных ресурсов. Критерии для определения лучших браузеров — максимум поддерживаемых сетевых и почтовых протоколов, а также языков, скриптов и прочих необходимых интернет-технологий (Java, Flash, RSS и т.п.) при минимуме запросов к мощи и быстродействию компьютера. Более экономичными и удобными считаются программы для веб-серфинга с многооконным MDI-интерфейсом.

Качественный браузер обычно поддерживает внешние модули сторонних разработчиков, включая мультимедийные плееры, обеспечивающие прием онлайн-телевидения и радио — Microsoft Media Player, Real

Player и Apple Quick Time. При оценке программ для веб-серфинга также учитывается их способность противодействовать всплывающим рекламным окнам. Помимо функций самих браузеров, при тестировании нужно сравнивать также возможности утилит, входящих в их состав менеджеров загрузки, почтовиков и т.п.

При тестировании специального задания у конкурирующих систем не было — с их помощью открывались насыщенные графикой и разнообразными скриптами сайты, и оценивалось, какая из программ загружает их быстрее, а какая — медленнее. При этом учитывался объем оперативной памяти, занимаемой каждым браузером.

Совокупные оценки тестирования в баллах представлены в таблице.

	Скорость веб-серфинга	Системные требования	Поддержка сетевых технологий	Возможность подключения плагинов, макросов и надстроек	Удобство интерфейса	Итого
Internet Explorer 6.0 SP1	3	3	5	5	3	19
Mozilla Firefox 1.0	5	4	4	4	5	22
Opera 8 beta	5	4	4	4	4	21
K-Meleon 0.9 beta 4	5	5	4	3	4	21
Netscape 7.2/Mozilla 1.7.5	4	3	4	4	4	19



Обзор программ фильтрации спама сделал О. Татарников в статье "Антиспамовые фильтры" (КомпьютерПресс, №2/2005, С.94...95).

Все, кто пользуется Интернетом, знакомы с таким явлением как спам — массовыми неадресными рекламными рассылками по электронной почте. Некоторые ученые даже считают, что компьютерные вирусы и спам приведут к разрушению Интернета.

По данным аналитических организаций, более двух третей всех сообщений, отсылаемых по e-mail, приходится именно на долю "электронного мусора". Как это ни печально, эффективных средств борьбы со спамом очень мало. Причем разного рода списки, фильтры и другие подобные службы иногда сами закрываются по вине спамеров. Решения, которые в той или иной мере могут помочь снять проблему спама, можно условно разделить на несколько групп:

1. Простейшие способы ручной или автоматической фильтрации почты по заголовкам. В принципе, любой пользователь может перейти с протокола POP 3 на IMAP 4 или на Web-интерфейс и оценивать письма только по их заголовкам, не получая текста. Во многих почтовых программах можно настроить автоматическую фильтрацию по заголовкам писем. Однако в последнем слу-

чае требуется очень тонкая и вдумчивая подгонка условий оценки, и можно получить много нареканий от своих "нерасторопных" корреспондентов по сбоям в работе такого фильтра.

2. Самыми надежными являются специальные службы фильтрации, которые могут находиться у почтового провайдера или на отдельном сервере (последние, как правило, платные). В некоторых случаях вся почта отправляется на определенный адрес, где фильтруется, и к пользователю приходит уже "чистой". Этот метод — самый простой для пользователя, но, как правило, и наименее контролируемый (то есть велика вероятность, что может потеряться часть полезной корреспонденции, о чем никто никогда не узнает).

3. Можно применять входные фильтры, основанные на анализе IP-адреса хоста, передающего спам (который можно узнать, например, "по отзывам пострадавших"), и использовать общие базы данных с адресами таких спамеров (DNSBL — DNS Black Lists). Однако сегодня это уже малозффективный способ борьбы с современными методами рассылки спама.

4. Существует фильтрация на основе автоматического пополнения access-листа адресами спамеров. Например, при такой фильтрации может

использоваться встречный анализ подозрительности отправляющего хоста, однако данный фильтр плох тем, что требует постоянного контроля и тонкой настройки. Причем первое письмо от потенциального спамера он в любом случае пропустит, что делает его работу малозффективной в современных условиях.

5. И наконец, существуют программы или встраиваемые модули для анализа содержимого письма. Программы для такой проверки (их может быть несколько) принимают информацию от почтовой программы, а возвращают, как правило, свою оценку и рекомендацию к дальнейшему действию.

Среди программ, предназначенных для борьбы со спамом, особенно интересны те, что работают по принципам Байеса и самообучаются в процессе анализа корреспонденции. Данная технология отличается использованием байесовских принципов для распознавания спама по образцу, моделирование которого происходит благодаря анализу самого спама. Однако простота применения байесовских принципов обманчива, так как отнесение письма к спаму производится по сложным алгоритмам выявления общих элементов в реальных посланиях. Таким образом, чем большее количество спама подверглось анали-



зу, тем лучше работает фильтр. Кроме того, метод Байеса обладает автокоррекцией, поскольку в случае изменения структуры писем фильтр изменится автоматически.

Программы Spamassassin (<http://www.spamassassin.org>) используют для защиты от спама пользователи UNIX-систем (Windows-версия — <http://wiki.apache.org/spamassassin/UsingOnWindows>). Она имеет развитую систему контекстного лингвистического анализа, обучения и самообучения. Кроме того, она может работать совместно с другими программами анализа корреспонденции. Система начинает действовать сразу с заранее предустановленными параметрами и вылавливает спам с вероятностью 60—70%. Для того чтобы увеличить этот процент, необходимо ее обучить — дать ей примеры “белых” (ham) и спамовых (spam) писем. После полного обучения системы появляется оценка “BAYES”. Любое письмо после запуска системы содержит в заголовке дополнительную информацию по анализу его на предмет спама — что-то типа ****SPAM****.

BayesIt! (<http://klirik.narod.ru> или <http://www.riftlabs.com>) — это антиспамский фильтр для почтовой программы The Bat!, реализующий технологию статистической фильтрации писем при помощи алгоритмов Байеса. BayesIt! позволяет отфильтровывать нежелательную корреспонденцию только средствами The Bat!. BayesIt! не может работать с письмами на удаленном почтовом сервере по принципу получения только выборочной корреспонденции. Он ничего “не знает” о почтовых протоколах и тонкостях доставки почты — он должен получить от The Bat! текст письма и может возратить ему лишь оценку “нежелательности” в шкале от 0 до 100. Фильтр не может изменять переданное ему письмо или добавлять в него какие-нибудь признаки типа фразы “SPAM” в заголовке. Сортировка корреспонденции — это забота почтовой программы. На основании полученной от фильтра оценки The Bat! может либо переместить письмо в папку Junk folder (“Макулатура”), либо просто удалить его. Потом письма в папке Junk Mail можно как-то обработать, но только вручную. К сожалению, никакие дополнительные средства фильтрования с этой папкой не работают.

Anti-Spam Filter (ASF) (<http://rus.spamliquidator.com>) — это российско-канадская система борьбы со спамом, основанная на составлении и проверке спамерских баз. Отличительной осо-

бенностью такого подхода является то, что при довольно высоком уровне фильтрации гарантируется полная сохранность полезной переписки. Фильтры системы постоянно обновляются, оперативно реагируя на изменения технологий спамеров. При этом в процессе создания фильтров и пополнения антиспамерской базы самое непосредственное участие принимают пользователи программы, подавая жалобы на спамеров (процесс подачи жалоб автоматизирован). Anti-Spam Filter отправляет жалобу на сервер программы, а если установлен флажок “Добавлять в личный черный список при отправке жалобы”, то вместе с отправкой жалобы спамер заносится в личный блок-лист пользователя. На основании жалоб формируются правила фильтрации, что происходит на сервере автоматически в режиме реального времени, однако существуют и сложные фильтры, которые специально создаются разработчиками. Подача жалоб поощряется — в качестве награды за ловлю спамеров предусмотрено бесплатное продление лицензии. Однако не всякий, на кого пожаловались, немедленно становится спамером — письма от него не будут блокироваться ASF, пока его спамерство не подтвердится. Так что если вы случайно или преднамеренно подали жалобу на своего приятеля, то ничего не произойдет (если, конечно, одновременно с вами он не успел досадить еще нескольким десяткам пользователей ASF). В данный момент доступна версия для работы в операционной системе Windows 98/Me/NT/2K/XP. Система оптимизирована для работы с почтовым клиентом Microsoft Outlook Express, однако успешно работает и с другими клиентами (Outlook 2000, TheBat!, Mozilla и т.д.). На данный момент поддерживается только протокол POP3, поддержка IMAP и APOP планируется в следующих версиях.

Mailbox cleaner (<http://www.realsofts.com>) — программный инструмент для борьбы со спамом фирмы Realsoft. Mailbox cleaner позволяет пользователям осуществлять фильтрацию входящей электронной корреспонденции непосредственно на сервере, то есть еще до ее загрузки в почтовый ящик. Mailbox cleaner реализует два режима — ручной и автоматический. Кроме того, Realsoft делает акцент на таких чертах своего продукта как простота и удобство. Незарегистрированная версия имеет ограничения по количеству удаляемых за один сеанс писем и останавливает работу сканера через произвольные промежутки времени. Что же касается стоимости регистрации, то

для пользователей из СНГ она составляет 150 руб.

SpamPal (<http://www.spampal.net>) — в основе программы лежит поиск отправителя письма в “черных” списках DNSBL, содержащих серверы, наиболее часто используемые спамерами. Алгоритм работы простой: почтовый клиент обращается к SpamPal, тот, в свою очередь, скачивает с почтового сервера пользователя заголовки писем и извлекает из них IP-адрес отправителя. Далее SpamPal обращается к DNSBL. Если адреса отправителя письма в спамерской базе нет, то корреспонденция без препятствий поступает к пользователю — в противном случае письмо помещается особой меткой. Избавиться от меченых писем можно при помощи фильтра в почтовой программе, который будет перемещать их в специальную папку или удалять. Если SpamPal допустил ошибку, и нужное письмо попало в папку для спама, то пользователь всегда может занести его в личный “белый список”. После этого проблем с данным отправителем уже не возникнет. Кроме того, SpamPal автоматически обучается, и адресаты, с которыми вы состоите в переписке, добавляются в список благонадежных. В качестве DNSBL рекомендуется использовать наиболее авторитетные базы SpamCop и ORDB.

Программа Spam Buster (<http://www.contactplus.com>) может отслеживать проникновение спама в почтовый ящик пользователя как в полуавтоматическом режиме, так и полностью автоматически (в последнем случае через определенные промежутки времени проверяется корреспонденция, входящая на почтовом сервере, и нежелательные письма уничтожаются). Spam Buster работает на основе мощного набора фильтров, в соответствии с которыми по заголовкам писем на сервере программа определяет, какие из них являются спамом. Фильтры программы могут работать с заголовком, темой, отправителем сообщения и его размером (например, можно задать такое правило: считать спамом все письма больше 500 Кб). В Spam Buster также можно настроить “белый” и “черный” списки адресов. Кстати, сразу после установки программы “черный” список уже содержит свыше 18 тыс. адресов. В дополнение к этому арсеналу пользователи без труда могут строить и собственные фильтры. К сожалению, русский язык в Spam Buster не поддерживается.

NoSpam (<http://igp.org.ua>) — эта разработка украинской компании IGP Systems пока не интегрируется ни в одну из почтовых программ и функци-



онирует полностью автономно: самостоятельно соединяется с почтовым сервером и, если находит письма, подходящие под категорию спама, то безжалостно удаляет их на месте. "Зачистка" осуществляется с указанным интервалом. NoSpam сама фильтрует

письма и по адресам отправителя, и по серверам, с которых присланы сообщения, и по тематике (заголовкам). В установках предусмотрены исключения, а также возможность автоматического обновления. Причем все настройки сохраняются в ini-файлы, так что конфигура-

цию программы несложно переносить с одной машины на другую. В принципе, при определенных усилиях со стороны пользователя из NoSpam может получиться хороший инструмент по борьбе с непрошенными письмами. Программа распространяется бесплатно.



Новейшие 64-битные процессоры производства AMD и Intel теперь в состоянии предотвратить исполнение разрушительного кода, внедренного на компьютер, сообщается в статье "Доступ запрещен" (CHIP, №2/2005, С.18...19).

После выпуска Service Pack 2 ОС Windows XP обрела наконец-то поддержку новейшей технологии антивирусной защиты.

Иметь надежную защиту от вирусов без установки специальных программ, их постоянного обновления, скачивания антивирусных баз и патчей — давняя мечта пользователей, уставших от борьбы со зловредными бациллами, разрушающими биты и байты компьютеров и уничтожающими ценную информацию. Первый шаг в этом направлении был сделан при появлении Service Pack 2 для Windows XP с модулем, названным Data Execution Prevention. Суть этой технологии состоит в аппаратном обеспечении антивирусной защиты, которая заключается в перехватывании зловредного кода и сведении к минимуму опасности умышленного переполнения буфера (Buffer Overflow). Разработанный корпорацией Microsoft для ОС Windows XP модуль предотвращения выполнения данных (Data Execution Prevention) эффективно работает

лишь при наличии 64-битного процессора. Он имеет флаг под названием No Execute (NX) — один бит в процессорном регистре, который обозначает, что находящимся в отдельных зонах памяти данным исполняться запрещено. Новая технология устраняет главный недостаток системы управления памятью процессоров с архитектурой фон Нойманна — память для данных и исполняемого программного кода является общей. Это явилось предпосылкой для работы "умных" программ, которые могли бы самостоятельно записываться и перезаписываться. Многие из широко распространенных вирусов и червей, таких как Sasser и Blaster, используют для атак дыры в программном обеспечении, чтобы спровоцировать переполнение буфера. Во время этой атаки нормальное содержимое стека заменяется адресом возврата, который указывает на вредоносный код. Этот код — он чаще всего располагается в этом же сегменте памяти — автоматически исполняется, что приводит к коллапсу компьютера.

Как раз это и предотвращает функция под названием No Execute. Конечно, злоумышленник может, как и прежде, внедрять код в память стека, однако как только этот код начнет исполняться, программа аварийно завер-

шится, выдав сообщение об ошибке, и система не пострадает.

В 64-битных версиях ОС Windows XP после инсталляции Windows XP Service Pack 2 автоматически будет активирована функция Execution Protection, и отключить ее пользователь не сможет. Это будет выглядеть несколько иначе, когда ОС работает в 32-битном режиме. В этом случае активируется расширение Physical Address Extension (PAE), которое эмулирует расширение адресации до тех же 64 бит, в которых присутствует NX-бит.

Microsoft сама указывает на то, что многие драйверы устройств имеют проблемы с используемой в серверных окружениях Physical Address Extension. Они просто могут не запускаться в 64-битном режиме. Если обновление драйвера не помогает, пользователю ничего не остается, кроме как отключить Execution Prevention для неработающего приложения.

Однако сама по себе функция защиты NX еще долгое время не сможет дать гарантию от заражения компьютера вирусами. Ведь злоумышленник может получить доступ в область памяти ПК не только через Buffer Overflow. К вирусам, прикрепленным к почтовым отправлением либо загружающимся при посещении веб-сайтов, NX совершенно "равнодушен".



Отчет о выборочном тестировании совместимости DVD-болванок с различными DVD/RW-приводами приведен в статье "Лучшие приводы и болванки" (CHIP, №2/2005, С.40...47).

Несмотря на широкое распространение DVD-рекордеров и болванок, процент брака при записи DVD в домашних условиях все еще слишком велик. Одна из главных причин этого неприятного явления — плохая совместимость используемых приводов и болванок. В отличие от CD-рекордеров, DVD-рекордеры пока еще не столь надежны и неприхотливы. Запись DVD-болванок напоминает игру в рулетку, и пользователь не может предсказать точно, повезет ему в очередной раз или нет. В ходе тестирования было испытано качество 14 различных

болванок от различных производителей (BenQ 8x -R, Intenso 8x +R, FujiFilm 4x +R, Traxdata 4x -R, Verbatim 8x +R, Verbatim 8x -R, Traxdata 4x +R, Memorex 8x +R, TDK 4x +R, Intenso 8x -R, Memorex 8x -R, Sony 4x +R, TDK 4x -R, EMTEC 4x -R) и их совместимость с 13 наиболее распространенными рекордерами (NEC ND3500A/ версия прошивки 2.16, Pioneer DVR-A08 XLA/1.10, BenQ DW-1620/ B7H9, LiteOn SOHW1633S/BS0A, Plector PX-712A/1.05, Teac DV-W58G/H4S3, Philips DVDRW885K/P1.5, Sony DRU-700A/VY06, LG GSA-4120B/207, Toshiba SD-R5272/1031, Teac DV-W24E/7.1A, Pioneer DVR K13/1.10, LiteOn SDW-431S/MCS3). При этом было записано и прочитано в общей сложности 175 дисков.

Во время тестирования каждое из устройств было проверено в качестве как пишущего, так и читающего привода. От расширенного тестирования было решено отказаться в пользу выявления наибольшего числа комбинаций рекордер-носитель. Это позволило проверить 182 комбинации на совместимость носителей и рекордеров. Конечной целью данного тестирования стало определение максимальной скорости и совместимости приводов и носителей. В то время как скорость записи определялась по продолжительности работы привода, совместимость выявляли путем измерения пропускной способности при чтении, а также с помощью измерений времени доступа к трем различным зонам диска. Эти величины напрямую



зависят от совместимости болванки и привода и являются надежными показателями долговечности записи.

Из 14 протестированных болванок различных типов всеми рекордерами безошибочно были записаны и прочитаны всего три: BenQ 8x -R, Intenso 8x +R, Traxdata 4x -R. Работая с данными болванками, вы всегда получите "неизменно превосходный результат". Хуже всех дела обстоят у болванок EMTEC 4x -R — они корректно работали с восемью из двенадцати приводов (тринадцатый по счету Philips 885 записывает лишь +R). Всего из 175 болванок не смогли записать три, а 22 болванки невозможно было прочитать. Процент брака составил около 12,5 — результат неутешительный. Бракованными бывают не только болванки — это доказал привод LG GSA4120B, который отказался работать с тремя типами болванок. Положительные эмоции были получены от общения с приводами NEC ND3500A и Pioneer DVR-A08 XLA, которые надежно записали все диски и споткнулись всего лишь раз на одной и той же болванке производства Sony. При этом NEC работает не спеша — эффективная скорость записи лежит в диапазоне от 4x до 6x. Самым быстрым однозначно был признан привод фирмы Plextor — каждую вторую болванку он пишет на большей скорости, чем об этом заявляет сам производитель данного носителя.

Итоги теста болванок

Лучшая болванка — BenQ 8x -R. Она убедительно выглядит по всем статьям: не только потому что ее без проблем "узнает" каждый привод, но и из-за впечатляющих скоростей. ATIP-код указывает, что производителем этих болванок является Sony.

Самый надежный носитель — Traxdata 4x -R. При скорости "только" лишь 4x она обладает самой высокой степенью совместимости. Все приводы как один записали ее без сбоев и затем легко прочитали. Для любителей надежности этот носитель — лучший выбор.

Самая быстрая болванка — Verbatim 8x +R. Зачастую приводы писали ее с большей скоростью, чем указано на коробке (8x). При этом с совместимостью у нее тоже все в порядке. Однако за все хорошее приходится платить — она не самая дешевая из представленных на рынке.

Решающим фактором выбора рекордера не должны становиться лишь его низкая цена и заявленные скоростные характеристики. Если у вас до сих пор не было DVD-рекордера, не

расстраивайтесь, сейчас его можно смело покупать, не переплачивая за "новую" технологию. Цена этих устройств упала до приемлемого уровня, а потенциал их совершенствования еще до конца не исчерпан. Обновленные прошивки позволяют приводам оставаться "в строю".

В этой связи приятное впечатление производят приводы от Pioneer и Plextor: на сайтах обоих производителей регулярно появляются обновленные прошивки, позволяющие им сохранять свою актуальность. Судя по всему, BenQ и Philips уделяют мало внимания вопросам поддержки — обновления прошивок появляются на их веб-сайтах очень редко.

Предыдущие, но совсем не устаревшие модели LG-GSA 4082B и Pioneer DVR-A07XLA — неплохой выбор для экономных пользователей. Эти 8x/8x-скоростные приводы подкупают своей надежностью. В связи с появлением усовершенствованной модели, "устаревшая" зачастую продается по сниженной цене, и, если не учитывать меньшую скорость записи, она вполне достойна внимания. Большая часть таких якобы "устаревших" приводов имеет все необходимые функции, которые потребуются вам и в будущем. Pioneer DVR-A08XLA и Plextor PX-716A записывают DL-болванки даже со скоростью 4x (вместо 2,4x). Если же такие болванки действительно упадут в цене, то вскоре копирование DVD-фильмов станет достаточно простым и недорогим делом. Относительно скорости записи: выигрыш во времени от повышения скорости записи с 12x до 16x составляет около 30 секунд. Конечно, приятно быть обладателем скоростного привода, но гораздо разумнее данный фактор не ставить во главу угла при выборе устройства. Разница между скоростями 8x и 12x немного больше — около полутора минут.

Тем, кто хочет использовать свой DVD-рекордер в качестве практически средства резервного копирования, придется по душе устройства от компании LG: из протестированных приводов устройства только этого производителя поддерживают такой стандарт как DVD-RAM. К сожалению, для 5-скоростного RAM-рекордера (GSA4120B) в продаже пока нет соответствующего носителя. Очень важный критерий для любого рекордера, используемого при воспроизведении DVD-фильмов — уровень шума. Устройства от компаний LiteOn и Samsung работают не очень громко (3,5 и 3,8 сон соответственно), но все же не бесшумно. Все остальные

приводы при работе практически не слышны.

8-скоростные DVD-рекордеры обеспечивают скорость передачи данных около 10 Мбит/с, поэтому ваш персональный компьютер должен иметь современный винчестер и процессор уровня Pentium 4 (2 ГГц). Большинство DVD-рекордеров имеют емкость буфера 2 Мб. Этого вполне достаточно для работы механизма предотвращения опустошения буфера, но если у вас старенький компьютер, то этого может не хватить. При скорости записи 8x этого объема хватает лишь на 0,2 с! Перед покупкой пишущего DVD-привода вы должны честно признать себе, насколько велики ваши познания в этой области, и четко определить круг задач, которые необходимо решить с помощью рекордера. Если вы собираетесь записывать диски от случая к случаю, можете смело покупать себе недорогую модель привода. Например, BenQ DW 1620 стоит около 75 USD и записывает болванки любых форматов (кроме DVD-RAM). Он удовлетворит желания практически любого пользователя. Для любителей видеофильмов собственного "изготовления" рекомендуется рекордер, производитель которого достаточно часто обновляет микропрограммы. Это позволит улучшить совместимость с носителями и покупать недорогие болванки, которые будут вполне совместимы с приводом. Кроме того, даже по прошествии пары лет для вашего рекордера найдется подходящая болванка. Однако стоит отметить, что такие приводы (Pioneer или Plextor) имеют более высокую цену, чем конкуренты.

Требуется соблюдать осторожность и при покупке болванок. Проверьте в первую очередь, какие носители качественно записываются вашим приводом, а скорость записи — это уже не очень важный показатель. Ведь если пара ошибок на диске с фильмом или коллекцией MP3-записей никого особо не волнует, то несовместимость привода и болванки при сохранении важной информации — это настоящая катастрофа. Поэтому профессионалы создают несколько резервных копий на разных носителях.



Организация памяти и структура данных

А.КОРБИТ, Т.КРИВОНОСОВА,
г.Минск, БГУИР, ФИТиУ, каф.ВмиП,
E-mail: a_korbit@mail.ru

Виды организации хранения данных в памяти

Практически все языки программирования предоставляют возможность манипулирования множеством разнообразных структур данных. Так, например, в языке Си рассматриваются скалярные переменные, массивы, строки, структуры и т.д. Все объекты программы требуют декларации (описания) их типа с последующим определением. Это позволяет компилятору зафиксировать в оперативной памяти (ОП) их конкретные размеры. Такие объекты, как правило, называют статическими.

Механизм хранения данных в ОП может иметь или векторную, или списковую организацию памяти.

Векторная память и адресная функция

Векторная память поддерживается почти всеми языками высокого уровня и предназначена для хранения массивов различной размерности. Каждому массиву выделяется непрерывный участок памяти указанного размера. При этом, например, элементы трехмерного массива X размещаются в ОП в следующем образом:

$X(1,1,1), X(1,1,2), X(1,1,3), \dots, X(n_1, n_2, n_3-1), X(n_1, n_2, n_3)$.

Адресация элементов массива определяется некоторой функцией (адресной), связывающей адрес массива и индексы элемента.

Пример адресной функции для однородного трехмерного массива $X(n_1, n_2, n_3)$:

$K(i,j,k) = n_1 \cdot n_2 \cdot (i-1) + n_2 \cdot (j-1) + k; i = \overline{1, n_1}; j = \overline{1, n_2}; k = \overline{1, n_3}$.

Для размещения такого массива потребуется участок ОП размером $n_1 \cdot n_2 \cdot n_3$ элементов или $n_1 \cdot n_2 \cdot n_3 \cdot \text{sizeof}(\text{type})$ байтов (операция $\text{sizeof}(\text{type})$ возвращает количество байтов для размещения элемента массива типа type). Рассматривая эту область как вектор (одномерный массив) $Y(1, \dots, n_1 \cdot n_2 \cdot n_3)$, можно установить соответствие между элементом трехмерного массива X и элементом одномерного массива Y :

$$X(i,j,k) \Leftrightarrow Y(K(i,j,k)).$$

Адресная функция двумерного массива $A(n,m)$ будет выглядеть так:

$$N1 = K(i,j) = n \cdot (i-1) + j; i = \overline{1, n}; j = \overline{1, m}.$$

Тогда справедливо следующее:

$$A(i,j) \Leftrightarrow B(K(i,j)) = B(N1), i = \overline{1, n}, j = \overline{1, m}.$$

Здесь B — одномерный массив размера $N1 = 1, \dots, n \cdot m$. Например, для двумерного массива $A(2,2)$ имеем:

	$A(1,1)$	$A(1,2)$	$A(2,1)$	$A(2,2)$
	1	2	3	4
$i=1, j=1$	$N1 = 2 \cdot (1-1) + 1 = 1$			$B(1)$
$i=1, j=2$	$N1 = 2 \cdot (1-1) + 2 = 2$			$B(2)$
$i=2, j=1$	$N1 = 2 \cdot (2-1) + 1 = 3$			$B(3)$
$i=2, j=2$	$N1 = 2 \cdot (2-1) + 2 = 4$			$B(4)$

Необходимость введения адресных функций возникает лишь в случаях, когда требуется изменить способ отображения с учетом особенностей конкретной задачи.

Например, следующая адресная функция обеспечит компактное представление верхней треугольной матрицы X с диагональю порядка n :

$$K(i,j) = (n - i/2) \cdot (i-1) + j, \text{ где } i = 1, 2, \dots, n; j = i, i+1, \dots, n.$$

Для хранения элементов такой матрицы достаточно выделить в оперативной памяти одномерный массив — $Y(1 \dots n \cdot (n+1)/2)$, а для доступа к его элементам использовать следующее соотношение:

$$X(i,j) \Leftrightarrow Y(K(i,j)), i = \overline{1, n}, j = \overline{i, n};$$

(допустимые комбинации значений индексов здесь строго фиксированы условиями задачи).

Списковая организация памяти

Некоторые задачи исключают использование структур данных фиксированного размера и требуют введения структур динамических, способных увеличивать или уменьшать свой размер уже в процессе работы программы. Основу таких структур составляют динамические переменные.

Динамическая переменная хранится в некоторой области ОП, не обозначенной именем, и обращение к ней производится через переменную-указатель.

Динамическое распределение ОП связано с операциями порождения и уничтожения объектов по запросу программы, т.е. захват и освобождение памяти производится программно в процессе работы. При этом в стандарте языка Си таких операций нет, а порождение объектов (захват памяти) и уничтожение объектов (освобождение памяти) производится при помощи библиотечных функций.

Функции для манипулирования динамической памятью в стандарте Си следующие:

- **void *calloc(unsigned n, unsigned size);** — выделение памяти для размещения n объектов размером size байтов и заполнение полученной области нулями. Возвращает указатель на распределенную область памяти или **NULL** при нехватке памяти;

- **coreleft(void)** — получение размера свободной памяти в байтах (тип результата: **unsigned** — для моделей памяти **tiny**, **small** и **medium**; **unsigned long** — для моделей памяти **compact**, **large** и **huge**);

- **void free(void *b)** — освобождение блока памяти, адресуемого указателем b ;

- **oid *malloc(unsigned n)** — выделение области памяти для размещения блока размером n байтов. Возвращает указатель на распределенную область или **NULL** при нехватке памяти;

- **void *realloc(void *b, unsigned n)** — изменение размера размещенного по адресу b блока на новое значение n и копирование (при необходимости) содержимого блока. Возвращает указатель на перераспределенную область памяти или **NULL** при нехватке памяти.

В Си++ введены две операции по захвату и освобождению участков динамической памяти: **new** и **delete**. Например:

1) **type *p = new type[n];** — возвращает указатель на ОП размером $n \cdot \text{sizeof}(\text{type})$;

...
delete []p; — освободили всю захваченную память;

2) **type *p = new type(5);** — захватили участок памяти размером $\text{sizeof}(\text{type})$, установив на него указатель, и записали в захваченную область значение 5;

...
***p=12;** — заменили 5 на 12;

delete p; — освободили захваченную под элемент память. Как правило, динамические переменные организуются в списковые структуры данных.

Списковая организация памяти применяется в случаях, когда возможность прямого доступа к любому именованному элементу памяти не используется, а решение задачи практически удовлетворяется лишь возможностью перехода от одного элемента данных к другому. Очевидно, что в таких случаях не обязательно размещать последовательно обрабатываемые элементы в смежных областях памяти.

Для адресации элементов данных достаточно каждый элемент дополнить полем ссылки (указателем) на область размещения следующего элемента, и отдельно задать ссылку на первый элемент списка.

Использование указателей для связи элементов данных позволяет легко отразить различные виды взаимосвязей данных. В графическом представлении списков указатели представляются стрелками, начало которых соответствует области хранения указателя, а окончание — адресуемому указателем элементу.

Пример из жизни: очередь на прием в какой-нибудь кабинет. Каждый запоминает человека, за которым занял очередь — все связаны в цепочку, но в пространстве распределены произвольно. Вновь пришедший человек занимает очередь за последним, запоминает его и садится на свободное место.

Так и в памяти элементы связанной структуры размещаются там, где есть свободное место, и каждый элемент должен знать, за кем он стоит, т.е. содержать ссылку (указатель) на предыдущий элемент цепочки.

Пусть один покидает очередь, это не требует перемещения в пространстве остальных, стоящий за покинувшим очередь человек просто запоминает того человека, за которым ушедший занимал очередь. Т.е. исключение элемента из цепочки не требует перемещения остальных элементов в памяти — элемент удаляется из списка, освобождается память, занятая им, и эту память можно использовать для других целей.

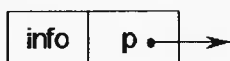
При помощи указателей легко добавить новый элемент — достаточно изменить значения двух указателей. При этом новый элемент может быть размещен в любом месте памяти.

Как правило, элементы таких данных (списков) имеют тип **struct**.

Описание связанных данных

Пусть необходимо создать линейный список, содержащий целые числа, тогда каждый элемент списка должен иметь информационную (*info*) часть, в которой будут находиться данные, и адресную (*p*) часть, в которой будут размещаться указатели связей (рис.1).

Рис. 1



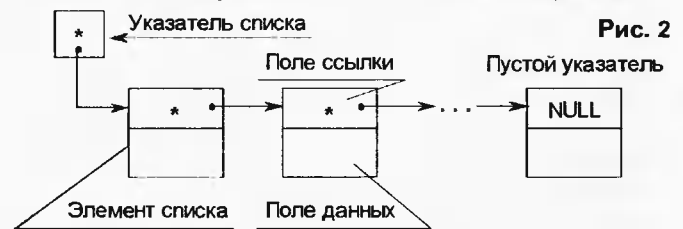
Шаблон объявления элемента такой структуры будет иметь вид:

```
struct spis {
    int info;
    spis *p;
};
```

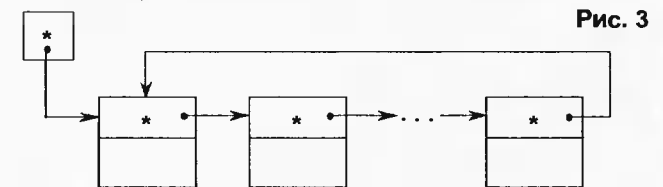
Важным понятием при манипулировании указателями является значение "пустого" указателя (*NULL*). Такой указатель используют для обнаружения окончания списковой последовательности.

Основные виды списковых структур

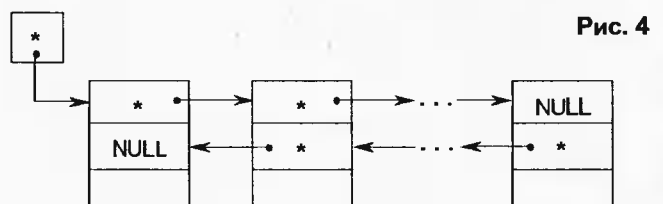
Линейный список (список) — организация некоторого множества элементов данных, при которой каждый элемент содержит собственно данные и указатель на расположение в памяти следующего элемента множества (рис.2).



Циклический список, в отличие от линейного, допускает многократное прохождение списка, начиная с любого элемента (рис.3).



Симметричный (двухсвязный) список позволяет организовать продвижение по связям в любом направлении и удобен для выполнения операций включения и исключения элементов в заданном месте. Очевидно, что нет принципиальных ограничений на вид структур, представляемых списками. Если элементом списка является другой список, говорят о списковых структурах (рис.4).



Таким образом, совокупность обрабатываемых элементов данных и взаимосвязей между ними без учета их физического размещения вводит новое понятие — абстрактная структура данных.

Приведем определения некоторых наиболее часто используемых абстрактных структур.

Множество — неупорядоченная совокупность элементов (данных). По определению, множество не содержит повторяющихся элементов. Если значения элементов повторяются, то говорят о комплекте.

Очереди и стеки — множества, элементы которых упорядочены по времени включения.

Очередь предполагает доступность только того элемента, который был помещен в нее ранее других.

Очередь с двумя концами (дек) допускает выполнение операций записи — выборки с обоих концов.

В стеке доступен только тот элемент, который был помещен позже других. Обращение к стеку связано с понятием "вершина стека".

Конечная последовательность элементов некоторого алфавита — строка.

Ориентированный граф — пара множеств вершин и дуг, где элементы множества дуг — упорядоченные пары вершин. Для нагруженного графа дополнительно определяется соответствующее каждой дуге значение (вес дуги).

Дерево — частный случай ориентированного графа, где имеется лишь одна вершина с нулевой степенью захода, а остальные вершины имеют единичную степень захода (степень захода — число дуг, входящих в вершину графа).

Таблица — множество элементов, организованное таким образом, что каждый элемент и, возможно, его расположение однозначно определяются его ключом. В отличие от списков, элемент таблицы идентифицирует только сам себя.

Таблица соответствует понятию файла, а ее элементы часто называют записями. Идентификация записей ключами оправдана тем, что обычно для различения записей существенна только небольшая часть записи, называемая ключом. Абстрактные структуры данных могут отображаться как в векторной, так и в списковой памяти. Выбор вида организации памяти определяется задачей использования представляемых данных.

Отображение структур данных в памяти

Итак, любые структуры данных могут отображаться в памяти, распределяемой как статически — объект будет создан на этапе компиляции, так и динамически — на этапе выполнения программы.

Статическое распределение реализуется средствами создания операционных объектов языка программирования на этапе трансляции и/или вызова программного модуля.

Примеры организации динамических данных

1. Создание динамической переменной

```
#include <alloc.h>

...
double *v; — декларировали указатель на участок па-
мяти, в котором можно хранить вещественные данные с
удвоенной точностью;
v = (double*)malloc(sizeof(double)); — захватили
участок памяти, установив на него указатель;
if (v==NULL) — контроль за возможной ошибкой;
{
    printf("\n Error! ");
    getch();
    return; // exit(1);
}
*v = 12.34; — косвенная инициализация;
... — работаем с динамической памятью;
free(v); — освобождаем память;
...
```

2. Создание одномерного динамического массива

```
#include <alloc.h>

...
double *x;
int i, n;
scanf("%d", &n); — ввод размерности массива;
x = (double*)calloc(n, sizeof(double));
if (x==NULL) { — контроль за возможной ошибкой (см. п.1);
    ... }
for ( i=0; i<n; i++) scanf("%lf", &x[i]);
— ввод элементов массива;
...
for ( i=0; i<n; i++) printf("\n %lf", x[i]);
— вывод элементов массива;
...
free(x);
...
```

3. Создание динамической переменной, используя опе-

рации **new** и **delete**

```
...
int *a;
a = new int(5);
... — работаем через косвенную адресацию;
delete a; — освобождаем память;
...
4. Создание одномерного массива, используя операции
new и delete
...
float *x;
int i, n;
scanf("%d", &n); — ввод размера массива;
x = new float[n];
if (x==NULL) { — контроль за возможной ошиб-
кой (см. п.1);
    ...
}
for ( i=0; i<n; i++)
    scanf("%f", (x+i)); — ввод элементов массива
(адреса);
...
for ( i=0; i<n; i++)
    printf("\n %f", *(x+i)); — вывод элементов
массива (значения);
...
delete [ ]x; — освободили память;
...
```

5. Создание двумерного массива, используя операции new и delete

Связь между указателями для двумерного массива $a[n][m]$ выглядит, как показано на рис.5.

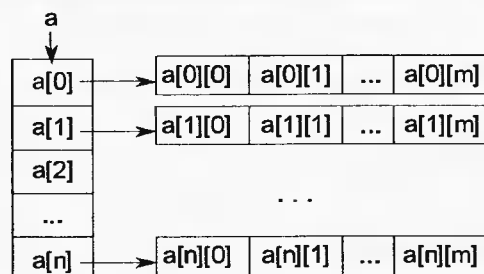


Рис. 5

И соответственно:

```
...
float **a;
int i, j, n, m;
scanf("%d%d", &n, &m);
a = new float* [n];
for ( i=0; i<n; i++)
    *(a+i) = new float [m];
...
for ( i=0; i<n; i++)
    for ( j=0; j<m; j++)
        scanf("%f", &a[i][j]);
...
for ( i=0; i<n; i++) delete [ ]a[i];
delete [ ]a;
```

Литература

1. Крячков А.В. и др. Программирование на С и С++. — М.: Горячая линия-Телеком, 2000.
2. Павловская Т.А. С/С++. Программирование на языке высокого уровня. — СПб.: ПИТЕР, 2002.



ИССЛЕДОВАНИЕ ПРОГРАММ

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

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КРЭКИНГА

(Окончание. Начало в №№8-12/2003, 1-12/2004, 1-2, 4-5/2005)

Выше мы рассматривали всевозможные приемы, позволяющие, образно говоря, залезать в чужое адресное пространство “через окно”. Но иногда разработчики программ предоставляют бесценную возможность попасть внутрь их детища “с парадного входа”, не таясь. И эта возможность — “родимое пятно” практически всех программ, использующих широко распространенный механизм расширения возможностей программы за счет плагинов. Плагин обычно представляет собой самую обычную DLL, построенную в соответствии с теми или иными стандартами, задаваемыми разработчиком программы. Однако **что именно** делает плагин, программу обычно не волнует — он может как выполнять штатные функции по обработке изображения, звука или чего-нибудь еще, так и быть “троянским коном”, созданным с одной лишь целью — в момент первого же обращения к нему исправить внутри программы “хозяина” несколько байтов. Поскольку ваш плагин будет “жить” внутри подопытной программы, для записи данных в адресное пространство вам не нужны будут никакие ухищрения, достаточно всего лишь поместить правильные значения по известным адресам при помощи самой обычной команды mov. Однако как добиться, чтобы ваш код выполнялся как можно быстрее? Точные сведения могут быть получены лишь через эксперименты, но приблизительное направление поиска следующее — многие программы после загрузки плагина первым делом “спрашивают” свежезагруженный модуль о его названии, назначении, поддерживаемых функциях и прочих подобных параметрах. А поскольку поиск и загрузка плагинов нередко выполняются сразу после старта программы (а если очень повезет — и до срабатывания защитных механизмов или параллельно с их работой), у вас есть хорошие шансы пропатчить программу уже на первых стадиях ее работы.

Общим требованием при патчинге программы в памяти является полная “неподвижность” программы, чего проще всего достигнуть, отправив все потоки модифицируемого процесса в состояние “замороженности” вызовом функции SuspendThread или при помощи старого фокуса “MySelf: jmp MySelf”. Связано это с тем, что модификация кода с точки зрения программы должна производиться одномоментно, чтобы во время патчинга программа случайно не попыталась выполнить не до конца модифицированный код (результатом чего почти наверняка будет сбой). Из этого правила есть важное следствие — если патчинг выполняется одной ассемблерной командой (а одной командой можно переписать один, два, четыре, а при большом желании — 8 или 10 байтов), то такая модификация будет одномоментной и потому вполне допустимой. К примеру, “волшебная” комбинация MySelf: jmp MySelf (в шестнадцатеричном виде этот код выглядит как EB FE) как раз является двухбайтной. Однако если вы выполняете патчинг чужого процесса при помощи функций WinAPI, вы не можете знать наверняка, блоками какого размера будет осуществляться запись в чужое адресное пространство, а потому пытаться переписывать больше чем один байт без “заморожки” процесса не стоит.

Самым старым подходом патчинга процессов под Win32

была загрузка программы с флагом CREATE_SUSPENDED с последующей записью в ее адресное пространство и “размораживанием” главного потока, однако применение его ограничено — очевидно, что этот метод практически неприменим в том случае, если программа упакована. Но что делать, если вы все же столкнулись с упакованной программой? Очевидно, что в момент запуска патчить ее бесполезно, поскольку тело программы все еще не декодировано. Стало быть, патчер должен дожидаться, пока программа будет распакована, и, как только этот знаменательный момент настанет, обрушиться на программу и привести в надлежащее состояние все “неправильные” байты. Осталось лишь выяснить, каким образом патчер может узнать о наступлении этого самого момента.

Самым оригинальным подходом к проблеме отличился, пожалуй, довольно старый патчер процессов, выпущенный ТЗХ — все обязанности по отслеживанию загрузки эта утилита возлагала на конечного пользователя. Созданный при помощи этой утилиты launcher просто запускал нужную программу, а затем выводил MessageBox с сообщением вроде “Нажми ОК, когда программа загрузится”. От пользователя, соответственно, требовалось кликнуть по кнопке ОК, когда он сочтет, что подопытная программа уже загрузилась и распаковалась. Несмотря на откровенный примитивизм, этот патчер вполне соответствовал духу популярных защит того времени и позволял решать насущные задачи.

Естественным следующим шагом в определении наилучшего момента для патчинга стала привязка к появлению на экране какого-либо окна, создаваемого модифицируемой программой. Поскольку большинство Windows-приложений являются оконными и при запуске создают окно, в котором предстоит работать пользователю, появление такого окна может служить недвусмысленным указанием на то, что распаковка программы уже завершилась, и потому можно приступать к патчингу. Одним из первых такой патчер выпустил uoda, причем его патчер позволял не только обнаруживать появление окна, но и принудительно его закрывать (что было полезным для борьбы с pag scgeen’ами). Однако этот инструмент имел ряд недостатков, в частности — требовал указания полного заголовка окна, что не позволяло работать с окнами с изменяющимся заголовком. А поскольку окна со счетчиками оставшихся дней “испытательного срока” встречались в программах все чаще и чаще, такая ситуация сподвигла меня на написание собственного инструмента — интерпретатора скриптов с механизмом поиска окон по содержащейся в заголовке окна подстроке, возможностью имитировать нажатие кнопок в окне, активировать отключенные управляющие элементы и т.п. Однако и этот инструмент не во всех случаях способен решить проблему своевременного патчинга — защита может быть устроена таким образом, что в момент появления окна будет уже поздно что-либо предпринимать. Кроме того, нельзя сбрасывать со счетов безоконные программы, которые хоть и редко, но все же встречаются в мире Win32-приложений. И тогда приходится применять старый, весьма расточительный с точки зре-



ния расхода ресурсов процессора, но проверенный временем и практически безотказный прием.

Традиционно проблема определения момента распаковки программы решается следующим образом — подопытная программа запускается при помощи launcher'a с флагом CREATE_SUSPENDED, этому процессу и его главному потоку назначается наименьший возможный приоритет, после чего главный поток "размораживается" функцией ResumeThread. Программа начинает распаковываться, но делает это очень медленно: во-первых, по причине минимального приоритета, а, во-вторых, потому что launcher в это время постоянно считывает содержимое байтов, которые предполагается пропатчить, и сравнивает их с эталонными значениями, заранее добытыми из "живой" программы при помощи дампера или отладчика вроде SoftIce. Цель этих действий заключается в следующем — исправление необходимо провести как можно раньше после того, как подвергаемые патчингу участки будут распакованы. Как только содержимое нужных ячеек памяти совпадет с эталонами, программу можно считать готовой к внесению модификаций. В этот момент патчер стопорит исполнение программы и модифицирует код, а затем возвращает в исходное состояние приоритеты, размораживает все потоки и позволяет программе выполняться дальше как ни в чем не бывало.

Блок, выполняющий постоянное сканирование, желательнее оптимизировать по скорости настолько, насколько это возможно — если наш launcher не успеет внести в программу изменения до того как управление будет передано на изменяемые участки, всю операцию по патчингу можно считать проваленной. Именно для того, чтобы снизить вероятность "слишком быстрого" запуска программы и увеличить промежуток

времени, пригодный для патчинга, мы и понижаем приоритет программы, отбирая у нее кванты процессорного времени в пользу нашего launcher'a и других программ. Из всего вышесказанного следует интересный практический вывод, который, к сожалению, не учитывается в большинстве существующих патчеров процессов — если патч предполагается использовать на мультипроцессорной системе, ваш launcher и подопытную программу (или хотя бы ее главный поток) лучше исполнять на одном и том же процессоре. Добиться этого под Windows NT можно при помощи функций SetProcessAffinityMask, SetThreadAffinityMask и GetProcessAffinityMask.

Дочитав предыдущий абзац, вы могли задаться вопросом, насколько вообще эффективно использование API'шных функций для чтения-записи в адресное пространство чужого процесса для наших целей, и нельзя ли воспользоваться каким-либо более "прямым" методом работы в чужом адресном пространстве. Такие вопросы вполне уместны — операции по "залезанию" в чужое адресное пространство вообще требуют значительных накладных расходов, из-за чего функция ReadProcessMemory действительно работает не слишком быстро (хотя обычно и такой скорости бывает более чем достаточно).

На этом наш экскурс в теорию и методологию патчинга можно считать законченным. Думаю, вы достаточно разобрались в основах самой важной и самой употребительной крэкерской операции. А стало быть, подходит к логическому завершению и эта работа. Единственная область, которая до настоящего момента мною почти не затрагивалась — это технологии будущего, "крэкинг завтрашнего дня". Подходы, только-только утверждающиеся на крэкин-сцене, а также идеи, которые пока еще никто не осуществил — это тема для отдельной статьи.

В.КАРЦЕВ,
Минск, БГУИР.

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

(Продолжение. Начало в №5/2005)

Пример управления исключительными ситуациями, характерными для языка C++

Рассмотрим подробнее применение обработки исключительных ситуаций C++ на довольно простом примере.

Пусть в окно редактирования Edit1 вводится текстовая фраза, и программа подсчитывает в ней количество пробелов, латинских букв, цифр и "специальных" символов, которыми будем считать запятые и точки. Кроме того, приложение будет рассчитывать отношение количества букв к количеству цифр. Результаты выполненных приложением действий будут отражаться в многострочном окне редактирования Memo1. Фраза не должна превышать заданного количества символов (оно указано в Edit2).

Текст файла с необходимыми обработчиками имеет вид:

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "UnitExceptCPP.h"
//-----
#pragma package(smart_init)
```

```
#pragma resource "*.dfm"
#include <math.h>
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    Edit1->Text="";
}
//-----
void TForm1::mystrlen()
{
    int n; n=StrToInt(Edit2->Text);
    if (Edit1->Text.Length()>n)
        throw "Такой длины фраза быть не может!";
}
//-----
void symbol(char c)
{
    if (!isalnum(c)&&c!=' ' &&c!='.' &&c!='(',')')
        throw "Нельзя вводить такой символ!";
}
//-----
Extended proportion(Extended a, Extended b)
{
    if(b) return a/b; throw 1;
}
//-----
void __fastcall TForm1::Edit1Change(TObject *Sender)
{
    static a,b,c,d,length=0;
    static char lastchar='\0';
    AnsiString s;
```



```
if (Edit1->Text!="" || (Edit1->Text=="&&lastchar!='\0'))
{
    try
    {
        Mem1->Clear();
        s=Edit1->Text;
        if (s=="") throw 2;
        symbol(*Edit1->Text.AnsiLastChar());
        mystrlen();
        if (length>s.Length())
        {
            switch(lastchar)
            {
                case ' ': a--; break;
                case '.': case ',': c--; break;
                default: if (isdigit(lastchar)) d--;
                        else b--;
            }
            length--;
        }
        if (length<s.Length())
        {
            switch(s[length+1])
            {
                case ' ': a++; break;
                case '.': case ',': c++; break;
                default: if (isdigit(s[length+1])) d++;
                        else b++;
            }
            length++;
        }

        lastchar=s.AnsiLastChar();
        Mem1->Lines->Add("Количество пробелов - "+IntToStr(a));
        Mem1->Lines->Add("Количество букв - "+IntToStr(b));
        Mem1->Lines->Add("Количество цифр - "+IntToStr(d));
        Mem1->Lines->Add("Количество точек и запятых - "+
            IntToStr(c));
        Mem1->Lines->Add("Кол_букв/Кол_цифр="+
            FloatToStr(proportion(b,d)));
    }
    catch (char* str)
    {
        ShowMessage(AnsiString(str));
        Edit1->Text=Edit1->Text.Delete(Edit1->Text.Length(),1);
        Edit1->SelStart=Edit1->Text.Length();
        if (length==0) Mem1->Lines->Add("Все символы из фразы
        удалены!");
    }
    catch (int i)
    {
        if (i==1) Mem1->Lines->Add("Фраза не содержит цифр!");
        else {a=b=c=d=0; length--; lastchar='\0';
            Mem1->Lines->Add("Все символы из фразы удалены!");}
    }
    catch (...)
    {
        ShowMessage("Неизвестная исключительная ситуация!");
    }
}

//-----
void __fastcall TForm1::Edit2Change(TObject *Sender)
{
    int n;
    AnsiString s;
    try
    {
        n=StrToInt(Edit2->Text);
        catch(const EConvertError&)
        {
            s=Edit2->Text;
            ShowMessage("В данное поле надо вводить число, а не"+
                s+"!");
            Edit2->Text=Edit2->Text.Delete(Edit2->Text.Length(),1);
            Edit2->SelStart=Edit2->Text.Length();
        }
    }
}

//-----
```

Выбрасывание исключительной ситуации (имеются в виду исключения C++) здесь может происходить (помимо непредвиденного случая) из трех функций:

- 1) метода (элемента-функции [2], функции-члена [3]) void TForm1::mystrlen(), проверяющего, чтобы вводимая фраза не оказалась длиннее разрешенной (окно редактора Edit2);
- 2) void symbol(char c), проверяющей, допустимый ли символ вводится пользователем приложения;

3) Extended proportion(Extended a, Extended b), в которой находится отношение количества букв к количеству цифр, содержащихся во фразе.

В первых двух случаях при возникновении условий для выбрасывания исключительной ситуации обработка исключительной ситуации выполняется в блоке catch(char* str). Это объясняется необходимостью выполнения одних и тех же действий: выдачи сообщения об ошибке и удаления последнего введенного символа. Таким образом, в данном случае хорошо видны преимущества использования обработчиков исключительных ситуаций — обработку похожих ошибок, которые встречаются в разных частях программы, можно компактно разместить в отдельном месте программы. Третий случай фактически демонстрирует то же преимущество. Действительно, достаточно вместо if (i==1)... использовать оператор switch, и в одном месте можно будет сосредоточить обработку ошибок, возникающих в самых различных точках программы. В данной же программе случай i==1 соответствует выбрасыванию исключительной ситуации в функции proportion(), когда фраза не содержит цифр, а случай i!=1 — выбросу исключительной ситуации, когда все символы оказываются удаленными из Edit1.

Следует отметить, что при формировании блока try...catch следует быть осторожным с порядком записи catch-обработчиков. Особенно внимательным надо быть, если имеется обработчик с формальным параметром void*. Его не следует размещать первым в списке catch-обработчиков. В противном случае последующие за ним catch-обработчики в данном блоке try...catch могут перестать выполняться.

Исключительные ситуации C++ и Standard C++ Library

Важность обработки исключительных ситуаций вообще и для языков C и C++ в частности стала понятна программистам уже давно. Не случайно ведь, что средства обработки исключительных ситуаций вошли в широко известную библиотеку Standard C++ Library, в состав которой, в частности, входят стандартная библиотека шаблонов STL, средства iostream и т.д. Следует отметить, что в C++ Builder 6 используется более современная версия стандартной библиотеки шаблонов, называемая STLport, хотя имеются необходимые средства для поддержания совместимости с так называемой Rogue Wave C++ Standard Template Library. В Standard C++ Library, в отличие от VCL-исключений, для которых базовым классом является класс Exception (потомок TObject), важнейшую роль для обработки исключений играет класс exception. Его потомками являются классы logic_error и runtime_error. Эти два класса созданы из соображений деления всех возможных ошибок на две разновидности, в зависимости от причин возникновения исключения. От logic_error происходят классы domain_error, invalid_argument, length_error и out_of_range. Производными от runtime_error являются классы range_error, overflow_error, underflow_error. Как видно из названий, класс exception связан с обработкой логических ошибок и ошибок времени выполнения программы. Если в приложении выполняется обработка исключительных ситуаций с использованием рассматриваемых классов, следует с помощью директивы препроцессора #include подключить заголовочный файл с именем stdexcept. Используя данные классы, можно перехватывать как исключительные ситуации,

выброшенные при работе компонентов, входящих в библиотеку Standard C++ Library, так и исключительные ситуации, созданные самим разработчиком программы с помощью ключевого слова `throw`. Например, пусть имеется некое приложение, у которого на форме, помимо других компонентов, есть окна редактирования `Edit1` и `Edit2`. Окна редактирования `Edit1` и `Edit2` предназначены для ввода двух величин (типа `float`). Приложение вычисляет, сколько процентов от большей величины составляет меньшая величина. Нулевые значения операндов считаются недопустимыми. Результат вычислений приложение отображает в `Мемо1`. Функция, позволяющая выполнить указанные действия, может выглядеть так:

```
void TForm1::Procent()
{
    float a,b;
    string st;
    try
    {
        a=StrToFloat(Edit1->Text);
        b=StrToFloat(Edit2->Text);
        if (a*b==0) throw out_of_range("Операнд не может быть равным нулю!");
        if (a==b) st.resize(st.max_size()+1);
        if (a>b) throw invalid_argument("Операнд а больше b!");
        else Mem1->Lines->Add("a - это "+FloatToStr(a*100/b)+"% от b");
    }
    catch (const invalid_argument& exlog)
    {
        ShowMessage("Внимание! "+AnsiString(exlog.what()));
        Mem1->Lines->Add("b - это "+FloatToStr(b*100/a)+"% от a");
    }
    catch (const exception& exlog)
    {
        if (!strcmp(exlog.what(),"basic_string"))
        {
            ShowMessage("Значения a и b равны между собой!");
            Mem1->Lines->Add("a равно b");
        }
        else
        {
            ShowMessage("Ошибка! "+AnsiString(exlog.what()));
            Mem1->Lines->Add("Непредвиденная ошибка!");
        }
    }
}
```

В файле, где находится текст данной функции, должны быть также строки, необходимые для корректной работы функции `Procent()`:

```
#include <stdexcept>
using namespace std;
```

В функции `Procent()` исключительная ситуация может возникнуть как благодаря наличию ключевого слова `throw` в тексте функции, так и в результате выполнения функции `st.resize(st.max_size()+1)`, содержащей некорректное значение аргумента. Такое значение аргумента указано специально, с целью искусственного выбрасывания исключительной ситуации класса `length_error`. Следует отметить, что `st` — переменная типа `string`. Такой тип представляет собой одну из двух версий шаблона класса `basic_string`. Тип `string` — составная часть Standard C++ Library. Сам `basic_string` работает подобно последовательному контейнеру STL.

При равенстве хотя бы одного из операндов нулю исключительная ситуация в функции `Procent()` выбрасывается с помощью ключевого слова `throw`, операндом которого является `out_of_range` ("Операнд не может быть равным нулю!"). В результате обработки такой исключительной ситуации на экран будет выведено окно с сообщением "Ошибка! Операнд не может быть равным нулю!", а в многострочное окно редактирования — строка, сообщающая о наличии непредвиденной ошибки. В случае, если операнд `a>b`, программа обработает исключительную ситуацию класса `invalid_argument`. В результате обработки на экране появится окно с сообщением "Внимание! Операнд а больше b!", а в `Мемо1` — строка, содержащая результат соответствующего вычисления. Как уже было отмечено ранее, некорректное значение аргумента функции `st.resize(st.max_size()+1)` приводит к возникновению исключительной ситуации класса `length_error`. Обработка последнего (как и исключительной ситуации класса `out_of_range`) будет выполнена в обработчике `catch (const exception& exlog)`, так как класс `length_error` является потомком класса `exception`.

Как видно из текста примера, при обработке исключительных ситуаций широко используется составная часть класса `exception` — виртуальная функция `what()`. Она возвращает заканчивающуюся нулем строку символов, содержащую информацию о возникшей исключительной ситуации.

(Продолжение следует)

Программа для форматирования текстов

И.РОЩИН,

г.Москва,

ZXNet: 500:95/462.53

E-mail: bestview@mtu-net.ru

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

Введение

Иногда приходится иметь дело с текстовыми файлами, в которых каждый абзац представлен в виде одной длинной строки (например, такие файлы получаются при использовании моего конвертора DOC → TXT [1]), а надо преобразовать их в "обычный" вид, когда абзацы разбиты на строки, ширина которых не больше указанной, а первая строка каждого абзаца выделена отступом заданной величины. Для этой цели я написал небольшую программу на Turbo C, которую предлагаю вашему вниманию.

Текст программы

```
/* =====
   File: "txt form.c"
   Compiler: Turbo C 2.01
   ===== */
#include <stdio.h>
```

```
#include <stdlib.h>
/* ===== Описание функций =====
*/
int fgetc_2 (FILE* f);
FILE* fopen_1 (char* name, char* mode);
void fcloseall_1 ();
int fgetc_1 (FILE* f);
void ungetc_1 (int c, FILE* f);
void fputc_1 (int c, FILE* f);
long ftell_1 (FILE* f);
void fseek_1 (FILE* f, long offset, int whence);

/* ===== Пошла сама программа =====
*/
int main (int argc, char *argv[])
{
    FILE *f_src, *f_dst;
    unsigned int text_width; /* Ширина формируемого текста (т.е.
                               максимальная длина строки). */
```



```

unsigned int text_indent; /* Добавляемый абзацный отступ. */
long begin_str; /* Смещение начала текущей строки в исходном
текстовом файле. */
long len_str; /* Длина текущей строки. */
long cut_str; /* Смещение последней (на данный момент)
позиции в исходном текстовом файле,
соответствующей возможному месту переноса
текущей строки (т.е. в этой позиции пробел).
Если cut_str=-1 - значит, такой позиции пока
не обнаружено. */
long i; /* Счетчик. */
int c; /* Текущий символ. */
int first_str; /* Признак "выводим первую строку абзаца" (1 -
да, 0 - нет. */
long n=0; /* Номер обрабатываемого абзаца. */

printf ("\n\n
-----\n\n
Text Formatter\n\n
-----\n\n
(c) Ivan Roshin, Moscow, 26 Sep 2004\n\n
E-mail: bestview@mtu-net.ru WWW: http://www.ivr.da.ru\n\n
L-----\n\n
");
if (argc!=5)
{
printf ("\nSyntax: txt_form text_width text_indent \\  

source_file destiny_file\n\n");
exit(1);
}
if (sscanf(argv[1],"%u",&text_width)!=1)
{
printf ("\nIncorrect text_width!\n\n"); exit (1);
}
if (sscanf(argv[2],"%u",&text_indent)!=1)
{
printf ("\nIncorrect text_indent!\n\n"); exit (1);
}
f_src=fopen_1(argv[3],"rb");
f_dst=fopen_1(argv[4],"wb");

/* ===== Главный цикл ===== */
while (c=fgetc_2(f_src),c!=EOF) /* Пока не конец текста. */
{
printf ("\rParagraph: %li",++n);
if (c==13) /* Пустой абзац. */
{
fputc_1 (13,f_dst);
fputc_1 (10,f_dst);
continue; /* К началу главного цикла. */
}
/* Непустой абзац. */
ungetc_1(c,f_src);
for (i=0;i<text_indent;i++) fputc_1(' ',f_dst); /* Отступ. */
len_str=text_indent;
first_str=1;
begin_str=ftell_1(f_src);
cut_str=-1;

/* Обработка абзаца. */
while (c=fgetc_2(f_src),(c!=13)&&(c!=EOF))
/* Пока не конец абзаца или текста. */
{
len_str++;

/* Если текущий символ - пробел, то в этом месте строка
может быть разорвана; запоминаем позицию пробела. */
if (c==' ') cut_str=ftell_1(f_src)-1;

/* Если длина строки превысила максимальную, и при этом есть
место, в котором строка может быть разорвана, то выдаем
в формируемый файл часть строки от начала до места
разрыва и затем код конца строки; после этого положение
начала строки в исходном файле - это позиция после того
пробела, где был разрыв. */
if ((len_str>text_width)&&(cut_str!=-1))
{
fseek_1 (f_src,begin_str,SEEK_SET);
for (i=begin_str;i<cut_str;i++)
{
fputc_1(fgetc_2(f_src),f_dst);
}
fputc_1 (13,f_dst);
fputc_1 (10,f_dst);
begin_str=cut_str+1;
fseek_1 (f_src,begin_str,SEEK_SET);
len_str=0;
first_str=0;
cut_str=-1;
}

/* Дошли до конца абзаца. Если есть остаток строки
(len_str!=0), то выводим его и код конца строки. */
if (len_str!=0)
{
long temp=ftell_1(f_src); /* Запомнили позицию. */
fseek_1 (f_src,begin_str,SEEK_SET);
for (i=0;i<len_str-first_str*text_indent;i++)
{
fputc_1(fgetc_2(f_src),f_dst);
}
fseek_1 (f_src,temp,SEEK_SET); /* Восстановили позицию. */
fputc_1 (13,f_dst);
fputc_1 (10,f_dst);
}

} /* Конец главного цикла. */

/* ===== Весь текст обработан ===== */
fcloseall_1 ();
printf ("\nOK!\n\n");
exit (0);
}

/* =====
Функция fgetc_2 - если в файле f с текущей позиции находится
пара байтов 13,10, то функция читает оба байта и возвращает
13, иначе - читает один байт и возвращает его значение.
===== */
int fgetc_2 (FILE* f)
{
int c;
c=fgetc_1(f);
if (c==13)
{
c=fgetc_1(f);
if (c!=10) ungetc_1(c,f);
return 13;
}
return c;
}

/* =====
Аналоги некоторых функций для работы с файлами (fopen,
fcloseall, fgetc, ungetc, fputc, ftell, fseek). При ошибке
прерывают выполнение программы, печатая соответствующее
сообщение; таким образом, после вызова этих функций не надо
проверять, была ли ошибка.
===== */
FILE* fopen_1 (char* name, char* mode)
{
FILE* f;
char* s;
char* s1="read";
char* s2="write";

f=fopen(name,mode);
if (f==NULL)
{
if (mode[0]!='r') s=s1; else s=s2;
printf ("\nCan't open file \"%s\" for %s!\n\n",name,s);
exit (1);
}
}

```

```

fseek_1 (f_src,begin_str,SEEK_SET);
for (i=begin_str;i<cut_str;i++)
{
fputc_1(fgetc_2(f_src),f_dst);
}
fputc_1 (13,f_dst);
fputc_1 (10,f_dst);
begin_str=cut_str+1;
fseek_1 (f_src,begin_str,SEEK_SET);
len_str=0;
first_str=0;
cut_str=-1;
}

/* Дошли до конца абзаца. Если есть остаток строки
(len_str!=0), то выводим его и код конца строки. */
if (len_str!=0)
{
long temp=ftell_1(f_src); /* Запомнили позицию. */
fseek_1 (f_src,begin_str,SEEK_SET);
for (i=0;i<len_str-first_str*text_indent;i++)
{
fputc_1(fgetc_2(f_src),f_dst);
}
fseek_1 (f_src,temp,SEEK_SET); /* Восстановили позицию. */
fputc_1 (13,f_dst);
fputc_1 (10,f_dst);
}

} /* Конец главного цикла. */

/* ===== Весь текст обработан ===== */
fcloseall_1 ();
printf ("\nOK!\n\n");
exit (0);
}

/* =====
Функция fgetc_2 - если в файле f с текущей позиции находится
пара байтов 13,10, то функция читает оба байта и возвращает
13, иначе - читает один байт и возвращает его значение.
===== */
int fgetc_2 (FILE* f)
{
int c;
c=fgetc_1(f);
if (c==13)
{
c=fgetc_1(f);
if (c!=10) ungetc_1(c,f);
return 13;
}
return c;
}

/* =====
Аналоги некоторых функций для работы с файлами (fopen,
fcloseall, fgetc, ungetc, fputc, ftell, fseek). При ошибке
прерывают выполнение программы, печатая соответствующее
сообщение; таким образом, после вызова этих функций не надо
проверять, была ли ошибка.
===== */
FILE* fopen_1 (char* name, char* mode)
{
FILE* f;
char* s;
char* s1="read";
char* s2="write";

f=fopen(name,mode);
if (f==NULL)
{
if (mode[0]!='r') s=s1; else s=s2;
printf ("\nCan't open file \"%s\" for %s!\n\n",name,s);
exit (1);
}
}

```



```

return (f);
}

void fcloseall_1 ()
{
    if (fcloseall()==EOF)
    {
        printf ("\nClose error!\n\n"); exit (1);
    }
}

int fgetc_1 (FILE* f)
{
    int c;
    c=fgetc(f);
    if (ferror(f)) {printf ("\nRead error!\n\n"); exit (1);}
    return c;
}

void ungetc_1 (int c, FILE* f)
{
    if (ungetc(c,f)==EOF)
    {
        printf ("\nUNGETC error!\n\n"); exit (1);
    }
}

void fputc_1 (int c, FILE* f)
{
    if (fputc(c,f)==EOF) {printf ("\nWrite error!\n\n"); exit (1);}
}

long ftell_1 (FILE* f)
{
    long a;
    a=ftell(f);
    if (a== -1L) {printf ("\nFTELL error!\n\n"); exit (1);}
}

void fseek_1 (FILE* f, long offset, int whence)
{
    if (fseek(f,offset,whence)!=0)
    {printf ("\nPosition error!\n\n"); exit (1);}
}

```

Комментарии

При компиляции этой программы непосредственно из среды Turbo C вы получите exe-файл. Но можно получить более короткий com-файл, если воспользоваться имеющимся в Turbo C компилятором tcc.exe. Ниже приведена командная строка для вызова этого компилятора (вместо c:\work\tc2 вы, конечно, должны подставить свой путь к каталогу Turbo C):

```

c:\work\tc2\tcc.exe -O -Z -mt -lt -Ic:\work\tc2\include
-Ic:\work\tc2\lib txt_form.c

```

Опции "-O" и "-Z" в этой строке — соответственно включение оптимизации переходов и включение оптимизации использования регистров, что также способствует уменьшению размера исполняемого файла программы.

Чтобы еще уменьшить размер, можно воспользоваться свободно распространяемым компрессором исполняемых файлов UPX (<http://upx.sourceforge.net>), который может сжимать DOS-файлы в форматах COM и EXE (этим, конечно, его возможности не ограничиваются). Я использую последнюю на момент написания этой статьи версию UPX — 1.24. Изучив указания в документации UPX на то, как достичь максимальной степени сжатия, я создал командный файл compress.bat, содержащий две команды (каждая команда сжимает файл своим методом):

```

upx --best --crp-ms=999999 --nrvtb --8086 -o result_2b.com txt_form.com
upx --best --crp-ms=999999 --nrvt2d --8086 -o result_2d.com txt_form.com

```

После запуска этого командного файла UPX создаст два сжатых файла — result_2b.com и result_2d.com. Из них надо выбрать наименьший по размеру, а другой — стереть. После этого исходный файл txt_form.com уже не нужен, его можно стереть и дать его имя оставшемуся сжатому файлу. Исполняемый файл программы, который выложен на моей web-странице и на сайте журнала, получен именно таким образом.

Между прочим, при сжатии com-файлов длины получающихся файлов result_2b.com и result_2d.com у меня почему-то всегда совпадали, а вот при сжатии exe-файлов — действительно различались.

И еще одно примечание: указываемая при вызове UPX опция "--8086" нужна, чтобы добавляемый к сжатой программе код распаковщика использовал только инструкции процессора 8086. Тогда сжатая программа сможет распаковаться даже на компьютере с таким процессором (например, на IBM PC XT). Если же эта возможность для вас не имеет значения, то опцию "--8086" можно убрать, и таким образом еще чуть-чуть сократить размер сжатой программы (так как распаковщик будет более коротким). В этом случае для запуска сжатой программы потребуется компьютер с как минимум 286-м процессором.

Руководство пользователя

Все очень просто. При запуске программы в командной строке сначала надо указать параметры форматирования (первое число — ширина получаемого текста в символах, т.е. максимальная длина строки; второе число — величина абзацного отступа, т.е. сколько пробелов надо добавлять в начале первой строки каждого абзаца), а затем — имя исходного файла (он должен находиться в текущем каталоге) и имя формируемого файла (он создается также в текущем каталоге). Если запустить программу без аргументов или с неправильным их количеством, то она выдаст краткую справку о том, какие аргументы и в каком порядке надо указывать.

Пример командной строки:

```
txt_form 64 4 text_1.txt text_2.txt
```

В процессе работы программа отображает порядковый номер формируемого в данный момент абзаца и таким образом дает понять, что она не зависла :).

Программа возвращает операционной системе код завершения: 0 — успешное завершение, 1 — произошла ошибка при работе с файлами (при возникновении такой ошибки программа выводит соответствующее сообщение и завершает свою работу).

Замечу, что если в обрабатываемом тексте попадется слишком длинное слово (длиннее, чем указанная при запуске программы ширина текста), то это слово не будет разорвано, и длина строки с этим словом окажется больше максимальной.

В обрабатываемых файлах код конца строки может быть как парой байтов 13,10, так и одним байтом 13 — это позволяет получать форматированный txt-файл из doc-файла в кодировке Windows-1251 (в таких doc-файлах, по моим наблюдениям, абзацы представлены как раз в виде длинных строк, заканчивающихся байтом 13).

В формируемом текстовом файле строки заканчиваются парой байтов 13,10.

Исполняемый файл программы выложен на веб-странице журнала в приложении к статье.

Литература

1. Рошин И. Простейший конвертер DOC → TXT. — Радиомир. Ваш компьютер, 2003, №10, С.27.

Адаптер для ПК

А.КАШКАРОВ,
г.С.-Петербург.

Задавшись целью расширить возможности ПК, можно на разнообразных CD и в Интернете найти программное обеспечение на любой вкус, и использовать ПК в качестве осциллографа, генератора импульсов различной формы и длительности, вольтметра, термометра, таймера, устройства, управляющего мощной нагрузкой по заданной программе, и т.д. Для этих целей лучше всего подходит устаревший по сегодняшним меркам ПК. В моем варианте используется Пентиум 1 с тактовой частотой процессора 166 МГц. Кроме P-1, подходит любой ПК, имеющий хотя бы один разъем LPT-порта. Такой ПК "не жалко" использовать в виде прибора-помощника (с ограниченным набором функций); стоимость его на вторичном рынке невелика, а польза от такой старой модели в умелых руках просто неоценима.

Через разъем DB-25 порта LPT1 к персональному компьютеру подключается внешний датчик. Для примера, таким датчиком может служить, в зависимости от конкретного назначения, терморезистор, фоторезистор, электронный схема обработки сигналов от счетчика Гейгера-Мюллера, управляющее напряжение и т.п. Однако, даже если величина напряжения, воздействующего на управляющий LPT-порт ПК, невелика — от 0 до +5 В, подключать внешние электронные устройства непосредственно к параллельному порту ПК без гальванической раз-

вязки или буферного каскада опасно. На рисунке показан простейший вариант буферного каскада на популярном операционном усилителе K140УД6. В данном случае выходной сигнал, поступающий на порт ПК, оказывается инвертированным (обратным) относительно входного сигнала. Схема рассчитана таким образом, что амплитуда входного напряжения на ОУ не превышает +6 В, поэтому от

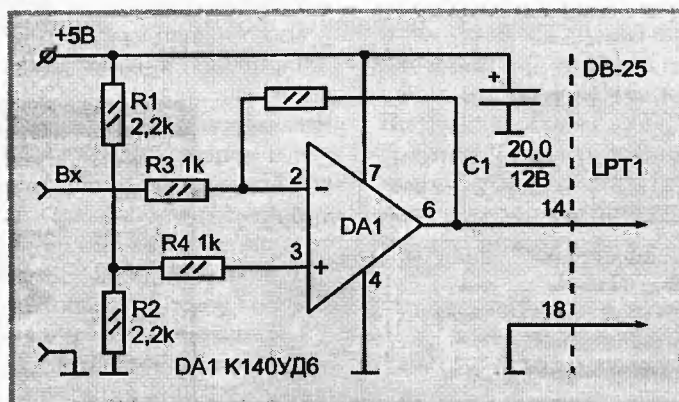
индикации состояния канала данную схему можно дополнить транзисторным ключом со светодиодом в качестве коллаторной нагрузки.

Элементы адаптера монтируются на перфорированной плате и закрываются корпусом размерами 3,5х5 см. Для уменьшения наводок переменного напряжения и для стабильной работы узла все проводящие стенки корпуса и экраны соединительных проводов подключаются к общему проводу. Схема не требует настройки, при исправных элементах она сразу начинает работать. Вместо операционного усилителя K140УД6 можно применить K140УД608, K140УД7. Используемые в схеме постоянные резисторы — типа МЛТ-0,125. Их номиналы могут отличаться от указанных в схеме на 20%.

Данный адаптер использовался мною в составе ПК, работающего в качестве устрой-

ства контроля и управления периферийными электронными узлами. В частности, мною использовались некоторые программы с ресурса www.ra3ggi.qrz.ru:
- низкочастотный частотомер (.../FILE/counter.rar);
- электронный частотомер с расширенными возможностями (.../wg.zip);
- аудиотестер и спектроанализатор Audio Tester v1.4 (.../at.zip);
- низкочастотный генератор с качением частоты (.../sineware.zip);
- двухканальный осциллограф со спектроанализатором по WIN Oscilloscope v2.51 (.../oscl_1.zip).

Питание ОУ — однополярное, в диапазоне 4,5...6 В, от стабилизированного внешнего источника постоянного напряжения. Ток, потребляемый ОУ в таком режиме работы, не превышает 5 мА. При необходимости визуальной



Как преобразить свой компьютер

А.АНИСОВИЧ,
г.Минск.

Все компьютеры "рождаются на свет" одинаковыми, точнее, с одинаковыми заставками, картинками, звуками и т.д. И в процессе повседневной, рутинной работы мы забываем о том, что на рабочем столе стоят стандартные "обои", включается стандартная заставка, туда-сюда ходит стандартный курсор, играет стандартная музыка. А ведь каждый пользователь ПК желает, чтобы его машина была непохожей на другие.

Эта заметка как раз посвящена тому, как преобразить свой компью-

тер, она пригодится тем, у кого установлена операционная система Windows XP.

Начнем, пожалуй, с рабочего стола. Если вы хотите поместить на рабочий стол свой рисунок, то можно просто щелкнуть правой кнопкой мыши по рисунку и выбрать пункт меню "Сделать фоновым рисунком рабочего стола". Но если вы хотите часто переключать свои картинки, то вам придется проделать некоторые операции: нужно свои любимые рисунки скопировать в пап-

ку C:\WINDOWS\Web\Wallpaper. Теперь эти рисунки будут отображаться в списке фоновых рисунков рабочего стола (рис.1).

Следующим шагом будет создание собственных иконок (значков) и курсоров. Очень неплохо смотрятся анимированные курсоры. Для создания иконок и курсоров хорошо подходит программа AWIcons Lite 9.2.0 (<http://www.awicons.com>, free, 1547 Кб). У программы дружелюбный интерфейс, и с ней можно очень быстро научиться рисовать значки и курсо-



Рис. 1



ант: количество кадров не ограничено, это уже зависит от вашей фантазии. Со значками дело обстоит намного проще. Рисуете — сохраняете.

Чтобы заменить стандартный курсор мыши на свой собственный, нужно войти в Панель управления, выбрать пункт "Принтеры и другое оборудование"/"Мышь" (или просто "Мышь") и в появившейся форме — вкладку "Указатели". На-

появится список событий Windows. Звук для каждого события можно изменить, нажав на кнопку "Обзор". После внесения изменений в звуковую схему их нужно сохранить, нажав на кнопку "Сохранить как..." (рис.4). В любой момент можно вернуть стандартную схему, выбрав в списке тем "Стандартная Windows".

Внимание! Автор не несет ответственности за случайно удаленные по вине пользователя ПК системные файлы и папки и измененные системные настройки, не упомянутые в данной статье.

Рис. 3

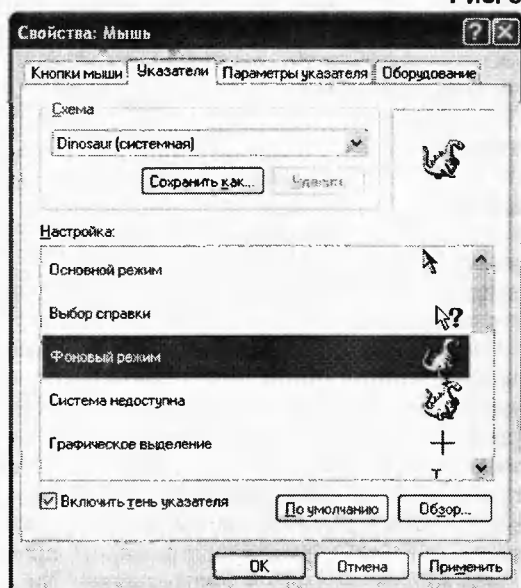


Рис. 2

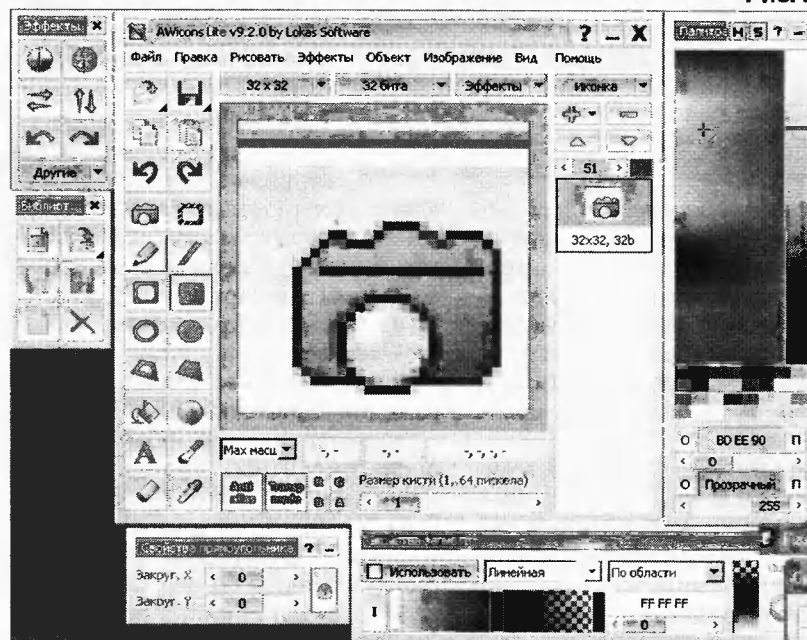


Рис. 4



ры. Чтобы нарисовать анимированный курсор, в меню "Файл" программы выбираем пункт "Новый ani-курсor". Перед вами появится серое поле для создания кадра. После того как первый кадр нарисован (рис.2), в меню "Объект" выбираем "Дублировать изображение". Появится второй такой же кадр, в котором можно произвести изменения. Теперь чуть выше голубого поля "Animated cursor preview" выберите для каждого из рисунков время смены кадров. Щелкните по этому полю. Вы увидите результат своей работы. Сохраните курсор, выбрав в меню "Файл" пункт "Сохраните объект". Для примера был рассмотрен очень простой вари-

жмте кнопку "Обзор" и выберите свой анимированный курсор (рис.3). Если захотите вернуть прежний курсор, просто нажмите кнопку "По умолчанию".

Ну, и наконец, займемся звуковой схемой. Изменить ее можно, войдя в Панель управления и выбрав пункт "Звук, речь и аудиоустройства". Далее выберите пункт "Изменить звуковую схе-

му". Появится список событий Windows. Звук для каждого события можно изменить, нажав на кнопку "Обзор". После внесения изменений в звуковую схему их нужно сохранить, нажав на кнопку "Сохранить как..." (рис.4). В любой момент можно вернуть стандартную схему, выбрав в списке тем "Стандартная Windows".



“Рабовладение” в iS-DOS на Sinclair

Е.ИЛЯСОВ,
г.Балашов
Саратовской обл.

Как известно, спецификация IDE (Integrated Drive Electronics) для интерфейса ATA (AT Attachment) предусматривает подключение и последующую совместную работу двух устройств с интегрированным IDE-контроллером. А расширенная спецификация E-IDE (Enhanced IDE) допускает совместное функционирование до четырех IDE-устройств. Но четыре IDE-устройства для Sinclair-совместимой машины, — как правило, ненужная жлобская роскошь. А вот для двух устройств — винчестера и CD-привода или двух винчестеров — вполне могут найтись целесообразные и оправданные применения.

Реальный, проработанный и программно поддерживаемый вариант использования двух винчестеров IDE “в одной упряжке” предоставляет для пользователей операционная система iS-DOS. Причем осуществляется это использование через абсолютно стандартный контроллер IDE-Drive разработки (c)Nemo, без каких-либо переделок или доработок. По большому счету, контроллер лишь предоставляет возможность обмена данными с IDE-устройством. А на одном шлейфе могут быть одновременно подключены и работать два IDE-устройства. И если оба эти устройства — винчестеры, то все аппаратные изменения сводятся к правильной установке переключателей-джамперов (Jumper), которые задают логику работы накопителей. Один винчестер должен быть выставлен переключками на нем как “Master”, а другой, естественно, как “Slave” (“раб”). Как правило, положение переключателей должно быть указано для каждого режима на наклейке, на корпусе винчестера, хотя в старых моделях это не обязательно. Встречались, например, винчестеры Seagate, у которых вообще ничего не указано на наклейке. В таких случаях приходится подбирать положение переключателей экспериментальным путем. Хорошо, если там всего лишь одна переключка, а если две или три — тогда их надо комбинировать. Но особых проблем быть, по идее, не должно. Единственная проблема, которая может встретиться — это если неверно выставить один из винчестеров переключками, тогда он будет конфликтовать со вторым, и, в итоге, откажутся работать оба.

Таким образом, вопрос только в программной поддержке. Второй винчестер выбирается именно программно. Среди регистров винчестера, доступных для программирования через IDE-контроллер, есть так называемый SDH-регистр. Изменяя маску, накладываемую на содержимое этого регистра, можно менять IDE-устройство, к которому будет происходить обращение через IDE-контроллер. Используя нужную маску SDH-регистра, можно переделать iS-DOS'ный драйвер винчестера таким образом, что этот драйвер станет работать со “slave”-накопителем. Так что, при необходимости, можно установить два винчестера и работать с ними — с каждым через свой драйвер. Это может найти применение, скажем, для быстрого копирования с одного винчестера на другой. Или на постоянной основе — для

использования второго накопителя в качестве архивного, для хранения нужной, но редко используемой информации. Необходимо уточнить, что в системе для нормальной работы нужно установить или один винчестер “Master”, или два винчестера — “Master” и “Slave”. Один винчестер “Slave” работать не станет. Возможно, это такая схемотехническая реализация контроллера (c)Nemo или изначальная особенность спецификации/интерфейса IDE/ATA. Но в любом случае, вряд ли это можно считать серьезной помехой для использования винчестера “Slave” под iS-DOS на Sinclair.

В качестве примера использования маски SDH-регистра для обращения к “Slave”-винчестеру здесь приводится комментированный ассемблерный текст сервисной утилиты статуса “freeware”, созданной на основе усовершенствованной версии “stanby.com” [1;43]. Формат исходника — iS-DOS Assembler.

```
-----
;
;                               SB_EXIT.COM
;
;
; Программа для торможения одного или двух винтов,
; выхода в Basic-48, TR-DOS, Монитор "JV KRAMIS" или
; в главное меню (аналогично нажатию кнопки <RESET>)
;
; Версия 1.73 от 15-го ноября 2004 года.
;
-----

ORG 24000                ;стартовый адрес

LD0 EQU #D0              ;метки портов винчестера
LF0 EQU #F0
LP1 EQU #1FFD            ;метки портов страниц RAM
LP7 EQU #7FFD
LPD EQU #DFFD

JR BEGIN

DEFM "UnCo"              ;вектор "униколора":
DEFB 2                   ;начальные цвета
UC1 DEFB 114             ;меню выхода в TRDOS
UC2 DEFB 1               ;при ассемблировании

BEGIN LD A,(UC1)          ;цвет меню
LD (WIND+4),A
LD C,#8A
LD A,(UC2)               ;цвет тени меню
LD (WIND+5),A
LD C,#8A
RST 16
EXX
DI                       ;запрещаем прерывания

FKEYS LD C,#43            ;ищем ключи
RST 16
RET C
RES 5,A                  ;приводим к одному регистру
JP NZ,STDBY              ;если нет - тормозим и на выход
CP "S"                   ;ключ "S"?
JP Z,STDBY               ;стоп-кран
JP KEYS2                 ;смотрим, какие есть еще

KEYS1 LD C,#43           ;ищем ключи
```



```

RST 16
RET C
RES 5,A ;приводим к одному регистру

KEYS2 CP "R" ;ключ "R"?
      JP Z,EX_GMN ;сброс (RESET)
      CP "T" ;ключ "T"?
      JP Z,EX_TRD ;выход в TR-DOS
      CP "B" ;ключ "B"?
      JP Z,EX_BAS ;выход в BASIC-48
      CP "M" ;ключ "M"?
      JP Z,EX_MON ;выход в Монитор "JV KRAMIS"

EX_IS XOR A ;обнуляем аккумулятор
      LD A,#F4 ;грузим код выхода
      EI ;разрешаем прерывания
      RET ;и выходим обратно в iSDOS.

STDBY IN A,(LF0) ;тормозим "винты",
      RLCA ;если они есть;
      JP C,STDBY
      LD A,#A0 ;stop "master"
      OUT (LD0),A
      LD A,#E0
      OUT (LF0),A
      LD A,#B0 ;stop "slave"
      OUT (LD0),A
      LD A,#E0
      OUT (LF0),A
      JP KEYS1 ;снова ищем ключи

EX_GMN LD BC,LP1 ;порт #1FFD Scorpion:
      LD A,0 ;ROM 0 на место RAM 0!
      OUT (C),A
      LD BC,LPD ;порт #DFFD Profi:
      LD A,0 ;ROM 0 на место RAM 0!
      OUT (C),A
      LD BC,LP7 ;порт #7FFD:
      LD A,0 ;ROM 0 — на место!
      OUT (C),A
      LD HL,0 ;и полный RESET.
      PUSH HL
      JP #0

EX_TRD LD IX,WIND ;вектор "окна"
      LD A,2
      LD C,$wt ;координаты окна
      RST 16
      LD C,$adrt ;координаты текста в окне
      RST 16
      EI ;разрешаем прерывания
      LD C,$key ;спрос клавиш:
      RST 16
      CP #10 ;если <SS/A>
      JR Z,EX_IS ;обратно в iSDOS;
      CP #20 ;или <Space> -
      JR Z,EX_IS ;обратно в iSDOS;
      JR NZ,RUN ;любая другая - в TR-DOS.

RUN LD BC,LP1 ;порт #1FFD Scorpion:
      LD A,0 ;ROM 0 на место RAM 0!
      OUT (C),A
      LD BC,LPD ;порт #DFFD Profi:
      LD A,0 ;ROM 0 на место RAM 0!
      OUT (C),A
      LD HL,0 ;и вылет в TR-DOS.
      PUSH HL
      JP #3D2F

EX_BAS LD BC,LP1 ;порт #1FFD Scorpion:
      LD A,0 ;ROM 0 на место RAM 0!
      OUT (C),A

```

```

LD BC,LPD ;порт #DFFD Profi:
LD A,0 ;ROM 0 на место RAM 0!
OUT (C),A
LD HL,0 ;и вылет в BASIC-48.
PUSH HL
JP #0

EX_MON LD BC,LP1 ;порт #1FFD Scorpion:
      LD A,10 ;ROM 0 на место RAM 0!
      OUT (C),A
      LD BC,LP7 ;установка ПЗУ ZX-128
      LD A,0
      OUT (C),A
      LD HL,0 ;и выход в Монитор JV "KRAMIS"
      PUSH HL
      JP #0

WIND DEFB 2,11,5,28,114,1,4,35 ;параметры окна
      DEFM | <SS/A> or <Space> - Quit|
      DEFB 13,13
      DEFM | <Any other key> - Exit to TR-DOS!|
      DEFB 3

```

;И на этом все! Всем спасибо.

;Alan Karin Riverman, 15.11.2004, понедельник.

Теперь, по традиции, шестнадцатеричная распечатка тела программы — для тех пользователей, кто по тем или иным причинам хотел бы получить работоспособную утилиту, не вникая в ассемблерные нюансы:

sb_exit.com, LSA:24000, Size: 292

block 0

```

#5DC0: 18 07 55 6E 43 6F 02 72
#5DCF: 01 3A C7 5D 32 9E 5E 0E
#5DD0: 8A 3A C8 5D 32 9F 5E 0E
#5DDF: 8A D7 D9 F3 0E 43 D7 D8
#5DE0: CB AF C2 0C 5E FE 53 CA
#5DEF: 0C 5E C3 F3 5D 0E 43 D7
#5DF0: D8 CB AF FE 52 CA 25 5E
#5DFF: FE 54 CA 41 5E FE 42 CA

#5E00: 70 5E FE 4D CA 85 5E AF
#5E0F: 3E F4 FB C9 DB F0 07 DA
#5E10: 0C 5E 3E A0 D3 D0 3E E0
#5E1F: D3 F0 3E B0 D3 D0 3E E0
#5E20: D3 F0 C3 ED 5D 01 FD 1F
#5E2F: 3E 00 ED 79 01 FD DF 3E
#5E30: 00 ED 79 01 FD 7F 3E 00
#5E3F: ED 79 21 00 00 E5 C3 00

#5E40: 00 DD 21 9A 5E 3E 02 0E
#5E4F: 61 D7 0E 66 D7 FB 0E 07
#5E50: D7 FE 10 28 B2 FE 20 28
#5E5F: AE 20 00 01 FD 1F 3E 00
#5E60: ED 79 01 FD DF 3E 00 ED
#5E6F: 79 21 00 00 E5 C3 2F 3D
#5E70: 01 FD 1F 3E 00 ED 79 01
#5E7F: FD DF 3E 00 ED 79 21 00

#5E80: 00 E5 C3 00 00 01 FD 1F
#5E8F: 3E 0A ED 79 01 FD 7F 3E
#5E90: 00 ED 79 21 00 00 E5 C3
#5E9F: 00 00 02 0B 05 1C 72 01
#5EA0: 04 23 20 20 20 20 20 3C
#5EAF: 53 53 2F 41 3E 20 6F 72
#5EB0: 20 3C 53 70 61 63 65 3E
#5EBF: 20 2D 20 51 75 69 74 0D

```



block 1

```
#5EC0: 0D 20 3C 41 6E 79 20 6F
#5ECF: 74 68 65 72 20 6B 65 79
#5ED0: 3E 20 2D 20 45 78 69 74
#5EDF: 20 74 6F 20 54 52 2D 44
#5EE0: 4F 53 21 03
```

Контрольная сумма в описателе полученного файла (если он не изменялся после набора дампа) — 34916 (#8864). Дата, соответственно, — 15.11.2004. Все это можно проконтролировать и, при необходимости, изменить в интерактивном режиме при помощи другой утилиты — "rename.com".

А теперь, в качестве "help"а, — краткое описание утилиты "sb_exit.com".

Данная программка тормозит до лучших времен один или два винчестера, если, конечно, они у вас есть (сначала тормозится "master", а за ним "slave"), а также обеспечивает выход, в зависимости от использованных ключей, в главное меню компьютера (аналогично нажатию кнопки RESET), или в TR-DOS, или в BASIC-48, или в монитор "JV" KRAMIS (если, конечно, у Вас компьютер KAY или модернизированный SCORPION ZS-256 с двумя ПЗУ) из системы iSDOS — как Classic, так и Chic.

Программа работоспособна в Chic'e на компьютерах KAY, SCORPION и PROFI, а в Classic'e — и на всех остальных. Срабатывают тормоза и "выезд на обочину" любым известным в iS-DOS способом: по "горячей" клавише, из меню и т.д.

Функции ключей

Ключ /s — тормоз винчестера перед выполнением основной функции.

Ключ /r — выход в главное меню (полный сброс — RESET); одновременно с ключом /s — с тормозом винчестера.

Ключ /t — выход в TR-DOS с запуском программы "boot" на диске A; одновременно с ключом /s — с тормозом винчестера.

Ключ /b — выход в BASIC-48; одновременно с ключом /s — с тормозом винчестера.

Ключ /m — выход в монитор JV "KRAMIS"; одновременно с ключом /s — с тормозом винчестера.

Сначала отрабатывается ключ /s (если он используется, то он должен стоять первым), а затем выполняется выход в указанном направлении. После отработки ключа /s следует команда установки на свое место нулевой банки ПЗУ, а затем отрабатывается команда, заданная вторым ключом:

- при выходе в главное меню происходит полный сброс компьютера, аналогичный нажатию кнопки <RESET>.

- при выходе в TR-DOS выполняется команда "RANDOMIZE USR 15619: RUN "boot", аналогичная той же в "EXIT_TRD.COM". Для того чтобы успеть сменить диск, если дисковод — системное устройство, выводится меню, и программа ожидает нажатия клавиши: <SS/A> или <Space> — выход обратно в iSDOS, любая другая клавиша — выход в TR-DOS;

- при выходе в BASIC-48 осуществляется переход на

адрес #0 ROM BANK 1, установленный по умолчанию в iSDOS;

- при выходе в монитор JV "KRAMIS" осуществляется переход на адрес #0 ROM BANK 0, в котором зашит "теневики".

Ключи можно задавать в виде /s/b или в виде /s /b, программе это безразлично. Регистр букв также не имеет значения.

Пример: Q:BOOT\sb_exit /s/b (выход в BASIC-48 с тормозом винта).

Если ключей нет ни одного, работа программы почти ничем не отличается от работы программы standby.com и во многих случаях может ее заменить.

Для устойчивого "торможения" винчестера и исключения его перезапуска при выходе из программы обратно в iS-DOS (когда, например, курсор находится в подкаталоге пятого-шестого уровня вложенности) лучше организовать вызов программы sb_exit с ключом /s через короткий bat-файл:

```
*L-
Q:SHELL\cache /29
CAT S:
Q:SHELL\pan /7
CAT S:
Q:SHELL\pan /7
Q:UTIL\sb_exit /s
```

Ну и, напоследок, можно напомнить, что эта программка, в лучших традициях фирмы (с)lskraSoft, является "клиентом" универсального настройщика цветов "unicolor.com", то есть позволяет настроить цвета единственного окна по вкусу пользователя.

При написании статьи использовались материалы: А.Леонтьева (г.Санкт-Петербург), С.Багана (г.Минск), А.Карныгина (г.Балашов), А.Кубрака и Е.Борисова (г.Старица).

Литература.

1. Е.Илясов. Тормоз для винчестера. — Радиомир. Ваш компьютер, 2003, №2, С.43.



Рисунок О.Попова



А.ПЕХИМЕНКО,
E-mail: paalex@inbox.ru

Обзор электронной прессы о Spectrum

Эта подборка ссылок появилась в результате моей обработки публикаций электронной Spectrum'овской прессы. На мой взгляд,

подобная подборка должна заинтересовать поклонников компьютера Spectrum — как железячников, так и программис-

тов. В этом материале в основном собрана информация о "железе" Spectrum (различных модификациях компьюте-

ров, портах, разных платах расширения, схемах доработок и т.д.), а также о схемах для радиолубителей.

Наименование	Тема
Adventurer1	Доработка компьютера ATM Turbo2+
Adventurer2	О работе с пленочной клавиатурой на Pentagon-128, доработка Sinclair-джойстиков для игры вдвоем, порт #FD на компьютере ATM TURBO 2+, стерео-ЦАП через АУ и программирование портов, АУ на ввод-вывод
Adventurer2	Как получить 2 HD, 32 сектора по 256 байтов, защита порта #FD
Adventurer3	Советы по ремонту и эксплуатации дисководов MC 5311(5313)
Adventurer7	Доработки компьютера Scorpion, схем COVOX, схема сопряжения АУ и COVOX
Adventurer9	Звуковая карта с прямым доступом
Adventurer11	Микросхемы программируемой логики, мышь
Adventurer12	FDD — новые возможности, турбо HD, защита порта #FD
Adventurer15	Схема контроллера жесткого диска 2
Alco news17	Схема соединения двух Spectrum'ов в сеть через АУ
Aspect 03	О компьютере Спрингер 97, доработка паяльника, принципиальная схема и монтажная плата усилителя на м/с K174УН4
Aspect 04	RS-фильтр для улучшения связи по модему
Aspect 06	Доработка компьютера Scorpion, одинаковое звучание каналов АУ
Baltic press	О дисководах SAMSUNG 5,25", о ZS-Scorpion-256
Black crow 1	Замена м/с РУ5 на РУ7, раскладка сигналов видеоконтроллера, доработка компьютера PROFI до Pentagon-512
Black crow 5	Питание для кэша, индикатор, слотовая система, буферизация, подключение контроллера дисковода к разным Spectrum, стандарты штекеров, о контроллерах винчестера
Black crow 6	Схема кемпстон-мыши на BB55 с разводкой платы, разводка платы под Soundrive v1.51+ на BB55
Black crow 7	ROM-диск, контроллер теневого ОЗУ, RS-232, команды CD-ROM, SIMM 4 Мб для компьютера Sintez-2
Black riwen2	Схема подключения HAYES-модема, схема турбирования CPU для компьютера Pentagon, схема турбирования м/с ВГ93, режим 512*192 точки
Black riwen3	Реализация кэша на HY62256ALP-10, описание ZX-дигитайзера, схема контроллера винчестера
Blackriwen 4	Схема VGA TV-converter for IBM, схема gigascreen для компьютера Pentagon
Body #01	Схема радиопередатчика с питанием 1,5 — 9 В на 50 м
Body #08	Схема радиопередатчика с питанием 1,5 — 12 В
Body #09	Схема тормоза компьютера на м/с 1533ЛН1 (изменение скважности импульсов на BUSRQ/)
Body #0D	Аппаратный мультиколор на компьютер "Байт"
Born#10	О ПК Pentagon 1024, о мыши, кэше, дисководах, TR-DOS
Buzz 4	Схема Covox
Buzz 5	Подключение Spectrum'a к телевизорам серии 700
Buzz 6	Блокировка порта #1FFD в компьютере Scorpion
Buzz 7	Схема подключения Kempston-джойстика на BB55
Buzz 18	Индикатор страниц ОЗУ, жучок в телефон
Bugs #2	Переделка джойстика от Sega в джойстик Spectrum, сканер из принтера
Dejavu 01	Исправление схемы подключения АУ в "старых" Scorpions
Dejavu 02	О XTR-модемах, о 3,5"-дисководах
Dejavu 05	Описание General Sound
Dejavu 06	Заставки от ПЗУ, схема для Pentagon, экран 512*192 точки
Dejavu 07	Подключение SIMM к ZX-Spectrum (буфер для SIMM), схема подключения SIMM 1 Мб к компьютерам Pentagon и Ленинград, схема дополнительного ОЗУ под ПЗУ, эмулятор ПЗУ, доработка Pentagon для работы с дисками высокой плотности
Dejavu 08	Схема часов на м/с 512ВН1, схема головки сканера, disc protector
Dejavu 09	Доработка контроллера Pentagon для работы с HD-дисками, дисковод 1,2 Мб, переделка "Taganrog 128" в "Taganrog 1024", программатор, о принтерах
Dejavu 10	Pentagon 1024, о дисководах
Dejavu 0A#	Схема порта #1FFD ON/OFF в ZS SCORPION-256, схема 128 цветов в ZS SCORPION-256/1024 TURBO+(ON/OFF), читаем порт #7FFD, индикатор треков, схема расширения ZS SCORPION-256 до 1 Мб на SIMM 1 Мб, дисковод 1 2 Mb, индикатор RD/WR, схема Kempston-мыши на BB-55, юстировка FDD 3,5 Мб
Depth 1	Доработки компьютера Scorpion, схема второго АУ
Depth 2	Схема 512*192 точки для Pentagon
Dni-pro 1	Схема Covox



Dni-pro 2	Порт на BB55 и его программирование, схема Kempston-mouse на BB55, разработка RGB-адаптера, фильтры улучшения модемного приема
Don news 10	Ремонт компьютера Pentagon
Don news 12	Как избавиться от wait в компьютере Taganrog 128, схема megascreen, как сделать из нее Pentagon
Echo-1	Подключение дисководов 3,5" и 5,25" (1,2 и 1,44 Мб)
Echo-2	О дисководах 3,5 и 5,25", доработки компьютера Балтик
Echo-3	Переключатель Profi/Pentagon на Profi
Echo-4	Super Autofire для компьютера "Байт"
Echo-5	Схема 128 цветов на черном фоне для компьютера "Байт", интерфейс на BB55
Echo-6	Схема Ultra DMA Sound Card с описанием работы, подключением, тестами, схема подключения 4-х дисководов, подключение принтера
Echo-7	Контроллер альтернативной памяти для компьютера "Байт", программирование ULTRA DMA SOUND CARD, схема подключения HAYES-модема
Energy 2	Устранение "тормозности" компьютера "Байт"
Energy 3	Доработка БП от компьютера "Байт"
Energy 4	Схема AZURE Soundrive v8.6
Energy 5	Разгон компьютера "Байт"
Energy 7	Схема C-DOS-модема
Energy 8	Описание C-DOS-модема и глюки схемы
Energy 8 5	Схема C-DOS-модема
Faultless 3	Доработки Spectrum
Faultless 9	Схема 128 цветов в Pentagon с полнейшим описанием и теорией
FlashTime 1	Схемы доработок Scorpions (AY, скорость), схема Soundrive 1.51, подключение 3,5"-дисковода
Full Pul1	Как сделать плату, джойстик от Sega к Spectrum, БП от IBM к Spectrum
Full Pul2	Как правильно паять, вентилятор на БП, дисководы с IBM 1,2 Мб на Spectrum
Full Pul3	Подключение CGA-, EGA-монитора к Spectrum'y
Full Pul4	БП +5 В/+12 В
Full Pul5	Ремонт клавиатуры
Full Pul7	Схема УМЗЧ 2*22 Вт на ТДА 1557, кулер
Full Pul10	О Scart-разъемах
Funeral 1 1	Схема Kempston mouse
Funeral 1 5	О Simm на компьютер Grm, турбо на компьютер Grm, телефонные карты
Hacker 1	1 Мб в Профи, решение проблемы с портом #FD
Hacker 6	Scar-разъем, подключение CD
Hacker 7	Схема Kempston mouse из электронного журнала Spectrofon
Impulse 8	Турбодовесок к компьютеру "Байт"
Impulse 9	Схема расширения памяти компьютера "Байт" до 1 Мб
Inferno 2	Ошибки АТМ, схема переделок, уменьшение шумов в телефонной линии, индикатор страниц ОЗУ, переделка палитры Spectrum в полноценную 16-цветную за счет бита Flash, прошивки ПЗУ, порядок файлов
Inferno 4	Активизация 0-го банка ПЗУ Pentagon, схема переделки Pentagon-128 для выхода по Reset в 0-й банк ПЗУ 27512, RGB-кодеры, расширенная клавиатура
Inferno 5	Мышь от Амиги, как переделать мышь, хвост для мыши, м/с 1533IE7, K561ТЛ1, 561IE10, защита м/с ВГ93 по питанию
Insty 07	Разводка шины PROF1
Issue 2	Схема компьютера ZX Scorpions 256, доработка Spectrum для работы с портами TR-DOS, схема второго окна проецирования памяти #S000-#BFFF
Izxnews 1	Схема Soundrive v1.02, схема Vicomm modem 3, Simm 1 Мб
Izxnews 5	Схема Vicomm3+ -модема
Izxnews 6	Схема переделки мыши от PC к Spectrum
Izxnews09	128 цветов на Spectrum, схема 2-х экранов
Izxnews10	Схема счетчика дорожек
Izxnews13	Три дисковода в компьютере Scorpion, ISA-слот на ZX
Knews 00	Жучок
Knews 01	Доработки компьютера
Knews 05	VICOMM modem с подъемом трубки, тормоз для процессора
Knews 10	Схема подключения AY
Knews 11	О flashcolor
Knews 14	Блокиратор порта #1FFD
Knews 16	Flashcolor для компьютера KAY, Scorpion
Lprint 3	Доработка C-DOS-модема, установка реле
Lprint 12	Доработка C-DOS-модема для поднятия трубки
Lprint 13	Схема теневого ОЗУ 32 Кб, схема входа в теневое ОЗУ по магии
Lprint 23	Схема расширения компьютера Балтик до 512 Кб
Nicron 6-25	Микропроцессор Z380
Nicron 24	Описание выводов и программирование м/с BB55

(Окончание следует)



Из истории компьютерных игр, или почему же IBM, а не ЕС?

Р.КАРПАЧ,

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

Однажды по каналу "Discovery" показали замечательный документальный фильм о зарождении и появлении популярных компьютерных игр. Все было превосходно, если бы не одно "НО", авторы ни разу не упомянули платформу, которая в начале 90-х была не менее популярной, чем ZX, NES, Commodore, — IBM PC. А ведь многие хитовые игры были написаны именно для нее, а уж потом продолжали свое шествие в других тогдашних системах.

Для чего все это нужно? Зачем вспоминать старые, на сегодняшний день ничемные по графике игры? Да потому, что их клоны, новые версии, называйте как хотите, выходят и по сей день. В разных вариациях, от разных производителей, но это все тот же Тетрис, тот же Warcraft, тот же Wolf, тот же Doom, и так далее, и тому подобное.

К сожалению, место в журнале ограничено, это же не канал "Discovery"! Поэтому речь сегодня пойдет о двух революционных игровых программах, без которых в начале 90-х невозможно было представить жесткий диск IBM-совместимого компьютера.

"Тетрис" (USSR)

Игра "Тетрис" была впервые написана Алексеем Пажитновым в июне 1985 г. на компьютере "Электроника-60" в ВЦ Академии наук. Шестнадцатилетний школьник Вадим Герасимов переписал игру для IBM PC.

Программа быстро распространилась по всей Москве. Когда игра дошла до Будапешта, венгерские программисты реализовали ее на разных платформах, и "Тетрис" был "обнаружен" британской софтверной компанией "Andromeda". Они попытались купить у Пажитнова права на версию для PC, но к тому моменту права уже были проданы компании "Spectrum Holobyte". После этого "Andromeda" попыталась купить права у венгерских программистов.

В 1986 г. "Spectrum Holobyte" выпустила версию для IBM PC в США, популярность игры была огромной, она сразу стала бестселлером.

Позднее ситуация с "Тетрисом" стала напоминать детектив. В 1987 г.

"Andromeda" заявила о своих правах на игру для PC и любых других домашних компьютеров. В 1988 г. советское правительство заявило о своих правах на "Тетрис" через организацию "Электроноргтехника" (ЭЛОРГ). К этому моменту ни Пажитнов, ни ЭЛОРГ не получили никаких денег от "Андромеды", но компания заявляла о своих правах на "Тетрис" и продавала лицензии другим компаниям. К 1989 г. полдюжины разных компаний заявили о своих правах на разные версии игры для разных компьютеров, игровых консолей и карманных электронных игрушек.

ЭЛОРГ заявила, что эти компании не имеют никаких прав на версии для игровых автоматов, и предоставила эти права компании "Atari Games", а права на версии для игровых консолей и карманных электронных игрушек — компании "Nintendo".

Однако "Tengen" (подразделение компании "Atari Games", занимающееся программным обеспечением для игровых консолей), выпустила свой вариант игры для консоли Nintendo Entertainment System, игнорируя соглашение. Многие игроки сочли версию "Тенген" лучше версии "Nintendo". Игру называли "Tetris". Но "Nintendo" подала на "Tengen" в суд и выиграла процесс. Всего через несколько месяцев после выпуска "Tetris" игру пришлось отозвать, продав всего 50000 экземпляров.

"Nintendo" выпустила "Тетрис" для консолей "Famicom" и "Game Boy" (версию для "Game Boy" написала по их заказу компания "Bullet-Proof Software", и эта версия стала, вероятно, самой известной версией "Тетрис") и продала более 3000000 копий. Судебные тяжбы между "Nintendo" и "Atari Games" по поводу версий для "Famicom" и "NES" продолжались до 1993 г.

Фирма "Nintendo" получила неплохую прибыль, но сам Алексей Пажитнов получил очень мало денег от сделки. В 1996 г. он с Хэнком Роджерсом (Henk Rogers) создал компании "The Tetris Company LLC" и "Blue Planet Software", пытаясь получить прибыль от бренда "Тетрис". "The Tetris Company LLC" (TTC) зарегистрировала слово "Tetris" как торговую марку. С тех пор несколько компаний

купили у TTC лицензию на торговую марку, но законность игр тетрамино, которые не используют название "Tetris", в суде не оспаривалась.

По американским законам, игру нельзя защитить копирайтом (только запатентовать), поэтому основным имуществом компании является торговая марка "Tetris".

Примечателен тот факт, что проблемы у создателя "Тетриса" были вызваны вовсе не наглостью отдельных болгарских компаний, а ситуацией в стране, в которой жил Алексей. В СССР не было такого понятия как авторское право, все принадлежало народу. Вспомним хотя бы историю со знаменитым олимпийским медвежонком 1981 г. Поэтому не удивителен тот факт, что "нашедшие" игру болгарские "друзья" просто "прихватизировали" ее, в надежде, что автор "Тетриса" никогда не объявится. Но популярность игры оказалась ошеломляющей, поэтому не удивительно, что правами на нее заинтересовались акулы игрового бизнеса. Так одна игра и один человек смогли переполнить полмира.

Wing Commander (USA)

Крис Робертс (Chris Roberts) не похож на обычного американца, он родился в Манчестере (Великобритания). В этом городе он прожил до 19 лет. Отец Криса преподавал социологию в университете. Когда парню было около 12 лет, дома появился компьютер — Commodore PET, и естественно, первым языком программирования для Криса стал BASIC. Сразу после освоения азов BASIC, Крис увлекся программированием движущихся объектов и в 13 лет написал свою первую игру о посадке вертолета на площадку. Позже в семье появился компьютер BBC с 32 Кб памяти и графическим разрешением видеосистемы 320x200x4 цвета (или 160x200x16 цветов). Именно эти возможности увлекли парня еще больше.

Один из старых учителей Криса Робертса стал в это время редактором журнала для пользователей компьютеров BBC "BBC Microuser". В те времена все уважающие себя журналы публиковали листинги программ своих читателей, и однажды он



позвонил Крису с предложением написать какую-нибудь игру. В ответ Крис написал игру "Kong" — гигантская обезьяна должна была залезать на здания и сбивать камнями вертолеты. "BBC Microuser" заплатил за нее 100 фунтов. Крису понравилось зарабатывать деньги своим хобби, и он создал еще ряд игр, написанных на языке BASIC.

Вскоре BASIC перестал его удовлетворять, так как для программ на нем в компьютере выделялось всего 12 Кб памяти. Так Крис пришел к ассемблеру. Один из друзей отдал ему недоделанный вариант игры, которую он никак не мог заставить работать. Крис потратил уйму времени на изучение принципов ее работы, а затем практически заново переписал. Это был "Wizardor" — обычная аркада с горизонтальным скроллингом, где главный герой должен был добраться до замка с драконом и убить его, собрав по пути различные вещи. На этот раз Крис решил продать игру серьезной фирме по производству игр. После долгих раздумий "Wizardor" был продан компании "Ocean". В 1984 г. эта игра вошла в списки игр-бестселлеров для компьютеров BBC. Достаточно большая денежная сумма приободрила юного программиста, и он решил продолжать заниматься созданием коммерческих игр.

После "Wizardor" последовала игра "Match Day" — футбольный симулятор. Вскоре ф."BBC Microcomputer" выпустила свою новую серию компьютеров "Master series", имевших RAM объемом 128 Кб. Игра "Striker's Run" — еще одна аркада — использовала на полную мощь возможности нового ПК. Ко времени создания этой игры Крис окончил школу и начал учиться в колледже. Но перед поступлением в университет г.Манчестер на специальность "Физика" парень подумал, что неплохо бы год отдохнуть и посвятить его написанию игр, хорошо при этом заработав.

Количество пользователей компьютеров BBC составляло что-то около 500000 человек, и Крис переключился на более распространенную модель — Commodore 64, начав писать "Ultra Realm" — предшественника "Times of Lore". Крису хотелось создать что-нибудь похожее на популярный тогда "Gauntlet", добавив герою возможность общаться с другими персонажами, собирать и использовать различные объекты и т.п.

В 1985 г. отцу Криса предложили место профессора социологии в университете штата Техас. Так вся семья Робертсов оказалась в США, в городе Остин (Austin).

Крис продолжил работу над "Ultra Realm". Сидя однажды в клубе любителей role-playing games, он заметил на стене большую картину, изображавшую сражающегося гладиатора, и спросил у знакомых, кто ее автор. Так Крис познакомился с Дэнисом Лубетом (Dennis Lubet), свободным художником и большим любителем фэнтези. Он попросил его заняться разработкой графики для игры.

Через некоторое время Дэнис был приглашен Ричардом Гэрриотом (Richard Garriot) для разработки упаковок игр серии "Ultima". Вскоре и Крис познакомился с ним и созданный Ричардом и его братом компанией "Origin". К этому времени "Ultra Realm" уже была закончена и предложена трем компаниям: "Electronic Arts", "Broderbund" и "Origin". Все три изъявили желание приобрести эту игру, и после долгих сомнений Крис продал ее "Origin", а вскоре и сам стал работать в этой компании. Переработанная для IBM PC версия "Ultra Realm" стала называться "Times of Lore".

Через некоторое время ф."Microprose" выпустила свой суперхит "F-19 Stealth fighter". Для реализации всех возможностей игры рекомендовалась машина с 386-м процессором, стоившая тогда около 15000 долларов. Тем не менее, почти не было людей, не приобретших теми или иными путями эту игру, в которую играли даже на старых XT с CGA-монитором. Тогда же Крис понял одну важную вещь — люди не спрашивают, какая игра лучше всего, скажем, для CGA, их интересует просто ЛУЧШАЯ игра, и неважно, что для этого требуется. Так родилась идея создания игры, максимально использовавшей все имеющиеся возможности компьютера. "Wing Commander" создавался именно в расчете на лучшие образцы техники того времени.

К сведению, самой первой игрой, использующей 3D bitmap, был симулятор "BattleHawks 1942" компании "LucasFilm". Крис использовал аналогичный подход к графике. Сначала с помощью программы "Sculpt 3D" на Commodore Amiga были проиграны предполагаемые картинки. "О! Это было класс!", — единогласно заявили все сотрудники "Origin", видевшие эти

наброски, после чего группа начала разработку. Масштабирование bitmap-образов было достигнуто за счет двухмесячного программирования в кодах по 16 часов ежедневно. Команда разработала свой собственный алгоритм, не требовавший громоздких математических расчетов. Результатом этой гигантской работы явилось около 600 Кб исходного текста на ассемблере.

При создании "Wing Commander" применялись ассемблер и C. Для "Strike Comander" "Origin" перешла на C++, при этом игра не использует 32-разрядную адресацию, но применяет 32-разрядные вычисления. Когда "Origin" начала разрабатывать "Wing Commander II", Крис сильно сомневался в том, что игра будет иметь такой же успех, как и "Wing Commander I" (ведь согласитесь, ничего особенно нового в ней не было). Но число копий игры, проданных за первую неделю продаж, показало обратное.

Вместо заключения

Насколько сложна и противоречива сама жизнь, настолько же судьба человека зависит от страны, в которой он живет. Алексей Пажигов сейчас является гражданином США, в конце 90-х годов он так и получил отчисления в свой адрес от компании "Nintendo". И теперь, я полагаю, живет, по крайней мере, не бедно. Крис Робертс продолжил разработку компьютерных игр. Когда-то СССР не смог удержать талантливых программистов у себя — видимо, не было ни условий, ни желания сделать это. Теперь, глядя на достижения индустрии IT на Западе и Востоке, власти многих стран СНГ загорелись желанием создать "Технопарки", "Силиконовые долины". Видимо, в надежде на то, что именно они поднимут скатившиеся в пропасть экономики бывших советских республик. Но, по моему глубокому убеждению, то, что создавалось десятилетиями, одним желанием и административно-правовыми актами не решишь. Как не могут громоздкие заводы конкурировать с конкретно-ориентированными малыми предприятиями, моментально реагирующими на изменение конъюнктуры IT-рынка. А ведь удачным примером выгодной узкой специализации и являются фирмы, создающие компьютерные игры!

*При создании публикации были использованы материалы сайта
www.romov.net*



А.ВЕНДИЛОВСКИЙ,
г.Минск.

Lord of the Rings. Battle of Middle-Earth

Над Средиземьем взмахнули огромные черные крылья, и грохот "Rammstein" прокатился аж до самого Минас Тирита, но даже музыка не

смогла заглушить тяжелый вопль, от которого холодело сердце — как говорится, не дай Бог такое похмелье, которым страдал пролетающий Назгул. Покачивая своей невидимой головой, он устало согнулся в седле — да уж, все эти переводы Гоблина до добра не доведут! Их послали в Шир за потерянными имуществами, и, преследуя носителя этого имущества, они остановились в кабаке "Гарцующий Пони", где позволили себе немного расслабиться вместе с Гендальфом.

После десятой бутылки все помнилось смутно — разгромленный номер гостиницы, требование продолжения банкета, поход за самогоном в курганы Умертвевших. Выяснение с Гендальфом отношений по поводу того, кто является более крутым магом, из-за чего половина леса была сожжена. Драка с каким-то бомжеватым придурком, который и меч-то где-то успел поломать. А позже — купание в реке на глазах у эльфов! Война началась, и обе стороны уже получили неплохие прибыли, продавая оружие третьим странам, которые еще не участвовали в конфликте, но хотели бы усилить оборону собственных границ. В это же время Гендальф наладил производство подделок Кольца Всевластия и вовсю промышлял этим бизнесом, несмотря на все протесты Саурона и длительное судебное разбирательство о праве владения на патенты по производству Волшебных Колец и Кольца Всевластия в частности. Выиграли от

этого пока что лишь адвокаты. В ответ Барад-Дур открыл туристический маршрут по Мордору, и теперь каждый желающий всего за 100 баксов мог

ветствующей пиротехникой, на который втихаря инкогнито прибывали даже Валары! А тут еще эта игра...

О ней много писали, и все же на греб-

не огромного количества уже вышедших игр на Средиземную тематику она запросто могла потеряться в толпе себе подобных. И я рад заявить, что, Слава Богам Потерянных Кластеров, этого не случилось.

Первый взгляд. Ура, нам позволяют играть и на стороне Саурона! Не подумайте чего, я всем сердцем на стороне Добра и Света, но иногда задумываешься, что такое милое колечко как Кольцо Всевластия совсем неплохо смотрелось бы и на собственной руке. Да

и как можно было совершать такие ошибки? Уж я бы им показал... А там и небольшое королевство с видом на море было бы тоже неплохо, а то зтот альтруизм светлых как-то набил оскомину... Ладно, отвлекся.

Графическая часть показала себя достойно, так как игра опиралась на фильм (кадры из фильма, голоса персонажей — прилагаются, альтернативный перевод Гоблина — возможен:), то и все персонажи похожи на тех, что мы видели в кинотрилогии. И я должен отметить, что все ключевые персонажи

очень легко различить в толпе себе подобных. Особо поразивших взор спецэффектов я не заметил, но для себя отметил, что, даже при средней настройке для максимальной производительности, местность и персонажи не превратились в хаотическое нагромождение пикселей (хотя, должен откровенно заявить, для максимально красивой картинки требуется ОЧЕНЬ мощная система).

Интерфейс нареканий не вызвал — прост и удобен, все необходимое для управления всегда под рукой, хватает нескольких минут, чтобы



бросить свое кольцо в недра Огненной Горы, примерить орчьи доспехи, посмотреть на Алмазные Ворота Барад-Дура и Башню Глаза. Толпы народа повалили со всех уголков Средиземья, и целые армии орков убрали за ними мусор, чтобы вернуть местности первозданно-зловонный вид. Время от времени приходилось прочищать Роковую Расщелину в горе — многие кольца даже не успевали расплавиться и забивали шахты. Не говоря уж о ежегодном концерте "Rammstein" в Минас-Моргуле с соот-



в нем разобраться и в дальнейшем не отвлекаться, посвятив все время игре.

Сама игра напоминает бешеную смесь всех хитов и времен. Наши войска переходят с уровня на уровень и прокачиваются (вот она, вездесущая РПГ-шная часть) вместе с главными персонажами. При накоплении очков силы, мы можем потратить их на дополнительные заклинания Кольца (или Деревя для светлых), которыми позже будем швыряться с небес (небольшая ремарка: вырубание деревьев с помощью Кольца — это несколько забавно...).

От "Дюны" досталась разбитая на лоскутки карта и возможность самому выбирать, на кого и когда нападать. А участки — это не просто захваченная территория, это дополнительные очки силы Кольца, а также этим определяется максимальное количество ваших войск на поле битвы и тех, кто перейдет с вами на следующий уровень. От "Дисайплов" в наследство перешло строительство — строить мы можем только там, где стоят специальные указатели, и никакой самостоятельности! Одна радость — битвы у нас многочисленные, и чтобы не тратить время на каждого воина в отдельности, строится и отдается под управление сразу целый отряд. Да, и конечно, апгрейды. Если с бойцами все понятно (выскочили, дал команду апгрейдиться, отправил дальше), то город меня несколько удивил. За каждого убитого сторожевыми башнями противника он тоже получает очки опыта, впрочем, как и за количество произведенных солдат. И когда это количество достигнет необходимого предела, здание ненавязчиво сообщает, что оно теперь круче, чем вчера :).

Любой RTS-ник уже бы задал вопрос: "А ресурсы в игре какие?" Наследие ЦыЦа и Варкрафта дало свои плоды. Ресурс один — деньги! Но получать их можно, перерабатывая сталь, мясо или древесину. Для мяса и древесины достаточно просто поставить специальные здания в городе, и они будут тихо и спокойно приносить стабильный доход. А вот с лесопилкой и рабочими придется помучиться первые несколько уровней.

Но когда прокачаете Кольцо до уровня "Рубка Леса", финансовые проблемы перестанут особо мучить, вернее, при правильной политике о них забудешь вообще.

AI и тактическая часть... Сразу вспоминаешь слова классика: "Сейчас рулят танковые клинья, а глазом сверху зыркать — это глупо". Единственное, что спасает игру от самой примитивной тактики "танкового кулака", это четкое позиционирование персонажей —



ваших меченосцев и арбалетчиков в прямом смысле раскидает во все стороны один, даже небольшой отряд конников, но Боже их упаси нарваться на сомкнутый строй копыеносцев — их тут же порубят в фарш. Страшный сон любого копыеносца — те же лучники на конях... И вот тут очень может пригодиться пехота.

Казалось бы, на этой ноте нужно закончить восхваления диагнозом — Хит! Если бы не огромная ложка дегтя, которая напрочь портит всю радужную картинку. Баланс, о котором так много говорилось выше, работает лишь в случае прохождения игры методом "кавалерийского наскока", сознательно не прокачивая персонажи и войска. Тогда игра становится интересной и... очень короткой. Если же вы будете вести себя как полноценный стратег, займетесь прокачиванием персонажей, Кольца и увеличением количества собственных войск, не говоря уже о правильном подборе заклинаний Кольца, то очень скоро, не пройдя даже середины, вы поймете, что прохождение становится ОЧЕНЬ простым. Заклинания Кольца решают все финансовые проблемы настолько, что вам было бы

проще выкупить все строения врага, чем воевать с ним :(. Придушив противника настолько, чтобы он не рыпался, но и миссия не заканчивалась, вы спокойно налаживаете конвейерное производство ваших солдат и одновременно превращаете их в Силу Кольца (деньги все равно на следующий уровень не переходят), а Саруман в это время "прокачивает" уровни регулярной армии. Военных заклинаний достаточно, чтобы выпутаться из до-

вольной сложной скриптовой передеряги. А когда появляется Барлог... Хотя он и появляется на короткое время, но запросто превращает любую нападающую армию в отчаянно обороняющуюся, а чуть позже и хорошо прожаренную, или может взломать защиту любой цитадели и практически стереть ее с лица земли. Если под прикрытием Барлога еще бросить пару отрядов наездников на волколаках, то на сопротивлении врага можно ставить большой и жирный крест. Назгулы прекрасно разгоняют толпы противников, давая вам время на подброску

резерва, и легко зачищают стены крепостей от катапульта. Так что последняя миссия по захвату Минаса Тирита прошла не как тяжелая битва, где против тебя выступали ВСЕ герои противника, а как легкая прогулка в окрестностях Барад-Дура. В миссиях за светлых — ситуация та же, отличающаяся лишь тем, что очень часто войска даются уже прокаченными.

Вывод — игра развлечет огромное количество фанатов книги и фильма, подарит несколько приятных вечеров и... вероятно, большинством игроков будет забыта, за исключением тех, кто будет использовать сетевую игру.

Рекомендуемые системные требования:

- Windows XP or Windows 2000;
- 1.3 GHz Intel Pentium IV or AMD Athlon processor;
- 256 MB RAM;
- Nvidia GeForce2 or equivalent video card with 32 MB RAM;
- DirectX 9.0b compatible sound card;
- 4 GB available hard disk space.

Диск предоставлен магазином "MediaCraft" www.mediacraft.shop.by



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

Для публикации бесплатных объявлений **некоммерческого характера** о покупке и продаже компьютерных комплектующих и программ, их текст можно присылать в письмо 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

Продаю литературу для ZX SPECTRUM.
Владимир.
E-mail: ua1ogy@atnet.ru

Ученик 9-го класса примет в дар любой отечественный компьютер (можно Spectrum), желательно в рабочем состоянии.

Тел. (0942)-41-12-36. Сергей.

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

Продам книгу Гомера Л. Дэвидсона "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



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. KB 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. KB 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. Наложенным платежом журналы не высылаются.

Наши платежные реквизиты:

- для жителей России и стран СНГ (кроме Беларуси) —
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-т, дом 76, корп. 4;
- для жителей Беларуси —
получатель: УП "РЛД", УНН 190218688
р/с 3012000004882 в Ф-ле №524 АСБ "Беларусбанк" в г.Минске код 121.
Адрес банка: 220028, г.Минск, ул. Физкультурная, 31.

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)		
	Радиомир	Радиомир. Ваш компьютер	Радиомир. KB и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2001	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@aiif.ru, <http://www.interpress-express.com>.

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

- г.Москва, ул.Глиаровского, 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.Нагатинская, д.6, эт.1, оф. VII (ст. метро "Нагатинская"), 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.