

радиомир

5 • 2004

Ваш компьютер

Microsoft
Windows xp
Professional SP1
русские версии

Microsoft
Office xp
Professional SP2



+FrontPage
РУССКАЯ ВЕРСИЯ
Загрузочный диск

ШКОЛЬНАЯ ПРОГРАММА В ОТВЕТАХ И РЕШЕНИЯХ

Мне не
спешка! Это
такое решение!

русский язык
алгебра
геометрия
физика



7
КЛА

УНИВЕРСАЛЬНЫЙ
4 СУПЕРЗАГРУЗОЧНЫЙ 2004



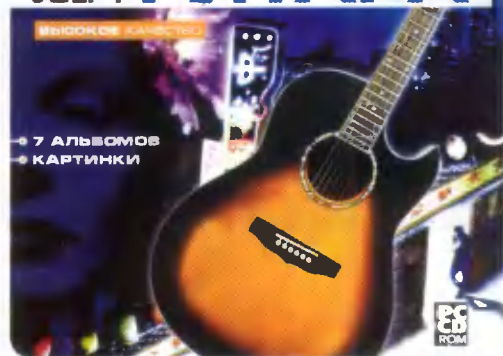
CORELDRAW 11
ADOBE ILLUSTRATOR CS
ADOBE STREAMLINE 4.0
MACROMEDIA FRESHMAN MX V11.0

ВЕКТОРНЫЕ
ГРАФИЧЕСКИЕ
РЕДАКТОРЫ

ВСЁ ДЛЯ СОТОВЫХ ТЕЛЕФОН SAMSUNG

- A800, T100, T400, S100, S200, S300, C100, Q200, V200, N600, N620
- Easy GPRS
- загрузка мелодий и картинок
- синхронизация с PC
- коллекция мелодий звонков

MP3 192K 44 KHZ STEREO
ТОЛЬКО ЛУЧШИЕ АЛЬБОМЫ
РОМАНТИЧЕСКАЯ
VOL. 1 ГИТАРА



7 АЛЬБОМОВ
КАРТИНКИ

ABBYY
SOFTWARE
HOUSE

RETROSOFT

>>> В ПОМОЩЬ ПОЛЬЗОВАТЕЛЮ

Киностудия
на твоём компьютере

СЛОВАРЬ: АНГЛО-РУССКИЙ
РУССКО-АНГЛИЙСКИЙ
Lingvo • 8.0



Adobe Premiere
Canopus
CyberLink Power Director
CyberLink Power VCR II 3.0.122
In-Sync
M2-edit
MPEG Softengines
Pinnacle
Edit
SONIC FOUNDRY VEGAS
Sonic
TMPGEnc Plus
DVD
Ulead
Morgan LSI MPE
QuickTime

библиотека фотоизображений ArtToday

SPORTS

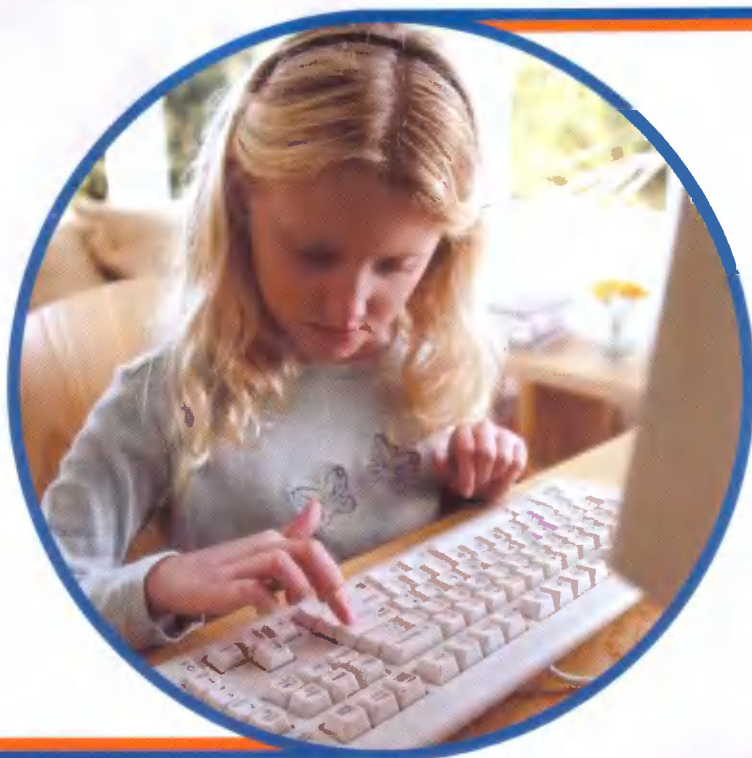
disc30



Компьютеры для дома и офиса

от компании **СитиПринт**

Интернет и правильно выбранный компьютер
раскроют перед вашими детьми
горизонты новых знаний.



Позаботьтесь об образовании ваших детей.

Предоставьте им возможность искать новую
информацию в сети, работать с
обучающим ПО и писать рефераты.

Приобретите компьютеры от компании
“СитиПринт” на базе процессора
Intel® Pentium® 4 с технологией HT
уже сегодня.



Учредитель и издатель:
ООО "Радиомир Пресс"

Свидетельство о регистрации
ПИ №77-9841 от 17.09.2001

Главный редактор
Ольга СТРИЖАНКОВА

Зам. гл. редактора
Иван БЕЛЬСКИЙ

Редколлегия:
Сергей ДРОЗДОВСКИЙ,
Алексей КОНДРАШОВ,
Анатолий КОРБИТ,
Владимир КУЦЕНКО,
Елена ЛЕВИТМАН,
Николай МЕДВЕДЬ,
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,
Александр ГОРАЮТИН,
Мария КАЛАБУХОВА

Оформление обложки —
Наталья БЕЛЬСКАЯ,
Надажда 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
рекламного 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
публикуемой информации несут ответ-
ственность рекламодатели и авторы.
Мнение редакции не всегда совпадает
с мнениями авторов

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

Подписано к печати 29.03.2004 г.
Формат 60х84 1/8.
Печать офсетная. 6 печ. л.
Цена свободная.

Отпечатано в типографии
ЗАО "Красногорская типография".
Заказ № 1080.

© ООО "Радиомир Пресс"

радиомир Ваш компьютер

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

№5/май/2004

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

НЕ ТОЛЬКО НОВИЧКУ

В.КУЦ. Сегодня и завтра компьютерных корпусов	2
В.КИРИЧЕНКО. Как сделать загрузочную заставку Windows	5
А.ГРИНЧУК. Анимация с Macromedia Flash MX Специальные типы анимации	7
А.ГОРЯЧКИН. Ifan-Тюльпан	10
Р.КАРПАЧ. Полезное и приятное. Для кого как	12
Е.ЛЕБЕДЕВ. Новый интерфейс	13
С.РЮМИК. Интернет-калейдоскоп, freeware #6	14

КОММУНИКАЦИИ

В.КУЦ. Факс в современном офисе — теперь это очень просто!	16
--	----

ДАЙДЖЕСТ

18

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

А.ИВАНЧИКОВ. COM — это не сон	22
А.КОРБИТ, А.ЩЕРБАКОВ. Программирование в среде Visual C++ 6.0. Элемент управления Tree Control	25

ДИАЛОГ ПРОГРАММИСТОВ

А. КАЛЮТОВ. Фрактальный анализ и увеличение изображений	27
CyberManiac /HI-TECH. Теоретические основы крэкинга	30
И.РОЩИН. Программа для разрезания файлов	32
И.ШИРКО. Microsoft Agent Часть III	34

РЕЦЕПТЫ

А.ПАВЛОВ. Терморегулятор для управления дополнительным вентилятором охлаждения	37
---	----

МИР 8 БИТ

Е.ИЛЯСОВ. Maxi-Editor	40
КОМПЬЮТЕРНЫЕ ХРОНИКИ	43

ИГРОТЕКА

А.ВЕНДИЛОВСКИЙ. SpellForse: The Order of Dawn	44
М.ВАЛЬПА. Lionheart: Legacy of the crusader	46

ДОСКА ОБЪЯВЛЕНИЙ

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

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



В.КУЦ,

E-mail: victor_kootz@mail.ru

Сегодня и завтра компьютерных корпусов

Корпус для компьютера является не только “упаковочным ящиком”, но и достаточно важным элементом, обеспечивающим размещение и жесткую фиксацию всех его устройств, их электропитание и защиту довольно хрупких “внутренностей” от воздействия окружающей среды. Еще совсем недавно практически все компьютерные корпуса, за исключением отдельных дорогих “брендов”, выглядели как близнецы-братья, поражая взгляд своим унылым однообразием одинаково серых коробок. Но в последнее время все резко изменилось — современные компьютерные корпуса обрели яркий, броский вид, обогатившись при этом целым рядом новых для них функций.

Практически все современные компьютерные корпуса соответствуют стандарту ATX. Наиболее широко распространены корпуса башенного типа (Tower), которые подразделяются (достаточно условно) на Micro-, Mini-, Midi- и Big-Tower, различающиеся по числу отсеков для 5,25"-накопителей. Соответственно корпуса типа Micro-Tower имеют 1 посадочное место под такие накопители, Mini-Tower — 2, Midi-Tower — 3 или 4 и Big-Tower — 5 и более. Самый распространенный сегодня формат корпуса — Midi (Middle)-Tower ATX обеспечивает использование большого количества накопителей и практически всех типоразмеров системных плат при приемлемых габаритных размерах самого корпуса. Разделение на Midi и Middle является условным, однако в последнее время сложилась практика обозначать полноразмерные корпуса с блоком питания, расположенным горизонтально, как Midi-Tower ATX. При всех своих достоинствах такие корпуса имеют и недостаток — большие габаритные размеры, особенно в высоту. С целью ее уменьшения в корпусах Middle-Tower блок питания развернут на 90 градусов и расположен параллельно системной плате. Такое решение, при незначительном увеличении ширины системного блока, позволило резко уменьшить его высоту и, сохраняя возможность установки трех 5,25" внешних устройств, свойственную корпусам типа Midi, вплотную приблизиться к типоразмеру Mini-Tower. В этом случае вентилятор БП расположен непосредственно над процессором, и процесс

отвода горячего воздуха от системы охлаждения CPU, по задумке разработчиков, облегчается. Но любая “палка” — “о двух концах”. В данном случае “концом”, бьющим по пользователю, выступает малая высота свободного пространства над системной платой, часто делающая невозможной установку крупногабаритных процессорных кулеров. Вторым недостатком такого решения является ухудшение общей циркуляции воздуха в корпусе, так как вентилятор БП работает в замкнутом пространстве, отгороженном спланированной корзиной с накопителями и шлейфами к ним, а снизу — платой видеоадаптера. Кроме того, погоня за уменьшением габаритов корпуса привела к тому, что большинство корпусов Middle-Tower позволяют устанавливать без особых проблем системные платы ATX размером 305 x 210...220 мм, а полноразмерные системные платы формата ATX (напомню, имеющие размеры 305 x 245 мм) часто не позволяют установить 5-дюймовые дисководы CD-ROM. Правда, большинство современных системных плат укладываются в эти ограничения, но, тем не менее, владельцам корпусов типа Middle-Tower надо быть внимательными при выборе новой материнской платы.

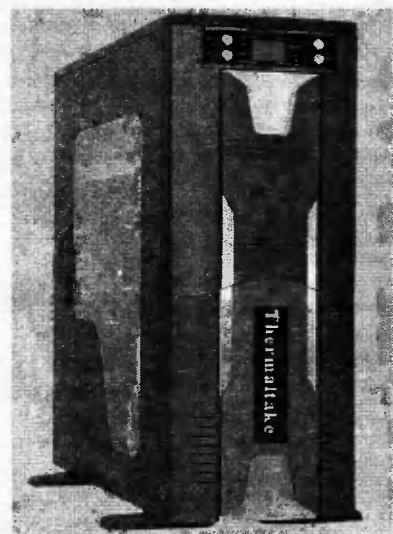
Крупногабаритные корпуса Big-Tower (иногда они называются Full-Tower) обеспечивают размещение системных плат любых размеров и самого большого количества устройств формата 5,25" (5 и более). Кроме того, обычно они комплектуются блоками питания повышенной мощности. Основная область применения таких корпусов — рабочие станции высокого уровня, небольшие серверы и “навороченные” компьютеры для фанатичных любителей всего самого-самого. Однако в связи с активной экспансией недорогих IDE RAID-контроллеров в массовые устройства, потребность в большом количестве посадочных мест для крупногабаритных дисковых накопителей может вывести корпуса Big-Tower в разряд наиболее распространенных устройств. Ведь хорошо известно, что для организации даже самого простого RAID-массива (0 или 1 уровня) необходимо минимум 2 диска, а современные высокоскоростные винчестеры в процессе работы ощутимо греются, и наиболее комфортно они себя “чувствуют”,

находясь в специальных 5-дюймовых боксах, хорошо охлаждаемых малогабаритными кулерами.

Несмотря на то что компьютерные корпуса принципиально не изменялись в течение нескольких последних лет (со времен принятия спецификации ATX), веяния изменчивой моды время от времени касаются и их. В последнее время это проявилось в увлечении производителей алюминиевыми корпусами, которые, сначала в качестве экзотической игрушки, а теперь, когда цена на них упала до 100 USD и ниже, и в качестве “рабочей лошади”, пусть и весьма породистой, все чаще и чаще стали появляться на российском рынке.

Алюминиевые корпуса обладают рядом серьезных преимуществ перед традиционными железными. Во-первых, алюминий обладает гораздо большей теплопроводностью, чем железо, поэтому алюминиевый корпус работает как один большой радиатор, эффективно отводя тепло от горячих элементов компьютера. Например, алюминиевые стенки корпуса вполне прилично отводят тепло от горячих жестких дисков, что в некоторых случаях позволяет отказаться от их дополнительного охлаждения. Кроме того, алюминий имеет меньшую массу, вследствие чего новые корпуса весят гораздо меньше, чем их железные собратья. Они почти не подвержены коррозии и гораздо легче в обработке, что значительно облегчает труд дизайнеров. На рис.1 показан футуристический дизайн корпуса Thermaltake Xaser III Super Tower V1000.

Рис. 1



Развитие идеи миниатюризации применительно к компьютерной области породило такое чудо как предельно интегрированные системные платы формата Flex-ATX и их естественное продолжение — Mini-ITX-корпуса то ли Slim, то ли Super Slim... В общем, все "супер", все "слим", тесно, крайне неудобно, возможностей — минимум, о модернизации я даже и не заикаюсь, но зато — сплошная оригинальность и эксклюзив! При использовании таких сверхкомпактных корпусов очень часто возникают проблемы с картами расширения — ведь в тонкий корпус чаще всего можно установить только специальную низкопрофильную карту, а они встречаются достаточно редко, да и возможности их порой куда скромнее, чем у полноразмерных аналогов. Выглядит Slim-корпус довольно красиво, но вот только стоит такой "малыш" гораздо дороже полнофункциональных корпусов, а рекламируется производителями — как недорогое решение для офисов, а порой и для домашнего (в том числе и для игр!) применения.

Долгое время являясь экзотическими, но малополезными игрушками, сверхминиатюрные компьютеры в последние год-полтора неожиданно-негаданно обрели "вторую жизнь". И произошло это с легкой руки компании Shuttle, своевременно предложившей на рынок так называемые скелетные, или barebone системы (не так давно получившие полуофициальное название Mini PC). Системы эти включают в себя миниатюрный корпус с установленной в него Micro-ATX-системной платой (до предела насыщенной интегрированными контроллерами) и дополнительно комплектуются еще и процессорным купером. После дооснащения (буквально в считанные минуты) такой системы процессором, памятью и жестким диском пользователь получает полнофункциональный, но при этом миниатюрный и, в дополнение ко всему, малошумящий компьютер. Как правило, в таких системах производители используют оригинальные компоненты, специально изготовленные для barebone-систем, поэтому замена материнской платы или добавление какого-нибудь компонента может вызвать определенные затруднения.

Современная barebone-система в обязательном порядке содержит полноценный набор портов, включая USB 2.0 и FireWire, и, как минимум, DVD-привод. Более того, наиболее продвинутые представители систем этого типа вплот-

ную приблизились по своим функциональным возможностям к бытовым аудио/видеоцентрам, и способны работать сразу в нескольких режимах без загрузки операционной системы. Довершает картину фактически ставший обязательным универсальный карт-ридер, необходимый для работы с цифровой периферией, а также зачастую встроенный интерфейс Wi-Fi — словом, все, что только может понадобиться в современном домашнем ПК. Наглядный пример продвинутой barebone-системы может дать Mini PC ASUS DiGiMatrix, который, благодаря выносу блока питания за пределы корпуса, имеющего более чем скромные габариты, больше напоминает бытовой видеомэгафон или какой-либо другой блок AV-комплекса, чем обычный компьютер (рис.2...4). Особенно этому способствуют расположенные на передней панели многофункциональный LCD-дисплей и большая ручка регулировки громкости, тогда как большая часть управляющих элементов и разъемов скрыты за декоративной крышкой.

Рис. 2

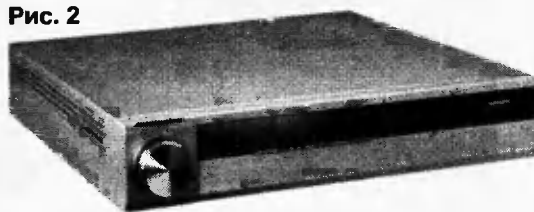


Рис. 3

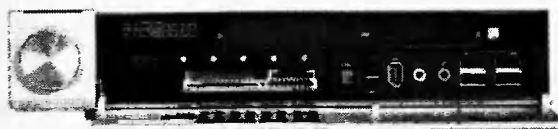


Рис. 4



Своего громкого названия — "медиа-центр" — DiGiMatrix удостоился благодаря возможности проигрывания и записи всех существующих типов оптических дисков, включая DVD, а также наличию встроенных TV- и FM-тюнеров. Общение с другими элементами домашнего мультимедийного комплекса облегчено благодаря наличию полного комплекта AV-входов и выходов, а также дистанционному управлению всеми основными функциями системы с помощью прилагаемого ИК-пульта. Роднит новинку с привычными аудиосистемами и то, что она способна воспроизводить FM-, CD-Audio- и MP3-записи без загрузки операционной системы. DiGiMatrix оборудован встроенным

мультистандартным карт-ридером, портами FireWire и USB 2.0 (как на передней, так и на задней панелях). Кроме того, благодаря наличию встроенных сетевых интерфейсов Gigabit Ethernet и IEEE 802.11b, DiGiMatrix может функционировать еще и как домашний сервер.

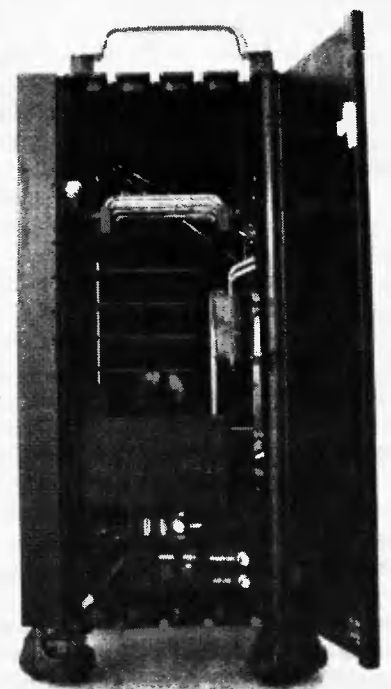
Таким образом, тенденция уменьшения размеров компьютерных корпусов проявляется в последнее время довольно четко. Однако традиционные корпуса форм-фактора ATX типоразмера Middle Tower в домашнем/корпоративном сегменте все еще занимают на порядок большую долю (в процентном отношении), чем экзотические изделия (Flex-ATX или Barebone).

Еще одной тенденцией последнего времени стала борьба за снижение уровня шума работающего компьютера. Наличие в современном компьютере большого числа устройств и компонентов, выделяющих в процессе своей работы много тепла, приводит к необходимости установки в корпус нескольких вентиляторов, обеспечивающих отвод горячего воздуха от них.

А каждый такой вентилятор — это дополнительный шум. Тем не менее, идея полностью бесшумных компьютеров постоянно бродит в умах наиболее продвинутых пользователей. Именно для таких бескомпромиссных борцов с шумом корейская компания Zalman — известный производитель высококачественных систем охлаждения — предложила абсолютно бесшумный корпус TNN 500A (рис.5).

Правая и левая боковые пане-

Рис. 5



ли TNN 500A представляют собой гигантские пассивные радиаторы, охлаждаемые конвекционными потоками воздуха, а тепло от внутренних устройств корпуса к наружным боковым панелям передается посредством системы тепловых труб. Помимо пассивных кулеров для процессора, видеокарты и жестких дисков, в данном корпусе используется еще и безвентиляторный блок питания, в котором реализованы те же методы для отвода тепла наружу.

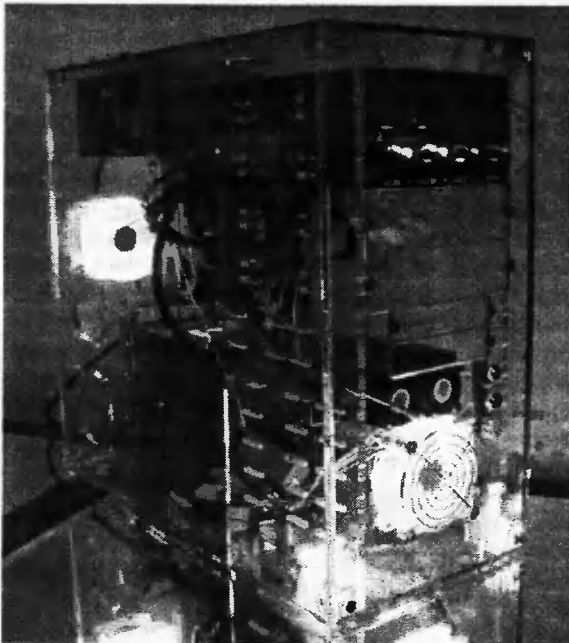
Может быть, черная ребристая коробка TNN 500A и способна обеспечить бесшумное, и при этом весьма эффективное охлаждение компьютера (по утверждениям разработчиков корпуса, система охлаждения способна рассеять до 100 Вт тепла от процессора и 50 Вт от видеокарты), однако она вряд ли способна украсить собой жилое помещение. А внешнему виду компьютерных корпусов в последнее время придается очень большое значение. Практически все наиболее значимые разработки последнего времени могут похвастаться запоминающимся дизайном или, по крайней мере, оригинальной расцветкой.

Более того, три-четыре года тому назад появился, сначала на Западе, а теперь уже докатился и до нас, так называемый моддинг — движение дизайнеров-энтузиастов, собственными силами превращающих корпуса своих компьютеров в настоящие произведения искусства. Почин любителей был с восторгом подхвачен индустрией, и сейчас в мире существуют сотни различных компаний и моддинг-студий, обслуживающих нужды моддеров. Если первые моддеры ограничивались, главным образом, раскраской стандартных корпусов в самые невероятные цвета, то сегодня на первое место выходит подсветка, осуществляемая разноцветными светодиодами и неоновыми лампами. Подсветка чего? Да всего без исключения. Неоновые лампы могут устанавливаться как на лицевой и боковых панелях корпуса, так и на периферийных устройствах и даже на железках и кабелях. Но наибольший эффект дает неоновая подсветка вращающихся лопастей вентиляторов. Лампы подсветки могут постоянно светиться, или мигать по определенной программе — все зависит от фантазии (и толщины кошелька) моддера. Все эти шедевры светотехники должны быть оценены по достоинству, поэтому в боковых стенках корпуса делаются большие прозрачные окна. Сейчас существует масса серийных Premodded-корпусов с окнами, выпускаемых известными

фирмами Chieftec, Thermaltake, Antec, Supermicro и т.д.

Но гораздо более эффектно выглядят полностью прозрачные корпуса — такие, как например, Acrylic Case от компании Sunbeam Technology — дорогой, но очень красивый и стильный вариант для истинных почитателей моддинга, выполненный целиком из оргстекла (рис.6).

Рис. 6



Но прогресс не стоит на месте, особенно в компьютерной индустрии. Современные многогиггерцовые процессоры, демонстрируя чудеса производительности, вместе с тем выделяют столько тепла, что обычные охладители уже не способны справиться со своей работой. Да и многие другие компьютерные элементы в процессе своей работы также выделяют слишком много тепла. При этом, как ни горько это признать, большинство технических решений, используемых на современном этапе для организации охлаждения компьютеров, окажутся (или уже оказались) неспособными справиться со своей задачей уже в ближайшее время, когда появится новое поколение высокопроизводительных процессоров. Зная это, лидеры компьютерной индустрии (во главе, естественно, с Intel) приняли решение о разработке новой спецификации материнских плат и корпусов, которая получила обозначение VTХ. Первые подробности о ней были озвучены на осеннем форуме IDF Fall 2003.

Типовая компоновка материнских плат форм-фактора VTХ 1.0 была пересмотрена, и это сказалось на незначительном увеличении глубины мате-

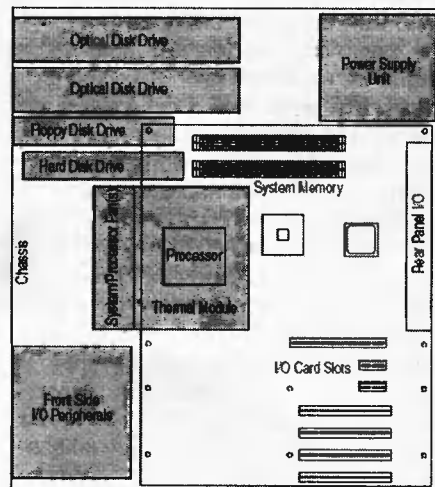
ринских плат. Теперь стандартная глубина составляет 266,7 мм, в то время как для ATХ 2.1 принималось значение 244 мм. Высота полноразмерной VTХ-платы теперь также превышает габариты ATХ-варианта на те же самые два сантиметра — 325 мм против 305 мм для ATХ 2.1.

На рис.7 показана типовая компоновка полноразмерного VTХ-корпуса.

Как видим, процессорный разъем переехал ближе к передней панели, над ним нависает система вытяжной вентиляции. Судя по всему, она будет крепиться на боковой крышке корпуса, что дает нам основание предполагать грядущий симбиоз системной платы и корпуса (вспомним современные barebone-системы). Тем более, что все мы хорошо знаем нелюбовь большинства вендоров к взаимной унификации своих разработок.

Полноразмерная плата VTХ сможет разместить 1 слот PCI Express x16 для графической платы, 2 слота PCI Express x1 для карт расширения, а также до четырех слотов PCI для ус-

Рис. 7



таревшего оборудования. Кроме «башен» формата VTХ, будут производиться еще и горизонтально ориентированные корпуса Desktop VTХ, в которых могут размещаться компактные платы стандарта Micro-VTХ (264 x 266,7 мм против 244 x 244 мм у Micro-ATХ) и самые миниатюрные Pico-VTХ (203 x 266,7 мм).

Серьезно изменятся и блоки питания нового формата, которые, кстати, теперь будут развернуты «поперек»



корпуса. Поскольку подробные спецификации блоков питания ВТХ сейчас все еще недоступны, мы можем лишь догадываться, что они будут в обязательном порядке иметь разъемы питания для винчестеров Serial ATA, старые разъемы 12 В для питания мощных графических плат, а главное — 24-штырьковый разъем питания, тогда как на платах формата АТХ этот разъем был 20-штырьковым. Ранее подобные 24-штырьковые разъемы питания использовались только на серверных материнских платах. Привычный разъем питания 12 В центрального процессора будет сохранен.

Учитывая уровень энергопотребления будущих процессоров (свыше 100 Вт) и графических плат (до 75 Вт), минимально допустимая мощность блока питания будет находиться на уровне 300 Вт, тогда как мощные системы станут использовать блоки питания на 420 Вт и выше.

Подводя итог вышесказанному, можно отметить, что современный компьютерный корпус из "ящика для железа" превращается (или уже превратился?) в самостоятельную, причем достаточно важную, деталь любого компьютера. Вместе с монитором он выполняет такую, абсолютно не

свойственную ранее компьютеру функцию как украшение интерьера, причем интерьера не только офисного, но все чаще и домашнего. Но, по моему глубокому убеждению, не стоит слишком увлекаться новомодными полупрозрачными, переливающимися всеми цветами радуги и поражающими взгляд вычурностью форм корпусами в стиле "техно". Такая мода приходит и уходит, мало что оставляя после себя. А вот строгие классические формы, дизайн, базирующийся на примате прямых линий, и однотонная неброская расцветка будут популярны всегда.

В.КИРИЧЕНКО,

E-mail: Steve_KEY@mail.ru

Как сделать загрузочную заставку Windows

Вам не надоела однообразная заставка при загрузке Windows, прославляющая "Windows'95 Microsoft Internet Explorer" или просто "Windows'98" с примитивной движущейся градиентной строкой в нижней части экрана? Мне — уже давно! И хотя эта заставка "держится" на экране всего несколько секунд (правда, на некоторых машинах — до минуты), все же хочется иметь и здесь что-то свое!

Сейчас я с улыбкой вспоминаю свои первые попытки как-то изменить эту заставку (когда узнал, что она "спрятана" в файле logo.sys) с помощью "графического редактора" Paint — любая попытка что-либо дорисовать на оригинальное изображение приводила к тому, что движущаяся градиентная строка наглухо останавливалась.

Сразу скажу, что делать картинки по типу тех, что "идут" с Windows, т.е. с движущейся полосой, мне было неинтересно. Поэтому появилась идея сделать движущимся все небо на каком-либо пейзаже.

Программисты Microsoft, видимо, просто "прикололись" с загрузочной картинкой, заставив Windows при загрузке "вращать" часть палитры для создания эффекта движения (по научному — ротация), ведь заставки, которые показываются при подготовке к выключению питания, и "Теперь питание компьютера можно отключить" — не двигаются! Весь же "фокус" изготовления таких картинок заключается в том, что изображение соответствующего размера (320х400 точек) в индексированных цветах (256-цветная палитра) должно быть обработано так, чтобы те места, которые мы хотим "оживить" (в нашем случае — небо), были определены цветами в конце палитры, и соответственно, в самом файле формата BMP число используемых для "неподвижной"

части цветов было указано явно. Изображение также не должно быть упакованным, иначе Windows его "не поймет".

Сложно так сразу понять? Далее будет пошагово описана процедура подготовки изображений для "оживления" загрузки. Команды будут приведены для пятой англоязычной версии Photoshop. Хотя изготовить необходимую картинку можно практически в любом "настоящем" графическом редакторе.

Для начала следует подобрать готовую картинку с пейзажем, на котором достаточно большую часть занимает небо с облаками (с ними картинка выглядит более эффектно), а цвета на самом пейзаже (т.е. на изображении земли, деревьев и пр.) отличаются от небесных оттенков (например, как на рисунке).



Сначала нужно перевести изображение в систему RGB, если оно закодировано в другой системе. Выберите из меню пункты Image/Mode, далее выберите RGB Color. Затем дважды быстро щелкните левой кнопкой мыши на изо-

бражении слоя Background на палитре Layers — вы получите диалоговый бокс Make Layer. Нажмите OK, согласившись с названием слоя Layer 0, или сначала введите любое имя слоя.

Размер исходной картинки может быть практически любым (оптимально — большим или равным 534х400), а соотношение высоты и ширины — лучше всего 3/4. Если исходное изображение имеет иное соотношение сторон, следует "обрезать" лишнее, выделив сюжетно важную часть (как правило, небо). Для проведения этой операции следует выбрать из меню пункты Image/Image size, далее в появившемся диалоговом боксе установить флажки Constrain Proportions и Resample Image (выберите Bicubic), в рамке Pixel Dimensions выбрать единицы измерения Pixels. Далее "поиграйтесь", задавая различные значения высоты и ширины картинки и подбирая оптимальные. Например, наберите в окошке Width число 640 (или 534), тут же автоматически изменится значение Height. Если Height получится больше 480 (400), нажмите OK, иначе задайте Height, равное 480 (или 400), и если Width получится более 640 (534), нажмите OK. Затем выберите из меню пункт Image/Canvas size и установите размер картинки 640х480 (или 534х400), при необходимости выбрав внизу диалогового положения оставшейся части изображения на целой картинке (т.е. определите, какая часть изображения будет сохранена), и нажмите OK (эту операцию придется еще подтвердить).

Теперь мы имеем изображение с соотношением сторон 3/4, т.е. равным соотношению сторон экрана монитора. Далее, снова выбрав пункт меню Image/Image size и отключив флажок Constrain Proportions, следует установить размер картинки равным 320х400. Есте-

ственно, изображение потеряет пропорциональность (оно станет вытянутым по вертикали), но именно такой формат и нужен для файла Logo.sys!

Далее следует любым способом отделить "небо" от "земли". Обычно я пользуюсь режимом быстрой маски. Чтобы включить этот режим, следует нажать "горячую" клавишу Q (обратите внимание, что "горячие" клавиши действуют только в таком режиме клавиатуры, когда вводятся английские буквы — индикатор EN около часов в панели задач, в то же время, в русифицированных версиях Windows по умолчанию включена русская раскладка — обычно русский/английский режим переключаются сочетанием клавиш Alt+Shift или Ctrl+Shift), или щелкнуть мышью на кнопке с изображением белого круга на сером прямоугольнике, которая находится на палитре инструментов во втором снизу ряду (ярлычок, появляющийся при наведении на нее курсора мыши, гласит — Edit in Quick Mask Mode). Теперь, если кликом мыши или "горячей" клавишей N выбрать инструмент "карандаш" (не "кисточку"!), то "рисование" черным цветом будет закрашивать изображение красным цветом (по умолчанию можно установить любой другой, если нужно), а "рисование" белым цветом (кстати, цвета переключаются "горячей" клавишей X) будет убирать закрашивание. Закрашенные таким образом участки изображения считаются покрытыми "Быстрой маской". Перед нами стоит задача закрасить, например, все детали пейзажа, кроме неба, которое будет сделано движущимся. При выполнении этой операции следует сильно увеличить изображение, чтобы были видны отдельные пикселы, из которых оно состоит, и работать самыми "тонкими" карандашами (выбрать размер карандаша можно, щелкнув правой кнопкой мыши на изображении, и из появившегося всплывающего меню выбирать пункты First Brush, Last Brush, Next Brush и Previous Brush).

Когда операция выделения будет закончена, нужно отключить режим быстрой маски, нажав Q или кнопку на палитре инструментов рядом с той, которой эта маска была включена. Мы увидим, что часть изображения окажется выделенной линией, состоящей из так называемых "бегущих муравьев" (стандартное выделение).

Теперь заберите в буфер выделенную часть изображения, нажав на выбор сочетание Shift+Del, Ctrl+X или выбрав из меню пункты Edit/Cut. Затем вставьте вырезанную часть, нажав Shift+Ins, Ctrl+V или через меню Edit/Paste. Скорее всего, эта часть вставится в центр картинки, но это не беда — выбрав инструмент Move Tool (второй в палитре инструментов или "горячая" буква V) и увеличив

изображение так, чтобы была видна граница удаленной части картинки, подвиньте вырезанную часть до совпадения ее края с краем оставшейся части.

У нас получилось изображение, состоящее из двух слоев, один из которых называется Layer 1 (при желании можно задать и другое имя слоя).

Для дальнейшей работы отключите "видимость" слоя, где находится небо, нажав на изображение глаза рядом со слоем в палитре Layers, и создайте палитру для основного изображения, выбрав из меню Image/Mode/Indexed Color (подтвердите сведение и удаление невидимых слоев — Flatten visible layers and discard hidden layers). Далее выберите Palette-Adaptive, Color Depth-6 bits/pixel (выберется 64 цвета, остальные будут использованы для "движущегося" неба — чем больше цветов отведено для движения, тем движение более плавное, и, соответственно, наоборот — чем меньше цветов, тем быстрее), Dither=None, Color Matching-Best и нажмите OK. Затем выберите Image/Mode/Color table и найдите самый черный квадратик в верхней четверти таблицы. Теперь ткните мышью в первый верхний квадратик и запомните соотношение цветов, из которых он состоит — напротив букв R, G и B. Далее нажмите Esc и ткните на тот самый черный квадратик — здесь следует задать запомненные на предыдущем шаге цифры и нажать OK, а затем ткнуть в верхний левый квадратик и задать там все три цвета "по нулям". Это нужно сделать для того, чтобы рамка вокруг изображения при загрузке Windows была черного цвета — так привычнее... Нажмите Save в этом же окне и сохраните палитру в файл с расширением ACT под любым именем — она нам понадобится в дальнейшем. Затем нажмите Esc и Ctrl-Z для отмены преобразования картинки в индексированные цвета — она нам еще нужна в RGB...

Включите изображение неба ("глаз" около изображения слоя в палитре Layers) и выберите для работы этот слой — ткните в него мышью. Выберите из меню Image/Adjust/Color Balance или просто нажмите Ctrl-B, затем включите в появившемся окне флажок Preview, если он был выключен, и, выбирая поочередно переключатели Shadows, Midtones и Highlights, "на глаз" подкорректируйте оттенки неба, так чтобы они были более яркими, более "живыми", с переходом от почти темно-синего до ярко-белого цвета. Обычно для этого нужно сдвигать верхний движок Cyan-Red влево на 35...45 единиц (процентов), а нижний Yellow-Blue — на столько же вправо (могут быть варианты — смотрите, как изменяется изображение). Флажок Preserve Luminosity

используйте как вам больше нравится. Добившись приемлемого результата, нажмите OK.

Теперь нужно отключить изображение слоя с "землей" и, снова выбрав из меню пункты Image/Mode/Indexed Color, установить Palette-Adaptive, Dither=None, Color Matching-Best и Colors-96 (половина от оставленных нами для неба). Далее, ничего более не меняя, выберите Palette-Custom, и вы увидите палитру, которую предложит для данного неба Photoshop. Обратите внимание, какой цвет самый темно-синий (ткните на него мышью и запомните значения) и какой самый светлый (тоже запомните) — может пригодиться потом. А для начала попробуйте такой трюк — выделите 6 строк палитры (нажмите на первую клеточку мышью и, передвинув мышь на последнюю клеточку, отпустите кнопку) — вам будет предложено определить первый и последний цвет градиента (остальные промежуточные цвета программа "придумает" сама). Определите первый цвет как синий (R и G — по "нулям", B — 255), а последний — белым (R, G, B — все по 255). Ткните OK. Дождитесь, пока компьютер "просчитает" изображение с новой палитрой (перестанет мигать курсор около флажка Preview) и посмотрите на результат. Если понравилось — нажмите OK, если же "не пойдет" — выберите еще раз (или несколько раз) Custom и переопределите палитру таким образом, чтобы при наличии градиента от примерно темно-синего цвета к почти белому, облака и небо смотрелись бы естественно (тут нужен определенный вкус и отсутствие дальтонизма, дихроматизма, астигматизма, алкоголизма и т.п.) ;—), можно также попробовать ввести запомненные цвета самого темного и самого светлого квадратиков (видите — они пригодились!). Запомните определенные вами цвета — они вам уж точно пригодятся!

Когда небо будет уже приемлемым, нажмите OK и выделите полученную таким образом картинку (Ctrl-A или в меню Select/All) и скопируйте ее в буфер обмена (Ctrl-Ins или Ctrl-C). Затем нажимайте Alt-Ctrl-Z до тех пор, пока картинка не вернется к своему первоначальному варианту — с двумя слоями. Включите отображение всех слоев, слой с изображением неба "бросьте" в корзину — она нарисована внизу палитры Layers (или удалите, щелкнув на нем правой кнопкой мыши и выбрав из всплывшего меню Delete Layer), а из буфера обмена вставьте то небо, которое поместили туда на предыдущем шаге. Если оно "затмит" собою все изображение, перетащите слой (в палитре Layers) с небом на самую нижнюю позицию, так чтобы слой с "землей" был выше его.

У нас опять получилась почти прежняя картинка с пейзажем и небом, только небо теперь несколько другого оттенка.

Переходим к завершающей стадии создания файла Logo.sys. Включите отображение всех слоев, выберите (в который уже раз) Image/Mode/Indexed Color. Подтвердите сведение. Выберите Dither-None, Palette-Custom и загрузите (Load) сохраненную ранее палитру, созданную для "земли". Начиная от пятой строки квадратиков, выделите следующие шесть строк и задайте на них градиент — тот, который ранее задавали для неба. Нажмите ОК и посмотрите на результат (подождите, пока перестанет мигать курсор около флажка Preview). Если все в порядке — нажмите ОК, если же нет — значит, вы где-то ошиблись, и попробуйте повторить все заново, внимательно следя за результатами...

Выберите из меню Image/Mode/Color table и сохраните полученную палитру в другой файл, затем для проверки заполните верхние четыре строки квадратиков черным цветом — выделите их и задайте для первого и для последнего черный цвет (R, G, B — по "нулям"). В идеале все, кроме неба, должно стать черным, но на небе не должно быть черных пятен (что говорит о неверном выборе палитры — цвета "земли" и "неба" перекрываются по цветовому диапазону), также на земле не должно быть пятен другого цвета, кроме черного.

Если все так, как здесь описано — поздравляю! Картинка практически готова! Осталось совсем немного...

Выберите Image/Mode/Color table и загрузите последнюю сохраненную палитру. Нижние шесть строк в ней до сих пор имеют черный цвет — это нужно исправить! Задайте для этих клеточек такой же градиент, как и для пре-

дыдущих шести строк, но в обратном порядке (т.е. от белого к синему). Нажмите ОК.

Сохраните изображение в файл BMP — выберите из меню File/Save As, выберите нужный формат (Save As-BMP (*.BMP; *.RLE)), название можно оставить прежним, нажмите ОК (или что там будет — в русской версии Windows, наверное, Сохранить). В появившемся окне BMP Options выберите File Format-Windows и снимите флажок Compress (RLE). Подтвердите операцию (OK).

Картинка, в общем, готова. Но для того чтобы при помещении ее в папку C:\, во время загрузки Windows небо на ней "двигалось", нужно немного "подправить" всего один байт в файле BMP.

Запускаем любой дамповый (шестнадцатеричный) редактор (RED, DUB или что-то подобное). Я лично предпочитаю в качестве такового использовать VC — Volcov Commander. Можно попробовать FAR, но с ним можно и "пролететь", если к его редактору прикреплены определенные "плагины" (Plugins). Кстати, обратите внимание на длину нашей BMP-шки — она должна быть 129080 байтов или 129078 байтов, а при просмотре в виде текста (Alt-F3 в VC, NC или FAR'e) первые байты должны быть BM6 или BM8. Любое другое значение говорит о том, что Photoshop неправильно сохранил палитру (обычно это бывает, когда в конце палитры идут черные квадратики-цвета) — при просмотре все будет, вроде бы, в порядке, а при использовании файла как Logo.sys — бардак!

Итак, загружаем нашу картинку в дамповый редактор (в VC, например, наводим курсор в панели на нужный нам файл и дважды нажимаем F4 — для загрузки в редактор и для переключения в дамповый режим, а в FAR'e — F4 и F11, далее выбрать HexEditor). Подправляем

байт по смещению 32H (шестнадцатеричное, 50 десятичное). Для тех, кто не знает, что такое смещение — нажмите три раза стрелку вниз и четыре раза стрелку вправо) с 00H на 40H (64 десятичное), т.е. нажимаем цифры 4 и 0. Сохраняем. Далее копируем файл на диск C:\ под именем LOGO.SYS. Перезагружаем машину и любимся.

Если любоваться нечем, лезем любым текстовым редактором (хоть "Блокнотом") в файлы MSDOS.--- (расширение — три минуса) и MSDOS.SYS и вписываем в раздел [Options] строку Logo=1 (или исправляем на вышеуказанное строку Logo=0). Снова перезагружаемся и теперь уж точно любимся!!!

Эффект довольно красив (сам обалдел, когда увидел первый раз) — небо на картинке постепенно становится как бы грозным, потом почти инверсным, а затем опять принимает нормальный вид. Кстати, для того чтобы вдоволь налюбоваться плодами вашего творчества, нажмите, когда появится заставка, клавишу Pause на клавиатуре — и картинка будет "жить" на экране до нажатия любой другой клавиши.

Итак, на мой взгляд, в статье приведено достаточно информации, чтобы "средний" пользователь (слово какое-то неправильное, оно мне не нравится, но другого пока не выдумали) программы Photoshop смог сделать свою движущуюся заставку на личный компьютер. Разумеется, эффектов можно сделать еще больше и еще лучше — пробуйте, экспериментируйте, и у вас получится!

В прилагаемом к статье архиве logos.zip на web-странице журнала приведено несколько изображений заставок, созданных мною по описанной в статье методике.

Особая благодарность за идею PopSOFT (PopSOFT@mail.ru).

Удачи!

А.ГРИНЧУК,

E-mail: gav@nihe.niks.by

Анимация с Macromedia Flash MX

Специальные типы анимации

Позади у нас остались основные типы анимации в Macromedia Flash: анимация движения и анимация формы [1]. Это как раз те базовые элементы, из которых в большинстве случаев и конструируется нечто более сложное. Впрочем, почти все, что можно сделать на рабочей области, мы уже изучили. Где же следует искать скрытые резервы?

Во-первых, рисовать и анимировать можно и программными методами. Не зря же от версии к версии значительно увеличиваются возможности встроенно-

го языка Action Script. Но если цель вашего изучения Flash — относительно простое создание анимированных картинок, то это явно не лучший вариант. Второй "скрытый ресурс" можно поискать в работе со слоями. Именно здесь мы познакомимся с новыми объектами — направляющими слоями (Motion Guide) и слоями-масками (Mask).

Движение по направляющей линии — одна из лучших находок Flash. В полном объеме и с той же легкостью повторить это в других программных продуктах не удастся. Разве что в род-

ственном Macromedia Dreamweaver с помощью временной шкалы удастся повторить часть из этих трюков. Идея движения по направляющей проста — в дополнительном слое создается линия, движение фигуры происходит вдоль этой линии. Осталось главное — на конкретном примере разобраться с деталями.

Продолжим наши манипуляции с автомобилем (C:\Program Files\Microsoft Office\media\cagcat10\j0212957.wmf). В первую очередь, импортируем файл (File/Import), а

Рис. 1

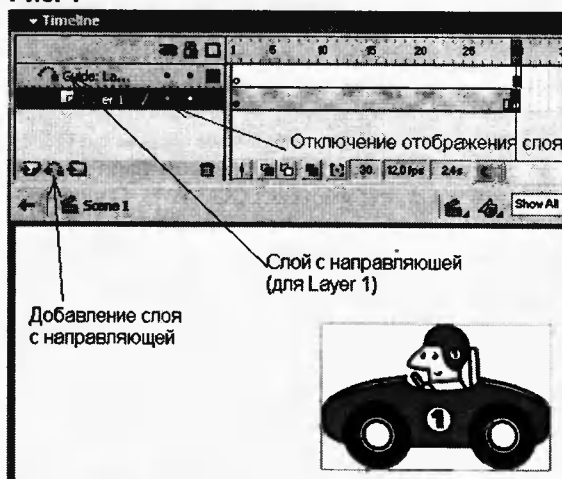
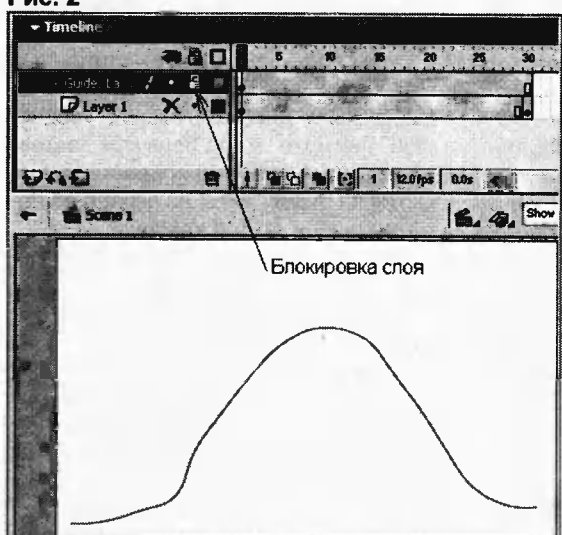


Рис. 2



затем — группируем (**Modify/Group**) и создаем конечный кадр анимации — выделяем 30-й кадр, **Insert/Keyframe**. Необходимо проверить, чтобы нужный нам слой был выделен, направляющий слой будет создаваться именно для выделенного. А вот дальнейшие действия требуют предельной аккуратности. Добавим слой с направляющей (**Insert/Motion Guide** или кнопка ниже названий слоев — рис.1). Для того чтобы избежать помех при рисовании линии, отображение основного слоя лучше отключить. Переходим в первый кадр дополнительного слоя и рисуем траекторию движения. В итоге получим набор статических кадров, о чем говорит серый цвет кадровой линейки. Это именно то, что нам нужно, поэтому, во избежание случайных повреждений линии, блокируем дополнительный слой (рис.2).

Но, естественно, основная работа будет происходить в основном слое (**Layer 1**). Настала пора включить отображение слоя и заняться первым кадром (не забудьте его выделить).

Наша задача — привязать рисунок к линии, для чего следует поместить курсор мыши как можно ближе к центру фигуры и нажать левую клавишу, здесь должен появиться (это обязательное условие) небольшой круглый маркер. Не отпуская клавишу мыши, мы должны переместить фигуру и попасть маркером на линию (рис.3). Снова выделяем первый кадр и открываем панель **Properties**, где задаем тип движения и устанавливаем флажок фиксации **Snap** (рис.4). В конечном кадре также необходимо привязать нашу картинку с автомобилем к линии в конечной точке траектории. О самой направляющей можно не волноваться, при просмотре (**Ctrl + Enter**) и экспорте анимации она отображаться не будет. Зато сам просмотр выявляет одно не очень приятное обстоятельство — автомобиль,

вопреки всяким законам природы, сохраняет горизонтальное положение (рис.5). Иногда такое поведение и

требуется, но не в нашей ситуации. Исправляется положение возвратом в первый кадр и заданием на панели **Properties** опции ориентирования по пути (**Orient to path** — рис.6). Поведение автомобиля сразу становится более реалистичным, особенно после добавления нового слоя с рисунком горки (рис.7).

Иногда нужно два или более объектов заставить двигаться по одной и той же траектории. Тогда просто создаются обычные слои и с помощью мыши и ярлычка для перетаскивания слоя перетаскиваются в новое положение (рис.8). И последнее замечание — направляющая линия может иметь вид даже мертвой петли и пересекать саму себя. Macromedia Flash правильно ведет себя и в таких трудных условиях. Просто не

Рис. 3

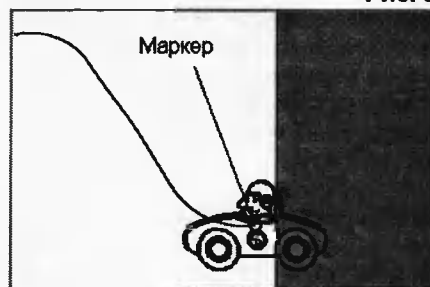


Рис. 4

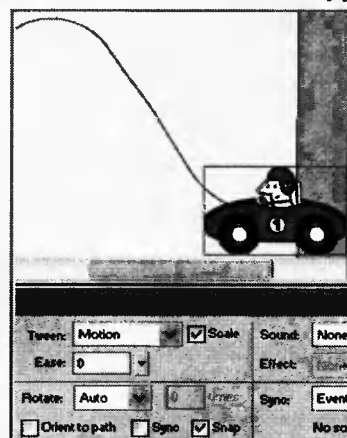


Рис. 5

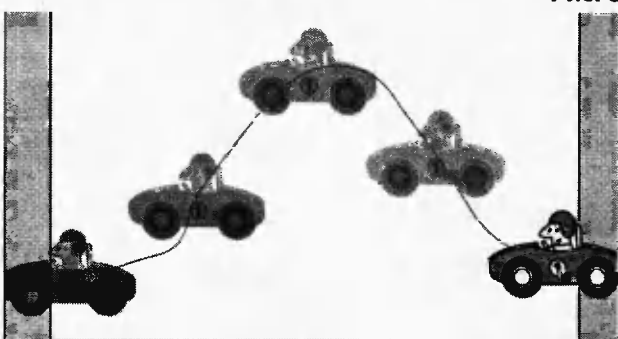


Рис. 6

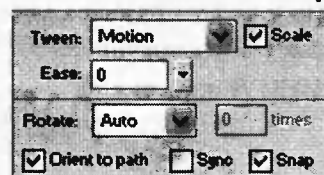


Рис. 7

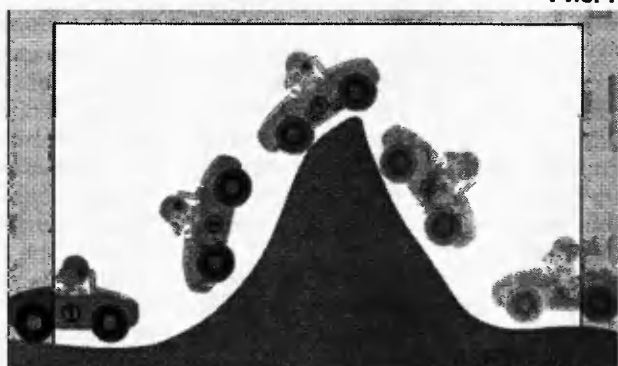


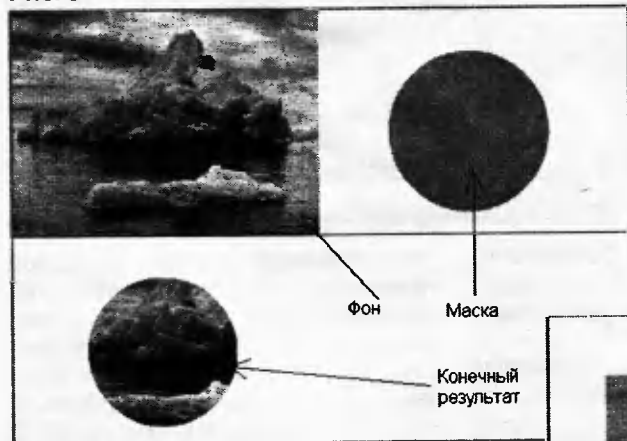
Рис. 8



стоит увлечься и рисовать очень заковыристую направляющую; слишком сложные мельтешения выглядят не очень приятно.

Другой тип слоя — слой-маска — используется намного реже. Выглядит это следующим образом — создается основное изображение и слой-маска с каким-либо рисунком, допускается и анимация. Место, занятое изображением на маске, в итоговом ролике будет прозрачной областью, через которую можно просматривать только часть основного изображения (рис.9). Цвет и прозрачность изображения на маске при этом не играют никакой роли.

Рис. 9



Особенно эффектно все это выглядит, когда в слое-маске используется анимация формы. Продемонстрируем это на небольшом примере. Создадим новый файл, первый слой назовем **Osnova** и импортируем фоновый рисунок. По традиции, сделаем ключевым 30-й кадр. А вот основная часть работы сейчас придется на дополнительный слой — **Insert/Layer**. Делаем щелчок правой клавишей мыши по новому слою и выбираем пункт меню **Mask**. Лучше сразу же снять блокировку слоев и второй слой переименовать в **Maska**. В первом кадре слоя **Maska** создаем надпись "Привет!" и разби-

ваем ее на точки двукратным применением операции **Modify/Break Apart** (рис.10). В том же слое (**Maska**) делаем ключевым 15-й кадр (F6), после этого на протяжении 15-и кадров у нас будет статическая маскирующая надпись. А вот во второй части анимации наша надпись трансформируется в прямоугольник, занимающий все пространство рабочего листа — переходим в 30-й кадр, делаем его ключевым, удаляем надпись и рисуем прямоугольник. Осталось в 15-м кадре задать тип анимации — **Tween: Shape** на панели **Properties**. В итоге получаем цветную надпись, которая через небольшой промежуток времени трансформируется в прямоугольник. Сам процесс трансформации выглядит достаточно причудливо (рис.11).

И чтобы завершить наш разговор о специальных типах анимации, нам придется в некотором роде сделать шаг назад и вспомнить, что на заре мультипликации каждый кадр рисовался отдельно, шаг за шагом. Наверное,

Flash никогда бы не стала такой популярной программой, если бы ее создатели не пытались упростить и сделать более удобными все возможные варианты работы. Научить работать на таком уровне, пожалуй, невозможно, но все же можно дать несколько общих советов. Во-первых, каждый кадр должен быть ключевым — требование естественное, так как ключевые кадры — это именно то, что рисуете вы сами, а не программа. Далее стоит немножко подумать, какие части вашего рисунка будут двигаться как единое целое. Их лучше выделить и сгруппировать (**Modify/Group**). Если вам понадобится вращать изображение вокруг определенной точки, не совпадающей с центром фигуры, то помо-

Рис. 10

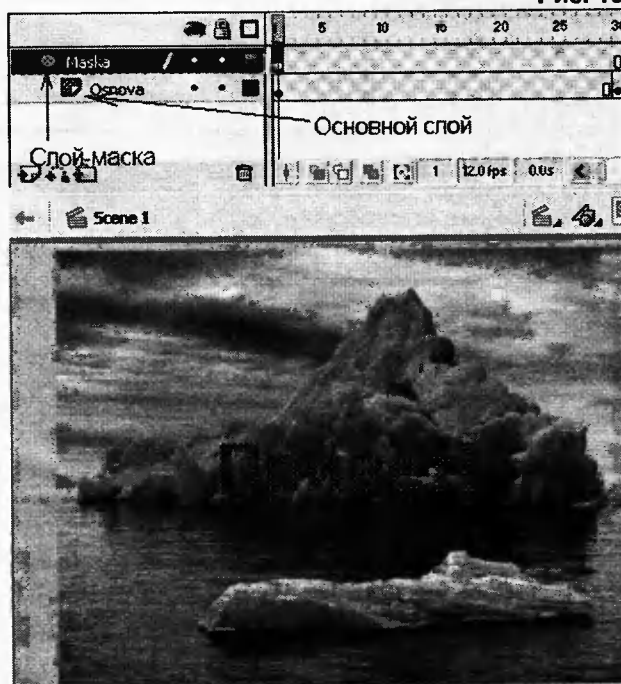


Рис. 11

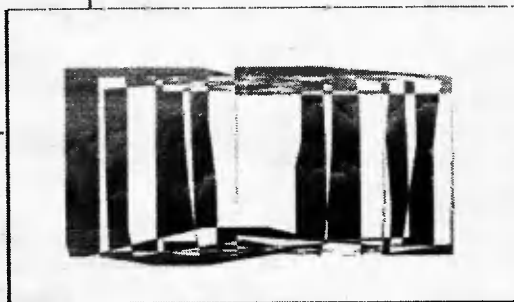
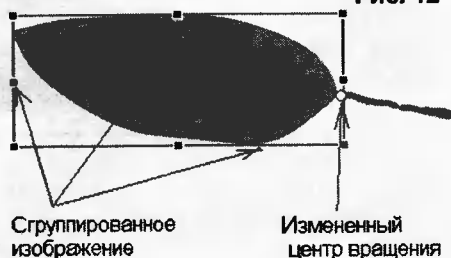


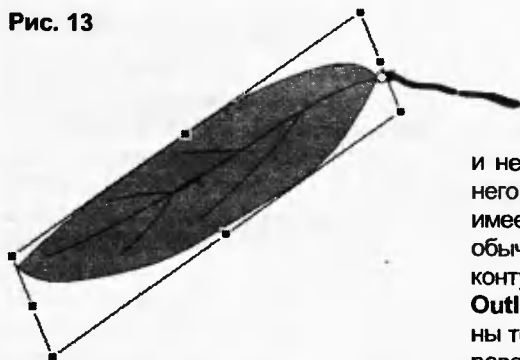
Рис. 12



жет режим свободной трансформации **Modify/Transform/Free Transform**. Так, на рис.12 показан листик с черенком, подготовленный для дальнейшей работы. Что нам остается сделать, так это создавать новые ключевые кадры, и трансформировать изображение листика. Его лучше поворачи-

вать и слегка добавлять скос (**Modify/Transform/Rotate and Skew** — рис.13). Осторожность нужна главным образом в тот момент, когда вы создаете новый ключевой кадр и начинаете трансформацию: по умолчанию выделены все элементы рисунка — не нужно забывать о снятии выделения. Только после этого выделяются и модифицируются отдельные элементы.

Рис. 13

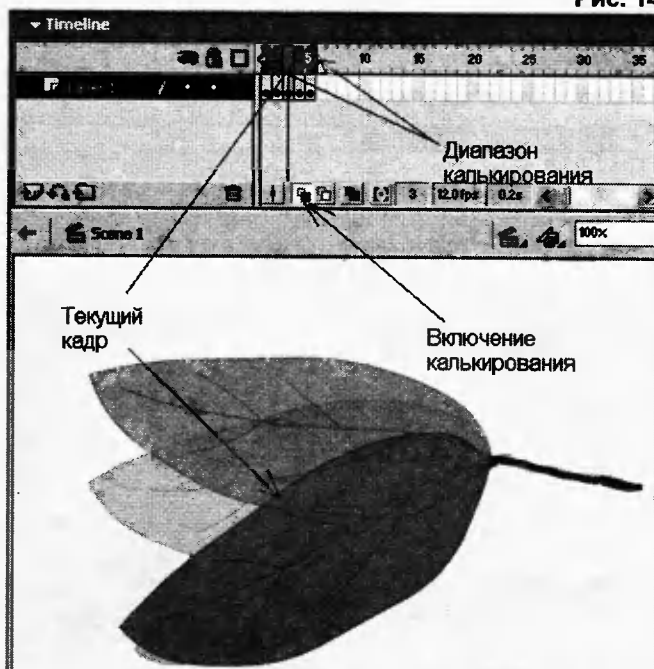


Все это проще, чем перерисовывать каждый раз все заново, но как же про-

контролировать ход нашей анимации? Для этого существует режим калькирования (**Onion Skin**). Можно просматривать не только текущее изображение, но и некоторый диапазон кадров вокруг него (**рис.14**). Ниже кадровой линейки имеется группа из трех кнопок: режим обычного калькирования (**Onion Skin**), контурного калькирования (**Onion Skin Outlines** — в остальных кадрах видны только контуры) и режим редактирования выделенного набора кадров (**Edit Multiple Frames**). Нетрудно теперь понять, как создавались иллю-

Рис. 14

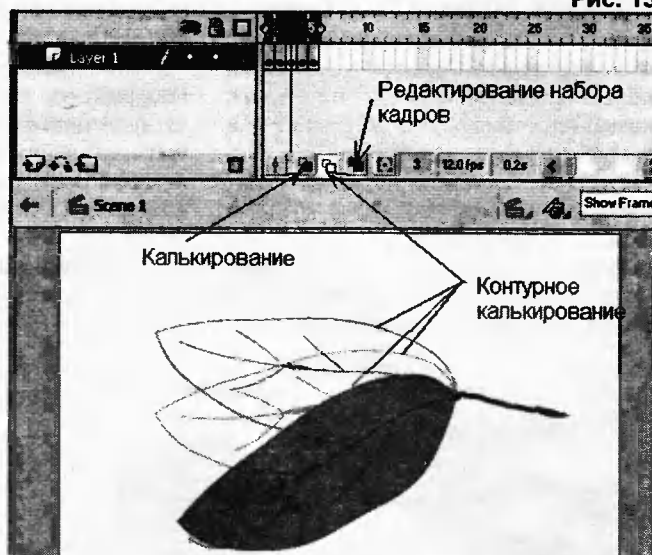
трации к статьям — просто был включен режим обычного калькирования.



Литература

1. А.Гринчук. Анимация с Macromedia Flash MX. — Радиомир. Ваш компьютер, 2004, №№1-4.

Рис. 15



Irfan-Тюльпан

А.ГОРЯЧКИН,

г.Кыштым, Челябинской области.
E-mail: anatoliy_goryach@mail.ru

Программа — IrfanView v3.80.

Назначение — графический व्यूер/конвертер.

Разработчик — Ирфан Скилджан.

Сайт разработчика — <http://www.irfanview.com>.

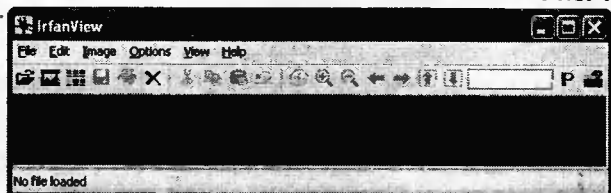
Условия распространения — freeware.

Объем дистрибутива — 821 Кб.

Операционная система — Windows 9x/NT/2000/XP.

Отказавшись от услуг канадской фирмы Corel, я обратил свой взор на софт, который был у меня под рукой. Среди прочей "мягкой рухляди" оказался IrfanView v3.80 (**рис.1**). Установив его и изучив меню программы, я понял, что нашел то, что искал. В меню Options — Capture было то, что мне нужно. Просмотрев установки меню Capture, я обнаружил в нем наличие опции Include mouse cursor

Рис. 1

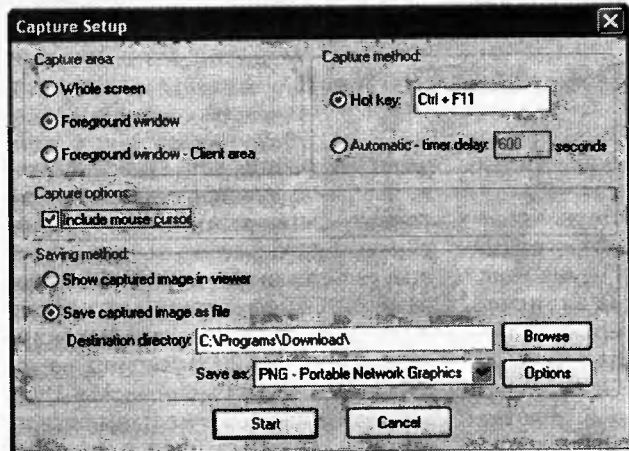


(**рис.2**). Слово capture, как известно, переводится с английского как захват. Испытав захват скриншотов на деле, я остался вполне доволен результа-

У меня, как и у других пользователей, иногда возникает потребность создавать скриншоты программ. Можно, конечно, воспользоваться встроенной функцией Windows, нажав клавишу Print Screen или клавиатурную комбинацию Alt+Print Screen. Только в этих случаях в скриншоте не будет изображаться курсор мыши. А если нужно получить скриншот именно с изображением курсора мыши? Одно время

я пользовался для этих целей программой Corel Capture из пакета Corel Draw девятой версии, но почему-то у меня эта программа "не пошла" из-под Windows XP. Вдобавок, меня не устраивало то, что для столь простых операций эта программа занимает слишком много места.

Рис. 2



тами работы программы IrfanView. Понравилось мне и то, что пользователь имеет возможность сам задавать в программе клавиатурную комбинацию "горячей" клавиши для функции захвата. Это удобно в тех случаях, когда в момент захвата требуется, например, нажатие левой кнопкой мыши в сочетании с клавишей Ctrl.

Делать скриншоты с помощью функции захвата изображения в программе IrfanView довольно просто. Сначала надо задать необходимую область сохранения экрана — весь экран или только активное окно. Далее, если в скриншоте требуется показать курсор мыши — ставим галочку в опции Include mouse cursor. Остается только указать папку, в которой будет сохраняться изображение и его формат. Затем кликаешь по кнопке Start и загружаешь программу, скриншоты которой собираешься получить. После этого остается нажать "горячую" клавишу (Hot key). Захват опробован и в некоторых играх, работающих в полноэкранном режиме, например, WarCraft 3. "Ложка дегтя" — захват видеоизображения с помощью этой программы, к сожалению, невозможен.

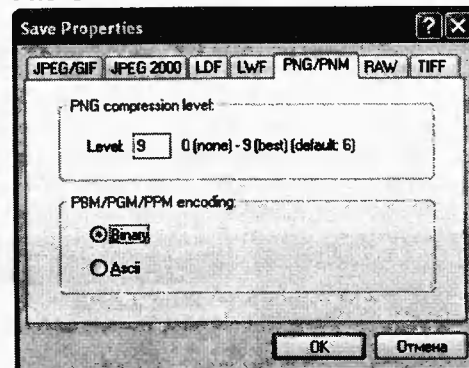
Работая с программой Virtual Dub, я обратил внимание на help этой программы. Он был сделан в формате HTML. В help'e были использованы скриншоты программы в графическом формате PNG. Изображения были хорошего качества и очень компактные по своему размеру.

Здесь я сделаю небольшое "пирическое отступление" и расскажу немного о графическом формате PNG. PNG (Portable Network Graphics) — переносная сетевая графика. Это относительно новый и весьма компактный графический формат. Он имеет 10 уровней компрессии (0...9), 9-й уровень обеспечивает наивысшее сжатие. Упаковка файлов работает по принципу ZIP-архивации, и сжатие

производится без потери качества изображения. Формат обладает широким диапазоном глубины цвета — 16,7 миллионов цветов (24 разряда), 256 цветов (8 разрядов), 16 цветов (4 разряда) и монохромный (1 разряд). Графический формат PNG удобен для передачи изображений по глобальным и локальным сетям, так как вместе с форматами

GIF и JPEG поддерживается многими Web-браузерами. По сравнению с файлами формата JPEG, PNG-файл не только не уступает ему по качеству, но и во многих случаях значительно меньше его по объему.

Рис. 3



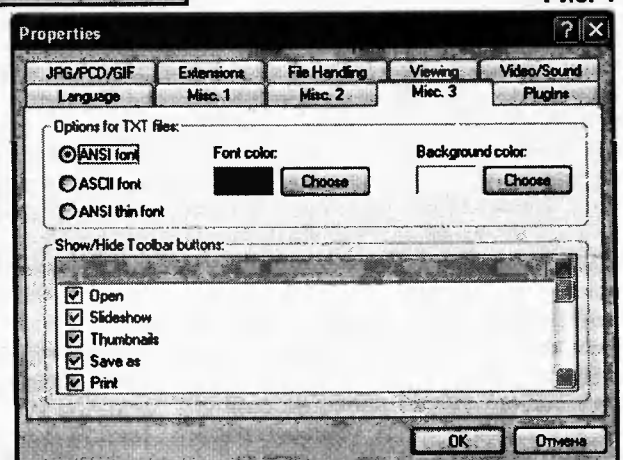
Направлялся вывод — формат PNG как нельзя лучше подходит для скриншотов и графического оформления гипертекстовых страниц. Я проявил интерес к этому формату и, соответственно, стал подыскивать программу, которая смогла бы конвертировать графические файлы в этот формат. В этом случае IrfanView опять оказался востребованным. Эта программа обладает функциями не только просмотра, но и конвертирования графических файлов. Он без особого труда справился с задачей конвертирования графических файлов в формат PNG. Есть опциональная возможность задавать уровень компрессии, что существенно влияет на объем получаемых файлов PNG (рис.3).

Отличную функциональность программе обеспечивает поддержка плагинов. Создано этих плагинов великое мно-

жество, найти и скачать нужные можно на странице <http://www.irfanview.com/plugins.html>. Благодаря им IrfanView становится универсальным средством не только для просмотра и конвертирования графических файлов. Многочисленные плагины позволяют программе проигрывать аудиофайлы MP3, WAV, WMA, MID, RMI, AU, SND, AIF и просматривать видеофайлы с расширениями AVI, MPG, MPEG, MOV, WMV.

Умеет программа просматривать изображения в папках, то есть обеспечивается работа встроенного браузера. Для этого надо на панели инструментов нажать кнопку Thumbnails, или просто нажать клавишу с буквой T. По умолчанию кнопка Thumbnails отсутствует на панели инструментов, в меню Options — Properties на вкладке Misc. 3 в поле Show/Hide Toolbar buttons: отмечают нужные пункты птичками (рис.4). Кроме того, IrfanView способен переименовывать и конвертировать графические файлы в пакетном режиме, то есть сериями (функция Batch Conversion/Rename в меню File). Есть в IrfanView и встроенные средства для показа слайд-шоу — режим, в котором производится автоматический показ всех изображений, находящихся в текущей папке, через заданное время в секундах. Конечно, в области просмотра изображений IrfanView в какой-то степени уступает своему конкуренту ACDSee, но он

Рис. 4



занимает меньше места и более быстр в работе. Не каждый графический редактор обладает такими богатыми возможностями по обработке изображений, применению различных эффектов и фильтров, как программа IrfanView.

Литература

1. А.Горячкин. Файловые форматы точечной графики. — Радиомир. Ваш компьютер, 2002, №9, С.8.

Р.КАРПАЧ,

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

Полезное и приятное. Для кого как

Наши руки — не для скуки...

Стоит ли говорить о том, что существует множество программ, облегчающих жизнь пользователю? Конечно, нет! Но вот какие из них выбрать? Так вот, совсем недавно, в поисках простого графического редактора, мне пришлось посетить не один сайт. И всякий раз я испытывал разочарование. Впору даже составить такой самодельный список самых неудачных программ. То программы слишком «глючные» в прямом смысле этого слова, то наоборот, сделанные так, что мало кто, кроме самих разработчиков, поймет, как с ними работать. Читатель может возмутиться, мол, пользуйся Photoshop или Corel. Но у меня просто натура такая, хочется попроще, и чтобы места занимало поменьше. Сразу скажу, вначале я искал для школьников, что-то наподобие Windows Paint. Эдакую простую программу для рисования. И действительно нашел. Она называлась Painter версии 4.5 beta (freeware,

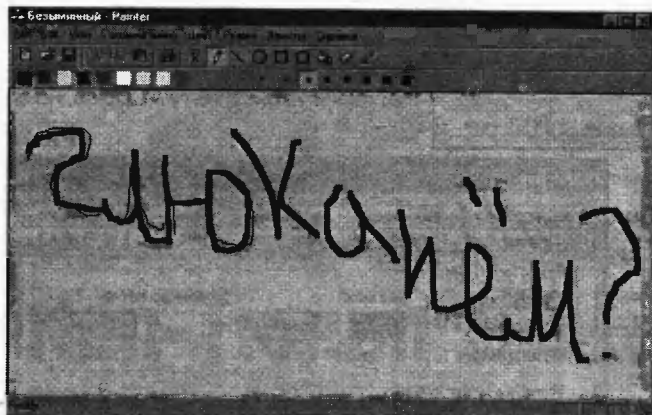
графики. Я уже чувствую, как читатель ехидно чешет руки. Мол, нечего ставить на старые компьютеры новые программы. А вот и нет! Программа работала на Celeron 950/64 Mb Ram/ WINDOWS 98SE. Одно радует, что это была всего лишь бета-версия. Надеюсь, мы дождемся выхода полноценной версии программы. Между прочим, исходники к Painter прилагаются.

Продолжая тему разного рода «глюков», хочу рассказать еще об одной программе под названием Blind Write (shareware, www.blindwrite.com). На сие творение рук программистов я наткнулся, когда искал простую программу для записи CD-RW. Она меня прельстила количеством поддерживаемых драйверов устройств и возможностью работы под Windows 95. У меня не столь уж новый привод TEAC CDW58E со скоростной формулой 8-8-32. Казалось бы, любая маломальски новая программа должна узнавать его в системе. Любая, но толь-

шина для работы хоть куда! Но какое было мое удивление, когда, прочитав документацию, я узнал, что на компьютере установлен, вы только не смейтесь, лицензионный MS-DOS версии 6.22. Самое обидное, что его там не оказалось! Вместо него был установлен Windows 2000. Видимо, не установился лицензионный MS-DOS 6.22 на компьютер с жестким диском емкостью 40 Гб и файловой системой NTFS. Не те времена... Но цена за лицензионное программное обеспечение была включена в цену компьютера. Как говорят, дешево и сердито...

Под занавес я хочу рассказать о программе, которая действительно в прямом смысле достойна внимания читателей. Это VicMan's Photo Editor (shareware, <http://www.vicman.net>) — замечательный программный продукт, при помощи которого был создан призовой экологический плакат (рис.3). Программа, в отличие от своих собратьев, имеет довольно понятный интер-

Рис. 1



www.titlize.ru). Казалось бы, вот она, замена Windows Paint (рис.1). Наскоро протестировав ее у себя дома, без задней мысли понес на работу. Там установил на 12 компьютеров и предложил ученикам альтернативу — Painter или Windows Paint, чему все присутствующие на уроке сильно обрадовались. Но спустя минут десять у меня начали волосы вставать дыбом. Один за другим компьютеры начали зависать. Оказалось, школьники очень рьяно взялись за работу и использовали инструменты, отчего и начались сбои. На некоторых компьютерах программа выдала окно, что не хватает памяти для отображения

ко не Blind Write. После ее запуска я узнал, что у меня, оказывается, привод CD-RW с интерфейсом SCSI (рис.2). Вот и покупай после этого программное обеспечение у независимых разработчиков — мало того, что Blind Write была не beta-версии, так еще и платная. Опять чувствую, читатель косится в сторону компьютера, чтобы написать письмо: «Руки у тебя «кривые!» Может, и кривые, однако, сколько багов можно сразу найти.

Еще одна хохма. Одна из минских компьютерных фирм поставила нам компьютер Celtron 1700/256 Mb RAM/40 Gb HDD /17" монитор AOC — вообще, ма-

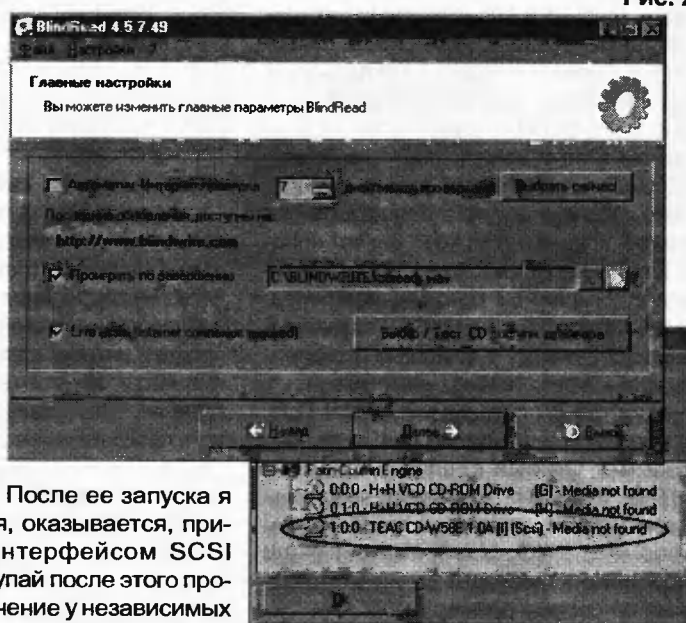


Рис. 2

фейс и такие же функции. Минимум наворотов, максимум производительности и функциональности. А что еще нужно для создания несложных коллажей, рисунков и для работы с фотографиями? Нет нужды описывать все функции VicMan's Photo Editor, на рисунке они и так видны. Хочу лишь остановиться на самом интересном. Так, мне очень пришелся по душе ин-

терфейс программы, он очень похож на интерфейс Photoshop. Впечатляет поддержка разнообразных плагинов, в том числе поддержка следующих расширений файлов: BMP, JPG, GIF, PCX, PNG, PSD, PSP, PCD, WMF, TGA, TIF, PBM, PGM, PPM, RGB, RLA, RLE, RPF, ICB, PCC, PDD, PIC, SCR, SGI, VDA, VST, WIN, EMF, ICO, BW, CEL, CUT, DIB. Также может порадовать стандартный набор инструментов — как в графическом редакторе, чем не могут похвастать многие программы подобного

Рис. 3



класса. Имеются великолепные функции коррекции гаммы и контрастности, возможность автонастройки рисунка. Предусмотрена и возможность создания формул для лучшей настройки изображений по вашему вкусу. Но самое главное — это системные требования: Pentium 166/32 Mb RAM/ Windows 95...XP. На компьютере в конфигурации AMD 450/128 Mb RAM/ 2 Mb video все работало великолепно. На этой оптимистичной ноте, пожалуй, и закончу свой рассказ.

Е.ЛЕБЕДЕВ,
г.Череповец

Новый интерфейс

С появлением ОС Windows XP для пользователей стала доступна возможность кардинально изменять вид своей любимой операционной системы. Раньше все изменения заключались лишь в смене цвета окошек и шрифта. А теперь новый, переработанный движок графической оболочки Windows позволяет изменять любые элементы, цвет, форму и дизайн в целом. В данной статье я хотел бы рассказать, как можно с минимальными усилиями создать красивый и стильный графический интерфейс в вашей любимой Windows XP.

С появлением XP на рынке программного обеспечения "объявились" разработчики ПО, предназначенного для изменения внешнего вида нового детища от Microsoft. Среди таких "контор" я особо выделил бы TGTSOFT и Stardock Software. Первая — разработчик популярной программы StyleXP, а вторая — разработчик не менее популярного WindowBlinds'a и "кучи" других полезных программ. StyleXP и WindowBlinds — очень хорошие программы, которые кардинально меняют дизайн вашей XP, но сегодня речь пойдет не о них, а о другом продукте Stardock Software. В каком-то российском журнале в разделе программного обеспечения я прочитал небольшое описание к программе YzDock. Эта чудо-программка при запуске "вешает" удобную панельку в одной из 4-х частей экрана. Программка заинтересовала меня, но вскоре

я узнал, что она, как говорится, "жрет много ресурсов", и интерес к ней пропал. Недавно я заглянул на сайт www.stardock.com и среди всевозможных утилит для XP, работающих с интерфейсом, заметил программку Stardock ObjectDock. Установочник занимает ≈4,67 Мб. Конечно, это приличный размер, но, скачав программу, я не пожалел о потраченном времени в Интернете.

Установка программы предельно проста, так что эту тему я миную и перейду сразу же к более детальному описанию. ObjectDock представляет собой очень красивую панель, которую пользователь может разместить в одной из 4-х частей экрана (снизу, сверху, слева или справа). После запуска панелька "по умолчанию" появляется в нижней части экрана (рис.1). На мой взгляд, это самое удобное место для ее размещения.

Рис. 1



В чем прелесть панельки? Пользователь размещает самые необходимые программки и папки на панели и тем самым получает быстрый доступ к ним. В дополнение, ObjectDock отображает открытые папки, размещая их слева от себя, а также показывает запущенные программы.

Перед тем как приступить к "загону" папок на панель ObjectDock, следует настроить программу. Затем в "Dock

Settings..." откроется диалоговое окно, в котором 5 закладок:

- Appearance — отвечает за размер иконок на панельке и их размер при наведении на них указателя мыши, настройку расположения ObjectDock'a, выравнивание иконок и фон иконок (лично я ставил его прозрачным);

- Appearance Tweaks — как говорится, "за все надо платить", следовательно, ObjectDock "съедает" определенный ресурс памяти, и для систем средней производительности работа с ObjectDock может сказаться на производительности системы. Закладка AT позволяет настроить эффекты (выбрать, включить/отключить) и настроить шрифт подсказок;

- Docklets — закладка, на которой производится управление всеми известными плагинами (plug-in) для программы;

- Dock Contents — закладка, на кото-

рой настраивается ObjectDock TaskBar (аналог панели задач в Windows XP), и отображение панели задач Windows (показывать ее или нет);

- Performance Tweaks — еще одна закладка, посвященная "разгону", на ней можно настроить главные эффекты программы в соответствии с возможностями вашей системы.

Те, кто имеет свободный доступ в Интернет, могут оперативно получать

информацию о погоде. Данная функция реализована в дополнении (доклете — docklet) — “Weather”. Чтобы включить это дополнение, следует войти в настройки ObjectDock и открыть закладку “Docklets”. В списке выбрать “Weather Docklet” и нажать кнопку добавления доклета на панель — “Add this Docklet to the Dock”.

В программах StyleXP, WindowBlinds, Talisman, IconPackager (как и в самой WinXP) существует понятие “тема” (theme). Тема включает в себя определенный набор рисунков, иконок, звуков и т.п. В ObjectDock есть поддержка тем. Здесь тема включает в себя набор иконок и фон для них. У пользователя есть возможность после создания своего набора иконок с любимыми программами создать свою тему для ObjectDock и тем самым сохранить труды своей работы с панелькой в качестве файла-темы.

В ObjectDock имеется приятная особенность — при наведении на любую из иконок мышкой, она увеличивается (рис.2), размер иконки можно настроить.

Добавление иконок на панель осуществляется путем нажатия правой клавиши на панельке — Add — New shortcut. При этом в конец панельки добавится новый элемент со значком вопроса. Теперь наводим на него курсор, нажимаем правую клавишу и выбираем “Dock Item Properties...”. Откроется диалоговое окно (рис.3).

Рис. 2

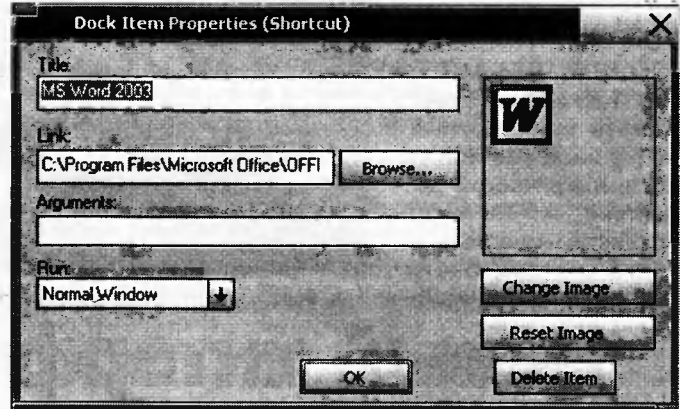


В поле “Title” вводится текст, который будет отображаться при наведении на иконку курсора. В поле “Link” вводится путь к файлу/папке. В “Arguments” вводятся параметры для запуска или ключи. Раскрывающийся список “Run” позволяет выбрать тип запуска папки/приложения: нормальное окно (normal window), свернутое (minimized), развернутое (maximized). Можно изменить иконку приложения/папки, кликнув по кнопке смены изображения — “Change Image”. “Reset image” позволяет вернуться к первоначальному изображению. “Delete Item” удаляет текущий значок. Также можно удалять значки по технологии “Drag-n-Drop”. Для этого достаточно нажать левую клавишу

мышью на значке и утащить его за пределы панельки, потом отпустить клавишу — элемент панели будет удален.

На сайте www.windowblinds.net расположена большая коллекция тем для WindowBlinds, IconPackager, WinAmp, ObjectDock и других популярных программ. Авторы тем стараются организовывать свои “творения” в целые интерфейсы. Например, зайдя в раздел тем для WindowBlinds,

Рис. 3



можно увидеть, что в комментариях к некоторым из них даны ссылки на иконки, панельки и обои к теме.

Источники информации

1. WindowBlinds, IconPackager, ObjectDock — www.stardock.com
2. StyleXP — www.tgtsoft.com

С.РЮМИК,
г.Чернигов.

*Дружба и доверие не покупаются и не продаются.
Г.Троепольский. “Белый Бим Черное Ухо”.*

Интернет-калейдоскоп, freeware #6

Согласно традиции, заведенной французами, красоту европейских столиц принято сравнивать с Парижем. Среди достопримечательностей “жемчужины Европы” — Лувр и Марсово поле, Эйфелева башня и Пантеон, Елисейские поля и... крупнейшие в мире “блошинные рынки”. Характерно, что и туристы, и коллекционеры с удовольствием тратят свободное время на посещение рынков Порт Сент-Уан (см. фото) и Порт де Монтрейль (никаких портов близко нет, “porte” по-французски — это “ворота”). Каждый из рынков, что отдельный город с проспектами, улицами, переулками, в которых можно найти антикварные и современные вещи — от прабабушкиной

иголки до ритуальной африканской маски. Говорят, кто приходит пораньше, тому достаются более редкие вещи, кто позже — более дешевые.



Интернет тоже можно сравнить с огромным “блошиным рынком”, где “сувенирами” являются программы и текстовые файлы. Особенно приятно, если они поставляются на условиях “freeware”, то есть без оплаты.

Девять причин, которые побуждают людей раздавать бесплатный софт, перечислены в [1]. К ним можно добавить еще две.

10. Бесплатная программа специально создается программистом как “портфолио” (от англ. portfolio — портфель) при поиске работы за рубежом через многочисленные “рекрутские” фирмы. Это своеобразная визитная карточка и аттестат зрелости.

11. Некоторые программисты при временном отсутствии постоянной работы, чтобы не терять профессиональную форму, работают над свободно распространяемыми проектами. При этом отзывы, критика и по-

желания пользователей являются мощным стимулом к изучению новых приемов программирования, что безусловно пригодится в будущем.

Как бы там ни было, но люди, которые делятся своими программами, заслуживают уважения. Интернет-странички некоторых из них представлены в этом обзоре.

Математика, бухгалтерия, логические игры

Все перечисленные в заголовке слова встречаются на домашней странице Антона Волнухина (г.Москва) <http://setia.ru>. Здесь размещены девять созданных им программ различной тематики, каждая из которых

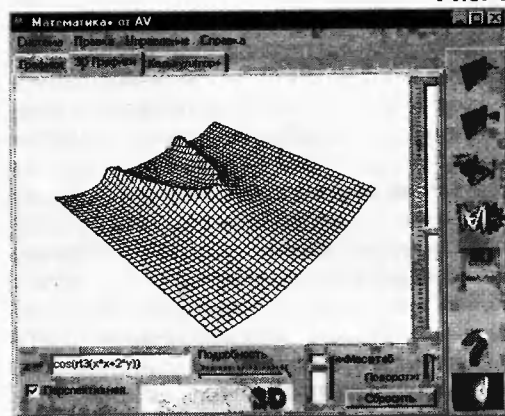


Рис. 1

выполнена со знанием дела. Например, "Математика+ от AV" (<http://old.setia.ru/Software/avmsetup.exe>, 928 Кб) — программа для визуализации двух- и трехмерных графиков (рис.1); "AV Баланс" (<http://old.setia.ru/Software/avbsetup.exe>, 1167 Кб) — бухгалтерские расчеты; "AV Sound-Manager" (<http://old.setia.ru/Software/snmsetup.exe>, 635 Кб) — музыкальный терминал; "Шашки AV" (<http://old.setia.ru/Software/avcsetup.exe>, 2378 Кб) — объемные, с музыкой; "Шашки Рэндзю" (<http://old.setia.ru/Software/rndzsf.exe>, 138 Кб) — "крестики-нолики" на доске 15x15 клеток.

Труд программиста не остался незамеченным. На престижном сайте <http://www.softbox.ru> его программы награждены знаками "Выбор редакции" и "Рекомендуем".

Ключевые слова для поиска: Математика+, AV, Шашки Рэндзю, Антон Волнухин.

"Шпорный" вопрос

Шпаргалка занимает достойное место в школьном и студенческом фольклоре. Различают следующие

разновидности шпаргалок: текстовые, графические, радиотехнические, бомба, гармошка, сломанный карандаш, сигареты, дискета, боди-арт, прыгалка, галстук, мобильный телефон, калькулятор, пейджер. Подробности можно прочитать на сайте с характерным адресом <http://myshpora.narod.ru>. Там же приведены программы: "Krizlog!" (<http://myshpora.narod.ru/krizlog.zip>, 835 Кб) — краткое изложение всех школьных литературных произведений с характеристиками героев; "Algoritms!" (<http://myshpora.narod.ru/algoritms.zip>, 574 Кб) — конспект математических формул и алгоритмов, полезных любому; "KNIGI!" (<http://myshpora.narod.ru/knigi.zip>, 353 Кб) — ускоренная "читалка" больших по объему книг.

На последней программе хотелось бы остановиться подробнее. Трудно судить, насколько эффективен способ обучения, когда слова быстро мелькают на экране монитора. Здесь интересен сам подход, схожий с изучением иностранного языка во сне. Если пофантазировать, можно усовершенствовать программу чтением слов "через одно" или

Ключевые слова для поиска: шпаргалка, Krizlog, myshpora, Help API.

Интернет-стрелочник

Знакомая ситуация — программ, выполняющих требуемую функцию, много, но каждая из них чем-то не удовлетворяет. Программист, разбирающийся в тонкостях дела, без сомнения, способен составить на основе массы аналогичных свою, "домашнюю", программу. Можно добавить к ней удобный пользовательский интерфейс и разместить в Интернете для всех желающих. Так, например, сделано на сайте <http://vexer.ru>. Программа "Стрелочник 1.4" (<http://vexer.ru/soft/linkwzrd.zip>, 226 Кб) была составлена "для дома, для семьи" (рис.3). Она позволяет "на лету" копировать и систематизировать интернет-ссылки посещаемых страниц. "Стрелочник", в отличие от программы докачки "FlashGet", справляется с любыми ссылками: html, ftp, mailto, адреса электронной почты.

Автор программы взял себе псевдоним "V.exeR". На его сайте размещены еще четыре работы: "VexSheduler" (<http://vexer.ru/soft/vexsheduler.zip>, 171 Кб) — одновременный запуск нескольких программ по сценарию; "VexPack for Delphi" (<http://vexer.ru/soft/vexpack.zip>, 19 Кб) — набор компонентов Delphi; "TexGen" (<http://vexer.ru/soft/texgen.zip>, 168 Кб) — генерация tile-текстур; "Досье" (<http://vexer.ru/soft/dossie.zip>, 364 Кб) — программа для хранения и систематизации данных о

Рис. 3

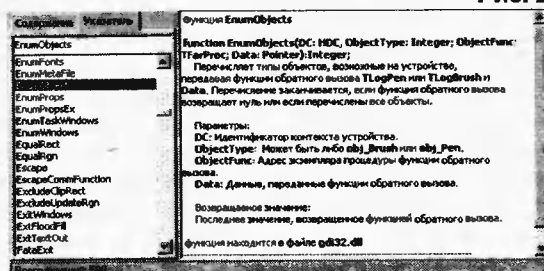
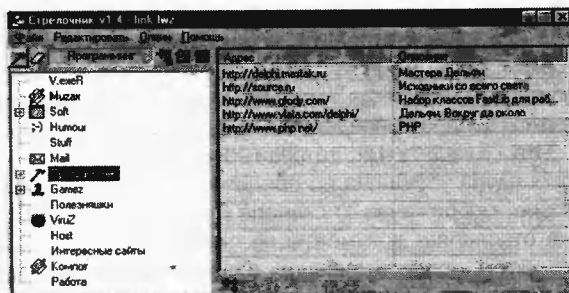


Рис. 2

"через два", главное, чтобы время индикации было не менее 0,25 с. Правда, далеко не всю литературу следует читать подобным образом, хотя понятным это становится не в школьные годы, а гораздо позже...

Еще одно собрание подсказок-шпаргалок для программистов размещено на домашней страничке Николая Мерзлякова <http://nikolayam.narod.ru>. Среди четырех его проектов можно выделить "Справочник по Win32 API" (рис.2) на русском языке (http://nikolayam.narod.ru/api_help.zip, 291 Кб).



людях, и не только.

Ключевые слова для поиска: V.exeR, программа "Стрелочник".

Литература

1. Рюмик С. С Интернета по нитке... — Радиомир. Ваш компьютер, 2002, №8, С.9-12.



В.КУЦ,

E-mail: victor_kootz@mail.ru

Факс в современном офисе — теперь это очень просто!

Любой мало-мальски приличный офис просто невозможно представить себе без факсимильного аппарата — это, сегодня уже практически незаменимого средства обмена текстовыми и графическими документами, количество которых, в связи со стремительным развитием “безбумажных” технологий и не менее стремительным повышением общего уровня бюрократизации нашей жизни, неудержимо растет. А так как бумага и прочие расходные материалы для факсов весьма недешевы, да и обслуживать такие приборы — дело весьма хлопотное, то в последнее время наметилась определенная тенденция по переводу большей части офисного документооборота (в том числе и “факсооборота”) на базу обычных персоналок. Еще одним аргументом в пользу использования ПК в качестве факсов может стать удобство учета и систематизации полученной информации. Распечатывать полученные документы при этом совершенно не обязательно.

Но вот незадача — практически все известные сегодня программные факсы ориентируются на работу исключительно на локальных компьютерах, в большей или меньшей степени эмулируя обычный факсимильный аппарат. А ведь не секрет, что автономные компьютеры давно уже стали анахронизмом в современном офисе, опутанном сверху донизу (в прямом и переносном смысле) локальными сетями.

Одним из наиболее эффективных решений данной проблемы может стать использование в корпоративных локальных сетях новейшего программного продукта Venta4Net, созданного российской компанией Вента на базе популярной программы VentaFax, являющейся его сетевой версией. Программа эта позволяет отправлять факсимильные и голосовые сообщения с любого компьютера локальной сети учреждения через модем (или несколько модемов), установленный на единственном компьютере этой сети — сервере Venta4Net. Принятые сообщения (в зависимости от настроек) могут быть адресованы одному или нескольким пользователям — клиентам Venta4Net. Связь между клиентскими машинами и сервером осуществляется по стандартному сетевому протоколу TCP/IP. Но о сетевых возможностях программы мы поговорим чуть позже, а сейчас я хочу кратко остановиться на основных, факсимильных, возможностях программы Venta4Net.

А они, эти возможности, действительно впечатляют. Так, при передаче факсов через Venta4Net, помимо обычных для автономных факсимильных аппаратов режимов автоматической или ручной отправки, можно рассылать как одиночные факсы, так и целые их пакеты по расписанию, указав промежуток времени, когда Venta4Net будет включаться и отправлять факсы заранее определенным абонентам.

Что же касается подготовки нового сообщения, то сама эта процедура упрощена до предела — Venta4Net получает готовое изображение от сканера (по протоколу TWAIN), а также отправляет документы из любых офисных приложений Windows, поддерживающих функцию печати (на виртуальный принтер VentaFax, устанавливаемый в систему при инсталляции программы). Для еще большего облегчения процедуры отправки факса можно воспользоваться преимуществами, предоставляемыми технологией Drag&Drop — просто перетащить пиктограмму имеющегося документа на ярлык Venta4Net на Рабочем столе или на ее открытое окно.

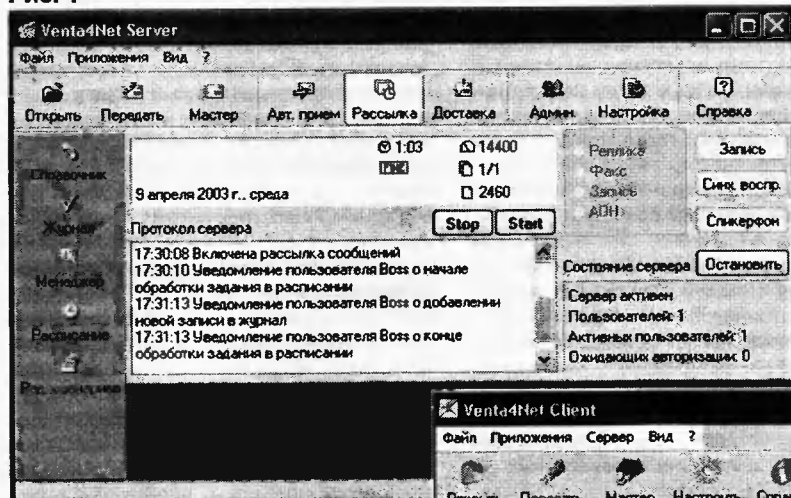
Так же просто, как и при использовании обычного факсимильного аппарата, может производиться прием факсов как в ручном, так и автоматическом режиме. Принятые факсимильные сообщения сохраняются в виде файлов стандартного гра-

фического формата (TIFF), и могут, если возникнет такое желание, автоматически распечатываться сразу по окончании их приема. Если автоматически распечатывать все принятые факсы нет необходимости, их можно просмотреть и, если нужно, отредактировать, используя приложение Менеджер сообщений. Этот Менеджер, конечно, далеко не Photoshop и даже не Paint, однако для выполнения самых элементарных операций он вполне сойдет. Принятые голосовые сообщения сохраняются также в стандартном для Windows голосовом формате (WAV). Эти две особенности позволяют избежать каких-либо проблем при последующей обработке полученных сообщений. Более того, в программе Venta4Net наконец-то воплотилась мечта всех конторских “рыцарей пера и скрепки” — благодаря тесной ее интеграции с системным почтовым клиентом, два наиглавнейших вида офисных коммуникаций — факсимильная связь и электронная почта — оказались объединены в рамках единого рабочего пространства программы.

Подвляющее большинство современных модемов поддерживают голосовые функции, а значит, и у компьютера с установленной на нем программой Venta4Net появилась возможность стать автоответчиком и АОНам в одном лице. При этом автоответчик имеет механизм дистанционного управления, позволяющий не только прослушать поступившие голосовые сообщения с любого внешнего телефона, поддерживающего тональный набор, но и организовать разветвленную систему голосовых меню, например, для автоматической голосовой и/или факсимильной справки или даже персональных голосовых почтовых ящиков. Но и это еще не все — наряду с рассылкой факсов по расписанию, программа обеспечивает такую же рассылку голосовых сообщений (телефонограмм), а при необходимости — еще и запишет ответ абонента. Использование голосовых функций модема позволяет расширить и возможности факсимильной рассылки — если при попытке передачи факса будет обнаружено, что трубку снял человек, то программа произнесет фразу “Барышня, примите, пожалуйста, факс” или любую другую, на ваше усмотрение. Запись и проигрывание голосовых сообщений производится с помощью очень удобной встроенной утилиты, а для всех тех, кому и это не под силу (или просто лень), в комплекте поставки программы имеется достаточный набор типовых речевых заготовок и целых фраз, произносимых приятным женским голосом и пригодных практически для всех случаев жизни. Программа Venta4Net может также просто записывать телефонный разговор. Совместная работа автоответчика и АОНа в рамках одной программы может обеспечить такую удобную функцию как способность автоответчика выдавать для разных абонентов разные сообщения. Радует и довольно большое количество настроек АОНа, что может оказаться совсем не лишним на наших, сами знаете каких, телефонных линиях — вручную можно подстраивать частоты тоновых сигналов, их задержки и длительность, а также определить, будет ли определение номера абонента осуществляться самой программой или встроенными средствами модема. Модемы со встроенной процедурой определения номера (например, знаменитый Courier) обеспечивают наиболее качественное распознавание номера абонента.

В программе Venta4Net предусмотрены средства управления ею из внешних приложений, помогающие стыковать ее с существующими в организации системами документооборота. Для максимальной автоматизации процесса подготовки факсимильного сообщения и включения его в расписание для

Рис. 1



дальнейшей отправки предусмотрен механизм запуска программы с параметрами, обеспечивающими создание и корректировку расписания и управление режимами рассылки. Существенно расширить возможности программы в части обработки и последующего сохранения полученных сообщений позволяет реализованный в ней механизм подключения плагинов. Плагин представляет собой файл динамически подключаемой библиотеки (*.DLL) и позволяет выполнять действия, специфические для конкретного пользователя, например, сохранять полученную информацию в собственных базах данных.

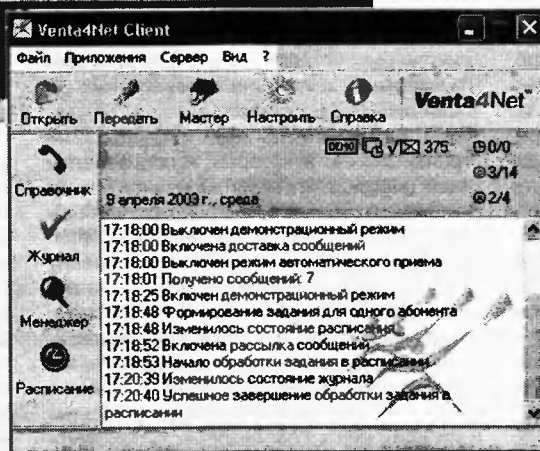
Как уже упоминалось выше, программа Venta4Net является комплексом, состоящим из одной серверной и нескольких клиентских частей. Наибольший интерес, конечно, представляют серверные модули, устанавливаемые на компьютерах, оснащенных модемами. В зависимости от потребности заказчика программы, серверы могут быть как однолинейные, так и многолинейные (обслуживающие от 2 до 32 телефонных линий), способные принимать и отправлять голосовые и факсимильные сообщения сразу по нескольким телефонным линиям (для этого, естественно, на компьютере должно быть установлено соответствующее количество модемов). При этом режим работы программы для каждой линии можно настроить индивидуально, вплоть до определения разных правил набора номера (например, с префиксом выхода в городскую телефонную сеть и без префикса — для внутренней) для каждой из телефонных линий. Для снижения сетевого трафика при общении сервера с клиентскими модулями в программе используется сжатие информации при передаче файлов журнала и расписания (сами же сообщения и без этого максимально сжаты).

Главное рабочее окно программы, лишенное традиционно для VentaFax изображения факсимильного аппарата, обрело гораздо более строгий вид (рис.1). Помимо панели с кнопками управления основными функциями программы, в левой части окна расположена вертикальная панель выбора приложений, позволяющая быстро просмотреть (или прослушать) принятое или готовое к отправке сообщение; просмотреть и, если понадобится, отредактировать расписание рассылки; открыть редактор сценариев дистанционного управления автоответчиком. Кроме того, из этой же панели доступны такие приложения как Журнал и Телефонный справочник.

Журнал является основным средством отчетности поль-

зователей программы и служит для хранения информации о всех действиях, совершенных программой Venta4Net в течение определенного времени. Каждая запись в журнале содержит данные о приеме или передаче сообщения, воспроизведении того или иного звукового файла в телефонную линию и т.п. Записи в журнале могут выделяться цветом и сортироваться по различным правилам. Другое важное приложение — Телефонный справочник — предназначено для ведения базы данных голосовых и факсимильных телефонов абонентов. Справочник может использоваться для автоматического заполнения

Рис. 2



полей окна передачи, а также при формировании расписания рассылки. Кроме того, Телефонный справочник предоставляет доступ к данным, хранящимся в стандартной Адресной книге Windows, правда, только в режиме просмотра.

Задания, созданные пользователями на клиентских компьютерах, передаются на сервер по протоколу TCP/IP и помещаются в общее расписание. Сервер обрабатывает расписание и осуществляет рассылку сообщений. Пользователи клиентской части программы, имеющие соответствующие права доступа, помимо

создания, отправки и приема факсов, могут просматривать (или даже редактировать) расписание рассылки и Журнал на сервере, а также создавать свои собственные, индивидуальные графики рассылки и телефонные справочники (рис.2).

Заканчивая разговор об интерфейсе, хочется лишь раз подчеркнуть, что в столь сложной программе как Venta4Net, насыщенной самыми различными настройками, использование русского языка как в оформлении рабочих окон, так и в файле помощи (который, на удивление, очень подробен и написан ясным и доходчивым языком) становится просто насущной необходимостью, и вполне может оказаться той решающей каплей, склонившей чашу весов в пользу отечественного продукта при выборе факс-программы для использования в вашем офисе. Еще одним достоинством программы Venta4Net, способным в существенной мере облегчить ее использование в тяжелых условиях российской действительности, являются просто исчерпывающие настройки всех необходимых рабочих параметров как самой программы, так и установленного на компьютере модема.

Демонстрационная версия пакета Venta4Net, свободно распространяемая через Интернет, предназначена исключительно для тестирования и, простите за тавтологию, демонстрации возможности взаимодействия серверной и клиентской частей программы в локальной сети, но никак не для нормальной работы. В этой версии к серверной части может быть одновременно подключено не более трех клиентов, причем сервер принимает сообщения от клиентов, но не отправляет их, а лишь имитирует процесс передачи. Сделано это, скорее всего, с целью борьбы с любителями использования "взломанных" программ, хотя трудно представить себе уважающего себя бизнесмена, унижающегося до использования в своем офисе "крякнутой" программы.

Более подробную информацию о Venta4Net можно получить на официальном сайте программы — <http://www.venta4net.ru>.



ЛИСТАЯ СТРАНИЦЫ

О новых перспективных технологиях и конструкциях транзисторов, разработанных специалистами ф. Intel, рассказывает С. Пахомов в статье "Революция в производстве транзисторов" (КомпьютерПресс, №1/2004, С.76...83).

Вся история электронной промышленности, начиная с 1959 г., когда была создана первая интегральная микросхема, развивалась по пути уменьшения размеров транзисторов и увеличения плотности их размещения на кристалле. Так, самые первые интегральные микросхемы содержали всего несколько десятков транзисторов, а современный процессор Intel Pentium 4 насчитывает уже 55 млн. транзисторов на кристалле. Считается, что плотность размещения транзисторов на кристалле подчиняется так называемому закону Мура, согласно которому количество транзисторов в одной микросхеме удваивается каждые 18 месяцев. Этот закон (а точнее, предсказание) был сформулирован Гордоном Муром (Gordon E. Moore) еще в 1965 г., когда по просьбе журнала Electronics он написал статью, в которой был дан прогноз относительно того, как будут совершенствоваться полупроводниковые устройства в течение ближайших десяти лет. Проанализировав темпы развития полупроводниковых устройств и экономические факторы за прошедшие шесть лет, то есть с 1959 г., Мур предположил, что к 1975 г. количество транзисторов в одной интегральной микросхеме составит 65 тыс., то есть за десять лет оно увеличится более чем в 1000 раз. А это означало, что каждый год количество транзисторов в одной микросхеме должно удваиваться.

Конечно, в те времена ни сам Гордон Мур, ни кто-либо другой не мог предположить, что опубликованный прогноз на ближайшие десять лет не только в точности сбывается, но и послужит основой для формулирования эмпирического правила развития всей полупроводниковой технологии на много лет вперед. Впрочем, с предсказанием Мура было не все гладко. К 1975 году рост количества элементов в одной микросхеме стал немного отставать от прогноза, и тогда Мур скорректировал период обновления до 24 месяцев, чтобы компенсировать ожидаемое увеличение сложности полупроводниковых компонентов.

Уменьшение размеров транзисторов положительно сказывается на их характеристиках. Так, если считать, что длина затвора МОП-транзистора, являющегося основным элементом современных цифровых интегральных схем, уменьшается в M раз, то в такое же количество раз уменьшаются и толщина слоя диэлектрика, отделяющего область затвора от кремниевой подложки, и ширина затвора, и рабочее напряжение затвора. Кроме того, в M раз возрастает скорость работы транзистора, в M^2 раз увеличивается плотность размещения транзисторов на кристалле, а рассеиваемая мощность уменьшается в M^2 раз.

Сегодня, когда размер транзисторов от поколения к поколению неуклонно уменьшается, а плотность их интеграции на кри-

сталле стремительно возрастает, разработчикам становится все труднее решать технологические проблемы, возникающие в силу действия естественных законов физики. Главными барьерами на пути миниатюризации транзисторов являются выделение тепла при работе транзистора и утечка электрического тока в том же процессе. Чем меньше транзистор, тем выше тепловыделение и тем больше ток утечки. Токи утечки возникают как через слой очень тонкого подзатворного диэлектрика, так и между стоком и истоком при выключенном состоянии транзистора.

При производстве процессоров по современному 90-нанометровому технологическому процессу, толщина слоя диоксида кремния, являющегося подзатворным диэлектриком, составляет 1,2 нм, то есть всего 5 атомных слоев, а в лабораторных условиях были продемонстрированы результаты экспериментальных исследований транзисторов, в которых использовался слой диоксида кремния величиной 0,8 нм (длина затвора — 15 нм). Такие приборы обладают хорошей управляемостью, однако фактически это уже близко к пределу для данного материала, поскольку в результате дальнейшего уменьшения самого транзистора и, как следствие, уменьшения толщины слоя диоксида кремния, ток утечки через диэлектрик затвора заметно возрастет. По оценкам экспертов корпорации Intel, уже в современных чипах почти 40% энергии теряется из-за утечек.

Для решения этой важнейшей проблемы корпорация Intel планирует заменить используемый в настоящее время диоксид кремния слоем диэлектрика с высокой диэлектрической проницаемостью. Применение такого материала позволяет использовать более толстые слои диэлектрика и, следовательно, снизить токи утечки при сохранении прежних характеристик управляемости транзистора.

В качестве подзатворных диэлектриков рассматривались различные варианты, в частности, Y_2O_3 , La_2O_3 , ZrO_2 , HfO_2 , Ta_2O_5 и др. Так, согласно информации компании Intel, большинство таких диэлектриков изготовлены на основе гафния и циркония. К примеру, диоксид циркония обладает диэлектрической проницаемостью, равной 25, что в 6,4 раза больше, чем диэлектрическая проницаемость диоксида кремния. Следовательно, при использовании диоксида циркония слой диэлектрика можно сделать в шесть с лишним раз более толстым, а это позволяет снизить ток утечки примерно в 10 тыс. раз.

Второй важнейшей проблемой является поиск нового материала затвора, который хорошо сочетается с новым диэлектриком. Именно в разработке такого сочетания и заключается революционный прорыв компании Intel. Естественно, о подробностях этого открытия не сообщается, известно только, что речь идет о сочетании диэлектрика со сплавом металла. По сведениям корпорации Intel, сочетание нового диэлектрика и нового сплава для изготовления затвора позволяет более чем в 100 раз по сравнению с диоксидом кремния снизить токи утечки, сохранив при этом высочайшие рабочие параметры транзисторов.

Специалисты корпорации Intel убеждены, что новые материалы, созданные в результате многолетних исследований, можно интегрировать в экономически эффективный массовый производственный процесс. Транзисторы, изготовленные на основе таких материалов и обладающие рекордными параметрами производительности, рассматриваются в качестве базового варианта для будущих процессоров Intel в рамках производственного процесса с проектной нормой 45 нм, который будет запущен в массовое производство уже в 2007 г.

В транзисторах следующего поколения будет использоваться еще одна новинка — технология напряженного, или растянутого, кремния (Strained Silicon Technology), которую корпорация Intel планирует использовать уже при производстве микросхем с проектной нормой 90 нм. Идея указанной технологии предельно проста. Известно, что создание напряжений в пространственной решетке — таких, чтобы атомы кремния находились друг от друга на расстоянии, чуть большем чем их естественное расстояние — приводит к ускорению переключений транзисторов с каналом n -типа (так же как сжатие пространственной решетки приводит к аналогичному эффекту для транзисторов с каналом p -типа). Это "растяжение/сжатие" именуют напряжением. Инженеры подразделения Logic Technology Development Division корпорации Intel разработали два различных способа для n - и p -канальных транзисторов. В n -канальных приборах поверх транзистора в направлении движения электрического тока наносится слой нитрида кремния (Si_3N_4), вследствие чего кремниевая кристаллическая решетка "растягивается". В p -канальных транзисторах это достигается за счет нанесения слоя $SiGe$ в зоне образования переносчиков тока — здесь решетка "сжимается" в направлении движения электрического тока, а поэтому "дырочный" ток течет свободнее. В обоих случаях прохождение тока значительно облегчается: в первом случае — на 10%, во втором — на 25%, а соединение обеих технологий дает ускорение на 20...30%.

Другим направлением исследования являются так называемые трехмерные tri -gate-транзисторы с тройным затвором, о разработке которых было объявлено в сентябре 2002 г. Такие транзисторы рассматриваются в качестве основы для 45-нанометрового технологического процесса 2007 г. Рассмотренные ранее приборы были планарными, то есть имели один плоский затвор, параллельный поверхности кремниевой подложки. Транзистор на основе технологии tri -gate представляет собой прямоугольный параллелепипед, выступающий над плоскостью кристалла ИС, а затвор как бы "обернут" вокруг трех его сторон.

Данная структура позволяет пропускать электрический ток как вдоль верхней поверхности тела транзистора, так и вдоль обеих его боковых частей. Благодаря подобной схеме распределения тока значительно увеличивается площадь, доступная для прохождения тока, а значит, снижается его плотность, а вместе с ней уменьшаются сопротивление транзистора в открытом состоянии



и утечки. Особенностью этой конструкции также являются поднятые исток и сток — в результате еще больше снижается сопротивление, что позволяет транзистору работать

при токе меньшей плотности.

Важно отметить, что транзисторы на основе технологий "новый материал диэлектрика/новый материал затвора" и tri-gate

не противоречат, а, скорее, дополняют друг друга. Поэтому транзистор следующего поколения, вероятно, будет воплощать в себе обе технологии.

Результатам тестирования жестких дисков с новым интерфейсом Serial ATA посвящена статья Н. Левского "Революция неизбежна" (CHIP, №1/2004, С.59...65).

Интерфейс Serial ATA, разработанный чуть более года назад, становится все популярнее. Практически все материнские платы, сходящие с конвейеров ведущих производителей, оснащаются таким контроллером. Доля выпускаемых жестких дисков с таким интерфейсом также увеличивается. Многим пользователям кажется, что смена интерфейса не прибавит производительности. Но тестирование, проведенное в редакции журнала "CHIP", доказывает, что такое мнение неверно. Компании, занимающиеся разработкой жестких дисков, работают над улучшением технических характеристик своих изделий. Поэтому каждый следующий продукт получается несколько лучше, чем предыдущий. Действительно, год назад жесткий диск Seagate Barracuda V, являвшийся первым накопителем, поддерживавшим новый интерфейс, мог равняться на лучших среди IDE-накопителей, а сегодня он проиграл по всем статьям современным устройствам с интерфейсом Serial ATA. Такое обстоятельство может свидетельствовать о том, что модели, рассмотренные в тесте, предоставляют пользователям больше возможностей, чем другие, работающие по старому интерфейсу.

Нельзя не отметить тот факт, что не все производители накопителей перешли на технологию Serial ATA. Так, в частности, как в России, так и во всем мире можно найти лишь продукцию компаний Seagate, Maxtor и Western Digital. Другие компании, такие как Hitachi и Samsung, пока еще продолжают производить жесткие диски с интерфейсом IDE.

Несмотря на то что в целом диски Serial ATA на российский рынок есть, сказать о том, что их ассортимент велик, нельзя. В лучшем случае, у производителя есть две линейки накопителей, а компания Seagate

предлагает продукты вообще лишь одной серии Barracuda 7200.7. Справедливости ради стоит сказать, что российский пользователь при желании еще может найти диски серии Barracuda V, но сделать это с каждым днем будет все сложнее и сложнее, поскольку производитель прекратил поставки этой модели.

Для проведения тестирования были отобраны все доступные в России модели жестких дисков Serial ATA. В их число вошли семь устройств: DiamondMax Plus 9 и MaxLine Plus II от Maxtor, Raptor WD360GD и WD1600JD от Western Digital, Barracuda 7200.7 ST380013AS, Barracuda 7200.7 ST312026AS и Barracuda V ST3120023AS от Seagate. Их технические характеристики приведены в табл.1, результаты измерений — в табл.2. Единственной общей чертой этих накопителей является объем буфера, который составляет 8 Мб. Однако наличие большого объема еще ни о чем не говорит, поскольку средняя скорость чтения или записи зависит также и от алгоритма управления эшшем.

Кроме того, стоит отметить и тот факт, что компания Western Digital первой в мире предложила жесткий диск с интерфейсом Serial ATA, частота вращения шпинделя которого составляет невиданную ранее величину 10000 об/мин. В недалеком прошлом, напоминает автор, пользователям такие скорости были доступны лишь на винчестерах с интерфейсом SCSI, объем которых был невелик, а стоимость могла в 3...4 раза превышать цену обычных накопителей с интерфейсом IDE. Накопитель Raptor унаследовал и другие черты SCSI-винчестеров. В частности, его объем составляет всего 34,5 Гб, а стоит он почти столько же, сколько Seagate ST310026AS, объем которого больше в три раза.

Для тестирования был использован компьютер на базе процессора Intel Pentium 4 3,20 ГГц (частота системной шины — 800 МГц), материнской платы AOpen AX4SG Max (чипсет Intel 865G) и видео-

карты AOpen Aeolus FX5600 (GeForce FX5600). Все тестируемые накопители подключались к интегрированному контроллеру Silicon Image Sil3112. Компьютер работал под управлением операционной системы Windows XP Service Pack 1.

В процессе тестирования оценивались максимальная, минимальная и средняя скорости чтения и записи, а также уровень загрузки центрального процессора при этих операциях. Для этой цели была использована программа HDtach версии 2.61. Определялось максимальное значение каждой группы величин, его переводили в процентную форму, после чего вычислялось среднее значение производительности для каждой модели.

С помощью шумомера AZ Instruments AZ8922 проводились замеры уровня шума (в децибелах), создаваемого устройством. Наивысшую оценку получал претендент, работающий максимально тихо в двух режимах — холостом и под нагрузкой. Чтобы смоделировать максимальную нагрузку, использовался бенчмарк, который нагружал жесткий диск.

Для оценки энергопотребления мультиметром BeeTech 70 измерялись пиковые значения потребления тока по обоим цепям питания.

По результатам тестирования оценку "Лучший продукт" получил Western Digital Raptor WD360GD. Шпиндель винчестера вращается с невиданной до этого скоростью десять тысяч оборотов в минуту. Данный ход, скорее всего, был продиктован маркетинговыми соображениями. Естественно, винчестер будет выделяться на фоне других, частота вращения шпинделей которых составляет всего лишь 7200 об/мин. Эта модель конечно быстрее остальных. Однако разница в производительности не столь велика, как хотелось бы потенциальному покупателю. Следует обратить особое внимание на среднюю скорость чтения и записи данных. По этому показателю Raptor действительно обходит все другие модели. Вторым важным

Табл. 1

Модель	DiamondMax Plus 9	MaxLine Plus II	Raptor WD360GD	WD1600JD	ST380013AS	ST312026AS	ST3120023AS
Емкость, Гб	57,3	233,8	34,5	149,1	74,528	111,8	111,8
Скорость вращения шпинделя, об./мин	7200	7200	10000	7200	7200	7200	7200
Цена, евро	65	245	100	125	70	90	110
Стоимость 1 Мб, центы	0,11	0,10	0,28	0,08	0,09	0,08	0,10

Табл. 2

Модель	DiamondMax Plus 9	MaxLine Plus II	Raptor WD360GD	WD1600JD	ST380013AS	ST312026AS	ST3120023AS
Суммарная потребляемая мощность, Вт	13,35	16,88	14,43	13,52	15,65	16,95	14,11
Средняя скорость чтения, Кб/сек	42738,8	48830,8	48673,4	47320	45122	45504,8	36442,8
Средняя скорость записи, Кб/сек	25271,6	20361,6	34851,9	27195	39982	44353,8	20364,3
Время доступа, мс	13,5	14,1	8,1	13,3	12,6	12,2	13
Загрузка CPU, %	0,6	0,6	0,9	0,6	0,8	0,7	6,6
Уровень шума в холостом режиме, дБ	36,3	36,4	35,8	36,7	35,5	36,8	35,2
Уровень шума в рабочем режиме, дБ	39,1	40,8	44,2	38,3	39,8	40,9	40,5



отличием от других устройств стоит признать малое время поиска данных.

К достоинствам WD360GD следует отнести качество исполнения корпуса. Помимо специальных радиаторов, расположенных с одной стороны корпуса, и двух разъемов питания, стоит отметить усиленную пластиковую конструкцию, предохраняющую все разъемы от поломки. Качество изготовления печатной платы тоже на высоком уровне. Единственным недочетом стоит признать наличие на внешней стороне платы микросхемы, которую можно повредить при неаккуратном обращении.

Оценку "Оптимальный выбор" получил Seagate Barracuda 7200.7 ST380013AS.

✂

Современный ПК занимает на рабочем столе достаточно много места, поэтому нередко для установки компьютера приходится приобретать специальную компьютерную мебель. Однако новейшие компьютерные технологии позволяют решить эту проблему кардинально — **разместить современный ПК в миниатюрном корпусе**. Примером является устройство, рассмотренное в заметке О.Денисова и С.Назарова "Shuttle SB65G2 XPC: стильная "малопитражка" для дома" (Подводная лодка, №1/2004, С.28).

В компьютере используется процессор Intel Pentium 4/Celeron с системной шиной, работающей на частотах 800/533/400 МГц, и технология Hyper-Threading; чипсет Intel 865PE и ICH5R и двухканальная системная память DDR400/333/266 SDRAM. На материнской плате имеются двухпортовый IEEE 1394 контроллер VIA VT6307, сетевой 100-Мбит/с Ethernet-контроллер Realtek 8100B, шестиканальный аудиокodeк Realtek ALC650, Wi-Fi-модуль беспроводной связи Wireless LAN PRISM IEEE 802.11b WM168 USB Module. Здесь установлены два гнезда DIMM, гнезда AGP и PCI, двухпортовый разъем USB, параллельный порт, порты IrDA и CD-In, по два разъема Serial ATA и Ultra ATA и разъем FDC для флоппи-накопителя.

На передней панели корпуса установлены микрофонный вход, линейный вход, линейный выход, четырехконтактный порт IEEE 1394, два порта USB; на задней пане-

✂

Недалек тот час, когда практически все бытовые устройства в вашем доме будут компьютеризованы и при этом смогут общаться друг с другом по радиоканалу. Это позволит, например, СВЧ-печи связаться с мобильным телефоном хозяина, который утром выгуливает собаку, и сообщить, что его завтрак уже готов. Аналогичным образом поступит автоматизированная кормушка для аквариумных рыбок. Она выдаст очередную порцию корма и через Интернет сообщит вам, находящемуся в отпуске на Таити, что все в порядке. О том, что **гиганты электронной индустрии уже работают в этом направлении**, рассказывает С.Пахомов в статье "Перспективы развития радиотехнологий" (КомпьютерПресс, №1/2004, С.84...87).

Еще в феврале 2002 г. на ежегодном Форуме корпорации Intel для разработчиков

Данное устройство очень похоже на сородича предшественника, Barracuda V, но при этом практически не имеет ничего общего с Barracuda IV. Вероятно, компания-производитель решила отказаться от фирменного стиля — двусторонней защиты жесткого диска посредством металлических пластин. Теперь металлическая пластина осталась только одна, и прикрывает лишь механические части накопителя.

Пластмассовый корпус разъемов выполнен качественно. Здесь нет четырехштырькового разъема питания, его место полностью закрыто пластиковой вставкой. В целом корпус создает впечатление массивной детали, которую трудно повредить. Вместе с

ли — шестиконтактный порт IEEE 1394, четыре порта USB, сетевой разъем RJ-45, последовательный порт, два порта PS/2, цифровой оптический S/PDIF вход-выход, три аналоговых аудиоразъема для вывода шестиканального звука, гнездо для Wi-Fi-антенны. Имеются один внутренний 3,5-дюймовый и два внешних — 3,5- и 5,25-дюймовый — отсека для накопителей. Габариты компьютера — 300x200x185 мм, его масса — 2,85 кг, в нем использован блок питания мощностью 220 Вт.

Внешне SB65G2 выглядит весьма стильно — компьютер заключен в алюминиевый корпус черного цвета, внутри которого "спрятаны" малогабаритная системная плата и фирменная I.C.E.-подсистема охлаждения ЦП и корпуса. Последняя работает так. На микросхеме ЦП закрепляется медная подошва с ребристым радиатором, охлаждаемая жидкостью. Нагретая жидкость поступает вверх по трубкам, охлаждается пластинчатым металлическим радиатором, который обдувается вентилятором, и, остыв, проходит вниз по трубкам к металлической подошве.

Таким образом, для охлаждения ЦП и корпуса ПК применяется только один вентилятор, что значительно снижает уровень шума. Более того, скорость вращения последнего регулируется автоматически — в соответствии с "разогревом" ЦП. Полупно авторы отмечают прекрасные возможности разгона ПК — с помощью настройки BIOS

старший вице-президент и технологический директор Intel Патрик Гелсингер впервые выдвинул идею о необходимости добавить любому устройству возможность беспроводной связи путем интеграции радиоэлементов в его микрочипы. При этом массовое производство подобных комбинированных микросхем позволит свести затраты на интеграцию дополнительных блоков практически к нулю. Концепция, предложенная Патриком Гелсингером, получила название "Бесплатное радио Intel" (Radio Free Intel).

Эта концепция предусматривает наличие радиопередатчиков, работающих в нескольких стандартах и на нескольких частотах, но на единой платформе, встраиваемой в недорогие кремниевые чипы. Суть идеи бесплатного радио заключается в том, что все беспроводные протоколы реализуются при по-

тем, подключать кабели достаточно удобно, поскольку никаких сложностей не возникает. Странным решением можно назвать лишь наличие контактов для джампера, предназначение которых нигде не указано, да и своего джампера в комплекте нет.

Качество оформления печатной платы оставило двойное впечатление. С одной стороны, достаточно удачно сделаны контакты и контактная площадка электродвигателя, сломать которые очень трудно. Но, с другой стороны, плохо то, что все элементы платы расположены с наружной стороны, и их очень легко повредить. Они расположены достаточно близко друг к другу. Пользоваться таким диском надо аккуратно.

можно вручную регулировать частоту системной и AGP/PCI-шины и напряжение питания ЦП, модулей памяти и AGP-шины.

В "периферийной" части SB65G2 практически не уступает полноформатному ПК. Однако возможности расширения этой системы значительно "урезаны", что, конечно, неизбежно при столь малых габаритах корпуса. SB65G2 можно оборудовать только тремя накопителями, двумя модулями памяти и одной PCI-платой.

В протестированном варианте компьютера были установлены ЦП Intel Pentium 4 3,0 ГГц с системной шиной 800 МГц, два модуля памяти DDR400 SDRAM по 256 Мб производства Samsung, видеоплата ASUS V9280S на наборе микросхем NVIDIA GeForce4 Ti 4200 8X с видеопамятью емкостью 128 Мб и жесткий диск Seagate Barracuda ATA V ST3120023A емкостью 120 Гб. Он отлично зарекомендовал себя в тестах PC Magazine Multimedia Content Creation Winstone 2003 1.0 (47,5 балла) и Futuremark 3DMark03 (1570 баллов). Столь высокие результаты способен показать далеко не всякий ПК аналогичной конфигурации.

В целом мини-платформа SB65G2 произвела очень приятное впечатление, и ее можно смело рекомендовать для построения одновременно скоростного, малогабаритного и малошумящего ПК. Но при этом, конечно, надо учитывать, что возможности расширения такой системы ограничены.

мощи одного набора компонентов, встраиваемых в компьютерный чип. Чтобы достичь подобной цели, аналоговые радиочастотные и логические части радиоустройств должны автоматически перестраиваться для адаптации к разным сетям и протоколам.

Аналогичные наработки есть и у лидеров телекоммуникационного рынка — компаний Texas Instruments и Motorola. Уже несколько лет трудятся в этом направлении Sandbridge, General Dynamics и Analog Devices.

Недавно компания PicoChip предложила программно-интеллектуальную платформу для базовых станций сотовых сетей. Сегодня базовая станция предназначена для работы в одном-единственном стандарте, при этом процесс смены стандарта или его обновления дорог, дорог и сложен. С программно-интеллектуальным радио, наоборот, все обстоит легко и про-



сто: не требуется менять никакое оборудование, необходимые операции можно совершить в операционном центре. По мнению экспертов, использование подобных технологий может сэкономить сотовым операторам десятки миллионов долларов на момент выхода на новые рынки.

Чтобы интегрировать возможности беспроводной связи в каждое полупроводниковое устройство, корпорация Intel предлагает использовать преимущества производства стандартных CMOS-полупроводников (Complimentary Metal-Oxide Semiconductor). Большие объемы и низкие издержки этого производства сделают "Бесплатное радио Intel" простым и удобным для конечных пользователей. Подход Intel к архитектуре программно-интеллектуального радио основан на использовании набора специальных элементов с определенными вычислительными функциями. Программируя различные связи между этими элементами, можно реализовать все необходимые протоколы радиосвязи. При этом технология MEMS (Micro-Electrical Mechanical Systems) позволяет значительно уменьшить размеры и стоимость пассивных компонентов (катушек индуктивности, конденсаторов и др.), используемых в радиопередающей части чипа.

Еще одна область, где ведутся исследования и разработки — использование так называемых умных антенн, позволяющих с помощью специальных каскадов усиливать радиосигнал, а также концентрировать его в ограниченном телесном угле, что не только улучшает качество приема, но и снижает возможность возникновения помех со стороны других передатчиков. Применение технологии умных антенн приводит к более эффективному использованию частотного спектра, а также к увеличению пропускной способности сети и ее емкости при одновременном уменьшении интерференции в переполненном частотном спектре.

Сочетание всех этих технологий позволит реализовать идею гибкого радио — устройства, способного выбирать, в зависимости от ситуации, тот радиопrotocol и ту частоту, которые ему удобнее использовать в данный момент.

С каждым днем идея "Бесплатного радио Intel" приближается к своему реальному воплощению. Исследователи из лаборатории цепей Circuit Research Lab (CRL) корпорации Intel в Хиллсборо (шт. Орегон) в сотрудничестве с лабораторией коммуникационных технологий разработали и уже изготавливают ключевые блоки для цифрового процессора автоматически перестраиваемого радио. Чтобы не дублировать контуры цепей для обеспечения работы на нескольких частотах и в нескольких стандартах, исследователи создали перепрограммируемые цепи, которые с помощью программного обеспечения позволяют реализовывать различные варианты беспроводных коммуникаций.

Однако, несмотря на колоссальный прогресс в CMOS-полупроводниковых и других технологиях, встроить радио в чипы, оказывается, совсем не просто. Дело в том, что радио — это аналоговая структура, работающая на частотах и с напряжениями, являющимися не дискретными, а изменяющимися

непрерывно в каком-либо диапазоне. Разработчики корпорации Intel столкнулись с неординарной проблемой: как превратить аналоговые величины в цифровые. К счастью, у них уже был накоплен определенный опыт решения аналогичных задач.

Так, например, создавая технологию "беспроводного Интернета на чипе", инженеры Intel преодолели большие сложности, связанные с совершенно различными путями оптимизации процессов производства для коммуникационных и для вычислительных устройств. Вместо того чтобы при создании отдельных чипов для коммуникации, вычислений и памяти идти независимыми путями, разработчики Intel использовали последние достижения в области создания флэш-памяти и интегрировали цифровые и аналоговые возможности. Правда, для этого пришлось объединить усилия и знания целого ряда экспертов из самых разных подразделений.

Не меньших успехов добились разработчики ARA (Adaptive Radio Architecture — архитектура адаптивного радио) — набора программ, который поддерживает целый ряд систем коммуникаций, как проводных, так и беспроводных. По мнению разработчиков Intel, эта программная платформа будет развиваться и вскоре сможет поддерживать и гибкое радио, способное выбирать нужный сигнал из широкого диапазона частот, и подстраиваемое радио, которое будет запрограммировано на самостоятельный поиск нужного сигнала. Эту интеллектуальную платформу с широким кругом возможностей называют программно-определенным радио (Software-Defined Radio, SDR).

Гибкая программная архитектура SDR обеспечивает пользователю возможность работы с различными протоколами передачи данных, быстрой перестройки и модернизации. Ноутбуки, сотовые телефоны и карманные компьютеры, снабженные таким программным обеспечением, смогут динамически программироваться для перестройки. Иными словами, одни и те же детали оборудования в таких устройствах смогут выполнять разные функции в различных обстоятельствах. Кроме того, многие функции будут реализовываться программно. В список таких функций входят генерация сигнала, его модуляция и обработка, использование многоуровневых протоколов.

В течение многих лет нормирующие органы по всему миру "нарезают" радиоспектр на небольшие "кусочки" и распределяют их между компаниями или же выставляют на аукционы для свободной продажи. Радиоспектр — это и телевидение, и высокочастотная связь и вещание, и радары, и коротковолновая связь, и спутниковая связь, и некоторые типы дистанционных пультов (например, те, что открывают двери в гаражах), и устройства для контроля за спящими детьми, и сотовые телефоны, и все более популярный сегодня беспроводной доступ стандарта Wi-Fi. Словом, все это те самые килоггерцы, мегагерцы и гигагерцы, которые вы не слышите, но активно используете в своей повседневной жизни. Поскольку с самого начала было ясно, что по мере своего распространения "Бесплатное радио Intel" столкнется не только с технологическими, но и с

административными сложностями, эксперты Intel активно работают в тесном контакте с регулируемыми органами по всему миру, чтобы создать условия для плодотворного развития этой технологии. Работа ведется с экспертами правительственных организаций США, стран Европы, Азиатско-Тихоокеанского региона, для того чтобы выработать единый подход к регулированию интеллектуального роуминга. Параллельно с разработкой технологических решений, корпорация Intel участвует и в обсуждении новых отраслевых стандартов, которые в будущем обеспечат эффективное внедрение программно-интеллектуального радио, поскольку одной из проблем внедрения является повсеместное заключение соглашений между различными операторами связи и Интернет-провайдерами.

В грядущем десятилетии радио станет неотъемлемой частью любого устройства. Ключевым в этом процессе является понятие адаптивности, то есть изменения поведения устройства в ответ на новое либо изменившееся поведение окружающей среды. В работе мобильных устройств можно выделить три сферы адаптивности.

Первая — адаптивность к физическим условиям. Прежде, когда связь была проводной, условия соединения по установленному каналу с течением времени практически не менялись. При организации беспроводной связи на многокилометровые расстояния приходится считаться с тем, что физические условия между приемником и передатчиком постоянно меняются, что и делает связь нестабильной. При помощи системы умных антенн и режима работы MIMO (Multiple Input Multiple Output) качество связи повышается многократно, а энергопотребление снижается в миллион раз. Суть данного метода заключается в том, что сигнал передается не с одного передатчика на один приемник, а делится между четырьмя передатчиками и четырьмя приемниками. Такое дробление позволяет избежать потери качества связи и усилить передающийся сигнал, причем в определенном направлении распространения сигнала.

Вторая сфера — адаптивность к сети. Многие владельцы мобильных телефонов попадали в ситуацию, связанную с перезагрузкой сети, поскольку множественное обращение пользователей к одной точке доступа снижает скорость передачи данных у каждого из них в отдельности. Intel предлагает иной способ организации связи: посылать информацию не напрямую к требуемой точке доступа, а от "соседа" к "соседу", резко повышая пропускную способность беспроводной сети.

Наконец, третье направление — адаптивность к пользователям. В прошлом году на осеннем Форуме Intel для разработчиков в Сан-Хосе был продемонстрирован образец нового класса мобильных устройств — "универсальный коммуникатор". При перемещении устройства по залу оно в режиме интеллектуального роуминга переключалось из стандарта 802.11 в сотовую сеть GSM, а затем переходило на технологию GPRS. При этом на экране коммуникатора сохранялось качественное видеозображение.



COM — это не сон

А.ИВАНЧИКОВ,
E-mail: iaa@rbcmail.ru

Продолжение. Начало в №4/2004)

Соглашение о вызовах

Заметим, что все методы интерфейса ICardsService описаны с ключевым словом `__stdcall`. Необходимость использования данного элемента описания заключается в следующем.

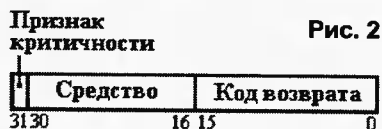
- при вызове функций по соглашениям C, параметры передаются через стек, начиная с последнего, при этом за удаление параметров со стека отвечает вызывающая процедура;

- при вызове функций по соглашениям Pascal параметры вызываемым функциям передаются через стек, начиная с первого, а за уничтожение параметров со стека отвечает вызываемая процедура.

Вызов API-функций Win32 происходит в соответствии со смешанными соглашениями, когда передача параметров выполняется в соответствии с соглашениями C, а за уничтожение параметров отвечает вызываемая процедура. Исключение составляют функции с переменным числом параметров, для которых по-прежнему используется соглашение языка C, или `__cdecl`. Функция, помеченная как `__stdcall`, использует смешанное или "стандартное" для Win32-функций соглашение о вызовах.

HRESULT

Как метод QueryInterface интерфейса IUnknown, так и подавляющее число методов всех остальных интерфейсов COM, а также и все методы интерфейса ICardsService возвращают значение HRESULT. HRESULT — это 32-разрядное значение, разделенное на три поля. Содержимое полей, составляющих HRESULT, пояснено рис.2.



Определенные системой значения HRESULT содержатся в заголовочном файле Win32 winerror.h. HRESULT похож на код ошибки Win32, но это не одно и то же. Старший бит HRESULT, как показано на рис.2, отмечает, успешно или нет выполнен метод. Это позволяет определить много кодов возврата и для успеха, и для неудачи. Последние 16 битов содержат собственно код возврата. Остальные 15 битов содержат дополнительную информацию о типе и источнике кода ошибки. В табл.1 приведены наиболее часто используемые коды. По соглашению, в названиях успешных кодов возврата содержится префикс S_, а в названиях кодов ошибок — E_.

Табл. 1

Название	Значение
S_OK	Метод отработал успешно. В некоторых случаях этот код также означает, что функция возвращает логическую "истину". Значение S_OK равно 0
S_FALSE	Метод отработал успешно и возвратил логическую "ложь"
E_UNEXPECTED	Неожиданная ошибка
E_NOTIMPL	Метод не реализован
E_NOINTERFACE	Компонент не поддерживает запрашиваемый интерфейс
E_FAIL	Ошибка по не указанной причине
E_OUTOFMEMORY	Компонент не может выделить требуемый объем памяти
E_INVALIDARG	Передаваемые в метод аргументы имеют недопустимые значения

Как правило, в зависимости от обстоятельств, функции возвращают различные коды как успешного, так и ошибочного завершения. Именно поэтому для анализа возвращаемого значения следует использовать макроопределения SUCCEEDED и FAILED, а не сравнивать их с конкретными значениями.

GUID

Строки:

```
DEFINE_GUID(IID_CardsService,
0xf345c351, 0x28fd, 0x4ee3, 0xab, 0xfa, 0x9d,
0xbf, 0x9b, 0xe9, 0xa3, 0x8a);
```

и

```
DEFINE_GUID(CLSID_CardsService,
0x100d2cc6, 0x16f0, 0x455b, 0xa5, 0xdd, 0xa0,
0xc5, 0x63, 0x76, 0x5a, 0xb4);
```

определяют, соответственно, GUID интерфейса и CLSID компонента.

GUID (Globally Unique Identifier — глобальный уникальный идентификатор) — это идентификатор интерфейса, определенный как структура длиной 128 битов. Для определения интерфейса такая "объемная" структура используется только по одной причине — чтобы гарантировать уникальность идентификатора интерфейса. Если два идентификатора совпадут, клиент может получить неверный указатель на интерфейс. Проблема усложняется тем, что компоненты создаются разработчиками по всему миру. Решение проблемы — генерировать уникальный GUID программно. Microsoft Visual Studio предоставляет для генерации GUID две программы: утилиту командной строки uuidgen.exe и оконное приложение guidgen.exe. При всяком новом запуске uuidgen.exe/guidgen.exe получается новый GUID. В Windows API для получения GUID служат функции CoCreateGuid и UuidCreate.

GUID по определению уникален "в пространстве и во времени". Для обеспечения уникальности алгоритм генерации GUID-значения использует случайные показания внутреннего таймера, параметры BIOS и уникальный номер сетевой карты (если в конфигурации компьютера сетевая карта отсутствует, то идентификатор сетевой карты эмулируется). Подробно алгоритм формирования GUID рассмотрен в [3].

При сборке компонентов и приложений-клиентов GUID должен быть многократно описан (в каждом из исходных файлов, который загружает компонент) и однажды определен. Для описания и определения GUID одним оператором (держать описание и определение в одном месте удобнее, чем в нескольких разных) используется макрос DEFINE_GUID, который определен в objbase.h. Подробно работа макроса будет рассмотрена в разделе "Описание/определение GUID". Для использования в DEFINE_GUID, внешнее представление GUID генерируется с помощью утилиты guidgen.exe.

Это приложение генерирует GUID в четырех различных форматах — следует выбрать второй из них.

Помимо уникальной идентификации интерфейсов, GUID используется и для уникальной идентификации компонентов. Такой GUID в COM называется идентификатором класса. Чтобы отличать идентификаторы классов от идентификаторов интерфейсов, для них используется тип CLSID.

```
// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~//  
// CardsService.h  
// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~// ~~~//  
// -----
```

```
//-----  
class CCardsService : public ICardsService  
{
```

```
virtual HRESULT __stdcall AnimateCards(  
    // описатель контекста граф. устройства  
    HDC hDC,  
    int nBackT,    // тип рубашки карты  
    // координата X левой стороны карты  
    int nXCoord,  
    // координата Y верхней стороны карты  
    int nYCoord,  
    int nState); // режим отображения
```

```
virtual HRESULT __stdcall ShuffleCards(
    int* pnCards, // массив значений карт
    // размерность массива
    int nCountCards);
```

```
virtual HRESULT __stdcall GetSuitRangCards(
    int nCardsNumber, // номер карты
    int* pnCardsRang, // масть карты
    int* pnCardsSuit); // ранг карты
```

```
private:
```

Класс компонента, кроме пяти ожидаемых указателей на экспортируемые из cards.dll функции, содержит дополнительно четыре члена-данных: m_hCardDll — описатель cards.dll, обеспечивающий загрузку и удаление системной библиотеки; m_lReference — счетчик ссылок на интерфейс; m_nWidth и m_nHeight — текущие значения размера карт, устанавливаемые при инициализации библиотеки.

```
// ~~~~~ CardsService.cpp ~~~~~  
// ~~~~~  
// CardsService.cpp  
// ~~~~~  
// ~~~~~  
// -----  
#include <time.h>  
#include <stdlib.h>  
#include "CardsService.h"  
#include "CardsFlags.h"
```

```

//-----
CCardsService::CCardsService() : m_lReference(1)
{
    ::InterlockedIncrement(&:g_lComponents);
    Init();
}

```

```

//-----
CCardsService::~CCardsService()
{
    if ( m_pfcddtTerm )
        m_pfcddtTerm();
    if ( m_hCardDll )
        ::FreeLibrary(m_hCardDll);
    ::InterlockedDecrement(&::g_lComponents);
}

```

```
//-----
HRESULT __stdcall CCardsService::QueryInterface(
    REFIID iid,
    void** ppvObject)
```

```
*ppvObject = NULL;
```

```
// запрос интерфейса IUnknown
if ( ::IsEqualIID(iid, IID_IUnknown) )
    *ppvObject =
        static cast<ICardsService*>(this);
```



```
// запрос интерфейса ICardsService
else if (::IsEqualIID(iid, IID_CardsService))
    *ppvObject =
        static_cast<ICardsService*>(this);

// запрашиваемый интерфейс не поддерживается
else
    return E_NOINTERFACE;

// увеличение счетчика ссылок на единицу
static_cast<IUnknown*>(*ppvObject)->AddRef();
return S_OK;
}

//-----
ULONG __stdcall CCardsService::AddRef()
{
    return ::InterlockedIncrement(&m_lReference);
}

//-----
ULONG __stdcall CCardsService::Release()
{
    if ( (::InterlockedDecrement(&m_lReference) )
        {
            delete this;
            return 0;
        }
    return m_lReference;
}

// реализация ICardsService
//-----
HRESULT __stdcall CCardsService::GetNormalCardsSize(
    int* pnXSize,
    int* pnYSize)
{
    if ( !(pnXSize && pnYSize) )
        return E_INVALIDARG;

    *pnXSize = m_nWidth;
    *pnYSize = m_nHeight;

    return S_OK;
}

//-----
HRESULT __stdcall CCardsService::ShowCards(
    HDC hDC,
    int nXCoord,
    int nYCoord,
    int nCardV,
    int nDrawT,
    int nColor,
    int nXSize /*= 0*/,
    int nYSize /*= 0*/)
{
    if ( !(nXSize && nYSize) )
    {
        if ( !m_pfcDraw )
            return S_FALSE;

        m_pfcDraw(hDC, nXCoord, nYCoord,
            nCardV, nDrawT, nColor);
    }
    else
    {
        if ( !m_pfcDrawEx )
            return S_FALSE;

        m_pfcDrawEx(hDC, nXCoord, nYCoord,
            nXSize, nYSize, nCardV, nDrawT, nColor);
    }

    return S_OK;
}

//-----
HRESULT __stdcall CCardsService::AnimateCards(
    HDC hDC,
    int nBackT,
    int nXCoord,
```

```
int nYCoord,
int nState)
{
    if ( !m_pfcAnim )
        return S_FALSE;

    m_pfcAnim(hDC, nBackT, nXCoord,
        nYCoord, nState);

    return S_OK;
}

//-----
HRESULT __stdcall CCardsService::ShuffleCards(
    int* pnCards,
    int nCountCards)
{
    if ( !(pnCards && nCountCards) )
        return E_INVALIDARG;

    int nDeckSize = abs(nCountCards) %
        (MAX_CARD_SIZE + 1);
    int nCardsIterator = nDeckSize;

    while ( -nCardsIterator >= 0 )
        pnCards[nCardsIterator] = nCardsIterator;

    nCardsIterator = nDeckSize;

    // srand((unsigned)time(NULL));
    while ( -nCardsIterator >= 0 )
    {
        int nCard1 = rand() % nDeckSize;
        int nCard2 = pnCards[nCard1];
        pnCards[nCard1] = pnCards[nCardsIterator];
        pnCards[nCardsIterator] = nCard2;
    }

    return S_OK;
}

//-----
HRESULT __stdcall CCardsService::GetNumCards(
    int nCardsSuit,
    int nCardsRang,
    int* pnCardsNumber)
{
    if ( !pnCardsNumber )
        return E_INVALIDARG;

    nCardsSuit %= (S_SPADES + 1);
    nCardsRang %= (R_KING + 1);

    *pnCardsNumber = nCardsRang * (S_SPADES + 1) +
        nCardsSuit;

    return S_OK;
}

//-----
HRESULT __stdcall CCardsService::GetSuitRangCards(
    int nCardsNumber,
    int* pnCardsRang,
    int* pnCardsSuit)
{
    if ( !(pnCardsRang && pnCardsSuit) )
        return E_INVALIDARG;

    nCardsNumber %= (MAX_CARD_SIZE + 1);
    *pnCardsRang = nCardsNumber / (S_SPADES + 1);
    *pnCardsSuit = nCardsNumber % (S_SPADES + 1);

    return S_OK;
}

//-----
void CCardsService::Init()
{
    m_pfcInit = NULL;
    m_pfcDraw = NULL;
    m_pfcDrawEx = NULL;
    m_pfcAnim = NULL;
```

А.КОРБИТ, А.ЩЕРБАКОВ,
г.Минск, БГУИР.

Программирование в среде Visual C++ 6.0

Элемент управления Tree Control

Цель лабораторной работы — изучить возможности элемента управления Tree Control и научиться составлять деревья произвольной структуры.

1. Принципы работы с элементом Tree Control

Элемент **CTreeCtrl** предназначен для работы с данными, имеющими древовидную структуру. Каждый элемент такой структуры имеет текст и картинку, а также может иметь подэлементы.

Для добавления нового элемента в дерево в классе **CTreeCtrl** существует функция-член:

HTREEITEM InsertItem(LPTVINSERTSTRUCT lpInsertStruct);

Здесь **lpInsertStruct** — указатель на структуру **TV_INSERTSTRUCT**, которая определяет атрибуты каждого добавляемого элемента. Члены этой структуры следующие:

- **hParent** — дескриптор родительского элемента. Если этот параметр установлен константе **TVI_ROOT** или **NULL**, то элемент добавляется в основание (корень) дерева;
- **Item** — структура, которая определяет внешний вид элементов.

Основные члены структуры **Item**:

- **mask** — флаг, определяющий, что будет включать каждый элемент (текст — флаг **TVIF_TEXT**, картинку — флаги **TVIF_IMAGE** и **TVIF_SELECTEDIMAGE**, числовой параметр — флаг **TVIF_PARAM**);
- **pszText** — указатель на строку, содержащую название элемента;
- **IParam** — целочисленный параметр, используемый для идентификации элемента;
- **ilImage** — номер картинки в списке **CImageList**;
- **iSelectedImage** — номер картинки, отображаемой при выделении элемента.

Для подключения **CImageList** к **CTreeCtrl** используется функция

**CImageList* SetImageList(CImageList * pImageList,
int nImageListType);**

где **pImageList** — указатель на объект класса **CImageList**.

2. Классы CImageList и CBitmap

Класс **CImageList** — это список изображений, он представляет собой набор изображений одинакового размера, к которым можно обращаться по индексу. Для создания класса **CImageList** служит функция **Create**, первые два параметра которой указывают ширину и высоту изображений в пикселах.

Для добавления картинки в список служит функция **Add**, в первый параметр которой передают указатель на класс **CBitmap**.

Объект класса **CBitmap** содержит функции для работы с битовым образом интерфейса графического устройства Windows. Функция **LoadBitmap** позволяет загрузить картинку, идентификатор которой передается в качестве первого параметра.

3. Пример написания программы

Необходимо написать программу, отображающую дерево, изображенное на рис.1. При выделении элемента дерева должно появляться его описание.

```
m_pfcdtTerm = NULL;  
m_nWidth = 0;  
m_nHeight = 0;  
  
char* szCardsDllName;  
if ( ::GetVersion() >= 0x80000000 )  
{  
    // Windows 95/98/Me, Win32s  
    szCardsDllName = CARDS32_DLL_NAME;  
}  
else  
{  
    // Windows NT/2000  
    szCardsDllName = CARDS_DLL_NAME;  
}  
  
m_hCardDll = ::LoadLibrary(szCardsDllName);  
if ( !m_hCardDll )  
    return;  
  
m_pfcdtInit =  
    (LPCDTINIT)::GetProcAddress(m_hCardDll,  
        CDTINIT_NAME);  
m_pfcdtDraw =  
    (LPCDTPDRAW)::GetProcAddress(m_hCardDll,  
        CDTDRAW_NAME);  
m_pfcdtDrawEx =  
    (LPCDTPDRAWEX)::GetProcAddress(m_hCardDll,  
        CDTDRAWEX_NAME);  
m_pfcdtAnimate =  
    (LPCDTANIMATE)::GetProcAddress(m_hCardDll,  
        CDTANIMATE_NAME);  
m_pfcdtTerm =  
    (LPCDPTERM)::GetProcAddress(m_hCardDll,  
        CDTTERM_NAME);  
  
if ( m_pfcdtInit )  
    m_pfcdtInit(&m_nWidth, &m_nHeight);  
}
```

Конструктор компонента инициализирует счетчик ссылок единичным значением и выполняет, посредством вызова метода **Init**, загрузку динамической библиотеки **cards.dll** (**cards32.dll** в случае Windows 95/98/Me, Win32s). Метод явной компоновки DLL, который использован для связи компонента с внешней DLL, подробно обсуждается в [2].

Из пяти экспортируемых из **cards.dll** функций обертку в качестве метода интерфейса имеют только две. Для функции **cdtInit** делать это не имеет смысла, так как предопределенные значения размеров карт сохраняются в конструкторе компонента и могут быть возвращены в любой момент методом **GetNormalCardsSize()**. Функцию **cdtTerm** можно вызвать в деструкторе компонента при завершении работы с ним. Методы-оболочки организованы однотипно: проверяется член-данное указатель на соответствующую функцию, и если он не равен нулю, т.е. предварительный экспорт функции был выполнен успешно, производится вызов функции.

Для функций **m_pfcdtDraw** и **m_pfcdtDrawEx** выполнен только один метод-оболочка **ShowCards**. Если один из двух последних параметров, объявленных по умолчанию, равен нулю, то производится вызов функции **pfcdtDraw**.

Дополнительный метод **ShuffleCards()** реализует простейший алгоритм перетасовки колоды карт — карты в колоде (колода эмулируется массивом целых) имеют значения от 0 до **MAX_CARD_SIZE-1**.

Методы **GetNumCards()** и **GetSuitRangCards()** предназначены соответственно для генерации значения карты по ее масти и рангу и для получения масти карты и ее ранга по значению от 0 до **MAX_CARD_SIZE-1**. Значение карты передается как параметр **nCardV** в метод **ShowCards()**.

(Продолжение следует)



Рис. 1 Для решения задачи проделайте следующие действия.

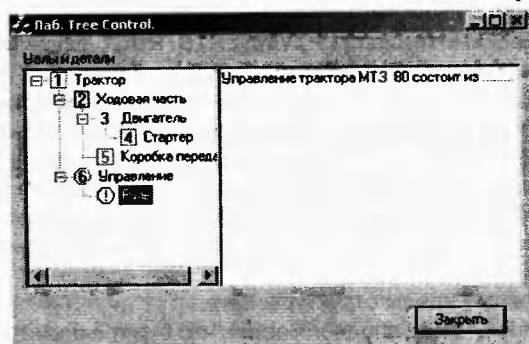
1. Создайте при помощи AppWizard новый проект, выбрав тип приложения Dialog Based.

2. Поместите на форму компоненты Tree Control, EditBox, StaticText в соответствии с рис.2.

3. В свойствах Tree Control в закладке Styles установите свойства Has buttons, Has lines, Lines at root.

4. С помощью ClassWizard, вкладки MemberVariables свяжите Tree Control с переменной m_tree, а EditBox, с переменной m_edit.

Рис. 2



5. Создайте для каждого элемента дерева картинку. Для этого в меню Insert выберите Resource... и в открывшемся диалоге — элемент Bitmap. Нажмите кнопку New. Двигая мышкой нижний правый конец изображения, установите его размер 16 на 16. Нарисуйте изображение. Наименование изображений должны быть: IDB_BITMAP1, IDB_BITMAP2, IDB_BITMAP3 и т.д.

В рассматриваемом примере использованы следующие изображения:

1 2 3 4 5 6 7.

Изображение 1 показывает текущий выделенный элемент.

6. В окне Workspase в классе окна диалога выделите функцию OnInitDialog. После комментария

// TODO: Add extra initialization here добавьте нижеприведенный код:

```
// TODO: Add extra initialization here
CImageList *pImageList;
CBitmap bitmap;
pImageList = new CImageList();
//загружаем картинки в CImageList
pImageList->Create(16,16, ILC_MASK,8,8);
for (int nID = IDB_BITMAP1; nID <= IDB_BITMAP8; nID++)
{
    bitmap.LoadBitmap(nID); //загрузка изображения из ресурса
    //добавление картинки в список
    pImageList->Add(&bitmap, (COLORREF)0xFFFFFFFF);
    //освободить объект bitmap для новой картинки
    bitmap.DeleteObject();
}
//связать список изображений с объектом класса CTreeCtrl
m_tree.SetImageList(pImageList,TVSIL_NORMAL);
TV_INSERTSTRUCT TreeCtrlItem;
//устанавливаем параметры для 1-го элемента Tree Control
TreeCtrlItem.hParent = TVI_ROOT;
TreeCtrlItem.item.mask=TVIF_TEXT | TVIF_PARAM | TVIF_IMAGE |
```

```
TVIF_SELECTEDIMAGE;
TreeCtrlItem.item.pszText = "Трактор"; // название элемента
TreeCtrlItem.item.lParam = 0;
//параметр для идентификации элемента в обработчике
//OnSelchangedTree1
TreeCtrlItem.item.iImage=0; //номер картинки в CImageList
TreeCtrlItem.item.iSelectedImage=7;
//номер картинки, если элемент выделен
HTREEITEM hTreeItem1 = m_tree.InsertItem(&TreeCtrlItem);
//добавляем элемент и запоминаем его дискриптор в hTreeItem1
```

```
//Для элементов "Ходовая часть" и "Управление"
//устанавливаем корневой элемент "Трактор";
TreeCtrlItem.hParent = hTreeItem1;
```

```
//устанавливаем параметры для 2-го элемента Tree Control
TreeCtrlItem.item.pszText = "Ходовая часть";
TreeCtrlItem.item.lParam = 1;
TreeCtrlItem.item.iImage=1;
HTREEITEM hTreeItem2 = m_tree.InsertItem(&TreeCtrlItem);
```

```
//устанавливаем параметры для 3-го элемента Tree Control
TreeCtrlItem.item.pszText = "Управление";
TreeCtrlItem.item.lParam = 2;
TreeCtrlItem.item.iImage=5;
HTREEITEM hTreeItem3 = m_tree.InsertItem(&TreeCtrlItem);
```

```
//устанавливаем параметры для 4-го элемента Tree Control
TreeCtrlItem.item.pszText = "Двигатель";
TreeCtrlItem.item.lParam = 1;
TreeCtrlItem.item.iImage=2;
TreeCtrlItem.hParent = hTreeItem2;
HTREEITEM hTreeItem4 = m_tree.InsertItem(&TreeCtrlItem);
```

```
//устанавливаем параметры для 5-го элемента Tree Control
TreeCtrlItem.item.pszText = "Коробка передач";
TreeCtrlItem.item.lParam = 1;
TreeCtrlItem.item.iImage=4;
m_tree.InsertItem(&TreeCtrlItem);
```

```
//устанавливаем параметры для 6-го элемента Tree Control
TreeCtrlItem.item.pszText = "Стартер";
TreeCtrlItem.item.lParam = 5;
TreeCtrlItem.item.iImage=3;
TreeCtrlItem.hParent = hTreeItem4;
m_tree.InsertItem(&TreeCtrlItem);
```

```
//устанавливаем параметры для 7-го элемента Tree Control
TreeCtrlItem.item.pszText = "Руль";
TreeCtrlItem.item.lParam = 2;
TreeCtrlItem.item.iImage=6;
TreeCtrlItem.hParent = hTreeItem3;
m_tree.InsertItem(&TreeCtrlItem);
```

```
return TRUE; // return TRUE unless you set the focus to a control
}
```

7. Для отображения описания выделенного элемента в окне EditBox, с помощью ClassWizard создайте обработчик события TVN_SELCHANGED для Tree Control. В обработчик добавьте код:

```
void Clab12Dlg::OnSelchangedTree1(NMHDR* pNMHDR, LRESULT* pResult)
{
    NM_TREEVIEW* pNMTreeView = (NM_TREEVIEW*)pNMHDR;
    // TODO: Add your control notification handler code here
```

```
int i=pNMTreeView->itemNew.lParam;
switch (i)
{
    case 0: m_edit="Трактор МТЗ 80"; break;
    case 1: m_edit="Ходовая часть состоит из двигателя и коробки передач."; break;
    case 2: m_edit="Управление трактора МТЗ 80 состоит из .."; break;
    case 5: m_edit="Стартер предназначен для ... "; break;
}
```

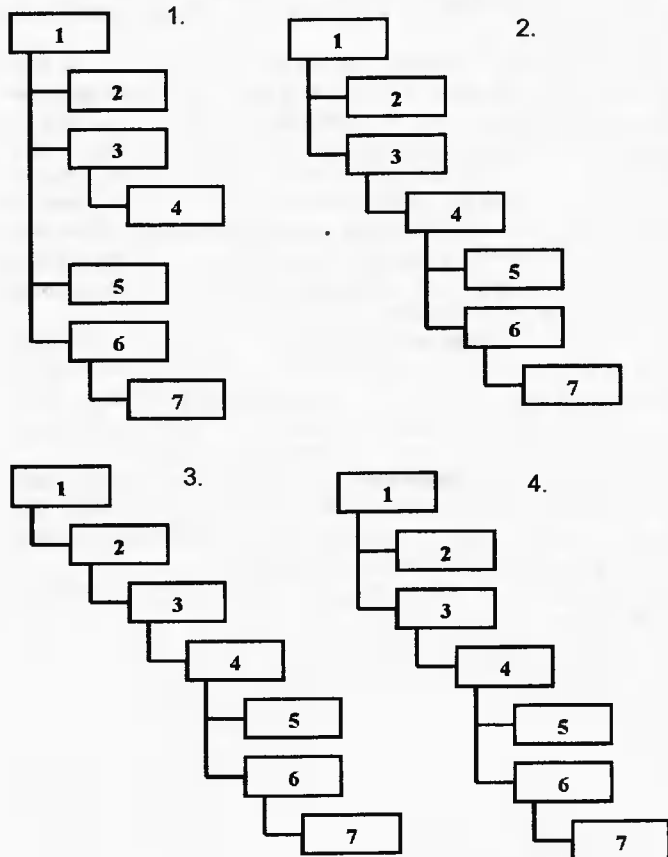


```
UpdateData(FALSE);
*pResult = 0;
}
```

4. Индивидуальные задания

Необходимо написать программу, отображающую дерево. При выделении элементов 2, 4, 5 и 6 в Edit Box должно выводиться одинаковое сообщение.

Варианты заданий:



Литература

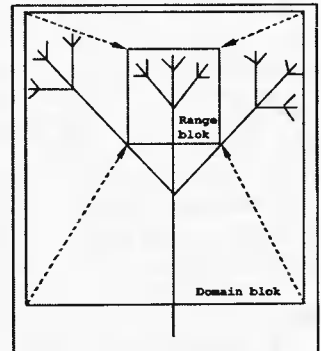
1. С.Холзнер. Visual C++ 6: учебный курс. — СПб.: Питер, 1999. — 576 с.
2. Н.Ю.Секунов. Самоучитель Visual C++ 6 (+ дискета). — СПб.: BHV-Санкт-Петербург, 1999. — 960 с.
3. К.Грегори. Использование Visual C++ 6. Спец.издание. — СПб.: Вильямс, 1999. — 864 с.
4. Ю.В.Тихомиров. Visual C++ 6 (+ дискета). — СПб.: BHV-Санкт-Петербург, 1999. — 496 с.
5. Майкл Дж. Янг. Visual C++ 6. Полное руководство (+ CD-ROM). — BHV-Киев, 2000. — 1056 с.
6. С.Гилберт, Б.Маккарти. Самоучитель Visual C++ 6 (+ CD-ROM). — М.: ДиаСофт, 2000. — 496 с.
7. К.Паппас, У.Мюррей. Visual C++ 6 для пользователя. — BHV-Киев, 2000. — 336 с.
8. К.Паппас, У.Мюррей. Visual C++ 6: руководство разработчика. — BHV-Киев, 2000. — 624 с.
9. И.Баженова. Visual C++ 6. 0. VISUAL STUDIO 98: Уроки программирования. — М.: Диалог-МИФИ, 2001. — 416 с.
10. Лэйси Дж. Visual C++ 6 Distributed. Сертификационный экзамен — экстерном. — СПб.: Питер, 2001. — 624 с.
11. А.Черносвитов. Visual C++ 7: учебный курс (с дискетой). — СПб.: Питер, 2001. — 528 с.

А. КАЛЮТОВ,
E-mail: k90012@hotmail.com

Фрактальный анализ и увеличение изображений

Фрактальный анализ — это совокупность математических методов для "расшифровки" фрактальной структуры анализируемого объекта или сигнала. Иными словами, фрактальный анализ позволяет обнаружить самоподобие объекта — повторяемость каких-нибудь свойств на различных масштабах. Например, в случае дерева таким свойством будет среднее число разветвлений любой ветки, то есть число тонких веток, отходящих от более массивной (см. рисунок).

На любом реальном дереве можно убедиться, что это число находится в достаточно узких пределах для каждого конкретного случая.



Термин "фрактальный анализ" ввел М.Бернслей (Michael Barnsley). Четкая формулировка основных понятий и теорем способствовала осознанию всей глубины и общности идеи о фрактальности окружающего нас мира. После разработки конкретных алгоритмов и методов пришла очередь практических приложений, причем многие из них были настолько многообещающими, что соответствующие идеи сразу были запатентованы. В частности, это относится к новым методам компрессии данных и масштабирования изображений.

Одним из многочисленных приложений фрактального анализа является обработка визуальной информации — изображений, трехмерных образов и т.п. Фрактальную структуру изображения можно описать с помощью небольшого числа параметров, поэтому наиболее привлекательной выглядела возможность использования нового раздела математики прежде всего для компрессии изображений и другой информации. В настоящей статье рассматривается другое применение фрактального анализа — один из алгоритмов увеличения изображения без потери качества.

Прежде всего следует выявить фрактальную структуру анализируемого изображения. Для этого были разработаны несколько способов. Наиболее простым является так называемый *метод разбиения на блоки*, предложенный М.Бернслеем, и впервые реализованный в программах А.Жаквиным (Arnaud Jacquin). Суть этого метода заключается в поиске *подобных* прямоугольных участков изображения, которые после масштабирования, всевозможных поворотов, зеркальных отображений и изменения средней яркости можно с некоторой небольшой ошибкой отождествить друг с другом. Меньшие по размеру участки изображения получили название *ранговых блоков* (Range block), а большие — *доменных* (Domain block) (см. рисунок). Ранговые блоки должны полностью покрывать все изображение. Таким образом, задача заключается в том, чтобы для каждого рангового блока найти преобразование доменного блока в ранговый с минимальными искажениями. Совокупность указанных преобразований для всех ранговых блоков называется *системой итерационных функций* (Iterated Function System, IFS). Для того чтобы восстановить исходное изображение, необходимо применить найденные преобразования к *любому* исходному изображению (например, серому фону).

Метод разбиения на блоки с последовательным сравнением всевозможных пар блоков обладает рядом недостатков, прежде всего — очень большой вычислительной сложностью. Обычно он используется только в тех случаях, когда требуется с высо-



кой точностью найти IFS для данного изображения.

После того как IFS найдена, исходное изображение можно очень легко увеличить, например, в два раза, следующим способом — каждый ранговый блок заменяется соответствующим доменным блоком, который масштабируется к удвоенному размеру рангового блока. В самом деле, каждое ранее найденное преобразование доменного блока в ранговый есть не что иное как наилучшее соответствие между двумя участками разного размера. Для того чтобы увеличить исходное изображение, достаточно заменить меньший блок большим.

Разумеется, чем более точно найдена IFS, т.е. найдены как можно более точные соответствия ранговый блок — доменный блок, тем более точным будет увеличение. Как правило, почти любое изображение можно увеличить на 200...400% без какого-либо заметного ухудшения качества. Описанный выше алгоритм нахождения IFS и увеличения будут малоэффективными только в двух случаях:

а) очень маленького по размерам изображения — из-за небольшого числа различных ранговых и доменных блоков и, как следствие, отсутствия "хороших" преобразований;

б) очень больших изображений — из-за астрономически большого числа пар блоков, которые необходимо проанализировать.

При использовании более совершенных алгоритмов первое затруднение обходится с помощью *словаря блоков*, куда помещаются ранговые блоки, для которых не удалось найти подходящие преобразования. Вторая проблема — эффективное нахождение *точной* IFS для произвольного изображения — в принципе, не решена по сей день, хотя существуют алгоритмы получения IFS, которые требуют значительно меньшего объема вычислений, чем традиционный метод перебора всех возможных сочетаний двух блоков [1, 2]. Один из способов заключается в уменьшении числа сравнений. Для этого все ранговые и доменные блоки разбиваются на отдельные классы. Сравниваются только те блоки, которые принадлежат одному и тому же классу. Каждый класс определяется таким образом, чтобы блоки из разных классов были *несводимы* друг к другу при всевозможных линейных преобразованиях, иными словами — сильно отличались бы по своей структуре. Чтобы идея была понятна, можно привести следующий пример. Допустим, в одной куче находится множество предметов различных цветов и формы, и необходимо найти два одинаковых предмета. Можно перебирать все возможные пары, а можно сначала разделить все предметы по какому-нибудь признаку на группы (например, по цвету), и затем искать нужную пару предметов в каждой группе.

В случае блоков классы могут определяться следующим образом — каждый блок разбивается на 4 квадранта, и вычисляется средняя яркость точек A_1, A_2, A_3, A_4 квадранта:

A_1	A_2
A_3	A_4

Класс 1: $A_1 \geq A_2 \geq A_3 \geq A_4$.

Класс 2: $A_1 \geq A_2 \geq A_4 \geq A_3$.

Класс 3: $A_1 \geq A_4 \geq A_2 \geq A_3$.

Подробно этот метод описан, например, в [2]. Там же приведены другие способы разбиения на классы.

Для иллюстрации простейшего алгоритма увеличения изображения далее приведен пример C-программы для удвоения размера любой полноцветной картинке. Программа написана под Windows, запускается из командной строки; в качестве параметра следует указать имя файла изображения в формате .bmp (truecolour).

Для увеличения быстродействия программа сильно оптимизирована и поэтому требует некоторых пояснений.

В основном, оптимизируются часто повторяющиеся операции. В частности, в ходе вычислений требуется сравнивать средние яркости ранговых и доменных блоков, поэтому они вычисляются заранее для всех ранговых блоков (массив Ravr). При сравнении ранговых и доменных блоков последний мас-

штабируется к размеру рангового блока. После этого производится изменение яркости каждой точки доменного блока таким образом, чтобы его средняя яркость была равна средней яркости рангового блока. Коэффициенты пропорциональности сохраняются в массиве c[] структуры (см. листинг программы). Сравнение блоков осуществляется по формуле

$$dist = \sum (r1 - r2)^2 + (g1 - g2)^2 + (b1 - b2)^2,$$

где $(r1, g1, b1)$ и $(r2, g2, b2)$ — цветовые компоненты точки соответственно 1-го и 2-го блоков. Суммирование ведется по всем точкам блока.

Все блоки имеют форму квадрата. Размер всех ранговых блоков выбран одинаковым. Поскольку ранговые блоки масштабировать не надо, цикл по перебору блоков данного типа — внутренний. Размер ранговых блоков — 2x2, доменных — 4x4, 8x8, 16x16, ... Информация о каждом преобразовании сохраняется в массиве ifs. Для ускорения работы в программе не выполняются дополнительные преобразования ранговых блоков типа поворотов и зеркальных отображений относительно осей симметрии квадрата. Дополнив текст соответствующими процедурами, можно найти более точную IFS.

Исходное изображение может иметь любое разрешение, причем ширина и высота изображения должны нацело делиться на удвоенный размер рангового блока (в нашем случае — на 4). Следует только иметь в виду, что увеличение слишком большого изображения может занять очень много времени.

Что касается времени работы, преобразование изображения 256x160x24 bit в 512x320x24 bit на Athlon 750 заняло 50 с (использовался компилятор для процессоров AMD, настройка на максимальное быстродействие).

Программа написана и отлажена в среде MS Visual C++ 6.0.

Исходный текст:

```
#include <stdafx.h>
#include <iostream.h>
#include <windows.h>
#include <math.h>

typedef struct{
    int size;
    int x,y;
    int dist;
    float c[3];
} rblockinfo;

int width,height,size;
//функция для сравнения 2-х блоков
inline int distance(int rsize, BYTE* a, BYTE* b){
    int df=0,rs2=3*rsize*rsize;

    for(int r=0;r<rs2;r++) df+=abs(a[r]-b[r])*abs(a[r]-b[r]);
    return(df);
}

// масштабирование блока: из size1 x size1 в size2 x size2
inline void resize(int size1,int size2,BYTE** block1,
                  int offset,int* block2){
    int t,s=size1/size2,s2=size2*size2;
    for (t=0;t<3*s2;t++) block2[t]=0;
    for(t=0;t<3;t++)
        for(int u=0;u<size1;u++)
            for(int v=0;v<size1;v++)
                block2[t*s2+size2*(u/s)+v/s]=block1[t][offset+width*u+v];
    s=s;
    for (t=0;t<3*s2;t++) block2[t]/=s;
}

int main(int argc, char* argv[])
{
    HANDLE hSrc, hDest;
    DWORD dwRead, dwWritten;
    BYTE R,G,B;
    int dist,Davr[3],nrbl,nwr,nhr,dres,s;
    int *dd,*dd2,*Ravr[3];
    char* name=new char[128];
    rblockinfo *ifs;
    float col[3];
    BYTE *pBuffer,*pic[3],*pic2[3],*range, *d;
```



```

int i,j,k,t,l,m,x,d,yd;
LPBITMAPINFO lpbi=
    (LPBITMAPINFO) new BYTE[sizeof(BITMAPINFOHEADER)];

int* pH=(int *)malloc(14);

if (argc != 2)
{
    cout<<"Usage: " << argv[0] << "sourcefile"; exit(1);
}

hSrc = CreateFile(argv[1], GENERIC_READ, 0, NULL,
    OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);

if (hSrc == INVALID_HANDLE_VALUE)
{
    cout<< "Unable to open " <<argv[1]<< endl; exit(1);
}

strcpy(name,argv[1]);
strcpy(&name[strlen(name)-4],"_copy.bmp");

ReadFile(hSrc,pH, 14, &dwRead, NULL);
ReadFile(hSrc,lpbi, 40, &dwRead, NULL);
// установка параметров изображения
width=lpbi->bmiHeader.biWidth;
height=lpbi->bmiHeader.biHeight;
nwr=width/2;nhr=height/2;nrb1=nwr*nhr;size=width*height;
for(t=0;t<3;t++){
    pic[t]=(BYTE *)malloc(size);
    pic2[t]=(BYTE *)malloc(4*size);
    Ravr[t]=(int *)malloc(nrb1*sizeof(int));
}
pBuffer=(BYTE *)malloc(3*size);

ifs=(rblockinfo *)malloc(nrb1*sizeof(rblockinfo));
range=(BYTE *)malloc(12*nrb1);
dd=(int *)malloc(12*sizeof(int));
dd2=(int *)malloc(4*12*sizeof(int));
d=(BYTE *)malloc(12);
if (!pic[0] || !pic[1] || !pic[2] || !pBuffer || !range
    || !dd || !d || !ifs || !dd2 || !pic2[0] || !pic2[1]
    || !pic2[2] || !Ravr[0] || !Ravr[1] || !Ravr[2]){
    cout<<"Not enough memory";goto ex;
}

ReadFile(hSrc, pBuffer, 3*size, &dwRead, NULL);

CloseHandle(hSrc);

for(i=0;i<size;i++){
    B=pBuffer[3*i+2];G=pBuffer[3*i+1];R=pBuffer[3*i];
    pic[2][i]=B; pic[1][i]=G; pic[0][i]=R;
}

for(i=0;i<nrb1;i++) ifs[i].dist=1000000;
// максимальный размер доменного блока -
dres=exp(int(log(min(height,width))/log(2))*log(2))+1;
// вычисление средней яркости каждого рангового блока
m=0;
for(l=0;l<height;l+=2)
for(k=0;k<width;k+=2){
    for(t=0;t<3;t++){
        Ravr[t][m]=(pic[t][width*1+k]+pic[t][width*(1+l)+k]+
            pic[t][width*1+k+1]+pic[t][width*(1+l)+k+1]);
        memcpy(range+t*4+12*m,pic[t]+width*1+k,2);
        memcpy(range+t*4+12*m+2,pic[t]+width*(1+l)+k,2);
    }
    m++;
}
// перебор всех возможных пар блоков
while(dres>4){
    for(i=0;i<height-dres;i+=dres)
        for(j=0;j<width-dres;j+=dres){//i,j - доменный блок
            resize(dres,2,pic,width*i+j,dd);
            for(t=0;t<3;t++){
                Davr[t]=dd[4*t]+dd[4*t+1]+dd[4*t+2]+dd[4*t+3];
                Davr[t]=(Davr[t]==0)?1:Davr[t];
            }
            m=0;
            for(l=0;l<nrb1;l++){
                for(t=0;t<3;t++){
                    col[t]=(float)Ravr[t][m]/Davr[t];
                    for (t=0;t<4;t++){

```

```

d[t]=(BYTE)dd[t]*col[0];
d[4+t]=(BYTE)dd[4+t]*col[1];
d[8+t]=(BYTE)dd[8+t]*col[2];
}
dist=distance(2,d,range+12*m);
if (ifs[m].dist>dist){
    ifs[m].dist=dist;
    ifs[m].x=j;
    ifs[m].y=i;
    ifs[m].size=dres;
    memcpy(&ifs[m].c[0],&col,3*sizeof(float));
}
m++;
}
dres/=2;
cout<<"Processing domain blocks "<<dres<<"x"<<dres<<endl;
}
//увеличение
size=4*height*width;
pBuffer=(BYTE *)malloc(3*size);

cout<<"Creating double-size image..."<<endl;
m=0;// обработка каждого рангового блока
for(i=0;i<2*height;i+=4)
    for(j=0;j<2*width;j+=4){
        yd=ifs[m].y;
        xd=ifs[m].x;
        dres=ifs[m].size;
        resize(dres,4,pic,width*yd+xd,dd2);//масштабирование
        for(t=0;t<3;t++){
            for(l=0;l<4;l++){
                for(k=0;k<4;k++){
                    s=dd2[t*16+1*4+k]*ifs[m].c[t];
                    pic2[t][(i+1)*2*width+(j+k)]=(s>255)?255:s;
                }
            }
            m++;
        }
        // увеличенное изображение помещается в буфер
        for(i=0;i<size;i++){
            R=pic2[0][i];G=pic2[1][i];B=pic2[2][i];
            pBuffer[3*i+2]=B;pBuffer[3*i+1]=G;pBuffer[3*i]=R;
        }
    }
hDest = CreateFile(name, GENERIC_WRITE, 0, NULL,
    CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, 0);

if (hDest == INVALID_HANDLE_VALUE)
{
    cout<< "Unable to create " <<name<< endl;
    exit(1);
}

// запись сигнатуры файла .BMP
WriteFile(hDest,"BM", 2, &dwWritten, NULL);
// запись заголовка и информации об изображении
pH[0]=3*size+54;pH[1]=0;pH[2]=54;
lpbi->bmiHeader.biSize = sizeof(BITMAPINFOHEADER);
lpbi->bmiHeader.biWidth = 2*width;
lpbi->bmiHeader.biHeight = 2*height;
lpbi->bmiHeader.biPlanes = 1;
lpbi->bmiHeader.biBitCount = 24;
lpbi->bmiHeader.biCompression = BI_RGB;
lpbi->bmiHeader.biSizeImage = 0;
lpbi->bmiHeader.biXPelsPerMeter = 0;
lpbi->bmiHeader.biYPelsPerMeter = 0;
lpbi->bmiHeader.biClrUsed = 0;
lpbi->bmiHeader.biClrImportant = 0;
WriteFile(hDest,pH, 12, &dwWritten, NULL);
WriteFile(hDest,lpbi, 40, &dwWritten, NULL);
WriteFile(hDest, pBuffer, 3*size, &dwWritten, NULL);
ex:
CloseHandle(hDest);
delete lpbi; delete name;
return 0;
}

```

Литература

1. J.Kominek. Algorithm for fast fractal image compression. In Proceedings from IS&T/SPIE 1995 Symposium on Electronic Imaging: Science & Technology, Vol. 2419 Digital Video Compression: Algorithms and Technologies, 1995. (электронная версия — на <ftp://axes.informatik.uni-leipzig.de/pub/Fractal/papers/Komi95a.ps.gz>).
2. Y.Fisher. Fractal Encoding — Theory and Applications to Digital Images, Springer-Verlag, NY, 1994.



ИССЛЕДОВАНИЕ ПРОГРАММ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КРЭКИНГА

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

Искусство разбивать окна

*Человек и кошка плачут у окошка,
Серый дождик каплет прямо на стекло.
К человеку с кошкой едет "неотложка" —
Человеку бедному мозг больной светло.
Федор Чистяков, "Человек и кошка"*

И вот, пройдя долгий и трудный путь исследователя ценных данных, упрятанных в недрах программ, мы, наконец, подошли к вратам "чистого крэкинга". Вспомните, с чего начинаются едва ли не все платные программы — с предложения зарегистрироваться. Это предложение может быть написано аршинными буквами в отдельном окне, появляющемся при запуске, или же в виде маленького MessageBox'a, выскакивающего в самый неподходящий момент во время работы. Оно может быть мерцающим баннером сверху или внизу экрана. Но, как бы оно ни выглядело, цель его существования во всех этих случаях одна — раздражать и давить на психику незарегистрированного пользователя, дабы он, зажав в кулак свои кровные, отправился на почту и перевел их автору программы в знак благодарности за будущее избавление от назойливого окна. Однако далеко не у всех людей, созерцавших эти безыскусные творения, возникали столь теплые чувства к разработчику. И эти раздраженные пользователи дали надоедливому окну то самое имя, под которым они и известны современному крэкеру. Называются эти окна "pag screens" (от английского "pag" — надоедать, приставать, придирааться), и в этой главе речь как раз пойдет о способах борьбы с такими окнами.

Я, увы, лично не застал рождение крэкинга под платформу Win32, но опоздал ненамного, а потому в моей коллекции имеются многие популярные руководства тех времен. В те времена первым словом едва ли не каждого крэкера была классическая команда `brx MessageBoxA` — другие варианты были очень-очень редким исключением. Тридцатидвухразрядные Delphi были лишь светлой мечтой, монстрообразный MFC, спроектированный на базе логики пришельцев с Альфы Центавра, не пользовался особой популярностью, и потому большинство программистов тогда обходилось одним лишь Win32 API. А из всех функций, WinAPI MessageBox была едва ли не самой популярной — ибо трудно было придумать более простое и универсальное средство вывести какое-нибудь нехитрое сообщение, да так, чтобы пользователь не смог это сообщение проигнорировать. Разумеется, разработчики shareware-программ не могли пройти мимо такой возможности, и первые "надоедливые экраны" были теми самыми MessageBox'ами. Мы же, в свою очередь, нещадно `brx`'или эти MessageBox'ы — и выискивали точку ветвления, в которой расходились жизненные пути зарегистрированной и незарегистрированной инкарнации программы. И если когда-нибудь появится мемориал в честь крэкеров Вселенной — на нем, несомненно, огромными золотыми буквами будет выбито `brx MessageBoxA`.

Но все-таки я начну рассказ о борьбе с нежелательными визуальными спецэффектами в программах не с MessageBox'ов. Ибо я стараюсь следовать принципу "от простого — к сложному", а стандартные окна с сообщениями — это все-таки не самое простое из того, с чем вы можете столкнуться. Согласитесь, чтобы удалить из программы лишний MessageBox, все-таки надо приложить некоторые усилия по обнаружению этого MessageBox'a в коде программы при помощи отладчика или дизассемблера. Я начну с рассказа о таких способах удаления `pag screen`'ов, которые не предполагают выхода на уровень ассемблера. Вы спросите — возможно ли такое? Конечно, возможно! И, как это ни удивительно, такую возможность подарила нам операционная система Windows.

Прежде чем переходить к рассмотрению методов борьбы с нежелательными окнами и конкретными примерами их применения, рассмотрим проблему баннеров и `pag screen`'ов, что называется, в перспективе. Я бы выделил четыре подхода к задаче ликвидации рекламных вставок в программах:

1. Изменение свойств объектов (окон, визуальных компонентов) таким образом, чтобы они не отображались на экране. На практике это может быть достигнуто тремя способами: выносом нежелательного объекта за пределы экрана (проще всего использовать отрицательные координаты — при любых размерах монитора объект все равно будет за пределами видимой части экрана); изменением размеров объекта (к примеру, дочерние окна нулевой длины и ширины на экране не видны); модификацией шаблонов диалогов с тем, чтобы придать нежелательным объектам свойства невидимости и неактивности.

2. Удаление шаблонов объектов из исполняемого файла.

3. Модификация кода с целью предотвратить создание или отображение нежелательного объекта.

4. Модификация кода таким образом, чтобы нежелательный объект самоликвидировался сразу после появления.

Вот о первых двух подходах, в основном, и пойдет речь в этой главе.

Ради того чтобы упростить жизнь программистам, в ОС Windows реализована возможность интегрировать в исполняемый файл ресурсы всевозможных типов — текстовые строки, иконки, изображения в стандартных форматах и, что самое интересное, так называемые "шаблоны" (templates) диалоговых окон. Эти шаблоны описывают внешний вид диалогов: размеры, атрибуты, находящиеся внутри диалога управляющие элементы (меню, кнопки, поля редактирования и т.п.) и надписи на этих элементах. При помощи соответствующих функций Windows API на основе этих шаблонов могут быть созданы реальные окна, внешний вид которых будет в точности соответствовать их описанию внутри шаблона. Открыв такой исполняемый файл при помощи какого-либо редактора ресурсов (наиболее известны Resource Hacker, Restorator и eXeScope), вы можете попытаться модифицировать ресурсы программы, например, перевести все надписи на русский язык (если быть до конца точным, то у ресурсов имеется еще один атрибут — идентификатор языка, который, возможно, также потребует модифицировать). Если вы все сделаете правильно, и редактор ресурсов корректно сохранит изменения, после запуска программы соответствующие надписи будут русифицированы. Таким же образом вы можете изменять расположение и свойства элементов интерфейса (например, наличие рамки и ее цвет) в диалоговых окнах. Более того, вы можете заменить не только надписи, но и хранящиеся в ресурсах изображения своими собственными. Нужно отметить, что ресурсы могут храниться не только внутри исполняемого файла, но и в любом другом файле формата Portable Executable. Чаще всего это делается для облегчения локализации — реализовать выбор одной из возможных ресурсных DLL гораздо проще, чем выпускать множество локализованных версий одного и того же исполняемого файла. Кроме того, ресурсные DLL — довольно удобный способ хранения "шкурки" (skin'ов) для приложений, использующих эту технологию модификации интерфейса.

Вообще все ресурсы делятся на типы, среди которых есть такие как иконки (RT_ICON), курсоры (RT_CURSOR), шаблоны диалоговых окон (RT_DIALOG) и многое другое. Для идентификации каждого конкретного ресурса среди других ресурсов того же типа используются числовые либо символичные идентификаторы. Именно на основе идентификаторов программа и различает ресурсы между собой. Что интересно, программе совершенно нет



дела до тех данных, которые прячутся за идентификаторами — поэтому, поменяв местами идентификаторы у пары однотипных ресурсов, можно получить весьма занятные эффекты — например, вместо сообщения об успешном завершении той или иной операции будет появляться сообщение об ошибке (и наоборот).

Теперь немного поговорим о том, какие тайны скрываются в недрах шаблонов диалоговых окон. Прежде всего, умение обращаться с шаблонами позволит вам полностью преобразить интерфейсы очень многих программ, причем сделать это с минимальными затратами сил. Большинство редакторов ресурсов умеют не только декодировать шаблоны в исходный текст, понятный компиляторам ресурсов (при желании вы можете “одолжить” особенно понравившееся окно из чужой программы), но и непосредственно показывать, как этот диалог будет выглядеть на мониторе. А наиболее продвинутые редакторы позволяют даже править эти шаблоны визуальными средствами. Тот, кто знаком с современными визуальными средствами разработки, наверняка догадался, о чем идет речь. А те, кто не догадался, могут просто взять Калькулятор из состава Windows 98 (с Калькулятором от Windows XP такой номер может не пройти — всяческие MUI могут отравить вам радость познания), программку Resource Hacker и вдоволь поэкспериментировать над первой программой при помощи второй — например, поменять местами все кнопки в Калькуляторе. Причем настоятельно рекомендую вам попытаться проделать эту операцию не только тасканием-бросанием кнопок а-ля Visual Basic, но и через ручную правку ресурсного скрипта (resource script).

Но махинации с кнопками — это, в общем-то, мелочи, несмотря на тот эффект, который производит “обработанный напильником” Калькулятор на неподготовленного пользователя. Гораздо важнее другое — то, что управляющие элементы шаблона также имеют свои собственные идентификаторы, по которым программа различает управляющие элементы между собой и “общается” с ними. И эти идентификаторы также можно менять при помощи редактора ресурсов. Попробуйте проделать над несчастным Калькулятором еще один опыт — поменяйте местами значения идентификаторов у двух кнопок, например, у единицы и семерки. Запустив Калькулятор, вы увидите, что единица и семерка действительно поменялись местами! Кроме идентификатора, вы можете менять и другие атрибуты кнопок (более корректно называть их не атрибутами кнопок, а стилями окна) — видимость, наличие рамки, способ выравнивания текста и многое другое. И из этой возможности следуют кое-какие выводы чисто практического свойства.

Довольно часто авторы shareware пользуются следующим приемом — сразу после запуска программы выводится pag screen, кнопка закрытия которого изначально неактивна, и активизируется лишь спустя несколько секунд. Внутри программы это может быть реализовано следующим образом — в шаблоне, по которому создается pag screen, стиль этой кнопки изначально определен как неактивный (он называется WS_DISABLED). После создания окна запускается таймер, который через установленное время активизирует нужную кнопку, и пользователь получает возможность закрыть окно. Однако если мы уберем у кнопки стиль WS_DISABLED, пользователю больше не потребуется ждать несколько томительных секунд, чтобы получить возможность нажать на кнопку. Аналогичным же образом иногда удается включить неактивные функциональные элементы не только на pag screen'ах, но и непосредственно в самой ломаемой программе. Увы, в связи с появлением некоторого количества программ, способных принудительно включать неактивные элементы управления, закрывать “лишние” окна и проделывать прочие столь же милые вещицы (ваш покорный слуга со своим проектом Sign Of Misery тоже отметил на этом поприще), авторы shareware все чаще вводят различные дополнительные проверки, призванные оградить навязчивую рекламу от священного права пользователя эту рекламу не смотреть.

Однако самое интересное заключено даже не в манипуляциях с идентификаторами и стилями. Куда более важным представляется вопрос — а что если просто взять и удалить из секции ресурсов “лишний” диалог? Будем рассуждать логически — создание диа-

логовых окон производится при помощи семейства функций CreateDialog*** (которые затем требуют принудительно отобразить окно при помощи ShowWindow) либо при помощи DialogBox*** (это семейство функций все операции по отображению окна берет на себя). Причем для создания модальных окон (т.е. таких, которые пользователь не смог бы проигнорировать) чаще используется именно второе семейство функций.

Предположим, что программа попыталась создать pag screen на основе шаблона диалога при помощи функции DialogBoxParam (или любой другой функции из семейства DialogBox***), и у нее это не получилось. Большинству разработчиков такой вариант обычно даже в голову не приходит — а потому и проверку на существование шаблона pag screen'а программа обычно не делает, а сразу переходит к созданию и выводу окна на экран. Что бы вы сделали, если бы вас попросили создать окно на основе “никакого” шаблона? Правильно, абсолютно ничего! Вот и Windows поступает точно так же — ничего не делает. Хотя на самом деле такое “ничего” все-таки может иметь далеко идущие последствия — к примеру, ни разу не выполнится оконная процедура выбранного с корнем диалога, что, в свою очередь, может повлечь за собой труднопредсказуемые эффекты. Вспомните принцип минимального вмешательства и ужаснитесь тому, что я предлагаю вам сделать — вырвать из программы особо ценный (с точки зрения разработчика, разумеется) ресурс, отключить “не глядя” оконную процедуру, спровоцировать передачу некорректных данных как минимум в одну функцию WinAPI...

Но “суха теория, мой друг...”, а потому все же посмотрим, что там у нас выросло на зеленеющем “древе жизни”. В качестве наглядного пособия я возьму программу SkyMap Pro 8 Demo, которая обладает одним весьма ценным для нас качеством — наличием pag screen'а, шаблон которого упрятан глубоко в недрах секции ресурсов и имеет идентификатор 3006. В качестве орудия, при помощи которого производилась ампутация надоедливой диалога, я воспользовался программой Resource Hacker (в принципе, подошел бы и любой другой редактор ресурсов). Сама операция заняла не более 15 секунд, после чего исполняемый файл “похудел” на 4 килобайта. После того как я произвел пробный запуск исправленной программы, оказалось, что pag screen как рукой сняло — и мне для этого не потребовались отладчики, дизассемблеры и прочая крэкерская “тяжелая артиллерия”! Видите, все далеко не так страшно, как может показаться. В принципе, если вы считаете, что выдирание диалога с корнем — слишком уж варварская операция, шаблон можно оставить на месте, лишь заменив его идентификатор на другой, не используемый в программе. Результат будет аналогичный — программа не сможет найти ресурс с измененным идентификатором и не создаст pag screen. Этот прием даже предпочтительнее — практика показала, что не всегда и не во всех редакторах ресурсов операция удаления проходит успешно.

А ведь есть еще и третий путь избавиться от диалога путем изменения идентификатора! Ничто не мешает исправить идентификатор не в секции ресурсов, а непосредственно в исполняемом коде, в том месте, где этот идентификатор передается в функции WinAPI. Поскольку наш идентификатор имеет довольно редкое значение 3006, он как нельзя лучше подходит для поиска внутри исполняемого файла. Найдя среди всех подходящих значений то, которое отвечает за создание pag screen'а — замените его каким-нибудь другим, не указывающим ни на один диалог, например, 12345. Когда вы запустите исправленное приложение, оно, конечно, попытается найти и загрузить шаблон с ID=12345, но у программ это все равно не получится. Впрочем, я считаю такой путь излишне изощренным и рассказал о нем главным образом в познавательных целях — гораздо проще “забить” вызов функции создания диалога и все, что с ней связано, пор'ами и получить тот же самый эффект.

Успешность такого метода борьбы с “лишними” окнами, вообще говоря, сильно зависит от внутренней логики приложения. Pag screen, к примеру, может оказаться не “пустышкой”, предназначен-



ной исключительно для раздражения пользователя, а исполнять какие-либо неочевидные функции. Например, на обработку сообщения WM_INITDIALOG в оконной процедуре pag screen'a может быть подвешена инициализация каких-либо критически важных для программы структур. В общем случае такая инициализация может происходить при возникновении практически любого события, связанного с этим окном: при нажатии на какую-либо кнопку в окне, при закрытии окна, в момент поступления первого сообщения WM_PAINT — перечислять варианты можно очень долго. И если просто удалить pag screen, это "волшебное" событие не произойдет — следовательно, не будет произведена инициализация, отчего, в итоге, программа будет работать некорректно, если вообще будет работать.

На практике такая защитная техника в чистом виде встречается достаточно редко — ведь программисту, использовавшему подобный прием, придется учитывать при программировании различия в работе между полной/зарегистрированной (без pag screen'a) и урезанной/незарегистрированной версиями программы. Тем не менее, если вы все же столкнетесь с реализацией такого алгоритма, вам вряд ли станет легче от осознания редкости встреченной проблемы. Поэтому давайте подумаем о том, как эту проблему можно было бы решить. В теории решение довольно простое — вам нужно найти функции, которые выполняют все нужные действия по инициализации, и вызвать их вручную. На практике вам скорее всего понадобится добраться до цикла выборки сообщений внутри оконной процедуры и проанализировать его содержимое. Обратите также внимание на следующий факт — если вам удалось обнаружить, что инициализация программы происходит в ответ на WM_INITDIALOG, из этого отнюдь не следует, что необхо-

димо выполнить **все** действия, которые программа выполняет при поступлении этого сообщения. Посмотрите на следующий пример:

```
.IF uMsg==WM_INITDIALOG
    invoke GetDlgItem, hWnd, TRIALUSE_BTN
    invoke SetFocus,eax
    invoke InitProgram
```

После того как мы умозрительно удалили диалог, которому должно было поступать сообщение, попытка установить фокус на кнопку TRIALUSE_BTN становится совершенно бессмысленна — поскольку больше нет диалога, нет и кнопки, которая в этом диалоговом окне находилась. Поэтому в нашем простейшем примере вместо вызова функций создания диалога было бы достаточно просто написать call InitProgram, а все лишние байты забить пор'ами. Однако представьте себе, что оконная процедура содержит какие-либо локальные переменные (например, вычисленное ранее число дней до истечения триального срока), используемые внутри InitProgram, или же продельывает какие-либо операции с созданным диалогом — и вы поймете, почему простейший путь не является лучшим. Что делать в таком случае? Из каждой безвыходной ситуации есть, как минимум, два выхода. Можно углубиться в дебри процедуры InitProgram и проверить, не обращается ли она к каким-либо объектам, которые мы так лихо снесли вместе с диалогом, и потом долго заниматься "художественной штопкой" машинного кода. Я сам пару раз применял такую технику, и потому могу сказать определенно — этот процесс мне совершенно не понравился. Поэтому мы поступим хитрее — попробуем поискать другое, более простое решение.

(Продолжение следует)

И.РОЩИН,

г.Москва,
ZXNet: 500:95/462.53
E-mail: bestview@mtu-net.ru
WWW: http://www.ivr.da.ru

Программа для разрезания файлов

Введение

Предлагаю вашему вниманию программу, с помощью которой можно разрезать заданный файл на части указанной длины, или получить часть файла указанной длины, начиная с указанного смещения, или пропустить первые X байтов файла, а остаток разрезать на части размером Y байтов, или разделить файл на части в X, Y и Z байтов и остаток... Как вы увидите дальше, программа позволяет разрезать файл на части и более сложным образом (замечу, что на самом деле "разрезание" — не совсем точный термин, так как программа не изменяет исходный файл, а создает новые файлы — части исходного.)

Руководство пользователя

При запуске программы надо указать в командной строке имя исходного файла, имя получаемых файлов (без расширения — оно будет сформировано автоматически) и команды, определяющие, как именно файл будет разбит на части.

Пример:

```
file_cut.exe 1.txt aaa 0x735-,92163,*
```

Исходный файл предполагается находящимся в текущем каталоге, там же будут сформированы и выходные файлы. Расширение первого сформированного файла будет "001", второго — "002", и так далее (если получится более 999 частей, будут записаны только первые 999).

Теперь о командах, определяющих, как именно файл будет разбит на части. Может быть указана одна или несколько команд; если их несколько, они разделяются запятыми. Всего можно использовать четыре различных типа команд:

- p — записать часть длиной p байтов, начиная с текущей позиции в исходном файле (если до конца исходного

файла осталось менее p байтов, размер записанной части будет равен оставшейся части файла);

- p — пропустить p байтов с текущей позиции в исходном файле (если до конца исходного файла осталось менее p байтов, будут пропущены все оставшиеся байты);

- p* — записывать части длиной p байтов, начиная с текущей позиции в исходном файле, пока не будет достигнут конец файла (таким образом, длина последней записанной части может оказаться меньше p);

- * — записать часть, начиная с текущей позиции в исходном файле и до конца файла. Эта команда, очевидно, может быть только последней.

Во всех командах p — число от 1 до 0x7FFFFFFF (2147483647), записанное в десятичном или в шестнадцатеричном виде (при записи в шестнадцатеричном виде число должно предвшаться символами "0x", регистр символов в записи числа не имеет значения).

Программа может обработать не более 64 команд (определяется значением константы MAX_COMM в исходном тексте программы), но столько команд вы все равно не введете из-за ограничения на длину командной строки в DOS.

Команды выполняются в порядке их записи. Перед выполнением первой команды текущая позиция в исходном файле — это начало файла. Работа программы заканчивается, если все команды выполнены, или если достигнут конец исходного файла, или если произошла какая-нибудь ошибка при работе с файлами.

Всего лишь четыре типа команд дают возможность достаточно разнообразной обработки файлов. Ниже приведены несколько примеров.



1. Получить первые 500 байтов файла:
500
2. Наоборот, пропустить первые 500 байтов и получить остаток файла:
500-, *
3. Получить часть файла, начинающуюся со смещения 0x1000 от начала и длиной 0x300 байтов:
0x1000-, 0x300
4. Разбить файл на четыре части — 100, 200, 300 байтов и остаток:
100, 200, 300, *
5. Разбить файл на части по 10000 байтов (кроме, может быть, последней части):
10000*
6. Пропустить 0x400 байтов файла, записать часть длиной 123 байта, потом пропустить еще 0x100 байтов и записать остаток:
0x400-, 123, 0x100-, *

Также замечу, что не очень сложно доработать программу, добавив еще какие-нибудь типы команд.

Исходный текст программы на Turbo C

```
/* =====
File: "file_cut.c"
Compiler: Turbo C 2.0
===== */

#include <stdio.h>
#include <stdlib.h>
#include <io.h>
#include <string.h>
#include <limits.h>

void save_part (char *nameonly, long begin_part, long len_part);
int work_com (char* s);
int get_val (char **s, long *v);

#define MAX_COMM 64 /* Максимальное количество команд. */

int n_comm=0; /* Количество команд. */
int comm_type [MAX_COMM]; /* Тип команды (0-3). */
long comm_val [MAX_COMM]; /* Число в команде (если есть). */
int n_part=1; /* Номер записываемой части. */
long len_src, point_src=0; /* Длина исходного файла и положение указателя. */

FILE *f_src, *f_dst;

void main (int argc, char *argv[])
{
    printf ("\nFile_Cut by Ivan Roshin, 19.03.2004.\n");
    if (argc!=4)
        {printf ("Syntax: file_cut.exe source-file destiny-file(without extension) commands\n"); exit(0);}
    /* Разбираем заданные команды. */
    if (work_com(argv[3])==1) {printf ("Incorrect commands!\n"); exit(0);}
    /* Проверяем, чтобы имя формируемых частей было не длиннее 8 символов. */
    if (strlen(argv[2])>8)
        {printf ("Error in destiny filename!\n"); exit(0);}
    f_src=fopen(argv[1], "rb"); /* Открываем исходный файл. */
    if (f_src==NULL) {printf ("Can't open source file!\n"); exit(0);}
    len_src=filelength(fileno(f_src)); /* Определяем длину исходного файла. */
    if {len_src==1} {printf ("Can't define source length!\n"); exit(0);}

    /* Выполнение указанных команд (если длина исходного файла больше 0). */
    if (len_src>0)
    {
        int i;
        for (i=0; i<n_comm; i++)
        {
            switch (comm_type[i])
            {
```

```
case 0: /* Запись части файла заданной длины. */
    if (comm_val[i]>(len_src-point_src))
        comm_val[i]=len_src-point_src;
    save_part (argv[2], point_src, comm_val[i]);
    point_src+=comm_val[i]; n_part++;
    break;
case 1: /* Пропуск части файла заданной длины. */
    if (comm_val[i]>(len_src-point_src))
        comm_val[i]=len_src-point_src;
    point_src+=comm_val[i];
    break;
case 2: /* Запись частей заданной длины, пока не кончится файл. */
do
{
    if (comm_val[i]>(len_src-point_src))
        comm_val[i]=len_src-point_src;
    save_part (argv[2], point_src, comm_val[i]);
    point_src+=comm_val[i]; n_part++;
} while (point_src<len_src);
break;
case 3: /* Запись остатка файла. */
    save_part (argv[2], point_src, len_src-point_src);
    point_src=len_src; n_part++;
}
/* Если файл уже обработан или номер следующей записываемой части больше максимального (999) - заканчиваем выполнение команд. */
if ((point_src==len_src)|| (n_part>999)) break;
}
if (fclose(f_src)!=0) printf ("Can't close source file!");
printf("\n");
}

/* =====
Функция save_part - запись файла-части.
Вход: nameonly - имя записываемого файла (без расширения),
n_part - номер записываемой части (по нему формируется расширение),
begin_part - смещение части от начала исходного файла,
len_part - длина части.
===== */

void save_part (char *nameonly, long begin_part, long len_part)
{
    long i;
    char dst_ext[4];
    char dst_name[12];

    /* Формируем имя записываемого файла с расширением. */
    itoa (n_part, dst_ext, 10);
    strcpy (dst_name, nameonly);
    strcat (dst_name, ".");
    for (i=strlen(dst_ext); i<3; i++) strcat (dst_name, "0");
    strcat (dst_name, dst_ext);

    /* Записываем. */
    printf ("Creating file: %s ", dst_name);
    f_dst=fopen(dst_name, "wb");
    if (f_dst==NULL) {printf ("Can't open destiny file!\n"); exit(0);}
    fseek (f_src, begin_part, SEEK_SET);
    for (i=0; i<len_part; i++)
    {
        int c;
        c=fgetc(f_src); if (c==EOF)
            {printf ("Can't read from source file!\n"); exit(0);}
        if (fputc(c, f_dst)==EOF)
            {printf ("Can't write to destiny file!\n"); exit(0);}
    }
    if (fclose(f_dst)!=0) {printf ("Can't close destiny file!\n"); exit(0);}

    printf ("OK\n");
}

/* =====
Функция work_com - разбор заданных команд.
Вход: s - строка с командами.
Выход: 0 - OK, 1 - ошибка в строке с командами.
===== */

int work_com (char *s)
```

```

{
m1: /* Главный цикл. */
/* Строка с командами закончилась? */
if (*s==0) {if (n_comm==0) return (1); else return (0);}

/* В массиве еще есть место для команд? */
if (n_comm==MAX_COMM) return (1);

/* Команда началась с числа? */
if ((*s>='0') && (*s<='9'))
{
if (get_val(&s,&comm_val[n_comm])!=1) return (1);
/* Ошибка! */
/* Проверим, имеется ли за числом символ "-" или "+", и
в зависимости от этого определяем тип команды. */
if (*s=='-') {comm_type[n_comm]=1; s++; n_comm++;
goto end_comm;}
if (*s=='+') {comm_type[n_comm]=2; s++; n_comm++;
goto end_comm;}
comm_type[n_comm]=0; n_comm++; goto end_comm;
}

/* Команда "?" */
if (*s=='?') {comm_type[n_comm]=3; s++; n_comm++;
goto end_comm;}

/* Команда не распознана. */
return (1); /* Ошибка! */

/* Текущая команда закончилась. Проверим: теперь должен быть
либо конец строки (код 0), либо запятая и далее ненулевой
символ (начало следующей команды). */
end_comm:
if (*s==0) goto m1;
if ((*s==',' && (*(s+1)!=0)) {s++; goto m1;}
return (1); /* Ошибка! */
}

/* =====
Функция get_val - преобразование строки с записью числа в это
число. Число может быть представлено в десятичном или
шестнадцатеричном виде и должно находиться в диапазоне
0<=x<=LONG_MAX.
Вход: **s - адрес указателя на строку,
*v - адрес, куда поместить число.
Выход: если число задано корректно, функция возвращает 0,
число помещено по адресу *v, а указатель на строку
указывает на первый символ после числа. Если число
задано некорректно, функция возвращает 1.
===== */

```

```

int get_val (char **s, long *v)
{
int d; /* Очередная цифра числа. */

/* Если первый символ строки - '1'-'9',
то началось десятичное число. */

if ((*s>='1') && (*s<='9'))
{
/* Началось десятичное число, обрабатываем. */
*v=**s-'0';

```

```

loop_dec:
(*s)++;
if ((*s>='0') && (*s<='9')) /* Продолжение числа (цифра 0-9)? */
*
{
d=**s-'0';
/* Добавляем цифру к числу (и следим за переполнением). */
if (*v>(LONG_MAX/10)) return (1); /* Ошибка! */ else *v*=10;
if ((LONG_MAX-*v)<d) return (1); /* Ошибка! */ else *v+=d;
goto loop_dec;
}
return(0); /* OK! */
}

/* Если первый символ строки - '0',
то началось шестнадцатеричное число (десятичное число
не может начинаться с нуля). */

if ((*s=='0')
{
(*s)++; if ((*s!='x') && (*s!='X')) return (1); /* Ошибка! */
*v=0;
loop_hex:
(*s)++;
if ((*s>='0') && (*s<='9')) || ((*s>='a') && (*s<='f')) ||
((**s>='A') && (**s<='F'))
{
/* Получаем значение очередной цифры числа. */
if ((*s>='0') && (*s<='9')) d=**s-'0';
if ((*s>='a') && (*s<='f')) d=**s-'a'+10;
if ((*s>='A') && (*s<='F')) d=**s-'A'+10;
/* Добавляем цифру к числу (и следим за переполнением). */
if (*v>(LONG_MAX/16)) return (1); /* Ошибка! */ else *v*=16;
if ((LONG_MAX-*v)<d) return (1); /* Ошибка! */ else *v+=d;
goto loop_hex;
}
/* Число кончилось. Теперь, если *v=0, то либо число равно 0,
либо в числе не было ни одной цифры. И то, и другое недопустимо. */
if (*v==0) return (1); /* Ошибка! */ else return (0); /* OK! */
}

/* Число в строке не обнаружено. */
return (1); /* Ошибка! */
}

```

Комментарии

Исполняемый файл программы вы можете скачать с сайта журнала или с моей web-страницы.

Кстати, если кто не знает, как выполнять обратную задачу — склеивать файлы из частей, напоминаю, что это можно сделать с помощью джосовской команды "copy". В командной строке надо указать параметр "/b" (двоичное склеивание), имена склеиваемых файлов в нужном порядке, соединенные знаком "+", и имя файла-результата.

Пример:

```
copy /b 1.txt+2.txt+3.txt all.txt
```

Обращаю внимание, что получаемый файл не должен совпадать с каким-либо из исходных файлов, иначе содержимое этого исходного файла может быть потеряно, и выполнение команды завершится с ошибкой!

И.ШИРКО,
г.Минск, БГУИР, ФИТУ.

Microsoft Agent Часть III

В этой заключительной части моего обзора технологии Microsoft Agent мы рассмотрим две интересные темы:

- управление персонажем с помощью голосовых команд;
- создание динамической библиотеки для удобной работы с MsAgent.

Управление персонажем с помощью голосовых команд

То, что Джин умеет говорить, конечно, здорово, но этим возможности технологии MsAgent далеко не исчерпаны. Оказы-

вается, Джин еще умеет **понимать** ваши слова, произнесенные в микрофон. Представьте, сидите вы за компьютером (или перед ним, как кому нравится) и командуете: "Запустить Winamp!", "Открыть CD-ROM!", "Выключить дискету!",..., а забавливый Джин летает по экрану и выполняет ваши приказы. В действительности, конечно, все не так безоблачно. Распознавание речи пока поддерживается только для английского языка. Но даже если вы на самом натуральном английском произнесете какое-нибудь слово, то нет никакой гарантии, что оно будет распознано правильно. Есть еще несколько маленьких проблем, на развее появление Джина на экране после того, как вы его позвали, не компенсирует все неудобства? Итак, очередное неудобство — для поддержки распознавания голоса нужно скачать и установить файл <http://activex.microsoft.com/activex/controls/agent2/actcnc.exe> (6 Мб).



Табл. 1

Параметр	Описание
Name	Строка, являющаяся идентификатором новой команды
Caption	Название команды, которое будет отображаться в выпадающем меню и в окне команд персонажа
Voice	Слово (или фраза), на которое будет "отзываться" данная команда
Enabled	Булево значение (ложь/истина), определяющее активность команды
Visible	Булево значение, определяющее видимость заголовка команды в выпадающем меню персонажа

Теперь немного теории. Объект персонажа содержит коллекцию *Commands*, хранящую все определенные для него команды. Рассмотрим некоторые методы для работы с этой коллекцией:

- **function** Add(Name, Caption, Voice, Enabled, Visible): IAgentCtlCommandEx; — добавляет новую команду в коллекцию. В табл.1 приведено описание допустимых параметров.

- **function** Command(Name: WideString): IAgentCtlCommand; — возвращает объект команды с идентификатором Name;

- **function** Insert(Name, RefName, Before, Caption, Voice, Enabled, Visible): IAgentCtlCommandEx; — добавляет в указанную позицию коллекции *Commands* новую команду. Если Before=Истина, то новая команда добавляется перед командой с идентификатором RefName, иначе — после. Остальные параметры — такие же, как и у функции Add;

- **procedure** Remove(const Name: WideString); — удаляет команду с идентификатором Name;

- **procedure** RemoveAll; — удаляет все команды из коллекции;

А теперь разберемся с объектом *UserInputIAgentCtlUserInput*, который мы будем получать при произнесении очередной команды. В табл.2 приведены некоторые его свойства.

А теперь запустите Delphi и откройте проект, созданный нами в [1]. Положите на форму новую кнопку "Добавить команды" и запишите обработчик события нажатия на нее:

```
procedure TForm1.ButtonClick(Sender: TObject);
begin
    //переключаем язык на английский, т.к. только для него
    //поддерживается распознавание речи
    Character.LanguageID:=5409;
    //добавляем команду "Test"
    Character.Commands.Add('Test', '&Test', 'Test', 1,
True);
    //добавляем команду "Go"
    Character.Commands.Add('Go', '&Go', 'Go', 1, True);
end;
```

Добавьте к форме компонент *Memo1:TMemo* и запишите процедуру, которая будет вызываться при произнесении команды в микрофон (событие *OnCommand* компонента *Agent1*):

```
procedure TForm1.Agent1Command(Sender: TObject;
const UserInput: IDispatch);
var
    ThisCmd:IAgentCtlUserInput;
begin
    //получаем данные о произнесенной команде
    ThisCmd := (UserInput as IAgentCtlUserInput);
    //убеждаемся, что идентификатор команды не пуст
    if ThisCmd.Name <> '' then
    begin
        //выводим имя распознанной команды
        memo1.Lines.Add('Персонаж '+ThisCmd.CharacterID + '
распознал команду ' + ThisCmd.Name);
        //выводим степень достоверности распознавания
        memo1.Lines.Add('с вероятностью ' +
inttostr(ThisCmd.confidence));
        //Если распознана команда "Test"
        if ThisCmd.Name = 'Test' then
        begin
            //если степень достоверности положительна, то аплодируем
            //пользователю, иначе просим его говорить более разборчиво
            if ThisCmd.Confidence > 0 then
                Character.Play('Congratulate')
```

Табл. 2

Параметр	Описание
CharacterID	Идентификатор персонажа, которому адресуется команда
Name	Идентификатор команды
Voice	Слово (или фраза), которое услышал персонаж
Confidence	Целое число, которое выражает степень достоверности распознавания команды
Count	Количество возможных альтернатив для услышанной фразы. Если Count=0, то команда неизвестна, если Count=1, то команда распознана однозначно, иначе существуют альтернативы. Две наиболее вероятные альтернативы определяются свойствами Alt1Name, Alt1Confidence, Alt1Voice, Alt2Name, Alt2Confidence и Alt2Voice.

```
else
    Character.Speak('Говори внятно!', EmptyParam);
exit;
end;
//Если распознана команда "Go"
if ThisCmd.Name = 'Go' then
begin
    if ThisCmd.Confidence > 0 then
    begin
        Character.Play('Sad')
        Character.Speak('Ну, я пошел...', EmptyParam);
        Character.MoveTo(300, 300, EmptyParam);
    end
    else
        Character.Speak('Говори внятно!', EmptyParam);
    exit;
end;
//если команда распознана неоднозначно, то выводим наиболее
//вероятные альтернативы
if ThisCmd.count > 1 then
begin
    memo1.Lines.Add('Существует ' + inttostr(ThisCmd.count) +
' альтернатив.');
```

```
memo1.Lines.Add('Услышано либо ' + ThisCmd.voice);
memo1.Lines.Add('либо ' + ThisCmd.Alt1voice);
memo1.Lines.Add('а может и ' + ThisCmd.Alt2voice);
end
else
    memo1.Lines.Add(ThisCmd.Voice + ' - неизвестная
команда!');
end;
end;
```

А сейчас самое время взять в руки микрофон (если его конструкция позволяет сделать это) и запустить программу. После успешного запуска щелкаем по кнопке "Добавить команды". Затем нажимаем и удерживаем клавишу *Scroll Lock* (это нужно делать перед подачей любой команды). Под Джинном должна появиться рамка с сообщением, что Джин вас внимательно слушает (см. рисунок). Вот теперь самый ответственный момент — четко произнесите слово "Hide". Это стандартная команда, которая приводит к исчезновению персонажа. Чтобы Джин опять появился на экране, следует нажать на *Scroll Lock* и ласково позвать: "Genie".

Это были стандартные команды, а теперь перейдем к тем, которые мы добавили сами. Произнесите в микрофон слово "Go". В компоненте *Memo1* появится строка "Персонаж Genie распознал команду Go с вероятностью N". Если $N > 0$ (т.е. слово произнесено достаточно разборчиво), то Джин загрузит и переместится поближе к центру экрана, иначе Джин попросит вас говорить внятнее. Еще можете дать команду "Test". Если все будет нормально, то Джин вам поаплодирует.

По описанному шаблону не составит особого труда написать программу для голосового управления компьютером. И все это безо всякого вникания в технологию распознавания речи.

Создание динамической библиотеки для работы с MsAgent

При использовании элемента *ActiveX* для удобной работы с технологией *Microsoft Agent* мы не проверяем, установлены ли

компоненты этой технологии в системе. Если они не установлены, то при загрузке программы возникнет ошибка, и запуск программы прервется. В связи с этим предлагаю несложный способ, который поможет обрабатывать такие ситуации. Нужно написать DLL, которая возьмет на себя всю работу с MsAgent, а сама программа будет ее динамически загружать. Создайте в Delphi новую DLL со следующим кодом:

```
library AgentDll;
uses
  OleCtrls, AgentObjects_TLB, ActiveX, Forms;
var
  Character: IAgentCtlCharacter;
  Request: IAgentCtlRequest;
  Agent1: TAgent;

Procedure InitAgent;
begin
  OleInitialize(nil);
  //динамически создаем объект Agent1
  Agent1:=TAgent.Create(nil);
  Agent1.Connected:=true;
end;

Procedure LoadChar(Agent, CharFile:string);
begin
  Request:=Agent1.Characters.Load(Agent, CharFile);
  Character:=Agent1.Characters.Character(Agent);
end;

procedure WaitFor;
var
  Status:LongInt;
begin
  repeat
    Application.ProcessMessages;
    Status := Request.Get_Status;
  until (Status <> 2) and (Status <> 4);
end;

Procedure ShowAgent;
begin
  Request := Character.Show(False);
end;

Procedure HideAgent;
begin
  Request := Character.Hide(False);
end;

Procedure PlayAgent(Anim:string);
begin
  Request := Character.Play(Anim);
end;

Procedure SpeakAgent(Frase:string);
begin
  Request := Character.Speak(Frase, EmptyParam);
end;

Procedure MoveToAgent(x,y:Smallint;speed:OleVariant);
begin
  Request := Character.MoveTo(x,y,Speed);
end;

Procedure GestureAtAgent(x,y:Smallint);
begin
  Request := Character.GestureAt(x,y);
end;

Procedure UnLoadAgent(Anim:string);
begin
  Agent1.Connected:=false;
  Agent1.Free;
end;

exports
  InitAgent,
  LoadChar,
  WaitFor,
  ShowAgent,
```



-- Genie is listening --
for commands.

```
HideAgent,
PlayAgent,
SpeakAgent,
MoveToAgent,
GestureAtAgent,
UnLoadAgent;
```

```
begin
end.
```

Данная библиотека экспортирует большинство основных функций, которые часто используются для работы с MsAgent. В ней мы динамически создаем объект типа TAgent, благодаря чему можем отслеживать ситуации, когда MsAgent не установлена. В принципе, мы могли бы же самое сделать и в самой программе, но при использовании DLL можно легко убирать поддержку MsAgent из программы, достаточно лишь удалить эту библиотеку. К тому же, если перенести этот код в программу, то ее размер намного увеличится. И еще одно достоинство такого подхода — мы научимся динамически загружать DLL. Создайте новый проект и в разделе типов запишите несколько строк:

```
//определяем типы, которые соответствуют процедурам,
//содержащимся в библиотеке
TProcessProc1=procedure;
TProcessProc2=procedure(st:string);
TProcessProc3=procedure(x, y:smallint; Speed:OleVariant);
TProcessProc4=procedure(x, y:smallint);
Объявите глобальные переменные:
```

```
var
  InitAgent, ShowAgent, WaitFor,
  HideAgent, UnloadAgent:TProcessProc1;
  LoadChar, PlayAgent,
  SpeakAgent:TProcessProc2;
  MoveToAgent:TProcessProc3;
  GestureAtAgent:TProcessProc4;
  lib:integer; //идентификатор нашей библиотеки
```

Теперь напишите процедуру обработки события создания формы:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  //загружаем динамически нашу библиотеку
  lib:=LoadLibrary(PChar('AgentDll.dll'));
  if lib>0 then
    begin
      //получаем адреса нужных нам процедур
      @WaitFor:=GetProcAddress(lib,'WaitFor');
      @InitAgent:=GetProcAddress(lib,'InitAgent');
      @ShowAgent:=GetProcAddress(lib,'ShowAgent');
      @HideAgent:=GetProcAddress(lib,'HideAgent');
      @UnLoadAgent:=GetProcAddress(lib,'UnLoadAgent');
      @LoadChar:=GetProcAddress(lib,'LoadChar');
      @PlayAgent:=GetProcAddress(lib,'PlayAgent');
      @SpeakAgent:=GetProcAddress(lib,'SpeakAgent');
      @MoveToAgent:=GetProcAddress(lib,'MoveToAgent');
      @GestureAtAgent:=GetProcAddress(lib,'GestureAtAgent');
    end;
    try
      //Джин появляется, аплодирует и исчезает
      TProcessproc1(InitAgent);
      TProcessproc2(LoadChar)('Genie.acs');
      TProcessproc1(ShowAgent);
      TProcessproc2(PlayAgent)('Congratulate');
      TProcessproc1(WaitFor);
      TProcessproc1(UnLoadAgent);
    Except
      //если возникла исключительная ситуация, то
      показываем соответствующее
      //сообщение
      ShowMessage('Компоненты MsAgent не установлены!');
    end;
    //освобождаем библиотеку
    FreeLibrary(lib);
  end;
end;
```

Вот таким образом мы можем без всяких проблем использовать возможности MsAgent. На этом мой обзор этой замечательной технологии завершен.

Литература

1. И.Ширко. Microsoft Agent. Часть II. — Радиомир. Ваш компьютер, 2004, №4, С.34.

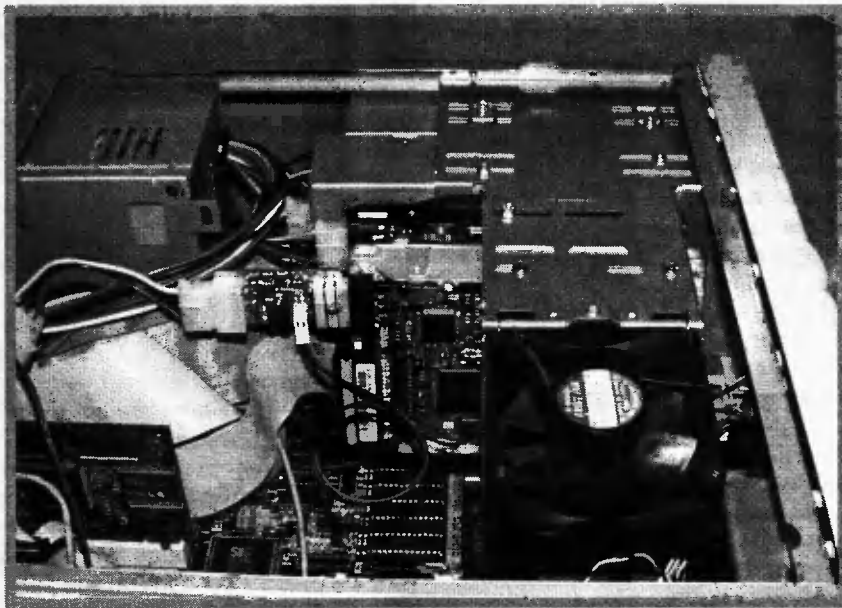
Терморегулятор для управления дополнительным вентилятором охлаждения

А.ПАВЛОВ,
г.Санкт-Петербург,
E-mail: pavlov@mail.loniis.ru

С повышением тактовой частоты, сложности электронных схем, количества периферийных устройств, устанавливаемых в современное электронное оборудование (в компьютеры, контроллеры, мини-АТС и т.д.), соответственно, повышается выделение тепла в нем, что ведет к повышению температуры внутри корпуса изделия. Избыточное тепло необходимо отводить в окружающую среду. Для этих целей, как правило, внутри корпуса устанавливается один или несколько дополнительных вентиляторов охлаждения. Но с повышением количества вентиляторов растет и уровень шума во время работы оборудования. Между тем, достаточно, чтобы вентиляторы дополнительного охлаждения включались только при превышении заранее заданной температуры, и отключались, когда температура уменьшится до необходимого, заранее установленного уровня. При таком режиме работы время, когда дополнительный вентилятор включен, снижается, что приводит к снижению шума, а также к уменьшению потребляемой устройством электроэнергии. Для управления работой вентилятора необходимо специальное устройство — терморегулятор, который следит за температурой и включает и выключает его. Надежный и несложный терморегулятор можно сделать самостоятельно. Конструкция такого устройства представлена в этой статье.

Внешний вид терморегулятора, установленного в компьютер, показан на рис.1. Устройство представляет собой

Рис. 1



Во-вторых, место расположения разъемов питания HDD или FDD находится приблизительно по центру корпуса ком-

пьютера, что дает возможность контролировать усредненную температуру внутри системного блока.

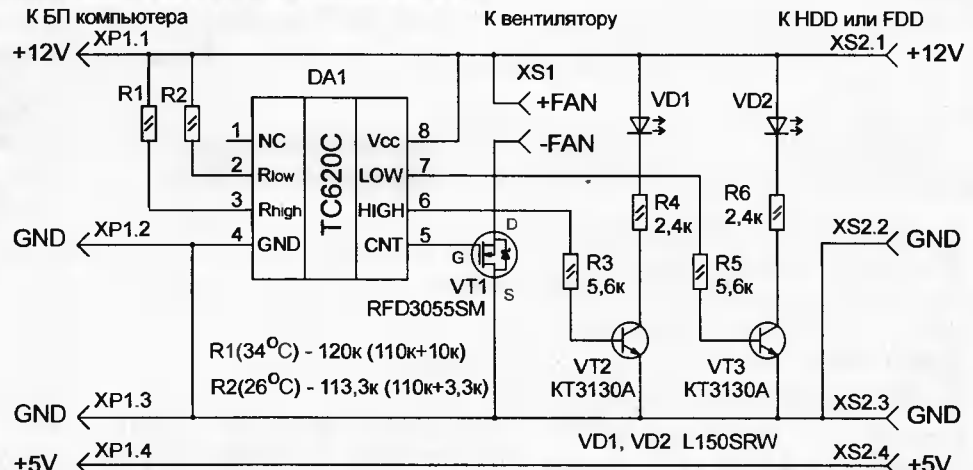
На рис.2 приведена принципиальная схема описываемого устройства. Его центральным элементом является микросхема TC620CEOА (DA1) производства ф. Microchip [1].

Фирма Microchip Technology Inc. знаменита своими PIC-микроконтроллерами. В последнее время рынок электронных компонентов динамично развивается, и Microchip не остается в стороне, увеличивая номенклатуру выпускаемых изделий и выходя на новые для себя области рынка электронной техники. Одной из таких областей является рынок аналоговой техники, и в частности — датчиков температуры.

TC620 — это микросхема, разработанная для использования в устройствах отслеживания и управления температурой. Она позволяет значительно уменьшить количество диск-

Рис. 2

печатную плату с тремя разъемами. Одним разъемом оно соединяется с разъемом питания винчестера (HDD) или 3-дюймового дисковода (FDD), другим — с разъемом кабеля питания периферийных устройств компьютера (+5 В, +12 В и GND). Третий разъем предназначен для подключения дополнительного вентилятора. Такая конструкция устройства выбрана по двум причинам. Во-первых, для регулятора не требуется дополнительных деталей крепления. По сути, он представляет собой «жесткий» удлинитель кабеля питания.





ретных элементов, необходимых для изготовления терморегулятора. Значение контролируемой температуры устанавливается посредством изменения номинала двух внешних резисторов. TC620 имеет встроенный датчик температуры. При проектировании устройств, регулирующих температуру, также возможно применение микросхемы TC621. В отличие от TC620, к TC621 подключается внешний датчик температуры — терморезистор. Он подключается одним выводом к первому выводу микросхемы, а другим выводом к V_{cc} . Эту микросхему удобно использовать для дистанционного контроля температуры.

В предлагаемом устройстве без дополнительных изменений в принципиальной схеме можно применить микросхемы, представленные в табл.1. Логика работы микросхемы TC620 показана на рис.3. Микросхема имеет два логических выхода (HIGH и LOW),

температуры. Светодиоды сигнализируют о превышении, соответственно, уровня максимальной и минимальной температуры. Ток через светодиоды составляет около 6 мА. Такого тока более чем достаточно для хорошей яркости свечения большинства светодиодов, не говоря уже о светодиодах с повышенной светоотдачей. Дополнительный выход CONTROL устанавливается в "1", когда температура превышает верхний предел (T_{high}) и сбрасывается в "0", когда температура падает ниже нижнего предела (T_{low}). Этот выход в данном устройстве используется для управления полевым п-канальным транзистором VT1, который включает и отключает вентилятор охлаждения.

Необходимо отметить, что у микросхем TC621 для выводов HIGH и LOW, в отличие от TC620, активным состоянием является низкий уровень. А у микросхем TC620H, в отличие от TC620C, сигнал на выводе CONTROL имеет инверсное значение.

Значение сопротивления внешних резисторов для TC620/621 рассчитывается следующим образом:

$$R_{low/high} = 0,5997 \cdot T^{2,1312},$$

где:

- $R_{low/high}$ — сопротивление резистора, Ом;
- T — величина контролируемой температуры, град. Кельвина.

Например, для температуры 35°C значение сопротивления составит:

$$R_{low/high} = 0,5997 \cdot (35+273,15)^{2,1312} = 120776,93 \text{ Ом}.$$

При величине сопротивлений, приведенных на принципиальной схеме, отслеживается две фиксированные температуры: минимальная — 26°C, и максимальная — 34°C. Различие значений максимальной и минимальной температуры (гистерезис) необходимо для того, чтобы на пороге срабатывания регулятора небольшие колебания температуры не приводили к многократным включениям и выключениям вентилятора. Производитель микросхем — ф. Microchip — рекомендует, чтобы значение максимальной контролируемой температуры отличалось от минимальной не менее чем на пять градусов [2]. При необходимости, для изменения значений контролируемых температур нужно при изготовлении устройства изменить номиналы резисторов согласно табл.2. Для точной подгонки номиналов $R1$ и $R2$, каждый из них набирается из двух последовательно соединенных резисторов: $R1.1+R1.2$ и $R2.1+R2.2$.

Микросхемы TC620 выпускаются в разных корпусах и могут применяться, в зависимости от исполнения, в температурном диапазоне от -40°C до +125°C. Напряжение питания микросхемы может быть от 4,5 В до 18 В.

Терморегулятор собран на печатной плате из двустороннего фольгированного текстолита. Чертеж и фотография платы представлены на рис.4 и рис.5. Она рассчитана на установку SMD-компонентов для поверхностного монтажа и микросхемы в корпусе SOIC (SO-8, 4303.8). Разводка печатных проводников сделана таким образом, что при изготовлении печатной платы в домашних условиях, т.е. без металлизации отверстий, достаточно впаять кусочки провода

Табл. 1. Микросхемы TC620

Наименование	Корпус	Предел измерения температуры
TC620CCOA	8-Pin SOIC	от 0°C до +70°C
TC620CCPA	8-Pin PDIP	от 0°C до +70°C
TC620CEOА	8-Pin SOIC	от -40°C до +85°C
TC620CEPA	8-Pin PDIP	от -40°C до +85°C
TC620CVOA	8-Pin SOIC	от -40°C до +125°C

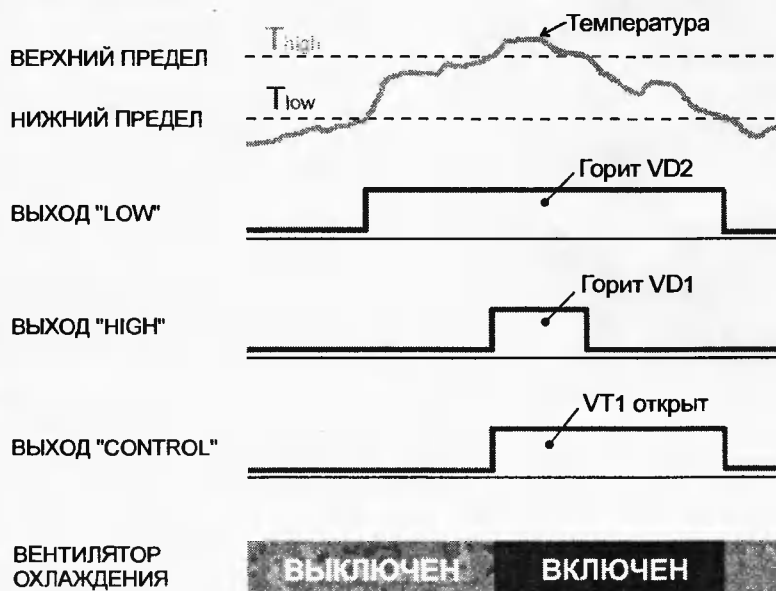


Рис. 3

каждый из которых программируется одним внешним резистором (R_{high} — это $R1$ и R_{low} — это $R2$). Для выходов LOW и HIGH активным является высокий уровень. Сигналы с этих выводов микросхемы в данном устройстве используются для управления светодиодами VD1 и VD2. Они служат для визуального контроля за состоянием

Табл.2. Соответствие величины контролируемой температуры номиналу внешнего резистора $R_{low/high}$

T, °C	25	26	27	28	29	30	31	32	33
R, кОм	112,6	113,4	114,2	115,0	115,8	116,6	117,5	118,3	119,1
T, °C	34	35	36	37	38	39	40	45	50
R, кОм	119,9	120,8	121,6	122,5	123,3	124,1	125,0	129,3	133,6

Рис. 4

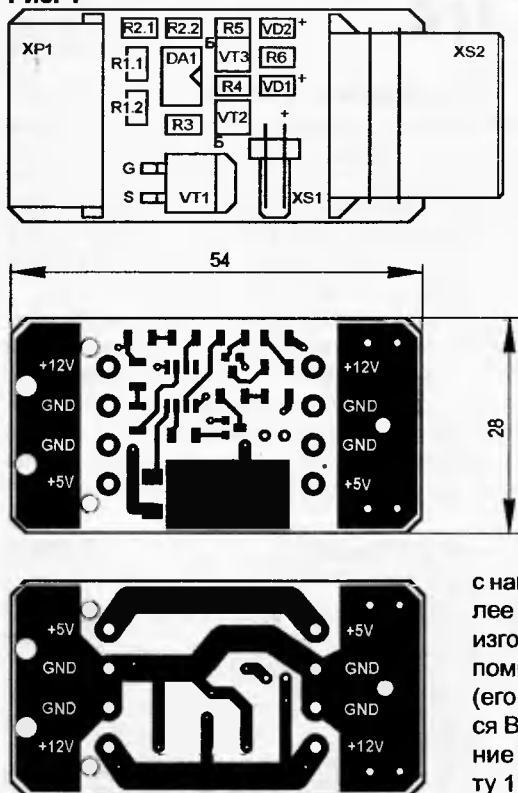
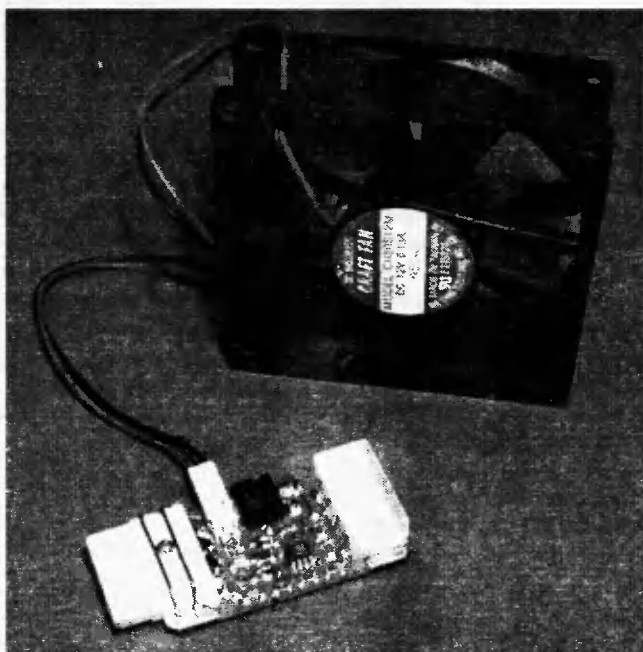


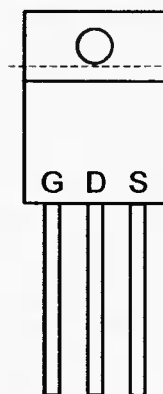
Рис. 5



только в переходные отверстия. В качестве VT1 можно применить транзистор NDT3055 (корпус SOT-223) или RFD3055 (корпус TO-251 или TO-252 (DPAK)). Также можно установить транзистор в корпусе TO-220 (KT-28), например, RFP3055 или MTP3055E, предварительно доработав корпус транзистора, как показано на рис.6. В терморегуляторе в качестве VT2 и VT3 можно применить

Рис. 6

Корпус
TO-220

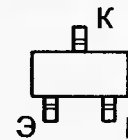


Линия удаления
лепестка крепления
транзистора
к радиатору

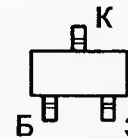
Корпус
DPAK



Корпус
SOT-23



Вариант 1



Вариант 2

любые маломощные п-р-п транзисторы в корпусе SOT-23 с напряжением коллектор-эмиттер более 20 В и током более 15 мА. При изготовлении печатной платы нужно помнить, что у транзистора KT3130 (его импортными аналогами являются BC846CR и BC847CR) расположение выводов соответствует варианту 1 (рис.6), а, например, у транзистора MMBT2222A — варианту 2. Поэтому, при использовании транзисторов с расположением выводов варианта 2, необходимо скорректировать печатную плату. Резисторы и светодиоды — типоразмера 1206. Применяемые разъемы: XP1 — TNP-4MR(5.08) угловой на плату; XS1 — WF-2R угловой на плату; XS2 — TNP-4F(5.08) прямой на кабель. В корпусе разъема TNP-4F между земляными выводами нужно просверлить отверстие диаметром 2,1 мм. С помощью винта M2 длиной 12 мм и гайки разъем крепится к плате. После этого, как показано на рис.5, разъем XS2 закрепляется двумя самодельными скобами, которые впаиваются в плату. Скобы можно изготовить из канцелярских скрепок. Ответная часть разъема XS1 (для подключения вентилятора) — HU-2.

Первое включение схемы желательно производить от какого-либо внешнего источника питания. Иначе, в случае наличия в схеме короткого замыкания, подключение терморегулятора к разъему блока питания компьютера может вызвать его повреждение. При правильном монтаже и исправных деталях устройство начинает работать сразу и в налаживании не нуждается.

В заключение, несколько советов по установке дополнительного вентилятора. Во-первых, воздушный поток от него не должен быть направлен на плату терморегулятора. Во-вторых, воздушный поток от дополнительного вентилятора не должен "бороться" с потоками стандартно установленных вентиляторов. В итоге такой "борьбы" в корпусе компьютера могут появиться "тепловые пробки" — области, где горячий воздух застаивается. В-третьих, по способу охлаждения вентиляторы делятся на вытяжные и нагнетающие. Нагнетающий вентилятор нужно использовать для охлаждения небольших областей или наиболее горячих компонентов, а вытяжной — для удаления горячего воздуха из корпуса компьютера. В-четвертых, вытяжной вентилятор устанавливается в верхней части корпуса компьютера, а нагнетающий — в нижней.

Хочется отметить, что микросхема TC620/TC621 может найти широкое применение в системах контроля температуры. Это, например, может быть контроль температуры в холодильнике, овощехранилище, в системе охлаждения двигателя и обогрева салона автомобиля и т.д.

Литература

1. Dual Trip Point Temperature Sensors. — DataSheet, Microchip Technology Inc., 2002.
2. TC620/TC621. — DataSheet, Microchip Technology Inc., 2002.



MAXI-EDITOR

Е.ИЛАСОВ,
г.Балашов,
Саратовской обл.

Одной из первых задач, которые приходится решать начинающим пользователям Sinclair-совместимых машин, является "обучение" своего компьютера общению на родном языке. Существует немало способов работы с символами национальных алфавитов. Самые распространенные из них — это добавление "недостающих" символов к уже имеющимся, "защитым" в системном ПЗУ, с помощью графики, определяемой пользователем (User Defined Graphic — UDG), или же полная замена всего используемого набора символов на альтернативный знакогенератор, размещаемый в свободной области оперативной памяти [1, 2]. У каждого из этих способов есть как свои достоинства, так и некоторые недостатки. Объединяет же их то, что и в том, и в другом случае новые символы приходится создавать самим пользователям (если только нет возможности работать с уже готовыми наборами символов, или они по каким-то причинам не подходят для применения).

Среди прикладного программного обеспечения встречается немало инструментов для облегчения такого рода творчества — как независимых утилит, зачастую "самодельных" [3], так и входящих в состав других, более мощных программных пакетов, как правило, графических редакторов [4]. Вниманию читателей предлагается еще одна разработка на эту тему.

Программа несколько необычна, и по своему замыслу заметно выделяется среди широко распространенного "софта" похожего назначения. Но прежде чем рассказать об этом подробнее, небольшое отступление.

Компьютеры системы Sinclair используют кодировку символов, в основном совпадающую в первой своей половине с кодовой таблицей ASCII, которая фактически являлась стандартом для персональных компьютерных систем начиная с 80-х годов и до наступления эры Windows (ей почти один к одному соответствует отечественная кодировка КОИ-7/ГО). А вот вторая половина "синклеровской" кодовой таблицы очень даже оригинальна — хранит в себе очертания символов блочной псевдографики, образы символов UDG, токены (ключевые слова) БЕЙСИКА и некоторые специальные символы. Всей этой коллекцией умело распоряжается процедура вывода на экран, находящаяся в системном ПЗУ BASIC-48.

Если же использовать независимую от ПЗУ процедуру вывода (например, описанную в [5]), то появляется возможность использовать по своему усмотрению не только 96 символов с кодами от CHR\$32 до CHR\$127, обычно применяемых для русификации, но и еще 128 символов с кодами от CHR\$128 до CHR\$255 включительно. Такой подход к решению задачи русификации позволяет без какого-либо вмешательства в схему компьютера полноценно работать с кириллицей, сохраняя при этом в полном объеме все знаки латинского алфавита. Это обстоятельство может быть немаловажным в практике работы компьютерных кружков, при самостоятельной разработке программ, "общающихся" с пользователем на национальном языке, либо использующих двуязычный, национально-латинский интерфейс.

Именно такую возможность — создание макси-знакогенератора — и предоставляет предлагаемая программа MAXI-EDITOR. Этот редактор макси-шрифта был состав-

лен одним из наших кружковцев и предназначен для работы в среде дисковой операционной системы TR-DOS. При его создании ставилась цель объять необъятное — совместить широкие возможности для редактирования символов с удобным сервисом, оставаясь при этом в пределах БЕЙСИКА. О том, насколько удалась такая попытка, судить прежде всего пользователям. Мы надеемся, что и у такого варианта найдутся свои приверженцы.

Управление программой организовано через двухуровневое псевдоменю, на самом деле представляющее собой два набора "горячих" клавиш. Первый из этих наборов дает возможность управлять дисковыми операциями, которые реализованы в этой программе (рис.1). Вторая группа клавиш позволяет редактировать символы вновь создаваемого или уже загруженного макси-шрифта (рис.2), который занимает

Рис. 1

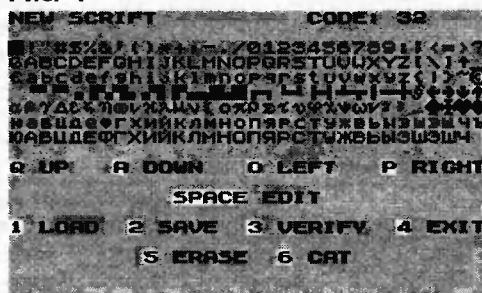
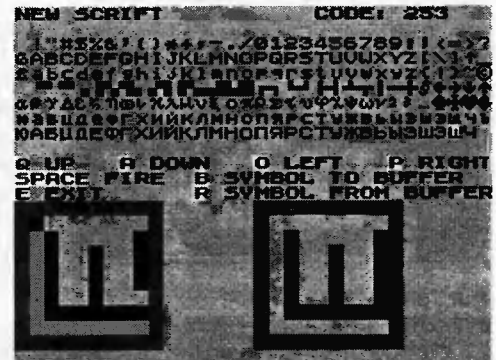


Рис. 2



теперь уже не 768 байтов, как традиционные фонты, а 1792 байта. Действия клавиш обеих групп, благодаря постоянным "подсказкам" на экране, вполне "прозрачны", а порядок работы с программой в целом достаточно наглядно отображен на приведенных иллюстрациях и в каких-либо особых комментариях не нуждается.

Отдельно стоит, пожалуй, отметить, что в макси-редакторе предусмотрена "приятная мелочь", почему-то ранее не встречавшаяся в программах аналогичного назначения, но значительно облегчающая жизнь дизайнерам шрифтов. Это не что иное как внутренний буфер размером в 8 байтов, куда можно на время поместить любой символ из текущей позиции, а затем при необходимости перенести сохраненный символ в другую выбранную позицию.

Листинг программы

"MAX_EDIT" (автостарт со строки 5)

```
1 DEF FN s(x,y,a,b,d,s$)=USR 63730
5 CLEAR 61169: RANDOMIZE USR 15619: REM : LOAD "max1" CODE
10 RANDOMIZE USR 15619: REM : LOAD "max2" CODE
11 GO SUB 9010
12 PAPER 7: INK 0: BORDER 7: CLS
15 INK 1: RANDOMIZE FN S(0,56,4,7,8,"MAX EDIT")
17 INK 2: RANDOMIZE FN S(16,120,2,2,8,"△Frolow S. V.")
18 IF INKEY$="" THEN GO TO 18
19 INK 0: CLS
20 CLS : LET ERROR=0: LET L=32
30 DIM b$(8,8): DIM e$(8,8): REM initialization
40 PRINT AT 0,0;"NEW SCRIPT": GO SUB 9100
50 PRINT AT 10,0;BRIGHT 1;"Q";BRIGHT 0;" UP ";
  BRIGHT 1;"A";BRIGHT 0;" DOWN ";BRIGHT 1;"O";
  BRIGHT 0;" LEFT ";BRIGHT 1;"P";BRIGHT 0;" RIGHT"
60 PRINT AT 12,1;BRIGHT 1;"SPACE";BRIGHT 0;" EDIT"
80 PRINT AT 14,0;BRIGHT 1;"1";BRIGHT 0;" LOAD ";
  BRIGHT 1;"2";BRIGHT 0;" SAVE ";BRIGHT 1;"3";
  BRIGHT 0;" VERIFY ";BRIGHT 1;"4";BRIGHT 0;" EXIT"
```



```
90 PRINT AT 16,9;BRIGHT 1;"5";BRIGHT 0;" ERASE ";
   BRIGHT 1;"6";BRIGHT 0;" CAT"
1000 REM
1005 LET Y=INT (L/32)-1: LET X=L-32*INT (L/32)
1007 LET L=32+32*Y+X: PRINT AT 0,20;" CODE: ";L;" ":
   POKE 23658,8: LET mem=22528+64+32*Y+X:
   POKE mem,128+PEEK mem
1010 LET S$=INKEY$: IF S$="" THEN GO TO 1010
1015 POKE mem,PEEK mem-128
1020 GO SUB 9200
1021 IF S$="Q" AND Y>0 THEN LET Y=Y-1
1022 IF S$="A" AND Y<6 THEN LET Y=Y+1
1023 IF S$="O" AND X>0 THEN LET X=X-1
1024 IF S$="P" AND X<31 THEN LET X=X+1
1030 IF S$="1" THEN GO SUB 6500
1040 IF S$="2" THEN GO SUB 2000
1050 IF S$="3" THEN GO SUB 3000
1060 IF S$="4" THEN PRINT #0;AT 1,0;FLASH 1;
   "ARE YOU SURE - PRESS Y TO YES !": PAUSE 0:
   LET S$=INKEY$: GO SUB 9200: INPUT "":
   IF S$="Y" THEN CLEAR 65367: RANDOMIZE USR 15619: REM : RUN
1070 IF S$="5" THEN GO SUB 7000
1080 IF S$="6" THEN GO SUB 7500: GO TO 40
1090 IF S$="" THEN LET L=32+32*Y+X: GO SUB 4000:
   POKE MEM1,PEEK MEM1-48
1100 IF S$="" THEN GO SUB 9000: RANDOMIZE FN S(8*(L-32*INT
   (L/32)),8*(1+INT (L/32)),1,1,8,CHR$ L): GO SUB 9010:
   GO SUB 7600: GO TO 50
1105 IF S$="1" AND error=0 THEN GO TO 40
1110 GO TO 1007
2000 INPUT "FILENAME ";F$: IF F$="" THEN GO TO 2000
2010 LET error=USR 15619: REM : SAVE F$ CODE 61938,1792
2015 IF error=4 THEN PRINT #0;AT 1,0;FLASH 1;
   " C A T A L O G   F U L L   ! ": BEEP 1.5,23:
   FOR i=1 TO 100: NEXT i: INPUT "": RETURN
2020 IF error=3 THEN PRINT #0;AT 1,0;FLASH 1;
   " D I S K   F U L L   ! ": BEEP 1.5,23:
   FOR i=1 TO 100: NEXT i: INPUT "": RETURN
2030 IF error=2 THEN PRINT #0;AT 1,0;
   "File exists. Erase - Y to YES !": PAUSE 0: LET S$=INKEY$:
   INPUT "": GO SUB 9200: IF S$="y" OR S$="Y" THEN GO SUB
   2100: GO TO 2010
2040 RETURN
2100 RANDOMIZE USR 15619: REM : ERASE F$ CODE
2110 RETURN
3000 INPUT "FILENAME ";F$: IF F$="" THEN GO TO 3000
3010 LET error=USR 15619: REM : VERIFY F$ CODE 61938,1792
3015 IF error=1 THEN PRINT #0;AT 1,0;FLASH 1;" NO FILE ! ":
   BEEP 1,23: FOR I=1 TO 100: NEXT I: INPUT "": GO TO 3030
3020 IF error THEN PRINT #0;AT 1,0;FLASH 1;" E R R O R   ! ":
   BEEP 1.5,23: FOR i=1 TO 100: NEXT i: INPUT "": GO TO 3030
3025 PRINT #0;AT 1,0;"Verification O. K.": FOR i=1 TO 250:
   NEXT i: INPUT ""
3030 RETURN
4000 REM EDITING !!!
4001 LET D$="": POKE 23658,8: GO SUB 7600: POKE MEM,PEEK MEM+48
4002 PRINT AT 10,0;BRIGHT 1;"Q";BRIGHT 0;" UP ";BRIGHT 1;"A";
   BRIGHT 0;" DOWN ";BRIGHT 1;"O";BRIGHT 0;" LEFT ";
   BRIGHT 1;"P";BRIGHT 0;" RIGHT"
4003 PRINT AT 11,0;BRIGHT 1;"SPACE";BRIGHT 0;" FIRE ";BRIGHT 1;
   "B";BRIGHT 0;" SYMBOL TO BUFFER"
4005 PRINT AT 12,0;BRIGHT 1;"E";BRIGHT 0;" EXIT ";BRIGHT 1;
   "R";BRIGHT 0;" SYMBOL FROM BUFFER": PRINT AT 13,0;PAPER 5;
   " EDIT ";AT 13,16;PAPER 2;" BUFFER ";#0;AT 0,0;
   PAPER 5;" ";AT 0,16;PAPER 2;" ":
   FOR I=14 TO 21: PRINT AT I,0;PAPER 5;" ";AT I,9;" ";
   AT I,16;PAPER 2;" ";AT I,25;" ";: NEXT I
4006 PRINT AT 0,20;" CODE: ";L
4007 FOR j=1 TO 8: LET o=PEEK (61938+8*(1-32)+j-1): FOR I=8 TO 1
   STEP -1: LET q=o-2*INT (o/2): LET E$(j,I)="П" AND q:
   LET o=INT (o/2): NEXT I: NEXT j
4008 LET X=1: LET Y=1
4010 FOR i=1 TO 8: PRINT AT 13+i,1;INK 1;e$(i);
   AT 13+i,17;b$(i);: NEXT i
4020 POKE 23658,8
4100 PRINT AT 13+Y,X;FLASH 1;INK 1;
   BRIGHT POINT (X*8,175-8*(13+Y));e$(y,x);:
   LET d$=INKEY$: IF d$="" THEN GO TO 4100
4110 GO SUB 9200
4120 PRINT AT 13+Y,X;INK 1;e$(y,x);
4130 IF D$="Q" AND Y>1 THEN LET Y=Y-1
4140 IF D$="A" AND Y<8 THEN LET Y=Y+1
```

```
4150 IF D$="O" AND X>1 THEN LET X=X-1
4160 IF D$="P" AND X<8 THEN LET X=X+1
4165 IF D$="E" THEN GO TO 6000
4166 IF D$="B" OR D$="b" THEN FOR i=1 TO 8: LET b$(i)=e$(i):
   NEXT i: GO TO 4010
4167 IF D$="r" OR D$="R" THEN FOR i=1 TO 8: LET e$(i)=b$(i):
   NEXT i: GO TO 4010
4170 IF D$<>" " THEN GO TO 4100
4180 IF e$(y,x)="П" THEN LET e$(y,x)=" ": GO TO 4200
4190 LET e$(y,x)="П"
4200 PRINT AT 13+Y,X;INK 1;e$(y,x);
4220 GO TO 4100
6000 LET MEM1=MEM: PRINT #0;AT 1,0;
   "SAVE SYMBOL TO CURRENT ? (Y=YES)"
6010 LET D$=INKEY$: IF D$="" THEN GO TO 6010
6020 GO SUB 9200: INPUT "": IF D$="Y" THEN FOR j=1 TO 8:
   LET o=0: FOR I=1 TO 8: LET o=o*2+(e$(j,i)="П"): NEXT i:
   POKE 61938+8*(L-32)+J-1,o: NEXT j: RETURN
6025 PRINT #0;AT 1,0;"PRESS N FOR AB ANDON SAVE."
6030 POKE 23658,8: LET Y=INT (L/32)-1: LET X=L-32*INT (L/32)
6040 LET MEM=22528+64+32*Y+X: POKE MEM,128+PEEK MEM
6045 PRINT AT 0,20;" CODE: ";L;" ":
6050 LET D$=INKEY$: IF D$="" THEN GO TO 6050
6060 GO SUB 9200
6070 POKE MEM,PEEK MEM-128
6080 IF D$="Q" AND Y>0 THEN LET Y=Y-1
6090 IF D$="A" AND Y<6 THEN LET Y=Y+1
6100 IF D$="O" AND X>0 THEN LET X=X-1
6110 IF D$="P" AND X<31 THEN LET X=X+1
6115 IF D$="N" THEN RETURN
6120 LET L=32+32*Y+X: IF D$="" THEN LET D$="Y": GO TO 6020
6130 GO TO 6040
6500 INPUT "FILENAME ";F$: LET ERROR=USR 15619: REM :
   LOAD F$ CODE 61938,1792
6510 IF ERROR=1 THEN PRINT #0;AT 1,0;FLASH 1;" NO FILE ! ":
   BEEP 1,23: FOR I=1 TO 100: NEXT I: INPUT ""
6520 RETURN
7000 INPUT "FILENAME ";F$: LET ERROR=USR 15619: REM :
   ERASE F$ CODE
7010 IF ERROR=1 THEN PRINT #0;AT 1,0;FLASH 1;" NO FILE ! ":
   BEEP 1,23: FOR I=1 TO 100: NEXT I: INPUT ""
7020 RETURN
7500 RANDOMIZE USR 15619: REM : CAT
7502 GO SUB 9200
7505 IF INKEY$="" THEN GO TO 7505
7507 CLS
7510 RETURN
7600 FOR I=10 TO 21: PRINT AT I,0;
   " ";: NEXT I: INPUT "":
   RETURN
9000 POKE 23606,242: POKE 23607,240: RETURN : REM new script
9010 POKE 23606,242: POKE 23607,237: RETURN : REM program script
9100 GO SUB 9000: FOR j=0 TO 6: FOR i=0 TO 31:
   RANDOMIZE FN s(i*8,16+j*8,1,1,8,CHR$ (32+j*32+i)): NEXT i:
   NEXT j: GO SUB 9010: RETURN
9200 BEEP .04,35: BEEP .08,23: RETURN : REM BEEP
9900 PRINT USR 15619: REM : ERASE "MAX EDIT"
9910 CLEAR : PRINT USR 15619: REM : SAVE "MAX EDIT" LINE 5
9920 PRINT USR 15619: REM : VERIFY "MAX EDIT"
9930 GO TO 5
```

По ходу работы этой программы, для вывода на экран расширенного набора символов произвольного размера, как раз и используется уже упомянутая выше машиннокодированная процедура "king_size", описанная в [5]. От опубликованного прототипа процедура отличается тем, что в данном случае она оформлена в виде отдельного кодового файла, и подгружается в выделенную область оперативной памяти (с десятичного адреса 63730) из основной программы.

Для тех пользователей, у которых нет возможности ознакомиться с публикацией [5], здесь приводятся краткие сведения о работе с этой машиннокодированной процедурой, позволяющие с успехом самостоятельно применять ее в собственных программах.

Управление процедурой организовано через функцию, определяемую пользователем —

DEF FN s(x,y,a,b,d,s\$)=USR 63730,

где переменные в скобках являются входными параметрами:



- s\$ — строка символов для печати на экране, допустимые коды выводимых символов — от CHR\$32 до CHR\$255;
- x — координата начала печати строки s\$ по горизонтали в пикселах. Может иметь значение от 0 (левый край экрана) до 255 (правый край);

- y — координата начала печати по вертикали, тоже в пикселах. 0 — верхний край экрана, 191 — нижний;

- a — коэффициент увеличения (масштаб символов) по горизонтали, от 1 до 32 раз. Символы, выходящие за правый край, дорисовываются слева;

- b — масштаб по вертикали, от 1 до 24 раз. Выход за пределы экрана не допускается, в этом случае практически гарантирована перезагрузка компьютера;

- d — самый интересный параметр — расстояние между символами в отмасштабированных пикселах.

Таким образом, вывод на экран, аналогичный стандартному, будет иметь место, если задать параметры a=1, b=1, d=8. Если задать d=7, то даже символами обычной ширины можно печатать на экране строки длиной 36 знаков. И, наконец, следует заметить, что в отличие от бейсиковского оператора PRINT, процедура "king_size" работает на всем экране, включая нижнее окно редактирования.

Также ниже приводится дамп (шестнадцатеричная распечатка) этого блока машинных кодов, который можно набрать и сохранить на внешнем носителе с помощью любого доступного шестнадцатеричного монитора.

Дамп кодового блока

<max1> <C> 63730,318

```
#F8F2: DD 2A 0B 5C 3E 80 21 2A    #F9B2: 20 DE 3A 26 5C 18 98 18
#F8FA: 5C 77 23 CB 3F 30 FA D5    #F9BA: E4 3A 8E 5C 5E EE FF 47
#F902: DD 5E 2C DD 5E 2D DD 6E    #F9C2: 3A 8D 5C A0 47 3A 28 5C
#F90A: 2E DD 66 2F 19 36 FF EB    #F9CA: E6 F8 6F 3A 27 5C FE C0
#F912: D1 7E 23 22 20 5C 6F 3C    #F9D2: 30 E5 1F 1F 1F E6 1F 67
#F91A: 20 10 DD 5E 2C DD 56 2D    #F9DA: CB 1C CB 1D CB 1C CB 1D
#F922: DD 6E 2E DD 66 2F 19 36    #F9E2: CB 1C CB 1D 3E 58 B4 67
#F92A: 22 C9 26 00 29 29 29 ED    #F9EA: 3A 8E 5C A6 B0 77 3A 27
```

```
#F932: 4B 36 5C 09 3E 08 32 24    #F9F2: 5C 47 E6 07 F6 40 67 78
#F93A: 5C DD 7E 0C 32 29 5C DD    #F9FA: 1F 1F 1F E6 18 B4 67 78
#F942: 7E 04 32 28 5C 3E 09 32    #FA02: 17 17 E6 E0 6F 3A 28 5C
#F94A: 25 5C 7E 23 22 22 5C 07    #FA0A: 47 1F 1F 1F E6 1F B5 6F
#F952: 32 26 5C 3A 25 5C 3D 20    #FA12: EB 21 2A 5C 78 E6 07 4F
#F95A: 30 3A 24 5C 3D 20 17 DD    #FA1A: 06 00 09 46 1A 21 26 5C
#F962: 7E 24 47 DD 7E 14 4F DD    #FA22: CB 46 28 04 B0 12 18 8F
#F96A: 7E 04 81 05 20 EC DD 77    #FA2A: 2F B0 2F 12 18 89
```

```
#F972: 04 2A 20 5C 18 9B 32 24
#F97A: 5C DD 7E 1C 47 3A 29 5C
#F982: 80 32 29 5C 2A 22 5C 18
#F98A: B6 32 25 5C DD 7E 14 47
#F992: 3A 29 5C 32 27 5C DD 7E
#F99A: 1C 4F C5 18 1C C1 3A 27
#F9A2: 5C 3C 32 27 5C 0D 20 F2
#F9AA: 3A 28 5C 3C 32 28 5C 05
```

Для вывода на экран текстовых сообщений в программе использован альтернативный подгружаемый знакогенератор, содержащий симпатичный "утолщенный" набор символов в "синклеровской" кодировке. В принципе, здесь можно применить любой подходящий шрифт с такой же кодировкой, либо обойтись стандартными символами, "защитыми" в ПЗУ. Но если есть желание после набора и отладки получить программу, максимально близкую к авторскому виду, то можно опять воспользоваться шестнадцатеричным монитором и повторить шрифт, использованный в оригинале макси-редактора:

<max2> <C> 61170,768

```
#EEF2: 00 00 00 00 00 00 00 00    #F072: 00 7E 63 63 7E 60 60 00
#EEFA: 00 18 18 18 18 00 18 00    #F07A: 00 3E 63 63 7B 6E 3B 00
```

```
#EF02: 00 36 36 00 00 00 00 00    #F082: 00 7E 63 63 7E 66 63 00
#EF0A: 00 36 7F 36 36 7F 36 00    #F08A: 00 3E 60 3E 03 03 7E 00
#EF12: 00 3F 6C 3E 1B 1B 7E 00    #F092: 00 7E 18 18 18 18 18 00
#EF1A: 00 63 66 0C 18 33 63 00    #F09A: 00 63 63 63 63 63 3E 00
#EF22: 00 3C 66 3C 67 66 3F 00    #F0A2: 00 63 63 63 63 36 1C 00
#EF2A: 00 1C 1C 0C 18 00 00 00    #F0AA: 00 63 63 6B 6B 6B 36 00
```

```
#EF32: 00 0C 18 18 18 18 0C 00    #F0B2: 00 63 36 1C 36 63 63 00
#EF3A: 00 18 0C 0C 0C 0C 18 00    #F0BA: 00 66 66 66 3C 18 18 00
#EF42: 00 00 36 1C 7F 1C 36 00    #F0C2: 00 7F 63 0E 38 63 7F 00
#EF4A: 00 00 18 18 7E 18 18 00    #F0CA: 00 1C 18 18 18 18 1C 00
#EF52: 00 00 00 00 1C 1C 0C 18    #F0D2: 00 60 30 18 0C 06 03 00
#EF5A: 00 00 00 7E 00 00 00 00    #F0DA: 00 1C 0C 0C 0C 0C 1C 00
#EF62: 00 00 00 00 18 18 00 00    #F0E2: 00 18 3C 7E 18 18 18 00
#EF6A: 00 03 06 0C 18 30 60 00    #F0EA: 00 00 00 00 00 00 00 FF
```

```
#EF72: 00 3E 63 6F 7B 63 3E 00    #F0F2: 00 1E 33 78 30 30 7F 00
#EF7A: 00 18 38 18 18 18 3C 00    #F0FA: 00 00 3C 06 3E 66 3E 00
#EF82: 00 3E 63 03 3E 60 7F 00    #F102: 00 60 60 7C 66 66 7C 00
#EF8A: 00 3E 63 0E 03 63 3E 00    #F10A: 00 00 3E 60 60 60 3E 00
#EF92: 00 0E 1E 36 66 7F 06 00    #F112: 00 06 06 3E 66 66 3E 00
#EF9A: 00 7F 60 3E 03 63 3E 00    #F11A: 00 00 3C 66 7E 60 3C 00
#EFA2: 00 3E 60 7E 63 63 3E 00    #F122: 00 0C 18 3C 18 18 18 00
#EFAA: 00 7F 63 06 0C 18 18 00    #F12A: 00 00 3E 66 66 3E 06 3C
```

```
#EFB2: 00 3E 63 3E 63 63 3E 00    #F132: 00 60 60 7C 66 66 66 00
#EFBA: 00 3E 63 63 3F 03 3E 00    #F13A: 00 18 00 38 18 18 3C 00
#EFC2: 00 00 18 18 00 18 18 00    #F142: 00 0C 00 0C 0C 6C 38 00
#EFCA: 00 18 18 00 18 18 08 10    #F14A: 00 60 66 6C 78 6C 66 00
#EFD2: 00 00 0C 18 30 18 0C 00    #F152: 00 38 18 18 18 18 3C 00
#EFDA: 00 00 00 7E 00 7E 00 00    #F15A: 00 00 7E 6B 6B 6B 6B 00
#EFE2: 00 00 30 18 0C 18 30 00    #F162: 00 00 7C 66 66 66 66 00
#EFEA: 00 3E 63 03 0E 00 0C 00    #F16A: 00 00 3C 66 66 66 3C 00
```

```
#EFF2: 00 3E 67 6B 6E 60 3E 00    #F172: 00 00 7C 66 66 7C 60 60
#EFAA: 00 3E 63 63 7F 63 63 00    #F17A: 00 00 7C 66 66 3E 06 06
#F002: 00 7E 63 7E 63 63 7E 00    #F182: 00 00 3E 66 60 60 60 06
#F00A: 00 3E 63 60 60 63 3E 00    #F18A: 00 00 3C 60 3C 06 7C 00
#F012: 00 7E 63 63 63 63 7E 00    #F192: 00 18 3C 18 18 18 0C 00
#F01A: 00 7F 60 7E 60 60 7F 00    #F19A: 00 00 66 66 66 66 3C 00
#F022: 00 7F 60 7E 60 60 60 00    #F1A2: 00 00 66 66 66 3C 18 00
#F02A: 00 3E 63 60 67 63 3E 00    #F1AA: 00 00 63 6B 6B 6B 36 00
```

```
#F032: 00 63 63 7F 63 63 63 00    #F1B2: 00 00 66 3C 18 3C 66 00
#F03A: 00 3C 18 18 18 18 3C 00    #F1BA: 00 00 66 66 66 3E 06 3C
#F042: 00 1E 0C 0C 0C 6C 38 00    #F1C2: 00 00 7E 06 3C 60 7E 00
#F04A: 00 66 6C 78 6C 66 63 00    #F1CA: 00 0C 18 30 18 18 0C 00
#F052: 00 60 60 60 60 63 7F 00    #F1D2: 00 18 18 18 18 18 18 00
#F05A: 00 63 77 7F 6B 63 63 00    #F1DA: 00 30 18 0C 18 18 30 00
#F062: 00 63 73 7B 6F 67 63 00    #F1E2: 00 3B 6E 00 00 00 00 00
#F06A: 00 3E 63 63 63 63 3E 00    #F1EA: 3E 63 6F 7B 6F 63 3E 00
```

Что же касается предпочтительной кодировки для проектируемого шрифта, то совершенно не обязательно "изобретать велосипед". Для традиционного способа русификации (подменой знакогенератора) чаще всего используются кодировки по стандарту КОИ-7/Н2 (НС — совмещенная латинско-русская), КОИ-7/Н1 (НР — только кириллица) или предпочитаемая очень многими синклеристами "вывернутая наизнанку" КОИ-7/Н1 (большие русские буквы на месте больших латинских, малые — на месте малых). При редактировании макси-фонта также можно воспользоваться давно известными и широко распространенными кодировками, к примеру, очень часто применяемой на просторах бывшего Союза кодировкой КОИ-8. Несмотря на то что она досталась нам в наследство еще от советских времен, ею и по сей день активно пользуется значительная часть русскоязычных web-серверов в Internet.

Также можно использовать кодировки, применяемые на персональных компьютерах клона IBM. Чаще всего это альтернативная кодировка, модифицированная альтернативная или кодовая страница (Code Page) DOS 866. Все это, по большому счету, одно и то же, так как они отличаются друг от друга лишь несколькими, редко используемыми символами [6]. Для отечественных пользователей-синклеристов они удобны тем, что содержат в себе наборы разнообразных псев-



дографических символов, с помощью которых можно изображать прямо в тексте всевозможные рамки, рисунки и даже довольно сложные электронные схемы. А желающие выглядеть идущими в ногу с прогрессом могут воспользоваться "русской" версией ANSI-кодировки, применяемой в различных версиях операционной системы Windows. Вот только удобную псевдографику там уже не найти...

На обоих приведенных рисунках показан вид программы с уже готовым макси-шрифтом. Подобный макси-фонт может оказаться достаточно удобным для практических целей, поскольку содержит самые разные символы, практически, на все случаи жизни. Но его кодировка далека от каких-либо стандартов, и этот набор приведен здесь в основном для демонстрации возможностей программы-редактора.

Hard'n'soft:

- KAY-1024 turbo, (c)Nemo, 2002;
- Basic-48, (c) Sinclair Research Ltd, 1982;
- iS-DOS 2000 (c)IskraSoft, 1992-2003;
- iS-Editor, (c)IskraSoft, 1996-2002;

КОМПЬЮТЕРНЫЕ ХРОНИКИ

BestView 2.18

Первая версия BestView (1.0) увидела свет около шести лет назад — 4 февраля 1998 г. Потом были версии 2.0, 2.1, ..., и, наконец, 29 января 2004 г. была выпущена версия 2.18 (рис.1) — получается, по счету она двадцатая. Вот такой юбилей :-).



Рис. 1

Что же нового по сравнению с прошлой версией появилось в этой свободно распространяемой универсальной программе для просмотра и запуска файлов на "ZX-Spectrum"? Во-первых, конечно, возмож-

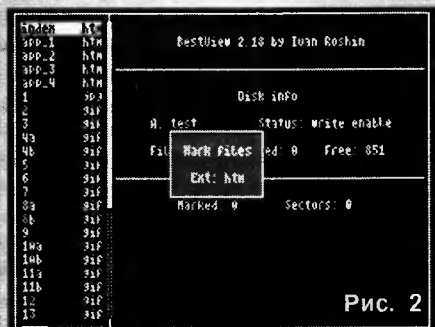


Рис. 2

ность пометки файлов, в том числе по маске расширения (рис.2). Количество помеченных файлов и их общий объем в секторах отображаются в правой панели, когда она находится в режиме показа информации о диске. Обычно в программах-файловых оболочках (и на "ZX-Spectrum", и на PC), когда помечается текущий файл, курсор автоматически перемещается к следующему файлу, и если удерживать нажатой клавишу пометки, то срабатывает автоповтор, и помечаются следующие файлы в списке. Но есть одно неудобство — если надо пометить группу файлов, начиная с какого-то и до конца списка, то, когда курсор доходит до последнего файла в списке, и этот файл помечается, при дальнейшем срабатывании автоповтора курсор уже не может сдвинуться дальше, и происходит снятие пометки с последнего файла. Потом этот файл опять помечается, и так далее, пока не отпустишь клавишу. Из-за высокой частоты автоповтора бывает трудно отпустить клавишу имен-

- bas_is.com, dump.com, is_ibm+.com, scr.com, srcopy.com, scr_bmp.com, etc.

Литература

1. Программирование в машинных кодах. Часть 3. Справочник по программированию в машинных кодах. Издание 2-е, исправленное и дополненное. — М.: МКП Инфорком, 1994, С.187...192.
2. Ларченко А.А., Родионов Н.Ю. ZX Spectrum и TR-DOS для пользователей и программистов. — СПб.: Питер, 1994, С.131...134.
3. Виктор Д.И. FONT EDITOR. — Радиолучитель, 1992, №5, С.10.
4. Как работать с графическим редактором ArtStudio. Методическое пособие для пользователей ПЭВМ "Спектр". — М.: Island, 1993-1994.
5. Илясов Е.В. REM? REM... REM!!! — Радиолучитель. Ваш компьютер, 2001. №2, С.36.
6. Фигурнов В.А. IBM PC для пользователя. Краткий курс. — М.: "Инфра-М", 1997, С.120...123, 368.

но в тот момент, когда последний файл оказывается помечен. Так вот, при реализации пометки файлов в BestView я предусмотрел корректную обработку такой ситуации — когда курсор доходит до последнего файла в списке, и этот файл помечается, программа отключает автоповтор, пока пользователь не отпустит клавишу.

Продолжу рассказ о нововведениях. Пометка файлов была добавлена, конечно, не просто так :-). Теперь можно удалять файлы не по одному, а сразу все помеченные (рис.3), причем удаление даже нескольких файлов происходит быстрее, чем раньше происходило удаление только одного файла.

Перед удалением файлов производится проверка файловой структуры диска. При обнаружении ошибок программа выводит соответствующее сообщение и не удаляет файлы с такого диска, чтобы не добавить новые ошибки.

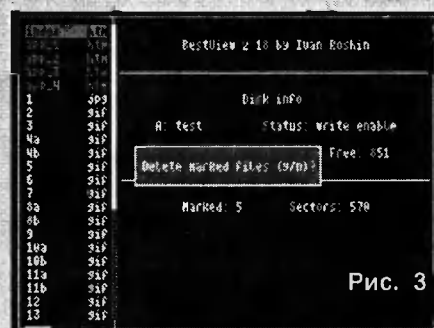


Рис. 3

Существенно ускорено автоопределение кодировки текста (в следующем номере журнала в дополнении к моей статье [1] вы сможете прочитать об оптимизированной версии соответствующей ассемблерной процедуры).

При всех нововведениях размер исполняемого файла программы вырос всего на полкилобайта.

BestView 2.18 и другие мои программы можно скачать с моей web-страницы — <http://www.ivr.da.ru>. Если вы по какой-либо причине не можете это сделать, напишите мне, и я вышлю BestView по E-mail. Также при желании вы можете получать по E-mail сообщения о выходе новых версий BestView.

Литература

1. И.Рощин. Автоматическое определение кодировки текста — 2. — Радиомир. Ваш компьютер, 2002, №№5, 6.

И.РОШИН,
г.Москва,

Fido: 2:5020/689.53

ZXNet: 500:95/462.53

E-mail: bestview@mtu-net.ru

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



SpellForse: The Order of Dawn

А.ВЕНДИЛОВСКИЙ,
г.Минск.

...Сознание медленно возвращалось, рассыпая перед ним обрывки неясных картин и голосов, смутно казавшихся ему знакомыми. Черный океан безвременья расступился, и он понял, что лежит на траве, в черных от копоти доспехах, которые теперь можно было спокойно выбросить, так как ни один кузнец не взялся бы за их ремонт, а от меча в руке вообще осталась только рукоять. Зато вторая рука изо всех сил сжимала старинный и явно магический фолиант. Он по-прежнему ничего не помнил — ни кто он такой, ни как сюда попал. Ощущение, что он здесь не один, накатило волной, и, не позволяя сознанию отчебучить какой-нибудь фортель, он осторожно отпустил ставшую ненужной рукоять меча и незаметно потянулся к спрятанному стилету.

— Я рад, что ты очнулся, — голос был старческий. Он и принадлежал старику с большим посохом, сидевшему напротив и внимательно его рассматривающему. Хотя старик и казался безобидным, внутреннее чутье подсказывало, что тот вполне может оказаться магом, с которым справиться в бою будет очень сложно, особенно в его нынешнем состоянии.

— Кто я? — неожиданно для себя спросил он старика.

— Ты? Ты Воин Руны, Сила Руны, Раб Руны — можешь называть себя как хочешь. Некоторые считают тебя Избранным, некоторые верят, что ты принесешь в мир спасение, другие — что только смерть. Вижу, что после перехода у тебя снова амнезия, но не волнуйся, это очень скоро пройдет.

— Что такое руна? — воспоминания опять рвались наружу. ...Огромные каменные монументы, на алтаре перед которыми яркими сполохами вырывалась магия, лица, звуки...

— Давным-давно, когда магия была подвластна людям, кроме стихийной, мысленной и некромантской, существовала еще одна ее разновидность. Ходят легенды, что это произошло после какой-то страшной войны, тысячу лет назад. Маги этого ордена создавали Руны. Зачем, толком никто не знает. Говорят, они боялись грядущего и искали пути к спасению. Сильнейшие воины и маги заключали с ними договор, чтобы найти путь к бессмертию. Они впечатывали сущность человека в кристалл с Волшебной Руной, и Руна считалась неразрушимой и вечной. Но за все нужно было платить. Пла-

той было вечное служение всякому, кто призовет человека руны на алтарь. Вас нельзя до конца уничтожить. Пока сильна твоя руна, ты будешь воскрешаться по воле ближайшего рунного камня. А если сила иссякнет, то ты вновь превратишься в маленький камешек, пока следующая твой владыка не положит его на Постамент Героев и не призовет тебя из Вечного Нигде. ...А потом пришел Великий Разлом, тринадцать величайших магов этого мира чего-то не поделили и устроили величайшее бедствие...



...Армии стояли друг против друга, на фоне возвышающейся к небесам башни магов, чья вершина была окутана всполохами молний. Бой барабанов, крики ярости — армия людей стояла совсем рядом с армией орков, чуть дальше полета стрелы. Воздух был насыщен ненавистью, яростью и страхом. Все ждали только сигнала, чтобы вцепиться друг другу в плотки. С небесной вышины на них взирал маг, укравший свое лицо во тьме сутаны. Он воздел руки к небу, и серебристые молнии разорвали небеса над башней. Даже на таком расстоянии ветер донес до него яростный вопль атакующих армий. Зеленая волна орков врезалась в блестящие ряды людей, звон стали, ударяющей о сталь, мог перекрыть только крикумирающего. Зеленая трава поля превратилась в грязное месиво, в которое были втопты и мертвые, и умирающие. Молодой рыцарь упал с коня и не смог подняться, он отчаянно барахтался, но сотни ног сражающихся просто втоптали его в это жуткое месиво, которое хлынуло через отверстие забрала в рот и нос, убивая также же надежно, как и сталь. Тучи стрел как с одной, так и с другой стороны закрыли небо, разя всех без разбора и пощады, пробивая шкуры и кожаные доспехи орков и стальную броню людей, выкашивая в их рядах целые

просеки, которые, несмотря на крики и стоны поверженных, тут же заполнялись новыми сражающимися. Здесь не было иных правил, кроме двух — выживи сам и убей врага. Многие, доведенные боем до фанатизма, рубили все, что попадалось им на глаза, не разбирая, кому принадлежат эти руки и головы — врагам или друзьям. По полю прекатывались удушливые волны страха, что следующим, кто упадет на окровавленную землю, крича или скуля от боли, прижимая к себе обрубок руки или ноги, или вообще со вспоротым брюхом, будешь ты сам. Всадник смерти правил тут свой бал, и Ад следовал за ним.

Некромант окинул взором сражающихся и усмехнулся. — С днем твоего возрождения, мой Повелитель! Теперь уже никто не сможет нас остановить! Не так ли, друг мой? — он плавно развернулся к человеку, шагнувшему на смотровую площадку башни через сияющий портал времени. — Ты тоже решил поприветствовать нашего Господина? Ты чувствуешь, как пробуждаются его силы в земных глубинах? Мы ждали этого столько лет! Теперь мы будем стоять на вершине власти вечно!

— Ты хоть понимаешь, что творишь, некромант? Его пробуждение означает лишь воцарение Хаоса и разрушения этого мира, он несет только гибель и смерть! После его пробуждения народы обратятся в прах, а живые будут завидовать мертвым. Ты этого жаждешь?

— Люди? — некромант слегка изумленно и иронично посмотрел на мага — чего стоит горстка смертных, чей удел быть нашими рабами, по сравнению с Вечностью и властью? Или, может быть, ты возомнил, что сможешь остановить меня? Ты забываешь, старик — смерть всегда побеждает жизнь, и теперь пришел ее час.

В этот миг из глубин земных недр раздался звук, напоминающий глухой стон, и незыблемая земная твердь зашаталась, раскрывая огненные глубины. Оттуда вылетело несколько десятков огненных шаров, с ревом устремившихся в небо.

— Закончим наш диспут цивилизованно, — сквозь зубы процедил некромант, в его руке вспыхнул небольшой комок мертво-зеленого огня, — твоей смертью! — и изо всех сил швырнул огонь в мага.

На поле битвы царил абсолютный хаос — разломы прошли прямо через армии сражающихся, и многие нашли свой последний приют в объятиях раскаленной магмы. Вырывавшиеся огненные шары



падали на несчастных, ни на земле, ни в воздухе спасения от них не было. Многие обращали свой взор к Башне, надеясь на помощь. Им было невдомек, что там, на вершине, среди облаков, два мага ведут свой, не менее смертоносный поединок.

— Глупец, — тихо промолвил некромант, когда фаербол пробил защиту белого мага и отбросил того к стене, — ты проиграл, святоша! — С его пальцев заструились молнии и окутали белого мага, который закричал в смертельной муке. И когда, казалось, исход этого поединка был предрешен, огромный пылающий шар, вырвавшийся из глубин адских расщелин, пронзив небесный покров облаков, ударил прямо в вершину Башни.

Пол, покрытый магическими символами и узорами на черно-белых плитках, задрожал и стал медленно, но неуклонно обваливаться вниз. Оба чародея, не стовариваясь, посмотрели на гримуарий, лежавший на полу, и бросились к нему. В этот самый миг Башня вздрогнула, и платформа, разломившись на несколько частей, полетела вниз, прихватив с собой книгу и сражающихся магов.

Уворачиваясь от пролетающих мимо обломков стен и колонн, перепрыгивая с одной падающей глыбы на другую, они продолжали сражаться, обмениваясь ударами огня и молний. Далеко внизу кровавой звездой пылал огонь, в котором проигравший должен будет окончить свой полет. С каждым прыжком они старались подобраться к гримуарию, воздух развевал их одежды и свистел в ушах. Белый маг рухнул грудью на остаток пола, и когда казалось, его пальцы уже схватили книгу, его интуиция заставила его тело отпрыгнуть в сторону.

Буквально через долю секунды туда приземлился некромант, и его кулак ударил в то место, где, казалось, только что находилась голова чародея. Книга сорвалась с ненадежной опоры и понеслась к ревущему пламени. Больше не обращая внимания на своего противника, маг последовал за ней. Некромант тоже заметил, что конец их путешествия уже близок, а книга могла даровать ему спасение, и последовал за своим врагом. Они летели практически рука об руку, в тщетной попытке дотянуться до столь нужных страниц. И когда до пламени оставалось несколько мгновений полета, произошло сразу множество вещей. Практически прямо над магмой появился еще один временной портал, из которого появился высокий статный воин, на лице которого читалось изумление от увиденной

картины. Маг успел ухватить книгу и швырнуть ее в воина, прокричав, чтобы тот спался. Вслед за книгой ринулось и пламя, прямо во внутрь сворачивающегося портала, когда прогремел оглушительный взрыв...

— Мой друг очень долго искал именно твою руну, начитавшись каких-то пророчеств об Избранном. Он по-прежнему считал себя виновным в Расколе и во всех бедствиях нашей эпохи.

— И что же с ним стало? — спросил воин. Внезапно сцена схватки над лавой промелькнула перед его глазами. Видимо, это столь отчетливо отразилось на его лице, что старик, внимательно рассматривавший его, лишь горестно вздохнул.

— Он обязательно просил передать тебе три вещи — стань свободным от Руны, спаси наш народ и... береги книгу!

Воин поднялся, он уже отчетливо видел, где воздвигнутся стены новых зданий, мысленно представлял будущие военные вылазки и знал, где его армия нанесет первый удар по неприятелю.

— Время приниматься за работу, — тихо промолвил он.



Последние несколько лет на прилавках постоянно появляются игры, пестрящие громкими рекламными слоганами — “стратегия с элементами РПГ” или “РПГ с элементами RTS”, в которых данные “элементы” всегда оставались на весьма зачаточном уровне. По сути, они абсолютно не влияли на прохождение игры, а подчас лишь создавали трудности для игрока. И только теперь нам представилась возможность попробовать свои силы в игре, где два таких жанра как RTS и РПГ представлены в полном объеме. Я бы даже сказал, что именно таким общали некогда сделать третий Варкрафт. Конечно, создать нечто подобное игральным, это просто титаническая работа — ведь от малейшего “перекаса” весь геймплей может быть убит напрочь раз и навсегда. С одной стороны — две час-

ти взаимно дополняют друг друга. С другой — игрок не может пройти всю игру “по старинке”, делая основной упор лишь на один из жанров, так как это неминуемо приведет его к поражению.

Начнем наш анализ с движка игры. В первый момент находишься под впечатлением качества графики — проработаны мельчайшие детали, видно, что даже на обычную елочку (я уже не буду говорить о людях) потрачено огромное количество полигонов. Если приглядеться, то можно заметить, что каждая ваша постройка имеет несколько источников освещения с динамически изменяемыми тенями!

Как и в любой подобной игре, камера подвижна, но мало кто может себе представить, что вид “от бога” или, как его еще называют, “три четверти” можно превратить в... экшен от третьего лица, закрепив камеру за спиной главного героя! При этом графика в игре легко может поспорить со многими экшенами, в массовом порядке разрабатываемыми на прилавки. Вероятно, разработчики хотели внести в игру еще и жанр экшена от третьего лица (a-la Лара Крофт), но просто “не потянули” (при переходе на такую точку обзора становятся активными дополнительные клавиши управления героем, известные всем любителям хороших перестрелок ;).

Но за все нужно платить, и платой за такую красоту становятся неумные системные ресурсы. Насладиться всей полнотой палитры смогут лишь обладатели самых последних новинок компьютерной техники, остальным придется заняться “пожертвованием” детализацией в угоду быстродействию.

Теперь давайте рассмотрим несколько особенностей, которые первыми бросаются в глаза. Я, честно говоря, так и не понял, почему создатели ими так гордятся. Первое — скроллинг карты. Привычное перемещение за мышкой здесь напрочь отсутствует, его приходится делать специальными управляющими клавишами. Второе — положение камеры. Чтобы держать под контролем стратегическую часть игры, мы привычно устанавливаем уже упомянутый выше вид “три четверти” (кстати, это очень хорошо помогает, когда вам требуется провести своего аватара из одного угла карты в другой). Но как только начнется битва, будьте готовы, что вам придется собственноручно сменить десяток положений камеры, чтобы добиться оптимального результата. Дело в том, что при этом размер персонажей на экране становится столь маленьким, что попасть в быстро суетящейся толпе в какого-то отдель-



ного противника очень сложно, хотя и очень надо. Тут еще один отход от обычно принятых правил. Как делаешь обычно? Выбрал своих воинов, кликнул на цель, и смотришь на результат. Здесь все наоборот — выбираешь противника, потом кликаешь по своему бойцу, выбираешь "метод воздействия" и затем смотришь на полученный результат. Вот и получается, что нужно и прицелиться, и общую картину боя не потерять.

Несколько слов об интерфейсе — он интуитивен и очень понятен. На его освоение тратится не более пары минут, при условии, конечно, что вы хоть раз в жизни играли в стратегию реального времени ;). Должен заметить, что вся необходимая информация находится на экране перед взглядом игрока, но при этом не загромождает игровое поле.

Теперь об игровом процессе. Игрок имеет своего аватара, вокруг которого и развивается сюжет. В начале игры мы имеем стандартную панель генерации героя. Правда, рассчитывать на большой выбор не приходится, играем только за расу людей, со всеми вытекающими отсюда особенностями. Как и в любой другой "ролевушке", изначально мы имеем некоторое количество очков, которые

распределяем между параметрами героя, определяя, по какому пути развития он пойдет. Я не стану долго перечислять слоты, которые мы можем заполнить амуницией и заклинаниями, просто скажу, что тот, кто играл в первый Дьябло, найдет здесь очень много знакомого. Выполняем квесты и уничтожаем врагов, за что получаем такую дорожку каждому ролевика "экспу". Время от времени уровень героя поднимается, ему становятся доступны новые умения, и он с новыми силами бросается на супостатов. А героя придется прокачивать, и даже серьезно прокачивать, выполняя побочные квесты, так как любимые методы "танкового ролака", принятые в RTS, здесь вам не помогут. Ради сохранения баланса игры, количество бойцов вашей армии довольно ограничено, а вот противник давно отвык ограничивать себя в чем-либо. Так что, без серьезной помощи со стороны вашего персонажа, от вашей армии очень быстро останутся только рожки да ножки. К особенностям RTS я бы отнес и то, что всех своих подопечных мы получаем не из казарм, как привыкли в других играх, а из особого Постаментов, которым мы можем только воспользоваться (в том смысле, что сами построить его не в состоя-

нии). Так что любое возведение базы вольно или невольно получается возле одного из таких Постаментов. Благо, что все необходимые ресурсы тоже, как правило, находятся под боком. И не забывайте, что этот Постамент есть первейшая цель противника, и если он его разрушит, на вашем лагере можно ставить большой жирный крест. Кстати, в связи с ограниченным количеством бойцов в вашем отряде, вы должны четко представлять себе его баланс — сколько необходимо меченосцев, сколько лучников, сколько лекарей. К тому же, так как постройка юнитов идет из одного здания, этот процесс часто бывает достаточно длительным, а вот противники ждать не будут. Рекомендую всегда проверять, не могут ли NPS, находящиеся на одной с вами карте, помочь вам ресурсами или людьми — их помощь всегда будет кстати.

Вывод — очень занимательная и захватывающая игра. Несмотря на некоторые ее недочеты, уверен, что вы проведете множество вечеров, помогая своему герою пройти весь его нелегкий путь спасителя человечества :).

Игра предоставлена интернет-магазином "MediaCraft" www.mediacraft.shop.by

Lionheart: Legacy of the crusader

М.ВАЛЬПА,

E-mail: valpa@vpost.ru

Название игры уже манит ее купить, а титул "лучшей RPG года" точно разрешит все сомнения по поводу покупки.

Итак, перед вами игра, сломавшая барьер между историей и фантазией, реальностью и выдумкой, былью и небылью. В игровом мире Lionheart'a действия происходят в 1536 году от Рождества Христова, в Испании. Но отличие в том, что Святая Инквизиция, свирепствовавшая в то время, охотится не на мнимых, а на настоящих колдунов, некромантов, ведьм, которые появились во время магического взрыва, а также на гоблинов, огров, гигантов, появившихся на Земле вместе с людьми.

Магический взрыв произошел за несколько десятков лет до начала действий игры, из-за неудачного эксперимента Инквизиции, вызвавшего появление магии и нескольких новых рас на Земле. К 1536 году Инквизиции удалось уничтожить большинство магов и ведьм в Европе, остатки которых прячутся в лесах и маленьких деревушках.

Вот на фоне всех этих действий и появляется ваш персонаж, являющийся потомком Ричарда Львиное Сердце, но не догадывающийся об этом. Причем появляется в тюрьме работорговцев, но быстро выбирается оттуда и телепортируется в Барселону, где встречает своего будущего наставника — не кого иного как Леонардо да Винчи. И это еще не все знаменитости, которых вы встретите в игре — вы можете помочь Шекспиру вернуть

его музу, вдохновляющую писать стихи; спасти душу Жанны Д'арк; помочь Кортесу в финансировании путешествия по поиску Америки; расспросить Нострадамуса о его пророчествах и так далее. Потом, по сюжету, вы узнаете, что идет война между Испанией и Англией, из-за того что английская королева дала приют всем магам, на которых охотится Инквизиция. И вам по ряду основных сюжетных заданий даже придется непосредственно участвовать в этой войне.

Игра имеет очень оригинальный главный сюжет и дает возможность проходить его за разные компании: за Инквизицию, за орден рыцарей темпларов, за орден Салладина и за волшебников (эту гильдию сложнее всего найти, но и проходить сюжет за нее наиболее интересно). От того, за какую компанию вы будете проходить игру, будет зависеть и ваше звание. К примеру, если вы будете играть за волшебников, то сначала получите ранг ученика волшебника, потом волшебника-умельца и так далее. Помимо этого, в игре есть мелкие фракции типа гильдии воров или нищих, и просто "гора" вторичных квестов, которые настоятельно советуем пройти каждому, так как за это вы получите много опыта, полезных вещей и повышений к характеристикам.





Кстати, разработчики "Lionheart" придумали свой, оригинальный интерфейс игры, а также систему повышения характеристик. Выглядит она следующим образом. Вначале имеется пять основных школ: боевая школа, школа воровства, школа божественной магии, школа волшебства и школа шаманства. При этом школа боя разделится еще на пять дисциплин, а школы магии и воровства — на четыре дисциплины, в каждой из которых есть по пять заклинаний (вы удивитесь, но этого за глаза хватает). Теперь будем перечислять и дополнять ;).

В школе боя все очень просто. Здесь находятся дисциплины кулачного боя, владения одноручным оружием (булавами, саблями, одноручными мечами), владения двуручным оружием (боевыми молотами и топорами, двуручными мечами), владения стрелковым оружием (луками, арбалетами) и мастерства уклонения. К сожалению, сколько я ни искал, посохов так и не нашел — похоже, разработчики забыли продумать такую неотъемлемую от RPG вещь. Кстати, за мага советую "раскачивать" дисциплину владения одноручным оружием, так как сюда входит владение жезлом и возможность носить щит. Еще один "плюс" этой игры заключается в том, что сохраняется баланс между разными видами оружия, т.е. какой бы вид оружия вы ни выбрали, вы на этом не проигрываете, будь то лук или кулаки. Приятный сюрприз "Lionheart" — при выборе оружия и брони, в магазине вам могут попасться средневековые бомбы (маленькие бочки с порохом и фитильком), которые нередко выручают в сложной ситуации.

Школа воровства тоже является полезной школой — именно за счет нее я заработал 10% всего "опыта" при прохождении игры. В этой школе четыре дисциплины: поиск секретов и ловушек, взлом замков, хождение в тени и общение с людьми. Последняя, в свою очередь, делится на убеждение людей и бартер. Но самое интересное в том, что вы получаете хорошую порцию "опыта" за нахождение ловушек, потайных дверей, скрытых сокровищ и взлом замков. При этом, чем выше сложность замка или секретность ловушки, тем больше "опыта" вы получаете за их взлом или обнаружение. Но и это еще не все — вы получаете 75% "опыта" от убийства твари только за то, что вы к ней подкрадетесь. Помимо этого, в игре довольно много потайных дверей и скрытых сокровищ, и вы получите неплохую прибыль, если будете активно заниматься их поиском.

Далее идет школа божественной ма-

гии, незаменимая для воина школа, так как это единственная школа, где есть лечащая магия. В ней находятся следующие дисциплины: божественная милость (здесь есть заклинания, улучшающие боевые характеристики и вызывающие на помощь рыцарей), стойкость (заклинания лечения и повышения сопротивления к различным магиям и ядам), разрушение (заклинания, понижающие характеристики врагов и приносящие урон) и божественное прикрытие (повышает значение брони и количество хит-поинтов). Также эта школа отлично подойдет тому, кто будет проходить компанию за Инквизицию или выберет путь священника.

После этого идет школа волшебства или школа магии мысли — самая классическая и мощная магия. Здесь находятся такие дисциплины как электричество, огонь, лед и защита. Последняя дисциплина школы волшебства похожа на последнюю дисциплину божественной магии и имеет заклинания с такими же функциями. С остальными дисциплинами этой школы все просто: большой урон, простые заклинания, стандартные эффекты. Если вы решили начать игру за мага, то выбирайте именно эту школу, потому что здесь нет ничего сложного в выборе и применении заклинаний.



И, наконец, самая мудреная из магии — школа шаманства. А находятся здесь следующие дисциплины: телепатия, некромантия, превращение и родовая защита. Ну, с телепатией все понятно — здесь различные виды управления вашими врагами и все такое. Некромантия тоже вещь объяснимая — поднимаешь из убитых армию скелетов, зомби и прочей нежити. А вот дальше идут сложности. Магия превращения дает герою возможность повышать свой кулачный урон и менять свою оболочку на оболочку различных зверей (медведя, волка), при этом какие-то характеристики повышаются, а какие-то понижаются, в зависимо-

сти от того, в кого вы хотели бы превратиться. И напоследок — магия родовой защиты. Сюда входят различные заклинания — от поглощения духа до улучшения характеристик, как при превращении, но без изменения оболочки и отрицательных эффектов. Поэтому начинать "раскачивать" данную школу магии советую только опытному любителю RPG.

Теперь поговорим о расах — их, ни много, ни мало, четыре. Первой, конечно же, идет раса людей. Здесь все просто, подойдет для новичка в мире RPG, потому что имеет самые универсальные для любого класса (воин, маг, некромант) характеристики. Потом идет раса демокинов. Они представляют собой смесь людей и духов, при этом имеют маленькие рожки и идеально подходят к классу лучника. Следующей расой идут феларкины — они являются чем-то средним между человеком и зверем, поэтому служат незаменимыми воинами. Последней расой хочу представить силвантов. Силванты — это чистокровные маги, подходящие к любому магическому классу и имеющие хороший магический потенциал.

Графическая часть игры тоже удалась на славу. Сохраняя классический RPG-вид, все текстуры и рельеф игры сделаны полностью в 3D. Но у этой игры есть небольшой глюк. Если вам попадется версия 1.0, то советую побыстрее найти патч до версии 1.1 (я нашел на диске "Игромания") и установить его, так как в первой версии сборки эта игра часто без причины "вылетает". Также желательно иметь видеокарту от nVidia, потому что игра тестировалась на видеокартах именно этой фирмы (хотя у меня стоит ATI Rage 128, и все идет нормально).

Музыкальная часть "Lionheart" просто идеально подобрана под игровую атмосферу. Средневековые мелодии словно уносят вдаль, и тебе хочется слушать их снова и снова. Здесь не к чему придираться,

потому что все сделано на высшем уровне, остается только пожелать остальным компаниям делать настолько же погружающую музыку для игр.

В итоге мы получаем отменную и оригинальную RPG с титулом "лучшая RPG года" и желание пройти ее несколько раз.

Минимальные системные требования:

- процессор Celeron 800 МГц;
- 2000 Мб свободного места на HDD;
- 256 Мб ОЗУ;
- видеокарта ATI rage 128, 32 Мб видеоОЗУ;
- 8-х скоростной CD-ROM;
- DirectX 9.0 (желательно).



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

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



119454, г.Москва, а/я 37 или
220095, г.Минск-95, а/я 199,
передавать по тел. в Минске (017) 249-41-47,
либо через E-mail:
rl@radiopage.by
rm@radio-mir.com
WWW: http://radio-mir.com

Куплю шлейфы (можно б/у) к принтеру "Электроника MC6312"; программы системные, игровые для "Профи 3+".

231800, Гродненская обл., г.Слоним,
ул.Шоссейная, 18 — 69, Синельников М.В.
Тел. 8-10375-156-24-74-81.

Продаю коллекцию CD-дисков для "Sega Dreamcast" и "Panasonic 3DO" (различные эмуляторы и программы, много игр и фильмов), кабель для подключения "Dreamcast" к компьютерному монитору. Вышло по почте. Могу выслать список имеющихся дисков (вложите в письмо конверт с обратным адресом).
442150, Пензенская обл., г. Нижний Ломов,
а/я 58, Сергей.

Продаю:

- микросхемы новые (серий от 134 до 1810);
- диоды, стабилитроны, транзисторы, (большой выбор);
- автомобильный телевизор "Мара-2", новый (270х140х70 мм, размер по диагонали — 7 см);
- дисконд MF580 (Венгрия) с дискеттами 5,25i;
- ПЗВМ "Сантака-001";
- блок питания импульсный +5В/17,5Вт +12В/48Вт;
- блок питания +5 В/10 Вт.

Тел. (017) 243-02-97, Саша.

Куплю процессор ATHLON только с разъемом

SLOT-A.
E-mail: santinal@yandex.ru

Школьник примет в дар дистрибутив Delphi версии 5 или более поздней. Дмитрий.
E-mail: svidinskii@ire.ru

Продам CD-диски. Список дисков:

www.m01.nm.ru/shop_cd.htm

Ищу техническую документацию на ноутбук ф. Siemens mod. Simatic PG 720P v.2148 или аналогичный.

211033, г.Могилев, ул.Гришина, 110 — 54,
Юношев В.Г.
Тел. (8-10375-0222) 23-80-00,
с 8 до 10 или с 20 до 24 часов.

Куплю контроллер жесткого диска и программное обеспечение для ПК "Сантака-002".
Приму в дар системный блок компьютера.

425568, Респ.Марий Эл,
Куженерский р-н, д.Шудумарь, Нагаев М.А.

Куплю в Москве HDD MK1403MAV TOSHIBA на 1443 Мб для notebook, можно "битый" (нужен только контроллер).
E-mail: vitpradv@hotmail.com

Куплю два картриджа (чистый оптический диск) типа OC301-1 или OC301-2, желательно фирмы "Maxell".
458785, г.Озерск, Челябинская обл., а/я 643.
E-mail: dkflbvhh@telecom.ozersk.ru

Продаю материнскую плату Epox EP-8KEM2 (встроенная видеоподсистема) с гарантией. Возможен торг.
Тел. в Минске 224-46-55, Михаил.

Подписные индексы:

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.pressa.ru);
- для жителей Беларуси: 48996, 489962 (для организаций) по каталогу РО "Белпочта" Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам").

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) по каталогу РО "Белпочта" Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам").

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) по каталогу РО "Белпочта" Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам").

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.

Получить подписку или заказать имеющиеся в наличии отдельные номера журналов можно из редакции. Текущие цены приведены в таблице.

Год	Можно заказать следующие номера журналов			Расценки на 1 экз. (с учетом пересылки)		
	Радиолобитель	Радиолобитель. Ваш компьютер	Радиолобитель. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
1999	11	1 — 6, 10, 11, 12	6	20	700	5
2000	3 — 6, 8 — 12	2 — 8, 10 — 12	6, 7, 9, 11, 12	20	800	5
2001	4 — 6	2, 5, 6	2, 3, 5, 6	25	900	5
	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2001	7 — 10, 12	7, 9 — 12	7, 8, 10, 11	30	1000	6
2002	1 — 12	1 — 12	1 — 3, 5 — 12	35	1500	7
2003	1 — 12	1 — 12	1, 5 — 8, 10 — 12	40	1800	8
2004	1 — 12	1 — 12	1 — 12	45	2100	9

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 30101810500000000109, 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.

Приобретение отдельных номеров журналов

В РОССИИ:

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

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

- г.Москва, ул.Гиларовского, 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.Ярославль, ул.Нахимсона, 12, тел. (0852) 27-57-15.

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

В ООО "ТРЭНТЭК": 111024, г.Москва, 4-я Кабельная, 2А (ст. метро "Авиамоторная"); тел. (095) 258-91-94, 258-91-95. E-mail: abook@mail.ru, abook@inbox.ru

На радиорынках: "Царицыно" (место 13/А), "Митино" (ряд 1, контейнер 17 и место Т-8); в ТК "Савеловский" (ст. метро "Савеловская", места А4, А5); на книжной ярмарке на "Тульской" (ст. метро "Тульская", место 515-19).

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.

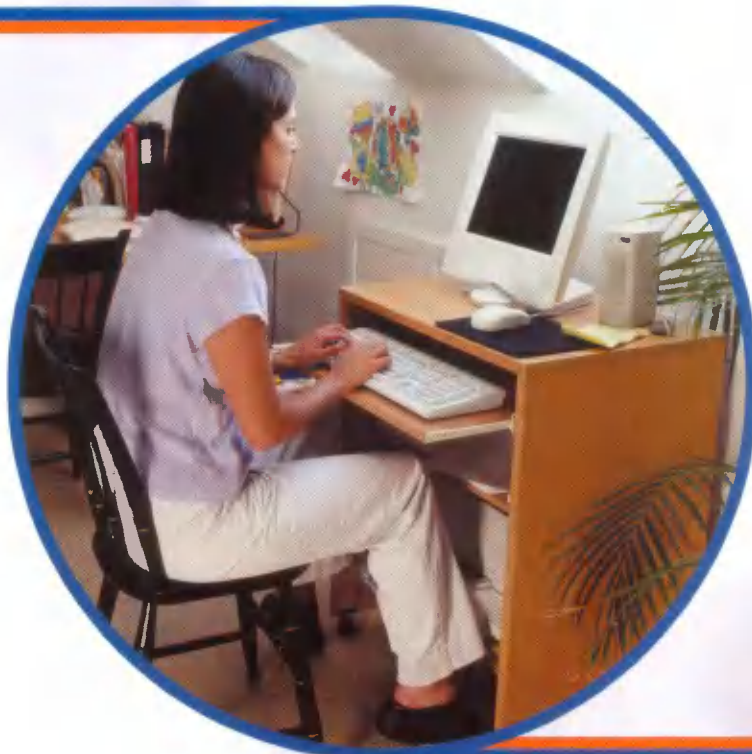
В БЕЛАРУСИ:

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").

Компьютеры для дома и офиса

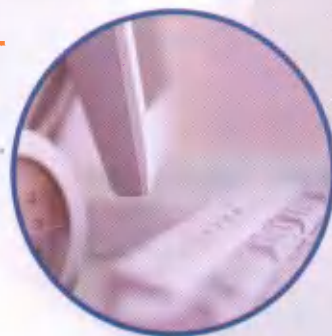
от компании СитиПринт

Нужен ключ к успеху?
Добейтесь одновременного повышения
эффективности и конкурентоспособности.



Начните использовать компьютеры от компании
“СитиПринт” на базе процессора
Intel® Pentium® 4 с технологией ИТ
уже сегодня, и Вы получите
возможность выполнять больше
работы за меньшее время.

Подарите Вашему бухгалтеру
точность высоких технологий.





- Ремонт и восстановление неисправных пейджеров
- Перепрограммирование пейджеров
- Большой выбор новых пейджеров (от **35000** р.)

7 способов отправки сообщений!

Стационарные охранные системы для дома и офиса

Получение на пейджер
полной информации о
состоянии охранной системы,
установленной в квартире,
офисе или коттедже



am® Атлант Моторс



Спутниковые автомобильные охранные системы

Осуществление мониторинга за
автомобилем при угоне:
определение его географических
координат по всему миру