

радиомир

10 • 2003

Ваш компьютер

PROMT XT + СЛОВАРИ

И другие переводчики



ЭНЦИКЛОПЕДИЯ

Комнатные растения



Microsoft Windows SERVER 2003 Corporate



Windows Server 2003 Standard Edition

БОЛЬШОЙ АНАТОМИЧЕСКИЙ АТЛАС



ЦИФРОВЫЕ ТЕХНОЛОГИИ:
КОДИРУЕМ ФИЛЬМЫ MPEG4

MPEG 4

закодируй свой фильм в MPEG 4

100%
MPEG 4

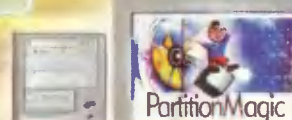
- Плееры MPEG4
- Кодирование MPEG4
- Редакторы видео
- Полезные программы
- Обзоры и руководства

ALEX SOFT

Partition Magic 8

СКОПОВЫЕ
ПЛИТЫ

Профессиональная работа с разделами



- Все для жесткого диска
- Клонирование данных
- Восстановление данных
- Резервирование данных



Jay

disc 14

КОЛЛЕКЦИЯ ИЗ 16 ДИСКОВ
Sound COLLECT

Garage + Progressive + Tec

Более 2500 сэмплов на каждом диске и
направлений специально для DJs

PRO100 SOFT ОФИС

- Microsoft Office XP 2002 русская версия
- АРХИВАТОРЫ
- АНТИВИРУСЫ
- УТИЛИТЫ ДЛЯ ОФИСА



2003

ЯЗЫКИ ПРОГРАММИРОВАНИЯ • СРЕДСТВА РАЗРАБОТКИ

Visual Studio .NET

Новые технологии от Microsoft



ALEX SOFT



БЛИСТАТЕЛЬНЫЙ Flash



Учредитель и издатель:
ООО "Радиомир Пресс"
Свидетельство о регистрации
ПИ №77-9841 от 17.09.2001

Главный редактор
Ольга СТРЕЖАНКОВА

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,
Надежда БОГОМОЛОВА

Адреса для писем:
119454, РФ, г.Москва, а/я 37,
факс (095) 131-97-17;
220095, РБ, г.Минск-95, а/я 199

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

Наши платежные реквизиты:
получатель: ООО "Радиомир Пресс",
ИНН 7729404741, КПП 772901001,
р/с 40702810900010000084
в ООО КБ "Агропромкредит"
ф-л Центральный, г.Москва,
корр. счет 3010181050000000109
БИК 044525109
Адрес банка: 125315, г.Москва,
Ленинградский пр-т, 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.Коштыяна, 6 — 233, тел. (095) 105-99-89

Подписано к печати 26.08.2003 г.
Формат 60 x 84 1/8.
Печать офсетная, 6 печ. л.
Цена свободная.

Отпечатано в типографии
ЗАО "Красногорская типография".
Заказ № 2459.

радиомир Ваш компьютер

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

№10/октябрь/2003

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

НЕ ТОЛЬКО НОВИЧКУ

В.КУЦ. Блистательный Flash	2
Б.КИСЕЛЕВ. MATLAB. Программирование в MATLAB	6
А.ГОРЯЧКИН. Total Commander	8

КОММУНИКАЦИИ

Р.КАРПАЧ. Локальные сети под Windows 98	10
И.РОЩИН. Добавление в HTML-документы информации об их кодировке	13

ДАЙДЖЕСТ 17

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

А.КОРБИТ, А.ЩЕРБАКОВ. Программирование в среде Visual C++ 6.0. Интерфейс графических устройств Windows	21
М.ДОЛИНСКИЙ. Решение задач на графах построением максимального потока	22

ДИАЛОГ ПРОГРАММИСТОВ

CyberManiac /HI-TECH. Теоретические основы крэкинга	24
И.РОЩИН. Простейший конвертер DOC → TXT	27
С.РЮМИК. Домашний "Screen Snapshot"	28
А.СИЗЕНКО. Работа с остатками в программе "1С: Бухгалтерия"	29
А.ЕВТЮХИН. Простой способ защиты программ от копирования	32
VINCE. Реализация криптоалгоритма Вигенера (1)	34
Б.ШИЛЬНИКОВ. DOS в среде Windows	37

РЕЦЕПТЫ

С.РЮМИК. "ONEchip3" на микроконтроллере AT89C2051	38
А.БУТОВ. Программное управление принтерами фирмы "Hewlett-Packard"	43
Р.КАРПАЧ. Назад в будущее	44

ИГРОТЕКА

А.ВЕНДИЛОВСКИЙ. Enter the Matrix	45
Е.КУРИЛОВИЧ. Коды, читы... ..	47

ДОСКА ОБЪЯВЛЕНИЙ

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

Дорогие друзья!

Страницы журнала "Радиомир. Ваш компьютер" открыты для всех, кто хочет и
умеет сделать общение с компьютером лучше, интереснее и полезнее. Поделитесь
с коллегами своими идеями, схемными решениями, программами и советами.

Присылайте нам и свои вопросы. — возможно, кто-то уже знает ответ.
Многочисленному форуму наших читателей под силу найти решения самых
сложных проблем. В общем, ждем ваших писем.

Редакция.

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

Блистательный Flash

В.КУЦ,

E-mail: victor_kootz@mail.ru

На потребительском рынке два последних года наблюдается неуклонный рост спроса на такие портативные носители информации для цифровых данных как флэш-карты, что обусловлено, в первую очередь, массовым поступлением на рынок цифровых фотокамер с матрицами высокого разрешения, позволяющих создавать снимки очень большого объема. Так, по данным компании IDC, в 2002 г. в мире было продано 52,7 млн. таких карт, а в 2006 г. продажи ожидаются уже на уровне 270 млн. При этом выручка от продажи флэш-карт увеличится с \$990 млн. в 2002 г. до \$2,2 млрд. в 2006 г. Согласитесь, рост впечатляющий.

Кроме цифровых фото- и видеокамер, флэш-карты самой различной емкости широко используются в таких устройствах как карманные ПК, MP3-плееры, диктофоны, сотовые телефоны. Постепенно область их использования расширяется, захватывая и бытовую технику — холодильники, микроволновые печи, детские игрушки и т.д. Развитие карт памяти идет столь стремительными темпами, что порой довольно трудно разобраться, какой продукт сегодня находится на “вершине” рынка. Идет постоянная эволюция как форматов карт памяти, так и их характеристик. Увеличиваются объемы, растет скорость, падает цена.

Flash-память относится к классу энергонезависимых типов памяти, хранящих данные даже в отсутствие напряжения питания. Физически каждая ячейка такой памяти представляет собой один транзистор, у которого под основным затвором помещен еще один, так называемый плавающий затвор, позволяющий хранить определенный заряд. Количество заряда определяет состояние ячейки: наличие заряда на транзисторе понимается как “логический 0”, а его отсутствие — как “логическая 1”. Использование только одного транзистора для хранения одного бита ведет к уменьшению площади памяти (и значит, к уменьшению цены) по сравнению с типами памяти, хранящей 1 бит информации на нескольких транзисторах (например, SRAM). Данные вводятся во flash-память побитно, побайтно или словами, эта операция называется программированием. Как только данные были введены, они остаются в памяти независимо от того,

подведено питание или нет. Очистка памяти производится при помощи операции стирания.

К достоинствам этого вида энергонезависимой памяти следует отнести высокую стойкость к механическим воздействиям, большое количество возможных циклов перезаписи информации, долговечность ее хранения (порядка 100 лет) и относительно невысокую цену. Чаще всего флэш-память можно встретить в виде карт памяти, которые подключаются к компьютеру с помощью специальных считывающих устройств — картридеров, или флэш-драйвов — полностью автономных запоминающих устройств, упакованных в миниатюрный корпус (что-то вроде брелка) и снабженных USB-интерфейсом, наличие которого позволяет устройству взаимодействовать с системой напрямую, без каких-либо дополнительных посредников. И те, и другие группы устройств, использующих флэш-память, имеют как достоинства, так и недостатки. В данном обзоре мы рассмотрим основные типы современных флэш-карт, а также бегло коснемся и флэш-драйвов.

Flash Card CompactFlash

Первые карты CompactFlash Card (CF) были выпущены в 1994 г. компанией Sandisk, и на сегодняшний день занимают лидирующее положение на рынке. CompactFlash (рис.1) по сути является эволюционным развитием карт типа PC Card (или, по старинке, PCMCIA) с уменьшенными габаритами

и напряжением питания 3,3 В или 5 В. Характерная особенность данного стандарта заключается в том, что карты CF включают в свой состав, помимо собственно флэш-памяти, еще и встроенный ATA-контроллер. Разъем, используемый в CF-картах, полностью соответствует спецификации ATA и аналогичен используемому PC Card-картами, но имеет уменьшенное до 50 количество контактов (против 68 у оригинала). Это дает CF-картам определенное преимущество перед другими типами флэш-карт — их можно легко установить в дешевый 68-контактный пассивный адаптер PC Card, а слотами такого типа оснащены практически все ноутбуки. Существует еще и вторая разновидность формата носителя данного типа — CompactFlash Type II, имеющая такие же размеры, но увеличенную толщину (5 мм). Устройства, предназначенные для работы с CF II, имеют обратную совместимость с носителями CF I, но не наоборот. Благодаря относительно большим габаритам, карты CF II имеют гораздо больший объем, чем CF I (емкость которых из-за физических ограничений застряла где-то на уровне 256 Мб). Кроме того, на базе CF II, помимо карт памяти, созданы такие интересные устройства как микровинчестеры Microdrive, имеющие емкость от 170 до 1000 Мб, адаптеры беспроводных сетей Wi-Fi или Bluetooth (включающие в себя даже антенны, а также вмещающие в себе карту памяти объемом до 128 Мб с беспроводной LAN) и множество других, самых разнообразных устройств.

Карты этого типа на сегодня обеспечивают максимальную скорость работы и имеют самую большую емкость — вплоть до 4 Гб, что позволяет им в отдельных случаях соперничать с накопителями на жестких дисках. Их характеристики отлично подходят для применения в самых разнообразных устройствах, включая цифровую фототехнику, где CF занимает ведущие позиции, несмотря на усилившееся давление со стороны формата SD, о котором речь пойдет ниже, и в карманных компьютерах. А вот в MP3-плеерах они встречаются крайне редко, причиной чему являются их относительно большие размеры и сложный и дорогой интерфейс — “честный” ATA, со всеми его глю-



Рис. 1

сами и минусами (включая и много-контактный разъем, который сложно разводить на микросплатах, зато тонкие штыри ответной части разъема при неаккуратном обращении с картой легко погнуть, а то и сломать).

Не могу не отметить и тот факт, что стандарт карт CompactFlash является открытым, что в немалой степени способствует значительному распространению карт этого формата.

SmartMedia

Стандарт флэш-памяти SmartMedia Card (SM) разработан корпорацией Toshiba в 1997 г. специально для использования в цифровых аудиоплеерах и фотоаппаратах (рис.2). Карты этого стандарта предельно просты и не содержат встроенный контроллер, что, конечно, уменьшает стоимость их производ-

титул "убийцы" формата SM. XD Picture Card (рис.3) на сегодняшний день имеет объем от 16 до 256 Мб, а в будущем планируется довести его до 8 Гб. Что касается скорости записи/считывания, то карты xD-Picture Card являются одними из самых "быстрых" на современном рынке и обеспечивают скорость записи до 3 Мб/с, а считывания — 5 Мб/с. Карты формата XD Picture по своим размерам значительно меньше SmartMedia, хотя и чуть потолще. Для облегчения проникновения новых карт на и так уже изрядно переполненный рынок флэш-карт, разработчики позаботились о совместимости их со многими современными цифровыми камерами, оборудованными слотами CompactFlash (правда, только через специальный адаптер).

ных (1 и 1,7 Мб/с соответственно). Поэтому, несмотря на широкую поддержку производителями и неплохую совместимость (карты MMC могут использоваться, кроме "родных" слотов, еще и в любом устройстве, поддерживающем формат SD), завтрашнего дня у формата уже нет. Потолок емкости, достигнутый MMC, ограничен 128 Мб, причем далеко не все существующие устройства поддерживают карты такой емкости — без ограничений поддерживаются лишь 64-мегабайтные карты. А по сегодняшним меркам этого уже не всегда хватает. Для защиты данных от случайного стирания в MultiMedia Card имеется механический переключатель разрешения/запрещения записи на карту.

Reduced Size MultiMedia

Несмотря на явные недостатки стандарта MMC, он не умер, а еще и продолжает свое развитие. Так, ассоциацией производителей MMCA (Multi-Media Card Association) созданы карты Reduced Size Multi-MediaCards (RS-MMC), идентичные MultiMedia Card во всем, кроме размеров (рис.4). Размеры карты RS-MMC составляют 18 x 24 мм, а ее вес — менее одного грамма. Такая "гиперминиатюризация" весьма актуальна для использования подобных карт в са-

Рис. 3



ства. Однако из-за переноса контроллера непосредственно в устройство возрастает не только его сложность, но и обостряется вопрос совместимости (карты, форматированные на одном устройстве, не всегда распознаются другими). Для соединения используется 18-контактная позолоченная панель. По длине и ширине эти карты примерно соответствуют CompactFlash, однако гораздо тоньше последних — это самые тонкие флэш-карты из всех известных на сегодня. Максимальный объем SM достигает 128 Мб и вряд ли когда-нибудь будет больше — низкая скорость работы карт этого типа привела к постепенному вытеснению их другими, более современными типами.

XD Picture Card

В частности, в августе 2002 г. японские компании Toshiba, Olympus и Fuji Photo Film, в цифровых фотокамерах которых использовались карты SmartMedia, представили новый стандарт флэш-карт — XD Picture Card (XD — сокращение от "eXtreme Digital"), которому многие пророчат

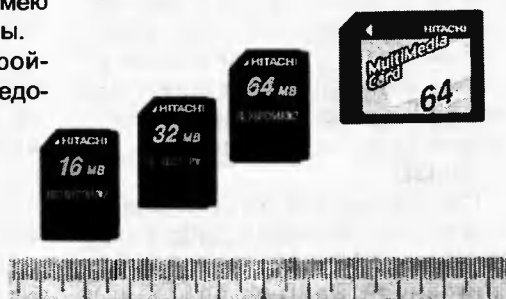
MultiMedia

Стандарт MultiMedia Card (MMC) был разработан в 1997 г. компаниями SanDisk и Siemens. Карты данного типа отличаются малыми габаритами, соизмеримыми с размерами почтовой марки, и весом. И это не удивительно, ведь изначально они проектировались для использования в мобильных телефонах. Но скромные габариты и малая потребляемая мощность пришлось по душе производителям других цифровых устройств, в первую очередь тех, где наиболее остро стоит проблема размеров носителя информации. Я имею в виду миниатюрные MP3-плееры.

Для обмена данными с устройством в MMC используется последовательный интерфейс, что позволило карте обойтись рекордно малым числом контактов — всего 7. Однако такая "скромность" имеет и свою обратную сторону — низкую скорость записи/чтения дан-

ных портативных устройствах — мобильных телефонах с цифровыми камерами, "наручных" PDA в форме часов и других подобных устройствах. В нынешнем году объем памяти таких карт планируется довести до 256 Мб, однако низкие скорости работы, доставшиеся RS-MMC от своего прародителя, вместе с необходимостью использования адаптера для подключения к любым имеющимся устройствам не позволяют нам отнести карты этого стандарта к перспективным.

Рис. 4





Secure Digital

Формат Secure Digital Card (SD) представлен компаниями Matsushita, SanDisk и Toshiba в 1999 г. По габаритам карта SD соответствует MMC, хотя и стала чуть толще (рис.5). По сравнению с MMC, в картах типа SD число контактов разъема достигло 9, однако устройства, использующие SD Card, без проблем понимают и MMC (но не наоборот). Такое незначительное, на первый взгляд, увеличение числа контактов дало



Рис. 5

возможность существенно повысить скорость передачи данных, так как появилась возможность пересылки нескольких битов одновременно, тогда как в MMC можно было пересылать только один бит за каждый такт шины. Емкость серийных карт доведена уже до 512 Мб, в ближайшее время ожидается выпуск карт емкостью 1 Гб, и уже были продемонстрированы образцы карт, содержащих 2 Гб флэша. В SD Card особое внимание уделяется вопросам защиты и безопасности, что как раз и отражено в названии карты — они имеют встроенные криптографические функции, позволяющие защитить информацию на карте от несанкционированного копирования, а также механическую защиту от случайного стирания. В общем, карты формата SD являются наиболее серьезным конкурентом CF, превосходя последний по многим параметрам. Количество устройств, поддерживающих этот стандарт, увеличивается с каждым днем. Помимо обычных областей применения флэш-карточек, SD начал применяться в таких довольно неординарных устройствах как детские игрушки, электронные книги, кухонные комбайны, холодильники, микроволновые печи и многое другое.

MiniSD

Специально для сотовых телефонов нового поколения, работающих с сетями 2.5G и 3G, весной 2003 г. компанией SanDisk разработан

сверхминиатюрный вариант карты SD — MiniSD, размеры которой не больше ногтя (рис.6). Объем новых карточек будет варьироваться от 16 до 256 Мб. По сравнению со стандартными картами SD, в интерфейсе MiniSD появились два дополнительных контакта, которые зарезервированы для нужд будущих изделий этого форм-фактора и пока не используются. Первое время карты MiniSD будут поставляться в комплекте с переходником, который позволит использовать их в стандартных SD-слотах.



Рис. 6

Memory Stick

Формат флэш-карт Memory Stick (MS) разработан в 1999 г. компанией Sony. Карта имеет 10-контактный разъем и оснащена механическим переключателем для защиты данных от случайного стирания. По скорости она примерно соответствует MMC, по уровню защищенности данных — SD, но по физическим размерам заметно больше обеих (впрочем, недавно появился вариант MS Duo с меньшими габаритами и весом). Из-за просчетов, допущенных при проектировании карты, ее объем не может превышать 128 Мб. Чтобы хоть как-то исправить этот просчет, Sony объявила о выпуске 256 Мб карт MS, имеющих организацию 128 Мб x 2, что подразумевает объединение в новой карте фактически двух автономных 128-мегабайтных карточек, используемых попеременно. Для того чтобы перейти от одного «банка» памяти к другому, карту необходимо изъять из слота, и перевести микропереключатель на корпус в положение А или В. Ничего не скажешь, сурово...

Но самое смешное еще впереди. Оказывается, что и цвет корпуса

карты MS, который бывает синим или белым, имеет для пользователя важное значение — ведь только «белые» карточки, поддерживающие фирменную технологию Sony Magic Gate, могут хранить музыку. Точнее, хранить-то можно на любых карточках, но воспроизводить файлы с «синей» аудиоплеер не сможет (совсем как дифференциация населения по цвету штанов в незабываемом фильме «Кин-дза-дза»). Стоит «белая», естественно, немного дороже «синей», поскольку в ее цену входят лицензионные отчисления звукозаписывающим компаниям. Кроме MP3-плееров, данный тип памяти используется в фотокамерах, некоторых моделях видеокамер, КПК и даже телевизоров (!?), причем исключительно производства самой Sony. Ввиду «закрытости» стандарта, а также небольшой емкости накопителя, говорить о широком распространении носителя данного типа не приходится.

Но Sony не была бы Sony, если бы так просто опустила руки. На смену стандарту флэш-памяти Memory Stick, который так и не смог стать сколько-нибудь популярным, спешит его новая версия — Memory Stick Pro (рис.7). Полностью повторяя своего предшественника по весогабаритным характеристикам, карты

Рис. 7



серии «Pro» имеют гораздо большую емкость (от 256 до 1024 Мб), работают значительно быстрее, но, к сожалению, их интерфейс не совместим со старым форматом Memory Stick. В таблицу сведены краткие технические характеристики основных типов флэш-карт.

Подводя итог всему вышесказанному, следует отметить, что основная «битва» производителей флэш-карт различных стандартов развернулась на двух «фронтах» — уменьшение размеров карт и увеличение их быстродействия. И на первом из них дела, похоже, в последнее вре-

	Compact Flash I/II	Secure Digital	MiniSD	Multi Media	RS-Multi Media	Smart Media	XD Picture	Memory Stick
Размеры, мм	42,8 x 36,4	32,0 x 24,0	20 x 21,5	32,0 x 24,0	18,0 x 24,0	45,1 x 37,0	20,0 x 25,0	50,0 x 21,5
Толщина, мм	3,3/5,0	2,1	1,4	1,4	1,4	0,76	1,7	2,8
Масса, г	11,4	2,1	1	1,4	0,8	2	2	4
Количество контактов	50	9	11	7	7	18	18	10
Напряжение питания, В	3,3 или 5	2,7...3,6	2,7...3,6	2,7...3,6	2,7...3,6	3,3 или 5	2,7...3,6	2,7...3,6

меня идут нормально — вероятно, уже достигнут определенный предел миниатюризации карт, когда дальнейшее уменьшение их размеров произойдет в ущерб удобству эксплуатации.

Со скоростью работы флэш-памяти дело обстоит сложнее. Уже сейчас скорость ее работы сравнима с накопителями на оптических дисках, но от современных винчестеров отстает весьма заметно. Такой скорости вполне достаточно для применения карт флэш-памяти в качестве средства переноса информации между компьютерами, однако современные цифровые камеры, оснащенные матрицами разрешающей способностью от 4 и более мегапикселей, для комфортной работы требуют от карт гораздо большего быстродействия, ведь иначе запись каждого кадра может занять слишком много времени. Именно недостаточное быстродействие, в первую очередь, и явилось причиной фактического «схода с дистанции» таких еще недавно популярных форматов карточек как SmartMedia или MMC.

Что до лидеров, то здесь нешуточная борьба разгорелась между «старичком» CompactFlash и «молодым и рьяным» Secure Digital. К плюсам CF можно отнести их большую емкость, невысокую стоимость хранения информации (что очень важно для больших карт) и высокую пропускную способность интерфейса, что, хотя бы теоретически, позволяет еще долго увеличивать скорость работы самих карт. Преимуществом же SD является их компактность, да и вопросы защиты хранимой на карте информации сегодня получают все большую актуальность.

Флэш-карты остальных форматов имеют лишь теоретические шансы завоевать хоть какую-то долю рынка, вероятнее всего, им уготована участь быть «нишевыми» продуктами, под-

держиваемыми на рынке исключительно усилиями своих разработчиков. Типичный пример тому — фирма Sony со своими карточками Memory Stick.

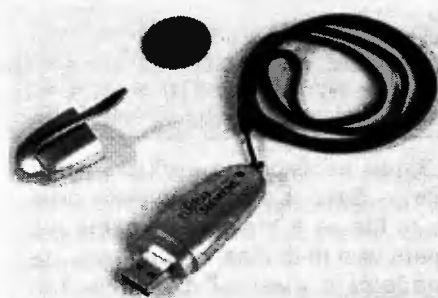
USB Flash Drive

Всем хороши флэш-карты — недорогие, надежные, миниатюрные. Но есть у них и достаточно серьезный недостаток — сложность с доступом к хранящейся на них информации. Для того чтобы передать информацию с флэш-карточки в компьютер, необходимо иметь специальный картридер (картридер), что, согласитесь, далеко не всегда удобно (особенно, если учесть большое количество несовместимых друг с другом типов современных флэш-карт) и не позволяет использовать эти карты в качестве средства для переноса информации между компьютерами. Если, конечно, не носить флэш-карты вместе с картридером — но тогда придется забыть обо всех их преимуществах, связанных с миниатюрными размерами. Наиболее простым и эффективным устройством для операций переноса относительно небольших объемов данных является ближайший родственник флэш-карт — USB флэш-драйв, представляющий собой чипы флэш-памяти, упакованные в миниатюрный корпус, напоминающий обычный брелок и оснащенный USB-интерфейсом.

Широкое распространение интерфейса USB, имеющегося на любом компьютере, если только он не представляет собой совсем уж музейную редкость, определило и удобство использования флэш-драйвов (рис. 8). Во-первых, не надо заботиться о драйверах (все современные ОС имеют в своем составе такие драйвера), во-вторых, питание устройства обеспечивает сам разъем USB. Все это, вкупе с возможностью «горячего» подключения,

делает USB-драйвы, емкость которых сегодня составляет от 16 Мб до 1 Гб, наиболее вероятным кандидатом на почетное звание «могильщика» 1,44-дюймовой дискеты. Единственно, в чем старая добрая дискета удерживала первенство по сравнению с новичком — это в возможности быть загрузочным устройством. Однако в последнее время пал и этот бастион — все современные флэш-драйвы обзавелись такой возможностью (естественно, если BIOS материнской платы поддерживает загрузку с USB-устройства). Так что теперь дисковод можно совершенно спокойно отправить на свалку истории.

Рис. 8



А как у флэш-драйвов обстоят дела со скоростью работы? Конечно, USB — это отнюдь не ATA, но и не дискета с ее 60 Кб/с. Ясно, что предельных скоростей, заявленных разработчиками интерфейса USB, современные флэш-драйвы не достигают, однако для накопителей с интерфейсом USB 1.1 скорость чтения достигает 700 Кб/с, а записи — 400 Кб/с, что вполне хватает для работы с небольшими объемами информации (до 64, а то и 128 Мб). Более современный USB 2.0 гораздо быстрее — читает со скоростью вплоть до 5 Мб/с, а пишет — до 4 Мб/с. Понятно, что приведенные показатели скорости устройств являются усредненными значениями и могут служить лишь для справки.



MATLAB. ПРОГРАММИРОВАНИЕ В MATLAB

(Окончание. Начало в №9/2003)

Б.КИСЕЛЕВ,
г.Минск,
БГАТУ, каф.ВТ,
тел.264-62-37.

Подготовка к организации собственных m-файлов

Напомним, что созданные пользователем m-файлы по умолчанию помещаются в рабочую папку **work** системы MATLAB.

Пример 1

Рассмотрим пример оформления m-функции. В меню **File** необходимо выбрать пункты **New, M-file** и в открывшемся окне набрать текст:

```
function w = rmovel(v)
% функция rmovel удаляет из вектора v совпадающие с точностью
% до 10^-6 элементы и формирует двухстолбцовую матрицу:
% 1-й столбец - неповторяющиеся элементы вектора v,
% 2-й столбец - количество повторений элемента, стоящего слева.
% вызов функции:
% w = rmovel(v)
% используемые функции:
% MATLAB: length, find
% функции пользователя: нет
% входные переменные:
% v - исходный вектор
% выходные переменные:
% w - матрица: 1-й столбец - неповторяющиеся элементы вектора v,
% 2-й столбец - количество повторений элемента, стоящего слева.
% Идея примера заимствована у В.Г. Потемкина, МИФИ.
w = [];
while length(v) > 0
    I=find(abs(v-v(1)) < 1e-6); J = length(I);
    w = [w; v(1) J]; v(I) = []; I = [];
end
```

Затем необходимо выбрать пункт **File — Save** и в открывшемся окне **Save file as** в строке **File name** набрать имя m-файла, оно должно совпадать с именем функции, т.е. **rmovel** (система должна предлагать его по умолчанию). Затем нужно нажать кнопку **Save**.

Приведенная программа работает следующим образом.

Первый исполняемый оператор определяет пустой массив **w**. Затем открывается цикл, который работает до тех пор, пока массив **v** не пустой (встроенная функция **length** возвращает длину массива). Дальше в цикле:

- формируется массив **I** из первого элемента массива **v** и совпадающих с ним элементов (встроенная функция **find** возвращает все элементы, отвечающие записанному в скобках условию);

- вычисляется переменная **J**, равная числу элементов массива **I**;

- к массиву **w** добавляется строка из первого элемента массива **v** и числа повторений этого элемента;

- из массива **v** удаляются все элементы **I** (которые совпали с **v(1)**), и очищается массив **I**.

К созданной m-функции можно обратиться из другого m-файла или командного окна. Например, определим в командном окне массив **A**, удалим из него повторяющиеся элементы и поместим оставшиеся члены с числом их повторений в массив **B**:

```
>> A = [2 4 3 2 7 3 3 9 0 5 0 8 8 8];
>> B = rmovel(A)
B =
     2     2
     4     1
     3     3
     7     1
     9     1
     0     2
     5     1
     8     3
```

Обратите внимание на то, что переменных из m-функции нет в рабочем пространстве (**Workspace**). Если запросить информацию об m-функции:

```
>>help rmovel
то получим ответ:
```

```
Функция rmovel удаляет из вектора v совпадающие с точностью до 10^-6
элементы и формирует двухстолбцовую матрицу: 1-й столбец -
неповторяющиеся элементы вектора v, 2-й столбец - количество
повторений элемента, стоящего слева.
вызов функции:
w = rmovel(v)
используемые функции:
MATLAB: length, find
функции пользователя: нет
входные переменные:
v - исходный вектор
выходные переменные:
w - матрица: 1-й столбец - неповторяющиеся элементы вектора v,
2-й столбец - количество повторений элемента, стоящего слева.
Идея примера заимствована у В.Г. Потемкина, МИФИ.
```

Определим этот же файл как сценарий. Все действия будут теми же самыми, только текст несколько изменится:

```
w = [];
while length(v) > 0
    I = find(abs(v-v(1)) < 1e-6); J = length(I);
    w = [w; v(1) J]; v(I) = []; I = [];
end
w
```

Сохранить его можно, например, под именем **rmovel_scr**. При обращении к нему, входные параметры задавать не нужно, но предварительно должен быть определен массив **v**. Все переменные сценария можно увидеть в рабочем пространстве.

Пример 2

Необходимо разработать функцию для задачи решения нелинейных уравнений. Ранее в статье [1] эта задача решалась чисто графическим путем. Там упоминалось, что задача решается в два этапа. Сначала должны быть отделены корни, а затем они уточняются с помощью какого-либо алгоритма. За теорией вопроса отошлем читателя к [2], здесь лишь напомним, что в общем случае уравнения записываются в виде

$$f(x)=0. \quad (1)$$

Будем считать, что $f(x)$ определена и непрерывна на интервале $[a,b]$. Значение x^* , обращающее функцию $f(x)$ в ноль, называют корнем этого уравнения или нулем функции (1).

Отделение корней — это приближенное их определение, т.е. нахождение отрезков, по возможности малых, которые содержат только один корень. Случай кратных корней здесь не рассматривается.

Общий алгоритм для решения задачи отделения корней построить затруднительно. Самый очевидный способ — нарисовать график функции $f(x)$. Точки, где $f(x)$ пересекает ось x , будут приближенными значениями корней.

Все встроенные программы определения корней предназначены для второго этапа. В MATLAB эту задачу можно решить, например, с помощью встроенной функции **fzero**, которая возвращает уточненное значение корня. Эта программа допускает несколько форм обращения. Рассмотрим простейшую:

fzero(@fun,x)

Здесь **fun** — имя функции $f(x)$ из условия (1); **x** — вектор $[x_1, x_2]$, где x_1 и x_2 — концы отрезка, содержащего корень; **tol** — заданная погрешность.

Найдем все корни уравнения

$$f(x) = 0.25x + \sin(x) - 1 = 0 \quad (2)$$

Очевидно, что при $x < 0$ $f(x) < 0$, а при $x > 10$, $f(x) > 0$, т.е. корни расположены на интервале $[0, 10]$.

Напишем m-функцию для вычисления выражения (2):

```
function f = root(x)
f = 0.25*x + sin(x) - 1;
```

Построим m-функцию для решения задачи, и сохраним ее под именем **Prim**:

```
function kor = Prim(x)
% Программа поиска и уточнения корней
plot(x, root(x), 'k'); grid on; % Построение графика функции
kor = []; % Создание массива для корней
%
% Вычисление корней
while 1
    xgr = input('Введите границы корня: '); % например, [2.5 3]
    % Добавление в массив очередного корня
    kor = [kor fzero(@root, xgr)];
    w = input('Уточнить другие корни? (Y/N)', 's');
    if w == 'N'
        break;
    end
end
disp('КОРНИ УРАВНЕНИЯ:');
disp(kor);
```

Наберем в командном окне:

```
>> x = 0:0.1:10;
>> Prim(x)
```

В результате получим графическое окно (рис.2). Из графика видно, что выражение (2) имеет три корня, расположенные на отрезках: $[0.5; 1]$, $[2.5; 3]$, $[5.5; 6]$. Перейдем в командное окно и увидим, что там появилось приглашение:

Введите границы корня:

Введем, например, первую пару:

```
>> [0.5 1]
```

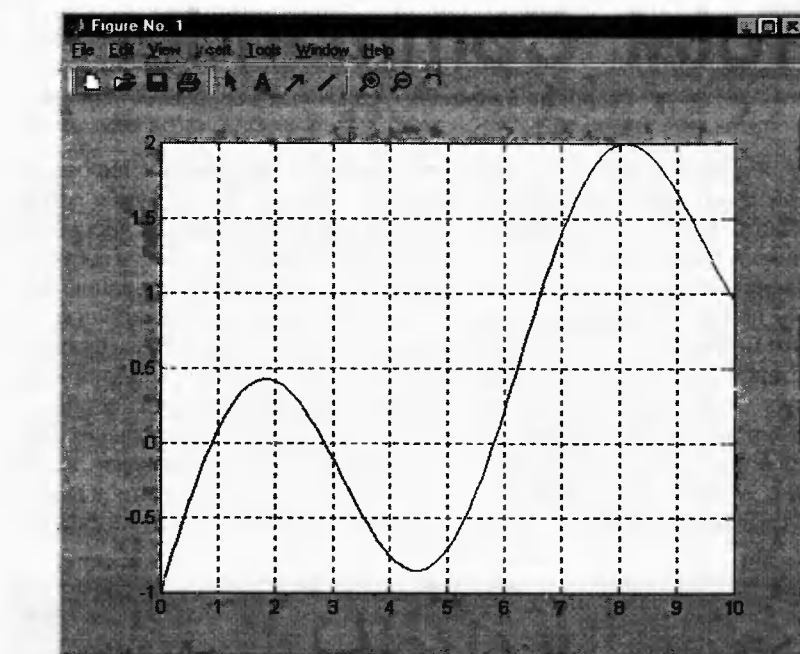


Рис. 2

После нажатия **Enter** появится приглашение:

Уточнять другие корни? (Y/N)

Если другие корни нужны, то вводим **Y** и аналогично задаем границы остальных отрезков, содержащих корни. В итоге получим сообщение:

КОРНИ УРАВНЕНИЯ:

0.8905 2.8500 5.8128

сания программ на различные темы можно найти в [3...5]. Заметим, что в более старой литературе тексты примеров ближе к традиционным универсальным алгоритмическим языкам, и наоборот.

Вопросы для самопроверки

1. Какие существуют основные типы данных?
2. Что такое m-файлы?
3. Отличия сценария от m-функции?
4. Что такое локальные, глобальные и статические переменные, а также переменные рабочего пространства?
5. Какая информация выдается по команде **help** имя файла.
6. Какие существуют операторы M-языка?
7. Правила записи m-файлов?

Литература

- 1 Киселев Б.М. MATLAB. Визуализация вычислений. — Радиомир. Ваш компьютер. 2003, №8. С.4.
- 2 Демидович Б.П., Марон И.А. Основы вычислительной математики. — М.: Наука, 1966.
- 3 Лазарев Ю.Ф. Matlab 5.x. — К.: Издательская группа BHV, 2000.
- 4 Мартынов Н.Н., Иванов А.П. Matlab 5.x. Вычисления, визуализация, программирование. — М.: КУДИЦ-ОБРАЗ, 2000.
- 5 Потемкин В.Г. Система инженерных и научных расчетов MATLAB 5.x. В 2 т. — М.: ДИАЛОГ-МИФИ, 1999.

Подстановка в уравнение, например, значения третьего корня показывает, что значение $f(x)$ практически равно нулю:

```
>> x=КОГ(3);
>> 0.25*x+sin(x)-1
ans =
-2.2204e-016
```

В качестве упражнения можете решить уравнения из [1]. Напомним, что поиск корней не всегда осуществляется так просто [2].

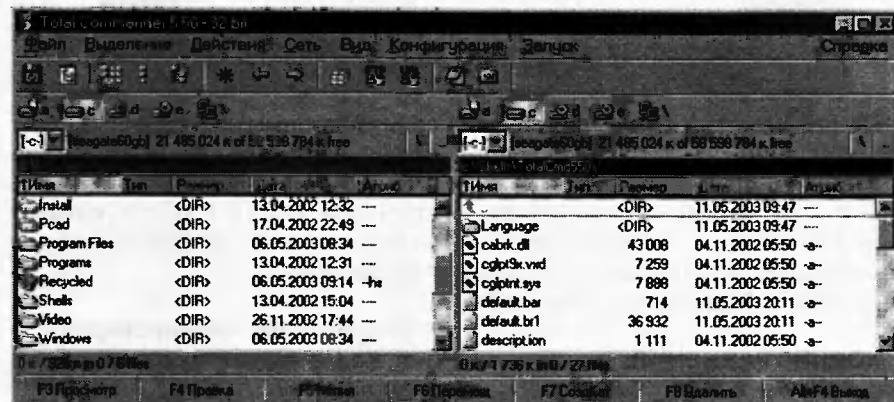
Достаточно много примеров напи-

Total Commander

Файловые менеджеры познаются в сравнении... файлов.
Изымылизм

О том, насколько хорош тот или иной файловый менеджер, можно сказать только по истечении определенного промежутка времени использования этой программы. Сегодня речь пойдет об обновленном файловом менеджере Total Commander (старое название менеджера — Windows Commander). И это не просто смена имени программы — в Total Commander в общей сложности исправлено 94 ошибки и добавлено 30 новых функций (рис.1).

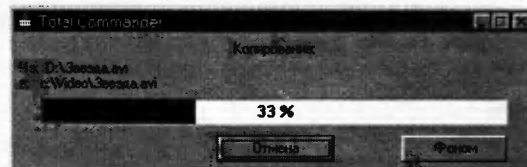
Рис. 1



Как и любой другой пользователь персонального компьютера, я в своей практике часто совершаю файловые операции — копирование и сравнение файлов, и должен признать, что Total Commander в этом нет равных. Судите сами. Допустим, пришел к вам приятель и принес для вашей коллекции компакт-диски с видеофильмами. Поскольку диски принесены не совсем, а лишь на короткое время, то придется их скопировать на жесткий диск своего компьютера, а уж потом, когда выпадет свободное время — посмотреть фильмы. Вы в это время заняты на компьютере какой-то срочной и не очень сложной работой, и не хотели бы отвлекаться на копирование файлов с фильмами. Total Commander берется вам в этом помочь с минимальными затратами времени. Для этого нужно в главном меню менеджера выбрать "Конфигурация — Настройки... — Разное — Собственные звуки — Настройка — Задать настройку звуков — Панель управления — Звуки — секция Total Commander" и все! Вы спокойно занимаетесь своими делами, а менеджер звуковым сигналом известит вас

об успешном окончании копирования файла. Удобно? Да, не нужно отвлекаться для наблюдения за процессом копирования файлов. А если вы желаете контролировать эту операцию, при старте копирования автоматически открывается отдельное небольшое окошко, в котором наглядно отображается процесс копирования, что особенно удобно при копировании больших файлов (рис.2). Кроме копирования, в Total Commander можно "озвучить" и другие события, например, удаление файлов.

Рис. 2



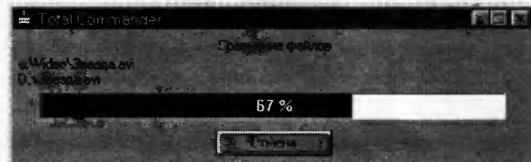
А теперь поговорим об операции сравнения файлов. Сравнить файлы по содержимому иногда приходится часто. Допустим, записал фильм на болванку и хочешь убедиться в том, что копия идентична оригиналу. И опять на помощь приходит Total Commander. В отличие от его "конкурента" FAR Manager, для которого при сравнении файлов обязательным условием является подключение внешнего плагина Compare, Total Commander имеет встроенную функцию сравнения файлов по их содержимому. Когда нужно сравнить файлы по их содержимому, Total Commander, как никто другой, успешно справляется с этой задачей. Причем сравниваемые между собой файлы не обязательно помещать в отдельные каталоги, как это было в

случае с файловой оболочкой DOS Navigator, а достаточно просто их выделить. Происходит операция сравнения довольно быстро, а для файлов большого объема — еще и наглядно (рис.3).

Рассмотрим конкретные примеры. Для сравнения двух файлов по их содержимому с помощью Total Commander нужно выделить эти файлы нажатием клавиши "Пробел", и в меню "Файл" задать команду "Сравнить по содержимому" (Compare by content). После завершения операции сравнения файлов, в случае их идентичности Total Commander выдает сообщение "Сравниваемые файлы идентичны!".

Сравнить файлы в Total Commander можно и по-другому, с включением режимов для текстовых или двоичных файлов. Если не выделять файлы перед подачей команды "Сравнить по содержимому", то откроется диалоговое окно, в котором нужно указать путь к местонахождению сравниваемых файлов, включить требуемый режим сравнения, например, двоичный, и нажать кнопку "Сравнить". При несовпадении в обоих вариантах сравнения будет указано количество найденных различий. Различия текстовых файлов в диалоговом окне будут отображены и выделе-

Рис. 3

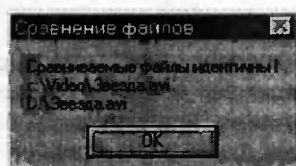


ны цветом в текстовом виде, а различия двоичных файлов — в шестнадцатеричном.

Справедливости ради стоит отметить, что никаких осложнений при сравнении файлов большого размера не возникает. Чтобы убедиться в этом, проведем небольшой эксперимент. Возьмем файл размером побольше, например, видеофильм "Звезда" в формате MPEG4. Сделаем с компакт-диска копию файла "Звезда.avi" на жесткий диск, и произведем сравнение файлов по их содержимому. На сравнение двух фай-

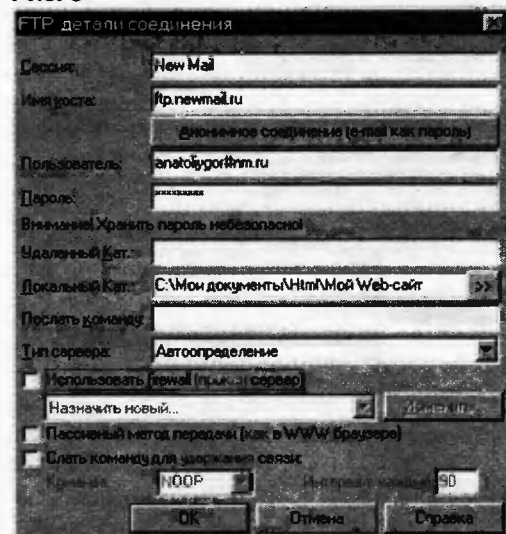
лов, каждый из которых "весом" в 666 МБ, Total Commander затратил всего 2 минуты 45 секунд, причем выполнил эту операцию без всяких проблем и осложнений (рис.4).

Рис. 4



Есть в Total Commander и встроенный FTP-клиент, которым удобно пользоваться при "заливке" и обновлении Web-сайта. Настройка FTP-клиента достаточно проста. Нажимаем клавиатурную комбинацию Ctrl+F — в результате появляется диалоговое окно FTP-клиента. Для того чтобы создать новое соединение, жмем на кнопку "Новое соединение" и начинаем ввод значений. В строке "Сессия" указываем название FTP-сервера, например, New Mail. В строке "Имя хоста" вводим адрес сервера, например, ftp.newmail.ru. В строке "Пользователь" указываем свое регистрационное имя (логин) и домен, а в строке "Пароль" — тот пароль, который был задан при регистрации. В остальных строках оставляем все без изменений (рис.5). Нажимаем на кнопку "OK" и даем команду "Соединиться". После этого вы попадаете в свою директорию на сервере ftp.newmail.ru.

Есть у Total Commander утилита пакетного переименования файлов (вызывается клавиатурной комбинацией Ctrl+T). С ее помощью можно изменять имена, расширения и регистр имен папок и файлов (рис.6). Может какой-либо другой файловый менеджер предоставить нам все вы-



шеперечисленные удобства? Нет.

Закончив с файловыми операциями, перейдем к тонкой настройке Total Commander. Приятно осознавать тот факт, что имеется возможность настроить программу "под себя". Все изменения вносятся в файл wincmd.ini, который находится в системной или в программной папке. Чтобы знать точное расположение файла wincmd.ini, к которому обращается Total Commander при загрузке, заходим в меню "Справка — О программе...". Редактировать файл wincmd.ini удобно в Блокноте (notepad.exe). После окончания редактирования файла wincmd.ini нужно произвести перезагрузку файлового менеджера Total Commander, для того чтобы новые параметры вступили в силу.

Вот некоторые опции Total Commander, которые можно настроить.

Защитный экран Splash Screen. При загрузке Total Commander на экране появляется заставка (splash screen). Это не очень удобно, поскольку не только замедляет загрузку Windows Commander, но и засоряет оперативную память вашего компьютера. Однако загрузку заставки можно отключить. Для этого в файле wincmd.ini в разделе [Configuration] необходимо вписать строку:

StartupScreen=0

Смена активной файловой панели. При запуске Total Commander по умолчанию всегда активна левая файловая панель. Вставив в файл wincmd.ini в разделе [Configuration] строку

ActiveRight=1,

мы добьемся того, что теперь активной станет правая файловая панель.

Квадратные скобки на именах папок. Если вы хотите убрать квадратные скобки, в которые взяты имена папок, вставьте в файл wincmd.ini в разделе [Configuration] строку

DirBrackets=0.

Измерение объемов катало-

гов. Нажав клавишей "Пробел" на имени любой папки, вы узнаете ее объем. А если такую процедуру нужно повторить для нескольких каталогов, то придется каждый раз перемещать курсор на нужную папку. Добавив в wincmd.ini в разделе [Configuration] строку

SpaceMovesDown=1,

можно исправить это неудобство. Теперь при нажатии пробела курсор будет автоматически опускаться на одну строку вниз (как при нажатии клавиши Insert).

Смена иконок. Владетельцы Windows 9x/Me при желании могут сменить внешний вид иконок файлов и папок, представив Total Commander в стиле Windows XP. Для этого нужно скачать файл winxp.icl, поместить его в папку, где расположен Total Commander, и в файле wincmd.ini в разделе [Configuration] вставить строку с указанием пути к месту нахождения файла winxp.icl:

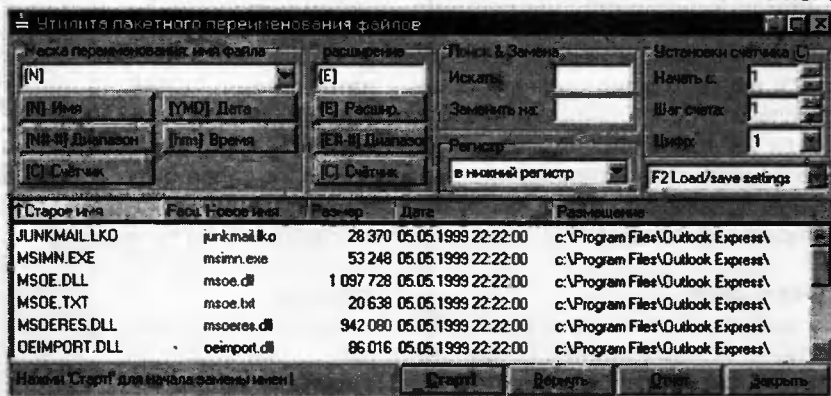
IconLib=C:\Shells\TotalCmd550\winxp.icl

Архив icons.rar можно скачать по адресу: <http://www.anatoliygor.nm.ru/download.htm>.

На этом наши настройки Total Commander не заканчиваются, потому что это всего лишь капля в огромном океане разнообразных "примочек" и "фишек". Очень много полезных и интересных сведений можно найти на сайте <http://wincmd.ru>.

И в заключение я хочу заметить, что, на мой взгляд, Total Commander (<http://www.ghisler.com>) на сегодняшний день является самым популярным файловым менеджером для среды Windows. Эту популярность у пользователей он заслужил благодаря огромному богатству функций, которыми он обладает. Привлекает и компактность Total Commander. После инсталляции программа занимает на жестком диске 1,68 Мб. Кроме того, для Total Commander создано немало плагинов, расширяющих его возможности.

Рис. 6



Локальные сети под Windows 98

Р.КАРПАЧ.
www.fdd5-25.narod.ru

Хотя на дворе уже вторая половина 2003 г., большинство "массовых" компьютеров, объединенных в локальную сеть, будь это компьютерный класс в каком-либо учебном заведении или компьютерный клуб, работают под управлением Windows 98. И первым вопросом, который задают все, получив новое оборудование, является: "А как же организовать работу и защиту этой самой сети?" Конечно, в этой статье я не претендую на истину в последней инстанции, но хочу отметить, что по характеру своей работы имею немалый опыт в решении данного вопроса.

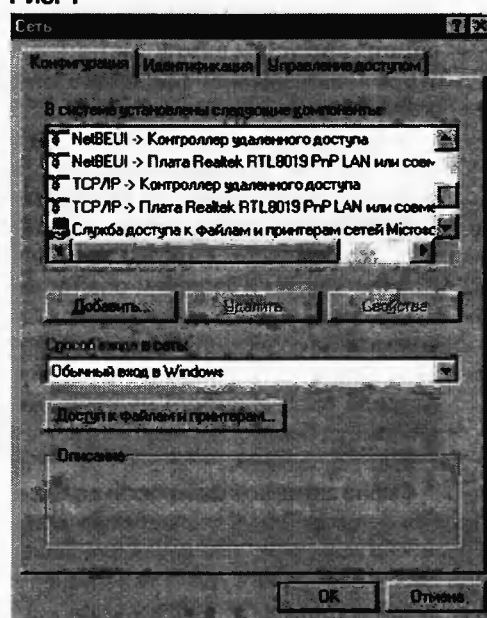
Итак, с чего же начать простому пользователю или учителю информатики? Вы можете сказать, что новое оборудование сдают под "ключ". Это верно, но вопрос, в каком виде. Ведь нам не нужна локальная сеть, организованная работой протокола NetBEUI (NetBIOS Extended User Interface — транспортный протокол, используемый Microsoft LAN Manager, Windows for Workgroups, Windows NT и другими сетевыми ОС). Сам по себе он неплох, но его не поддерживают большинство программ и игр, работающих в сети. По-другому дело обстоит с TCP (Transmission Control Protocol — основной транспортный протокол в наборе протоколов Internet, обеспечивающий надежные, ориентированные на соединения, полнодуплексные потоки), а также IPX/SPX (Internet Packet eXchange/Sequenced Packet eXchange). IPX используется в качестве основного протокола в сетях Novell NetWare для обмена данными между узлами сети и приложениями, работающими на различных узлах. Протокол SPX содержит расширенный по сравнению с IPX набор команд, позволяющий обеспечить более широкие возможности на транспортном уровне, и обеспечивает гарантированную доставку информации. Вот о них-то мы сегодня и поговорим.

Установка

Первое, что нам нужно будет сделать — это установить новые настройки сети. Подразумевается, что

у вас уже установлены драйвера к вашей сетевой карте. Для установки новых протоколов нам понадобится диск с Windows 98 и некоторое терпение. На рабочем столе кликнем правой кнопкой мыши по иконке сетевого окружения. Затем выбираем опцию "Свойства" и попадаем в настройки сети (рис. 1).

Рис. 1



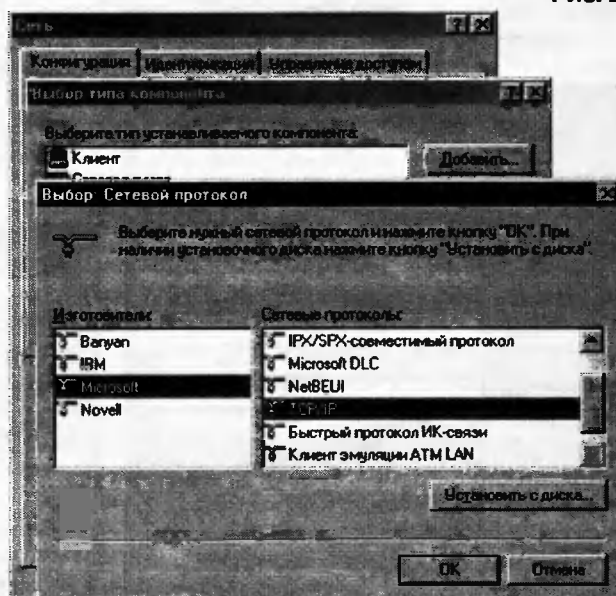
Там же вы увидите список установленных компонентов. На рисунке видно, что в системе уже есть протокол TCP/IP. Но если он у вас все же не установлен, то вам следует нажать кнопку "Добавить", затем выбрать меню "Протокол" и вновь нажать "Добавить". Затем выбрать позицию Microsoft и выбрать протокол TCP/IP или IPX/SPX. Все зависит от вашего желания. Я же далее буду рассматривать локальную сеть на базе протокола TCP/IP (рис.2).

После всех описанных манипуляций нам останется только на-

жать кнопку "OK", и Windows 98 попросит вас вставить компакт-диск с системой. После этого, скопировав нужные файлы, она попросит перезагрузить компьютер. Но мы ей в этом на данном этапе откажем, нажав кнопку "Нет" в диалоговом окне. Вы спросите: "почему же так?" Просто в большинстве случаев нам нужно будет создавать, помимо всего прочего, новых пользователей, для того чтобы разделить задачи рабочих столов. Например, у меня на работе они называются "Ученик", "Игры", "Учитель". Но об этом чуть позже. Мы вновь заходим в свойства "Сетевого окружения" и выбираем "Добавить", "Клиент". Затем переходим в опцию "Microsoft" и нажимаем на "Семейный вход в систему".

Кликнем на кнопку "Ok". После этого Windows 98 вновь запросит диск с системой и предложит перезагрузить компьютер. В этот раз мы ей предоставим эту возможность. Теперь по умолчанию у вас будет высккивать удобное окно со списком пользователей. Дождавшись окончания перезагрузки компьютера, мы переходим к другим действиям.

Рис. 2



Конфигурирование сети

После того как у нас загрузился рабочий стол Windows, мы вновь заходим в свойства "Сетевого окружения". У нас должны быть установлены следующие компоненты:

- сетевая плата (в моем случае — это RTL 8019);
- протокол TCP/IP или другой, на ваше усмотрение;
- общий доступ к файлам и принтерам. Без него ваш компьютер не будет виден в локальной сети, и вы не сможете делать ресурсы вашего ПК доступными для других пользователей. Если этот компонент отсутствует, вам следует его установить. Для этого нужно выбрать меню "Добавить", затем "Служба", после чего выбрать "Служба доступа к файлам и принтерам сетей Microsoft" (рис.3).

Рис. 3

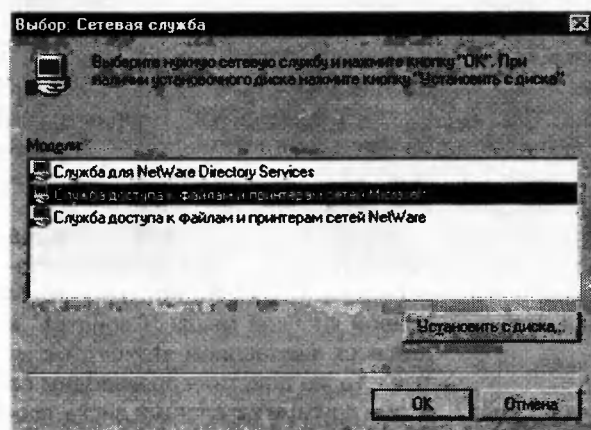
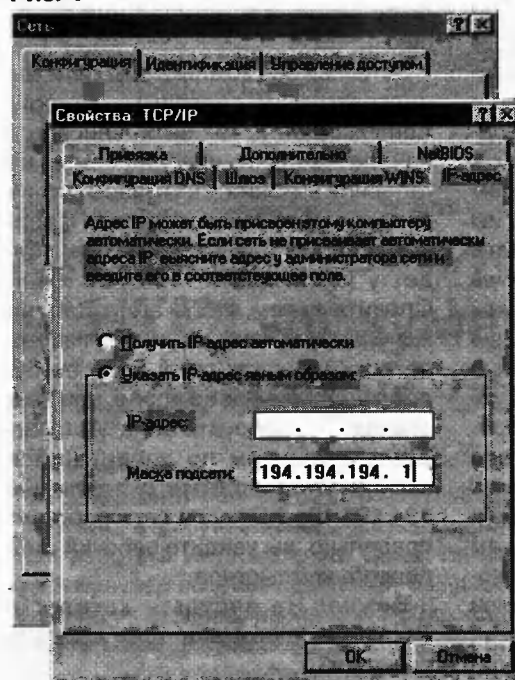


Рис. 4



Сейчас же мы будем конфигурировать протокол TCP/IP (рис.4).

Выберем "TCP/IP" — "Плата" (название вашей сетевой карты). Нажимаем кнопку "Свойства" и попадаем в настройки протокола. Выбираем закладку "IP-адрес" и установим флажок в пункте "Указать IP-адрес явным образом". Обычно в локальных сетях, чтобы администраторам легче было ориентироваться, последняя цифра указывает номер компьютера. Назовем главный компьютер вашей сети 194.194.194.1, а другие — по номерам 194.194.194.2, 3, 4, 5, 6 и так далее. Нажимаем "OK". Переходим на закладку "Дополнительно" и ставим галочку напротив надписи "Использовать протокол по умолчанию". Теперь мы вновь перезагрузим компьютер — первая часть нашей задачи выполнена.

Пользователи

После всей вышеописанной работы мы создадим несколько рабочих столов. Условно поделим их на "Ученика", "Учителя" и "Игры". Для того чтобы обеспечить разделение задач, нам

пример, хочу, чтобы под этим именем в систему могли заходить все, то код ставить не буду, и просто нажму "Далее". После всего нам следует выбрать "Создавать новые элементы" и выставить везде галочки. Затем еще раз нажмем "Далее" и "Готово" (рис.6).

Этот шаг повторим несколько раз, в зависимости от того, сколько пользователей нужно создать. В принципе, ничего сложного нет. Добавлю лишь, что все рабочие столы и главные меню будут храниться в каталоге C:\WINDOWS\PROFILES\.

Самые распространенные ошибки в компьютерных клубах

Этот раздел введен в статью с единственной целью — заставить администраторов компьютерных клубов не почитать на лаврах, а начать "ловить мышей" и закрывать явные "дыры" в своем хозяйстве.

Как это ни забавно, но именно для локальной сети под Windows 98 написано множество так называемых SHELLS, проще говоря, оболочек, которые призваны защитить диски компьютера от любопытных глаз. Чаше всего такого рода программы устанавливаются в компьютерных клубах. У нас в Белоруссии они лицензион-

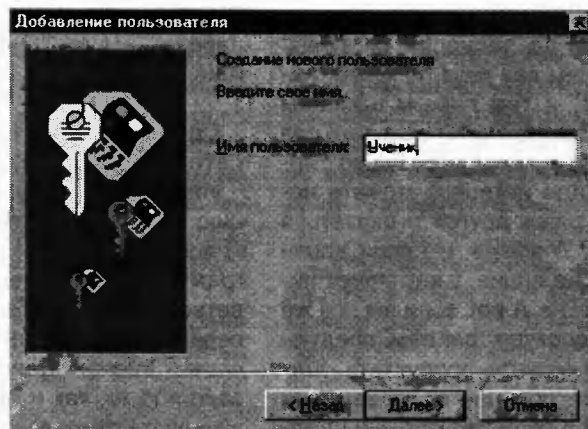
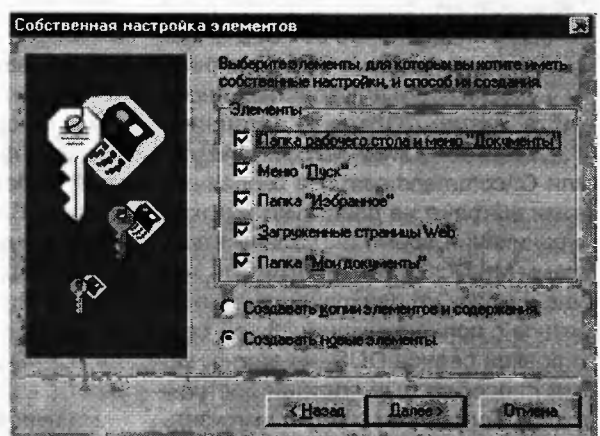


Рис. 5

следует зайти в "Панель управления", выбрать иконку с надписью "Пользователи", после чего включится диалоговое окно. Мы нажмем "Далее" и введем имя пользователя (рис.5).

Затем в появившемся меню введем пароль. Но поскольку я, на-

Рис. 6





ные и стоят от 30 USD. Но действительно ли они стоят таких затрат? Стандартный SHELL закрывает от любопытных глаз все, кроме ярлыков с играми. Вроде бы, все хорошо и красиво. Но стоит зайти, к примеру, в игру COUNTER STRIKE и выбрать меню CS MANUALS, как у нас загрузится Internet Explorer. В нем, не долго думая, печатаем C:\. И что у нас появляется? Вот так дела... содержимое диска C:\. Если у вас такой финт не получился, не отчаивайтесь, есть еще и другая возможность. В любой программе имеется встроенная помощь. Заходим в нее, родимую, и из нее вызываем полную справку Windows. Между прочим, отсюда можно вызвать множество утилит. А поскольку не на все программы действует блокировка опции "файл", то мы преспокойно открываем каталог Windows. Правда, такой фокус проходит не со всеми программами.

Существуют в природе и не ленивые администраторы игровых залов, на все замечательные SHELL они реагируют своим интеллектом и сами вручную изменяют через реестр доступ к файлам и дискам. Правда, и это все обходится довольно легко. В меню "Пуск" всегда есть ярлыки. И даже если все-все закрыто, и выключены диски, имеется огромный шанс докопаться до ресурсов системы. Как? Кликнем два раза по меню "Автозагрузка" и тем самым вызовем окно Explorer. Нажмем несколько раз "Backspace", и окажемся в меню с ярлыками. Ради интереса можете нажать еще разок-другой и оказаться в каталоге Windows. Но если администратор не ленив, у вас может ничего не выйти. А вот если вы все же получили доступ к файлам, следующий совет можете пропустить. Когда ничего не получается, то вам следует зайти в свойства ярлыка нажатием кнопочки в меню Explorer. Если таковой нет, доставьте ее ("Вид — Панель инструментов — Обычные кнопки"). После этого в "Свойствах ярлыка" в опции "Объект" запишите C:\WINDOWS\COMMAND\command.com или C:\command.com. На созданный вами rif-файл не действуют никакие ограничения, кроме как на запуск программ MS DOS (следует отметить, что мне ни разу не приходилось сталкиваться с полнофункциональной блокировкой сеанса DOS, не важно, каким способом был получен доступ к диску). Теперь нам следует выгрузить

из памяти то, что нам мешает. Если это просто SHELL, то отредактируем файл C:\WINDOWS\SYSTEM.INI, заменив строку shell=оболочка на shell=explorer.exe. Даже если меню "Файл" закрыто, программа предложит сохранить изменения.

Со времен Windows 3.11 в системе имеется замечательная замена Проводника под названием winfile. Эта программа всегда остается без внимания администраторов. А ведь на ее работу тоже не действуют никакие ограничения. Единственное, что нужно будет сделать — это включить поддержку "скрытых" файлов ("Вид" — "Выбрать типы файлов" — "Скрытые/системные"), и все — вы властелин системы. Затем мы быстро переходим в каталог C:\WINDOWS\SYSTEM и запускаем программу msconfig.exe, в которой не напрягаясь выключим автозагрузку программ. Именно там всегда прячется агент, учитывающий время, проведенное за компьютером. Затем идем к администратору и с расстроенным видом докладываем о том, что у нас повис компьютер, и надо бы его перезагрузить. И, как всегда у меня бывало, получаем разрешение. Все — сидите и играйте.

Но бывает и так, что при загрузке выскакивает пароль из BIOS. Как нам быть? Для Bios Award универсальный пароль — AWARD_SW, для Ami — AMI_SW.

Своими руками защищаем свою сеть

Итак, представим обычную ситуацию. Вам установили компьютерный класс. Все хорошо. Но каким программным обеспечением воспользоваться, да так, чтобы было дешево и сердито? Выход нам как всегда подскажет Internet. Далее будет рассказано про то, как и от кого мы будем защищать компьютер. Первое, что мы сделаем — это скачаем с сайта www.w98c.prv.pl программу Windows Configurator (рис.7). Зачем она нам понадобится? Именно с ее помощью мы установим те настройки, которые нам будут нужны для обрезания функций Explorer. Лень — двигатель прогресса. Ведь не вписывать же все своими руками? Далее мы скачаем с сайта <http://www.famatech.com/> программу Remote Administrator, она нам позволит с использованием протокола TCP/IP управлять любой машиной в сети.

Но перед тем как начать блокиров-

ку избранных функций пользователь, мы должны произвести кое-какие хитрые манипуляции. Условно переходим на рабочий стол под названием "Ученик" и выключаем поддержку скрытых файлов на этом рабочем столе ("Пуск — Настройки — Свойства папки — Вид — Не показывать скрытые и системные файлы"). Теперь, даже если кто-нибудь и попытается просмотреть диск, то увидит только обычные, а не скрытые файлы — папку C:\Мои документы или что-то наподобие "Мои рисунки".

После этого мы должны удалить с этой машины все "ненужное", с помощью чего можно перемещаться по дискам — оболочки (Norton Commander, Far, Windows Commander), программы конфигураций C:\WINDOWS\msconfig.exe (конфигуратор загрузки), C:\WINDOWS\winfile.exe (старый проводник времен 3.X), C:\WINDOWS\progmam.exe (среда времен 3.X), C:\WINDOWS\SYSTEM\sysedit.exe (редактор всех системных файлов).

Копируем ярлыки нужных нам программ в меню "Пуск". Например, на рабочем столе "Ученик" — это MS WORKS, MS PAINT, обучающие программы и кое-какие игры.

Нажатием комбинации клавиш "Ctrl+A" выделяем все файлы и каталоги в папке WINDOWS. Вызываем правой кнопкой мыши "Свойства" и для всех выставляем атрибут "скрытый". Те же действия производим со всеми другими каталогами диска, кроме C:\Мои документы. Эта папка должна быть всегда видна, чтобы можно было сохранять разную информацию.

Затем на все компьютеры сети мы устанавливаем программу Remote Administrator, и, как было описано в начале статьи, конфигурируем ее на головной машине. Указываем в поле IP адрес конкретной машины. Например, первый компьютер условно будет у нас иметь IP 194.194.194.1. Следите за тем, чтобы агент Remote Administrator был загружен в память на всех машинах сети. Если он будет отсутствовать на других рабочих столах — скопируйте его в "Автозагрузку". Кликнем по появившейся иконке в программе с подписью 194.194.194.1, и если сеть работает правильно, вы увидите рабочий стол первого компьютера.

Важно! Не забудьте каталог с Remote Administrator тоже сделать скрытым!

Рис. 7



После того как мы удалили все "опасные" программы и убрали иконку "Назначенные задания", начинаем блокировку Explorer. Для этого запустим w98c и выставим галочки везде, кроме опций меню "Пуск — Завершение сеанса работы". После того как мы закрыли все, что возможно, поставим на программу пароль, благо, она это позволяет.

Теперь мы перезагрузим компью-

тер и зайдем на рабочий стол "Ученик". Чтобы удостовериться, что все в порядке, и что блокировки работают, попробуйте повторить действия, описанные в разделе о "дырках" в софте компьютерных клубов. Если вы все сделали правильно, то у вас ничего не должно получиться. Обратите внимание и на тот факт, что сеанс MS DOS недоступен.

Осталось закрыть самую замечательную "дыру" Windows 98. Когда вы нажимаете "крестик" в логоне при

входе в рабочие столы, вы получаете доступ ко всей системе. Чтобы этого не произошло, мы при помощи все той же w98c.exe выставим все блокировки, предварительно удалив все из меню "Пуск". Оставить следует лишь "Завершение сеанса".

Если после прочтения этой статьи у вас все же возникли проблемы, то можете смело писать мне на e-mail: commander-norton@tut.by или звонить в г.Минск по тел.(017) 297-54-41-85.

Добавление в HTML-документы информации об их кодировке

Введение

В силу разных причин, не во всех HTML-документах присутствует атрибут "charset", указывающий их кодировку. При работе в Интернете это может не иметь значения, если браузер получает сведения о кодировке непосредственно от сервера, передающего документ. Но если записать такой документ на диск и затем открыть его для

просмотра, то пользователю придется самому указывать кодировку, так как браузеру больше неоткуда будет ее узнать (если только браузер не умеет автоматически определять кодировку, анализируя содержимое документа).

Столкнувшись с этой проблемой, я решил написать программу, которая анализировала бы сохраненные пользователем на диске HTML-документы, и при отсутствии в

И.РОЦИН,

г.Москва,

Fido: 2:5020/689.53

ZXNet: 500:95/462.53

E-mail: bestview@mtu-net.ru

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

каком-либо документе атрибута "charset", автоматически определяла бы кодировку этого документа и добавляла бы в него информацию о кодировке.

Определение кодировки

Все имеющиеся у меня HTML-документы были либо в кодировке Windows-1251 (далее WIN), либо в KOI8-R (далее KOI). Видимо, документы в других кодировках встречаются в русскоязычной части Интернета очень редко. Таким образом, программа должна уметь определять, какая из этих двух кодировок соответствует обрабатываемому документу.

В своих статьях [1, 2] я уже рассматривал различные алгоритмы автоматического определения кодировки текста. В [1] был рассмотрен алгоритм выбора между кодировками ALT и WIN, основанный на особенностях этих кодировок и неодинаковой частоте встречаемости различных букв, а в [2] — модификация этого алгоритма для кодировок WIN и KOI, а также другой алгоритм для выбора между кодировками ALT, WIN и KOI, основанный на том, что не все возможные сочетания букв могут встречаться в словах. Но при написании рассматриваемой программы я решил использовать еще один алгоритм, совершенно отличный от предыдущих и основанный на одной особенности кодировок WIN и KOI.

Коды прописных букв в кодировке WIN являются кодами строчных букв в кодировке KOI, и наоборот (рис.1 и 2). Т.е. при неправильном определении кодировки прописные буквы станут строчными, а строчные — прописными.

Рис. 1

Кодировка WIN	
0	1 2 3 4 5 6 7 8 9 A B C D E F
С	А В Г Д Е Ж З И Й К Л М Н О П
Д	Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ъ Ю Я
Е	а б в г д е ж з и й к л м н о п
Ф	р с т у ф х ц ч ш щ ъ ы ѡ ю я

Рис. 2

Кодировка KOI	
0	1 2 3 4 5 6 7 8 9 A B C D E F
С	ю а б в г д е ф г х и й к л м н о
Д	п я р с т у ж в ь ы з ш э щ ч ь
Е	Ю А В Ц Д Е Ф Г Х И Й К Л М Н О
Ф	П Я Р С Т У Ж В Ъ Ы З Ш Э Щ Ч Ъ

В обычном тексте (рис.3а — кодировка определена правильно) строчная буква может встретиться после прописной (например, в начале предложения), а прописная после строчной — не может. Если же в результате неправильного определения кодировки (рис.3б — WIN распознана как KOI и рис.3в — KOI распознана как WIN) прописные буквы стали строчными, а строчные — прописными, то в тексте можно будет встретить прописную букву после строчной, и нельзя — строчную после прописной.

Таким образом, алгоритм определения кодировки оказывается очень простым. Просматриваем текст, пока не обнаружим два соседних символа, код одного из которых принадлежит диапазону C0h...DFh, а код другого — диапазону E0h...FFh (т.е. один из этих символов — прописная буква, а другой — строчная). Если код первого символа принадлежит диапазону C0h...DFh, то кодировка текста — WIN, иначе — KOI (т.к. первый символ должен быть прописной буквой, а второй — строчной).

Рис. 3а

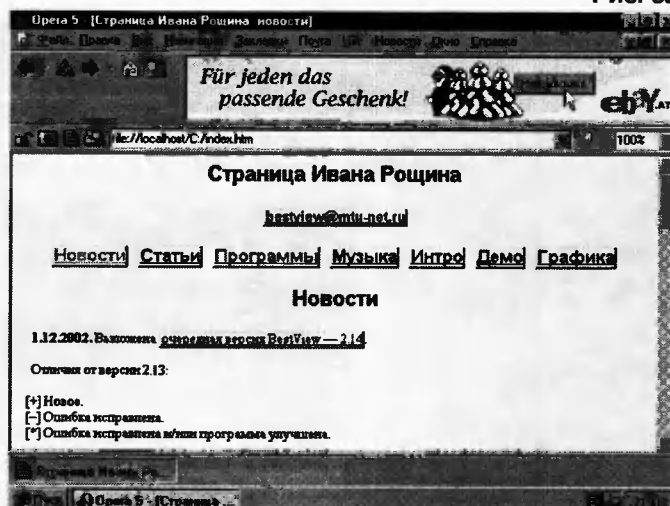


Рис. 3б

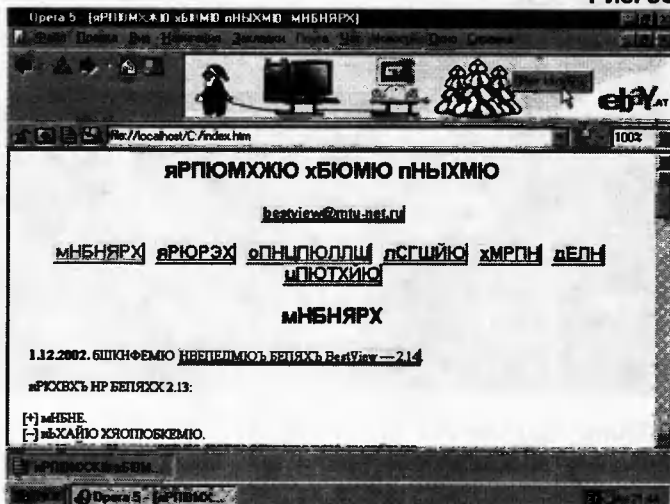
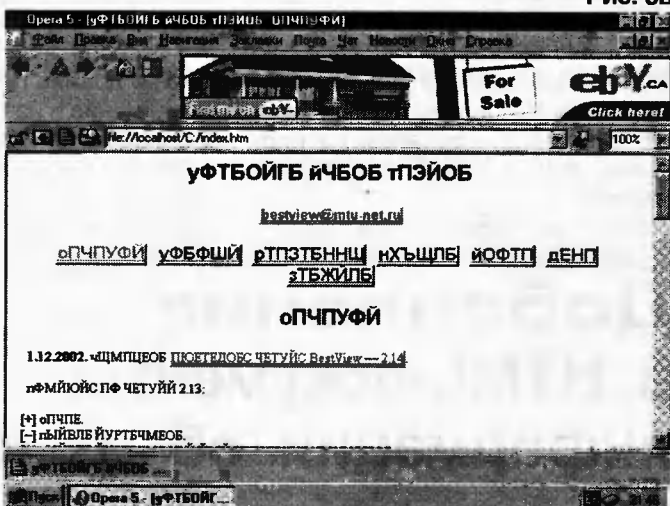


Рис. 3в



Если весь текст просмотрен, но требуемых двух символов не обнаружено — алгоритм не может определить кодировку текста.

Этот алгоритм, в отличие от описанных в [1, 2], может определить кодировку текста по единственному слову из строчных букв, начинающемуся с прописной буквы, и не

требует набора статистики — значит, определение кодировки будет более быстрым и точным. Но он годится только для выбора между кодировками WIN и KOI.

Текст программы

Полный листинг программы имеет значительный размер и поэтому не публикуется. Основой программы является функция `def_winkoi`, осуществляющая непосредственное определение кодировки, ее текст приведен ниже.

```
/*=====
Функция def_winkoi — определение кодировки (WIN или KOI)
указанного файла.
Выход: 0 — кодировку не удалось определить, 1 — WIN, 2 — KOI.
=====*/

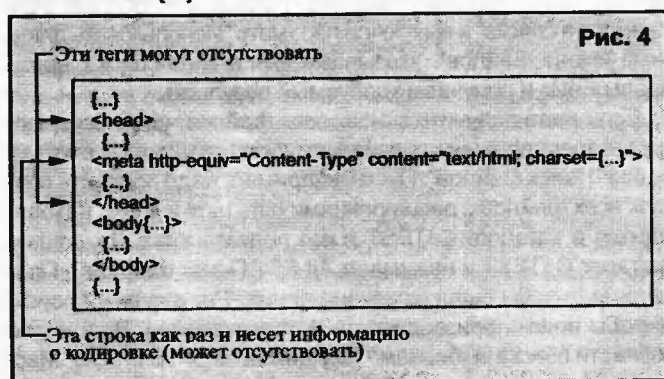
int def_winkoi (FILE* f)
{
    int c1,c2;

    rewind(f); /* Позиционирование на начало файла. */
    c1=fgetc_1(f); if (c1==EOF) return (0);
ml:
    c2=fgetc_1(f); if (c2==EOF) return (0);
    if (((c1>=0xc0) && (c1<=0xdf)) && ((c2>=0xe0) && (c2<=0xff)))
        return (1);
    if (((c2>=0xc0) && (c2<=0xdf)) && ((c1>=0xe0) && (c1<=0xff)))
        return (2);
    c1=c2;
    goto ml;
}
```

Полный листинг программы (`charset.c`), а также исполняемый файл (`charset.exe`) вынесены на веб-страницу журнала <http://radio-mir.com>

Работа программы

Как программа обрабатывает HTML-документ (HTML-файл)? На рис.4 приведены необходимые сведения о структуре HTML-документа. Для простоты показаны только те элементы структуры, которые имеют значение при обработке документа; остальные элементы (некоторые из них могут отсутствовать в конкретном документе) обозначены как {...}.



Сначала программа просматривает файл, пока не встретит одну из следующих строк: “charset=”, “<head>” или “<body>”. В процессе поиска информация о том, сколько процентов файла уже просмотрено, отображается на экране. Поиск производится без учета регистра символов, так как теги “head” и “body” могут быть записаны как прописными, так и строчными буквами.

В результате поиска возможны четыре различных случая — когда найдена (т.е. встретилась раньше других) первая, вторая или третья строка, и когда ни одна из них

не найдена. Рассмотрим эти случаи более подробно.

1. Если найдена строка “charset=”, то программа считает, что кодировка в этом файле уже указана. Программа выводит соответствующее сообщение, и на этом обработка файла заканчивается.

2. Если найдена строка “<head>”, то текущая позиция в файле запоминается (так как именно после тега “head” будет, при необходимости, добавлена информация о кодировке), и программа начинает поиск (с текущей позиции) одной из двух строк: “charset=” или “</head>”. Если первой встретится строка “charset=” — значит, кодировка в этом файле уже указана. Если же первой встретится строка “</head>”, или файл кончится до того, как будет найдена какая-то из искомых строк, то программа посчитает, что кодировка в этом файле не указана, и перейдет к определению кодировки.

3. Если найдена строка “<body>”, то текущая позиция в файле запоминается (так как именно перед тегом “body” будет, при необходимости, добавлена информация о кодировке), и программа переходит к определению кодировки.

4. Если ни одна из трех строк не найдена, программа делает вывод, что файл не является корректным HTML-документом. Тогда программа выводит сообщение “некорректный формат файла”, и на этом обработка файла заканчивается.

Определение кодировки происходит по алгоритму, рассмотренному в начале этой статьи. Полученный результат (“windows-1251”, “KOI8-R” или “не распознана”) печатается на экране.

Если кодировка не распознана, обработка файла заканчивается. Иначе программа вставляет в файл с позиции, запомненной ранее, одну из следующих двух строк:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<meta http-equiv="Content-Type" content="text/html; charset=KOI8-R">
```

Вставка строки производится следующим образом. Сначала обрабатываемый файл копируется во временный файл “\$\$\$\$\$.tmp”, находящийся в том же каталоге, затем временный файл открывается для чтения, а обрабатываемый файл — для записи. В обрабатываемый файл копируется часть временного файла от начала до позиции, с которой будет добавлена строка, затем записывается добавляемая строка, и копируется остаток файла. После этого временный файл удаляется.

Руководство пользователя

В командной строке необходимо указать либо имя одного обрабатываемого HTML-файла, либо символ “@” и сразу за ним — имя текстового файла-списка, где содержатся имена обрабатываемых HTML-файлов. Если вызываемый в командной строке файл находится в текущем каталоге, то путь к нему можно не указывать. В противном случае должен быть указан полный путь к файлу.

Ниже приведены несколько примеров запуска программы:

```
charset.exe C:\WORK\doc.htm
charset.exe doc.htm
charset.exe @C:\WORK\list.txt
charset.exe list.txt
```

В каждой строке файла-списка должно находиться одно имя обрабатываемого HTML-файла. Если все указанные в списке файлы находятся в том же каталоге, что и сам файл-список, то путь к этим файлам в списке можно не указывать. В противном случае для каждого файла в списке должен быть указан полный путь к нему.



Длина каждой строки в списке не должна превышать 128 символов, код конца строки должен быть (13,10), в последней строке списка код конца строки может отсутствовать.

На каждом из дисков, где находятся обрабатываемые файлы, свободного места должно быть не меньше, чем занимает самый длинный из обрабатываемых файлов, находящихся на этом диске. Дело в том, что, как уже упоминалось выше, при обработке файла программа создает в том же каталоге временный файл такой же длины, как и обрабатываемый. Также надо учесть, что размер обрабатываемых файлов возрастет за счет добавления информации о кодировке — длина каждого файла может увеличиться на 74 или 68 байтов.

Впрочем, даже если при работе программы не хватит места на диске, содержащее обрабатываемого в этот момент файла не будет потеряно (более подробно об этом будет рассказано ниже, при рассмотрении ошибок файловых операций).

В процессе работы программа выводит информацию о результате обработки каждого файла. Выглядит это примерно так:

```
C:\WORK\test_1.htm кодировка: KOI8-R.
C:\WORK\test_2.htm кодировка: windows-1251.
C:\WORK\test_3.htm кодировка не распознана.
C:\WORK\test_4.htm кодировка уже указана.
C:\WORK\test_5.htm некорректный формат файла.
C:\WORK\test_6.htm 54%
```

Если обрабатывается много файлов, и вам надо знать, каковы результаты обработки каждого файла (а на экране вся информация не умещается), можно использовать перенаправление вывода в текстовый файл. Например, если вы хотите перенаправить вывод программы в файл n.txt, надо просто добавить в конце командной строки "> n.txt" (без кавычек).

Если вывод перенаправлен в файл, при выполнении программы на экране не будет видно, какой файл в данный момент обрабатывается, а будут печататься только проценты для текущего файла. Впрочем, этого достаточно, чтобы видеть, что программа не зависла.

Программа возвращает операционной системе код завершения: 0 — признак успешного завершения, 1 — произошла ошибка при работе с файлами. При возникновении такой ошибки выполнение программы сразу же прерывается с выводом на экран монитора соответствующего сообщения. Каковы же причины некоторых ошибок, и что делать при их возникновении?

Если появилось сообщение об ошибке открытия на чтение одного из обрабатываемых файлов — возможно, этот файл отсутствует, или в списке почему-то оказалось длинное имя файла (например, если при создании списка в Dos Navigator'e вы забыли выключить параметр "Использовать длинные имена файлов").

Если появилось сообщение об ошибке открытия на запись одного из обрабатываемых файлов, это может говорить либо о том, что соответствующий диск защищен от записи (соответственно, защиту надо снять), либо о том, что для этого файла установлен атрибут "Read only" ("только чтение"). Тогда надо снять этот атрибут. Если в том же каталоге есть и другие подлежащие обработке файлы, лучше проверить и их — вдруг у них тоже установлен атрибут "Read only"? Для изменения атрибутов файла или группы помеченных файлов в Dos Navigator'e служит комбинация клавиш Alt-F.O

Если при обработке HTML-файла появилось сообщение "Ошибка открытия файла "\$\$\$\$\$\$.tmp" на чтение" — скорее всего, на диске, где находится обрабатываемый

файл, не хватило места для создания временного файла "\$\$\$\$\$\$.tmp", поэтому программа и не может его открыть. Попробуйте увеличить количество свободного места на диске и повторно запустить программу.

Если появилось сообщение об ошибке записи в обрабатываемый файл или об ошибке при закрытии этого файла — возможно, на диске не хватило места, чтобы поместился обработанный файл, и записать его не удалось. Но не беспокойтесь, первоначальное содержимое этого файла не пропадет, так как программа сначала копирует его во временный файл "\$\$\$\$\$\$.tmp", находящийся в том же каталоге. Таким образом, для восстановления исходного файла надо удалить исходный файл, а потом присвоить его имя временному файлу.

Если программа обрабатывала длинный список файлов, и ошибка возникла при обработке какого-то файла из середины списка, то после устранения причин ошибки не обязательно для продолжения обработки файлов удалять из списка имена уже обработанных файлов. Можно запустить программу, не изменяя список, при этом для уже обработанных файлов программа просто обнаружит, что в них уже присутствует атрибут "charset", и не изменит их.

Создание списка файлов

Для создания списка обрабатываемых HTML-файлов можно использовать команду DIR, перенаправив ее вывод в файл (но эта команда должна быть выполнена не в Windows, а в чистом DOS, иначе в полученном списке окажутся длинные имена файлов, а описываемая в этой статье программа не работает с длинными именами).

Например, если надо получить в файле C:\list.txt список всех файлов с расширением htm, находящихся в каталоге C:\WORK и его подкаталогах, то вводим такую команду:

```
dir c:\work\*.htm /s /b >c:\list.txt
```

Более удобный способ создания списка — с помощью DOS Navigator'a (я использую DN OSP 2.7.0 — www.dnosp.ru). Помечаем нужные файлы, нажимаем Alt-W — появляется окно "Создать список файлов". В этом окне должен быть включен параметр "Сохранять полные пути файлов в списке" и выключен параметр "Использовать длинные имена файлов". Указываем имя файла-списка, нажимаем Enter и получаем требуемый результат.

Если надо получить список всех файлов, удовлетворяющих определенным условиям, то сначала используем функцию "Поиск файлов". Пусть, например, надо получить список всех файлов с расширениями htm, html и shtml, находящихся в каталоге C:\TEST и его подкаталогах. Заходим в каталог C:\TEST и нажимаем Alt-F7 ("Поиск файлов"). Проверяем, чтобы была включена опция "Рекурсивный поиск" (чтобы поиск производился и в подкаталогах). В качестве области поиска выбираем "Текущий каталог". В строке "Маска файла" указываем "*.htm;*.html;*.shtml" (без кавычек). После окончания поиска помечаем найденные файлы (если ни один файл не помечен, то наиболее быстрый способ пометки всех файлов в панели — нажать "*" на цифровой клавиатуре) и нажимаем Alt-W, чтобы перейти к созданию списка помеченных файлов.

Литература

1. И.Рощин. Автоматическое определение кодировки текста. — Радиомир. Ваш компьютер, 2001, №10, С.34.
2. И.Рощин. Автоматическое определение кодировки текста — 2. — Радиомир. Ваш компьютер, 2002, №№5, 6.



ЛИСТЯ СТРАНИЦЫ

Краткая информация об интересном устройстве, позволяющем во многих случаях заменить ноутбук, приведена в заметке Майкла С.Лэски "Полезный гибрид КПК и ноутбука" (Мир ПК, №4/2003, С.47).

Компания AlphaSmart выпустила сверхпортативное устройство, названное Dapa. Оно имеет полноразмерную клавиатуру, работает под управлением операционной системы Palm OS 4.1 и способно сохранить 8 Мб информации, причем память можно увеличить за счет карт расширения Secure Digital. Его вес — 900 г, а стоимость в США — 399 USD.

Dapa объединяет в себе удобство нала-

донников с какой-то долей производительности и функциональных возможностей ноутбука. Его сенсорный монохромный экран с разрешением 560x160 пикселей имеет размер 19x5,7 см, т.е. в 3,5 раза больше, чем у большинства наладонных компьютеров, и отображает 11 строк текста 12-го кегля. Этого оказалось более чем достаточно, для того чтобы пользоваться входящими в комплект поставки полнофункциональной программой редактирования текстов AlphaWord и пакетом электронных таблиц QuickSheet. Альбомную ориентацию экрана можно изменить на книжную. В этом случае на нем помещается 42 строки, только гораздо более коротких.

Опытный образец, который опробовал ав-

тор, проработал от своей аккумуляторной батареи более 20 ч. В отсутствие сетевого адаптера он может работать и от трех батарей типа AAA.

AlphaWord и основные приложения для Palm OS (Date, Address, To Do и Memo) настроены таким образом, чтобы можно было целиком использовать весь экран устройства, а любые другие приложения Palm — в пределах меньшего по размеру (160x160 точек) экрана, типичного для устройств класса Palm.

Благодаря входящему в комплект поставки кабелю стандарта USB 1.1, перенос информации между Dapa и внешним ПК очень прост. Интересно и то, что в этом случае Dapa может работать в качестве клавиатуры ПК.

В 7-м номере нашего журнала за 2003 г. мы рассмотрели результаты тестирования 15-дюймовых ЭЛТ-мониторов. Предлагаем читателям ознакомиться с **результатами тестирования ЖК-мониторов** с аналогичной диагональю, приведенными в статье М.Бабенкова "Тестирование ЖК-мониторов с малым размером экрана" (КомпьютерПресс, №5/2003, С.114...125). Это позволит вам сделать правильный выбор при покупке очередного компьютера.

В последнее время ЖК-мониторы начали достаточно активно дешеветь, в то же время, их технические характеристики постоянно улучшаются. Так, средняя цена ЖК-монитора, обладающего разрешением 1024x768 (размер диагонали — 15 дюймов), сегодня в Москве составляет порядка 350 USD. Сейчас на рынке ЖК-мониторов присутствуют как крупные компании, работающие в области ЭЛТ- и ЖК-технологий, так и мелкие производители. Поэтому в обзор, наряду с мировыми лидерами в этой области, включены и пока еще мало известные у нас производители. И хотя по качеству картинки ЖК-мониторам пока не удалось догнать ЭЛТ-собратов, разрыв этот постоянно сокращается.

Первое и самое главное усовершенствование в ЖК-мониторах — это постоянное увеличение угла обзора. Именно угол обзора являлся самым слабым местом ЖК-мониторов. В лучших моделях, представленных в настоящем тестировании, угол обзора достигает значения 160° по горизонтали и 150° по вертикали.

Из улучшений следует также отметить и время отклика, сильно сократившееся за последнее время. В результате столь серьезный недостаток ЖК-мониторов как послесвечение достиг приемлемого уровня. Таким образом, сейчас ЖК-монитор позволяет вполне комфортно работать как с графическими приложениями, так и при просмотре динамического видео и в играх.

Для тестирования были отобраны популярные модели наиболее известных производителей с размером экрана 15 дюймов и рабочим разрешением 1024x768 пикселей. Мониторы не разделялись по наличию цифровых и аналоговых входов, а также встроенных динамиков. В тестировании участвовали как мультимедийные, так и обычные мониторы. Главное внимание уделялось качеству воспроизведения изображения, а наличие различных конструктивных особенностей позволяло модели завоевать допол-

нительные очки в конкурсе.

Тестировать ЖК-мониторы становится с каждым разом все сложнее, поскольку качество изображения у этих мониторов неуклонно улучшается, и оценить этот параметр невооруженным глазом зачастую просто невозможно. С точки зрения рядового пользователя, это означает, что отличить два монитора друг от друга по качеству воспроизведения картинки он будет не в состоянии. И самое удивительное во всем этом то, что качество монитора никак не зависит от его цены (во всяком случае, то качество, которое воспринимается визуально). Цена же в основном определяется различными дополнительными опциями. Это и цифровой вход, и наличие USB-концентратора, и встроенные колонки, и возможность поворота ЖК-матрицы на 90°, и т.д. Кроме того, сами ЖК-матрицы могут выполняться по различным технологиям. Например, есть недорогие TN+film-матрицы, а есть и дорогостоящие MVA-матрицы, для которых характерны большие углы обзора. Впрочем, подчеркивает автор, вопрос о том, какие матрицы лучше, лишен смысла. Во-первых, технологию изготовления матриц можно выяснить далеко не для всех мониторов. Во-вторых, у мониторов с одним и тем же названием могут быть разные матрицы. В-третьих, человеческое восприятие качества цветопередачи достаточно субъективно.

Итак, какой же из всего этого следует вывод? Прежде всего, поскольку, не вооружившись специальными измерительными приборами, заметить разницу между современными ЖК-мониторами практически невозможно, то руководствоваться нужно их стоимостью. Второе обстоятельство, на которое следует обратить внимание — это наличие дополнительных функций монитора. К примеру, кто-то собирается подключать монитор по цифровому входу, поэтому для него наличие этого входа является принципиальным условием. Для кого-то важна возможность одновременного подключения монитора сразу к двум компьютерам с выбором источника сигнала — в этом случае ему нужен монитор с двумя входами. А кто-то не желает переключать за довольно убогие встроенные колонки.

Автор подчеркивает условность выбора лучшего монитора в обзоре, поскольку любое сравнительное тестирование мониторов субъективно по своей природе, тем более, что, как уже отмечалось, все тестируемые

ЖК-мониторы практически неразличимы по качеству при визуальном наблюдении. Это означает, что мониторы, не удостоенные знака "Выбор редакции", или мониторы, оказавшиеся в конце рейтинга, вовсе не являются некачественными.

Тестировались мониторы Acer AL532, AOC LM520A, BenQ FP591, BLISS 1540, EIZO FlexScan L365, Hyundai ImageQuest Q15, IBM T541H, Iiyama BX3814UT, LG FLATRON L1510B, NEC MultiSync LCD1560VM, Philips 150B3Q, RoverScan SlimFlat Digital, Samsung SyncMaster 152T и ViewSonic VX500+. При этом все они одновременно подключались к одинаковым системным блокам с ОС Windows XP Home Edition. Их технические характеристики приведены в табл. 1 и 2, результаты тестирования — в табл. 3 и 4.

При тестировании одними из важнейших измеряемых характеристик мониторов являются яркость и равномерность яркости (максимальное отклонение от среднего значения в процентах) по всему полю экрана. Эти параметры измерялись в ходе испытания при помощи специального высокочувствительного прибора люксметра-яркомера ТКА-04/3. Фактически данный прибор заменял глаза, но, в отличие от них, был способен количественно оценить то, что человек пытается объяснить словами "более ярко" или "менее ярко". Кроме того, чувствительность яркомера значительно превосходит чувствительность глаз, поэтому он позволяет улавливать неравномерности, не заметные невооруженному глазу. Нужно учитывать и нелинейную природу человеческого восприятия яркости. К примеру, если визуально человеку кажется, что яркость одного объекта в два раза выше яркости другого, то с физической точки зрения их яркости в кд/м² должны различаться почти в десять раз! Этот пример наглядно показывает, насколько осторожно нужно относиться к цифровым измерениям приборов. Неравномерность по экрану может быть достаточно высокой, но на глаз вы этого просто не заметите.

Субъективность подхода заключается в том, что перед измерением необходимо одинаково настроить мониторы. Вариантов такой условно одинаковой настройки может быть несколько. Например, можно было бы установить максимальное значение яркости и контрастности и проводить измерения в этом режиме. Но практическое значение таких измерений крайне мало, поскольку при максимальных значе-



Табл. 1

Модель	Acer AL532	AOC LM520A	BenQ FP591	BLISS 1540	EIZO FlexScan L365	Hyundai ImageQuest Q15	IBM T541H
Экран	TFT LCD, 15"	TFT Active Matrix LCD, 15"	TFT LCD, 15"	TFT LCD, 15" Torisan	TFT LCD, 15"	A-Si TFT Active Matrix, 15"	TFT LCD, 15"
Максимальное количество цветов, млн.	16,7	16,7	16,2	16	16,7	16,2	>16
Угол обзора по горизонтали/по вертикали, град.	140/120	120/100	160/150	140/120	160/150	150/120	160/130
Контрастность	500:1	300:1	450:1	450:1	450:1	300:1	300:1
Яркость, кд/м ²	300	250	500	250	300	250	250
Время реакции пиксела, мс	25	40	16	-	25	-	40
Видеосигнал	D-Sub	D-Sub	D-Sub/DVI-D	D-Sub	D-Sub/DVI-D	D-Sub	D-Sub/DVI-D
Энергопотребление в обычном режиме/в режиме энергосбережения, Вт	36/-	50/-	50/-	-	35/3	40/-	25/-
Габариты, мм	348x341x131	356x360x165	401x392x182	-	340x369x157	350x357x201	356x365x202
Вес монитора, кг	3,2	3,3	5,3	3,2	4,1	3,5	4,3
Цена, USD	390	340	520	367	435	320	435

Табл. 2

Модель	Iiyama BX3814UT	LG FLATRON L1510B	NEC MultiSync LCD1560VM	Philips 150B3Q	RoverScan SlimFlat Digital	Samsung SyncMaster 152T	ViewSonic VX500+
Экран	TFT LCD, 15"	TFT LCD, 15"	TN+film TFT, 15"	TFT LCD, 15"	TFT LCD, 15"	A-Si TFT/TN, 15"	15.0" color TFT active matrix XGA LCD
Максимальное количество цветов, млн.	16,2	>16,1	16,7	16,7	16,7	-	16,7
Угол обзора по горизонтали/по вертикали, град.	120/100	120/120	160/150	150/110	150/110	160/150	135/125
Контрастность	300:1	350:1	450:1	350:1	450:1	450:1	500:1
Яркость, кд/м ²	320	250	300	250	300	350	300
Время реакции пиксела, мс	30	-	25	<40	13	-	25
Видеосигнал	D-Sub	D-Sub	D-Sub/DVI-D	D-Sub	D-Sub/DVI-D	D-Sub/DVI-D	D-Sub/DVI-D
Энергопотребление в обычном режиме/в режиме энергосбережения, Вт	48/3	28/3	35/2	23/2	48/0,8	41/-	25/-
Габариты, мм	384x376x180	356x229x380	344x333x181	343x342x165	321x338x152	357x346x184,5	360x342x140
Вес монитора, кг	3,4	4	5,1	4,3	3,25	2,9	3,9
Цена, USD	310	320	350	441	329	403	440

ниях яркости и контрастности работать за монитором невозможно. Поэтому использовался несколько иной принцип настройки мониторов. Прежде всего, на всех мониторах устанавливалась одна и та же цветовая температура 9300°K и никакой подстройки по отдельным цветовым каналам не производилось (использовались заводские настройки). Настраивались лишь яркость и контрастность монитора. Для осуществления такой настройки на экран монитора выводилась шкала, представляющая собой три поля черного цвета с заполнением 100, 95 и 90% и три поля белого с заполнением 100, 95 и 90% (или 0, 5 и 10% заполнения по черному цвету). Путем регулировки яркости и контрастности добивались того, чтобы стали различимыми переходы между градациями черного и белого на данной шкале. Важно, чтобы настройку всех мониторов производил один человек в присутствии представителей компаний, что обеспечивает относительную одинаковость настройки. Данная настройка характеризуется тем, что при установленных значениях яркости и контрастности достигается комфортность работы за ЖК-монитором (зрение не напрягается), поэтому такая настройка считается комфортной.

После настройки мониторов проводилось измерение яркости (в кд/м²) на белом фоне, для чего использовалась утилита Monitor Matter CheckScreen v.1.2. Измерения производились в пяти точках — в центре и в углах экрана. Далее определялись средняя яр-

кость и неравномерность яркости, рассчитываемая как максимальное отклонение от среднего значения в процентах.

Кроме этих параметров, рассчитывался запас по яркости. Для этого измерялась разность между средней комфортной яркостью и максимальной яркостью, которая может быть установлена для данного монитора. Данная характеристика представляется важной прежде всего потому, что комфортная настройка монитора зависит от внешнего освещения. Если вы работаете в темной комнате, то вам потребуется один уровень яркости монитора, при котором можно добиться комфортной работы, если же речь идет о комнате с ярким дневным освещением, то для создания комфорта вам придется добавить и контрастности, и яркости. В этом случае и сказывается тот самый запас по яркости, которым вы сможете воспользоваться. Второй аспект, подчеркивающий важность данного параметра, — разные приложения (особенно игровые) требуют различного уровня яркости. И тот уровень комфортной яркости, который необходим при работе с офисными приложениями, может показаться явно недостаточным для какой-нибудь игры. В данном случае опять критичен запас монитора по яркости. Ну и последний момент, возможно, не столь актуальный, как два первых, — время жизни монитора. Лампы подсветки, используемые в мониторах, со временем (измеряемым годами) теряют свою яркость. Как следствие, тот уровень яркости, которого

было достаточно при покупке монитора, может оказаться неудовлетворительно низким через три года. В этом случае может помочь хороший запас по яркости, если он есть.

Кроме измерительных тестов на определение неравномерности яркости по экрану монитора и запаса по яркости, использовались также и традиционные субъективные тесты на качество цветопередачи. Для этого на экраны мониторов выводилось изображение электронного оригинала обложки журнала КомпьютерПресс, которое визуально сравнивалось с печатным вариантом полиграфического качества. Осмотр производился как под прямым углом к экрану, так и под другими углами. Целью этого эксперимента было дать общую оценку насыщенности изображения оттенками различных цветов, а также определить степень ухудшения качества изображения при изменении угла зрения.

С помощью пакета Monitor Matter CheckScreen оценивались правильность передачи цветовых переходов (градиентов) и четкость контрастных переходов, а также определялась скорость реакции при быстром перемещении по экрану, поскольку, как известно, ЖК-мониторы обладают некоторой инертностью при перерисовке изображения.

В завершение тестирования оценивались дизайн монитора, его конструктивные особенности (мультимедийность, наличие цифрового интерфейса, USB-хаба, возможность поворота экрана и т.д.) и удобство настройки, то есть функциональность



Табл. 3

Модель	Весовой коэффициент	Acer AL532	AOC LM520A	BenQ FP591	BLISS 1540	EIZO FlexScan Q15	Hyundai ImageQuest Q15	IBM T541H
Нормированное максимальное отклонение яркости	12,5	0,24	0,36	0,08	0,19	0,29	0,15	0,15
Нормированный запас яркости	11,5	0,44	0,33	0,86	1,0	0,62	0,17	0,40
Качество цветопередачи при выводе графического изображения	10,8	0,94	0,96	0,98	0,95	0,89	0,96	0,81
Качество графического изображения при просмотре по углом 45° по горизонтали	8,5	0,88	0,88	1,0	0,88	0,84	0,93	0,86
Правильность передачи цветовых переходов (градиентов)	10,6	0,97	1,0	1,0	1,0	0,97	1,0	0,97
Четкость контрастных переходов	9,5	0,96	1,0	1,0	0,98	0,94	1,0	0,98
Скорость реакции при быстром перемещении по экрану	9,8	0,96	1,0	1,0	0,98	1,0	1,0	1,0
Удобство регулировок монитора	6,8	0,93	0,82	0,91	0,84	0,98	0,84	0,80
Дизайн	9,4	0,98	0,69	1,0	0,75	0,96	0,73	0,65
Мультимедие	4,2	0,58	0,33	0,92	0,48	0,73	0	0,15
Другое	6,4	0,27	0,14	1,0	0,14	0,14	0,14	0
Интегральный показатель качества	100	74,15	70,35	85,73	76,70	76,07	65,44	64,01
Показатель "качество/цена"		19,01	20,69	16,49	20,90	17,49	20,45	14,72

Табл. 4

Модель	Iiyama BX3814UT	LG FLATRON L1510B	NEC MultiSync LCD1560VM	Philips 150B3Q	RoverScan SlimFlat Digital	Samsung SyncMaster 152T	ViewSonic VX500+
Нормированное максимальное отклонение яркости	0,18	0,28	0,23	1,0	0,23	0,91	0,72
Нормированный запас яркости	0,18	0,27	0,31	0,71	0,52	0,58	0,78
Качество цветопередачи при выводе графического изображения	0,98	0,85	0,91	1,0	0,89	0,99	0,96
Качество графического изображения при просмотре по углом 45° по горизонтали	0,88	0,91	0,93	0,95	0,88	1,0	0,91
Правильность передачи цветовых переходов (градиентов)	0,97	0,97	0,97	1,0	0,97	0,99	0,97
Четкость контрастных переходов	1,0	0,94	0,98	0,98	0,96	0,97	0,94
Скорость реакции при быстром перемещении по экрану	1,0	0,96	1,0	1,0	0,98	0,98	0,98
Удобство регулировок монитора	0,64	0,98	0,86	0,95	0,64	1,0	1,0
Дизайн	0,60	0,81	0,83	0,94	0,73	1,0	1,0
Мультимедиа	0	0	0,60	1,0	0	0	0,73
Другое	0	0,24	0,44	0,95	0,14	0,24	0,14
Интегральный показатель качества	63,42	67,96	72,74	94,82	68,48	84,26	84,89
Показатель "качество/цена"	20,46	21,24	20,78	21,50	20,81	20,91	19,29

и навигация по пунктам меню настройки.

Чтобы сравнить мониторы не только по отдельным характеристикам, но и в целом, применялась методика учета оценок отдельных характеристик монитора в едином интегральном показателе качества. Для этого каждой отдельной характеристике монитора придавался некоторый весовой коэффициент, определяющий ее значимость (первый столбец табл.3). Для получения интегрального показателя качества весовые коэффициенты умножались на соответствующие им нормированные оценки отдельных характеристик и складывались. Нормирование оценок

проводилось таким образом, что лучший результат в данном тесте принимался за единицу. Таким образом, если для какого-либо монитора нормированная оценка в тесте составляет 0,8, это означает, что данный результат на 20% ниже максимального.

Оценки в каждом субъективном тесте выставлялись экспертной группой по десятибалльной шкале: чем выше оценка, тем лучше показатель; в случае отсутствия какого-либо из критериев ставилась оценка 0.

Помимо определения наиболее качественного монитора, определялась также оптимальность покупки монитора, для чего

подходят и для домашнего ПК — ведь при такой большой емкости они имеют весьма малые габариты, они тише и потребляют меньше энергии, чем их 3,5-дюймовые собратья. Правда, стоят они дороже. Адаптер для подключения 2,5-дюймовых винчестеров к ПК можно купить примерно за 15 евро.

Все протестированные жесткие диски имеют интерфейс Ultra ATA/100. Тем не менее, на практике такая скорость недостижима, так как в настоящее время IDE-диски имеют скорость передачи данных немногим более 50 Мб/с и даже не приближаются к пропускной способности Ultra ATA/100 (100 Мб/с). Од-

интегральная оценка качества монитора делилась на его цену. Чем выше качество монитора и ниже его стоимость, тем больше соотношение "качество/цена", определяющее выгодность покупки.

В номинации "Самый качественный монитор" знаком "Выбор редакции" были отмечены модели Philips 150B3Q, BenQ FP591 и ViewSonic VX500+. В номинации "Оптимальная покупка" отмечены мониторы Philips 150B3Q, LG FLATRON L1510B и Samsung SyncMaster 152T, у которых соотношение цены и качества получилось наиболее привлекательным.

Однако при включении двух дисков в RAID-массив такая пропускная способность шины становится ограничением для передачи данных.

Имеет место некоторое плутовство с гигабайтами, к сожалению, принятое в данной области. Все производители жестких дисков указывают завышенное значение емкости привода. Это связано с тем, что в данной области принято определять 1 Гб как 1 млрд. байтов. Однако в двоичной системе счисления 1 Гб = 1073741824 байта. Из этого следует, что нужно вычитать примерно 7% из указанной производителем емкости, для того чтобы рассчитать реальную емкость, поэто-

Результатам тестирования 2,5-дюймовых жестких дисков. используемых в ноутбуках, посвящена статья "Высокая плотность" (CHIP, №3/2002, 54...59).

Увеличение плотности записи положительно сказывается на емкости 2,5-дюймовых жестких дисков. Наибольшая емкость на данный момент составляет 60 Гб, но уже готовы и более емкостные модели. Так, например, компания Fujitsu в начале марта анонсировала винчестеры серии MTH нового поколения емкостью до 80 Гб. Плотность записи на таких дисках составляет 69,3 Гбит/кв.дюйм. Имея емкость 60 и 80 Гб, маленькие диски вполне



му в сводной таблице характеристик приведены точно такие же значения емкостей дисков, какие отображаются при загрузке Windows и являются реальными.

Современные жесткие диски теоретически должны обладать быстрыми буферами объемом 2, 8 или 16 Мб и быстро передавать многократно считываемые данные. Однако эффективность кэша определяется не только его емкостью, но и его алгоритмом. Более четко просматривается влияние скорости вращения привода диска: чем быстрее крутится диск, тем меньше время доступа и выше скорость передачи данных и рабочая температура.

Для «ноутбучных» винчестеров, в отличие от высокоскоростных «настольных», стандартными скоростями вращения шпинделя являются 4200 и 5400 об/мин. Так как тепловыделение у винчестера пропорционально потребляемой мощности, следует обратить внимание на диски для ноутбука с энергопотреблением по возможности менее 2 Вт, чтобы продлить время использования аккумулятора и снизить неприятные ощущения от нагрева корпуса.

Пока проблема барьера в 128 Гб не касается 2,5-дюймовых винчестеров, вследствие того что их емкость не достигает этого предела. Материнские платы предыдущего поколения имеют 28-битную адресацию, которая позволяет обращаться только к 2²⁸ секторов по 512 байт — в целом точно 128 Гб. Если диск более объемистый, старые контроллеры узнают лишь первые 128 Гб, оставшаяся же часть будет неиспользованной, по крайней

мере, при начальной загрузке ПК. В Windows теоретически возможно после загрузки обращаться к более высоким областям, так как операционная система с помощью собственных драйверов получает доступ ко всему диску. Тем не менее, на практике зачастую это может не получиться. Для совсем старых ноутбуков есть еще более жесткое ограничение емкости винчестера, составляющее 8,4 Гб. Преодолеть упомянутый предел в 128 Гб позволит 48-битная адресация, которую имеют все новые контроллеры и которая в течение многих лет не будет создавать ограничения.

Для всех ПК с материнскими платами на базе чипсета Intel 8xx имеется очень простой способ оптимизации работы жесткого диска. Утилита Intel Application Accelerator обладает рядом особенностей, улучшающих работу диска. IAA поддерживает 48-битную адресацию данных, уменьшает время загрузки ОС, включает в себя утилиту диагностики, позволяет автоматически выбирать наилучший режим DMA и т.д. Начиная с версии 2.3, Intel отказалась от поддержки чипсетов для мобильных компьютеров, сконцентрировав усилия на высокопроизводительных ПК. Так что остается довольствоваться более ранней версией 2.2.2. Ее можно загрузить со страницы www.intel.com/support/chipsets/iaa/mobile.htm. Перед инсталляцией также необходимо обратить внимание на то, что должна быть установлена современная версия утилиты Intel Chipset Setup, поддерживающая все версии чипсетов (www.intel.com/support/

chipsets/inf). Уже это повышает производительность на 20% по сравнению со стандартным драйвером Windows.

IAA также позволяет регулировать шум жесткого диска, разумеется, за счет потерь в скорости. Управлять можно только головкой чтения/записи и тем самым влиять на типичный шум при обращении к диску. На постоянный шум мотора управление громкостью работы не влияет.

Владельцы материнских плат с чипсетом не от Intel с этой целью могут использовать утилиты для дисков Hitachi. Со страницы www.hgst.com/hdd/support/download.htm можно загрузить множество утилит, позволяющих протестировать и оптимизировать работу жесткого диска. Для владельцев мобильных компьютеров будет полезна утилита Power Booster, предназначенная для управления электропитанием.

Усиливается тенденция к миниатюризации. Фирмы Hitachi и Toshiba скоро выпустят жесткие диски нового 1,8-дюймового формата емкостью 20 и 40 Гб. Первая, кроме того, использует разработанную IBM технологию MicroDrive для выпуска мини-винчестера 4 Гб в крохотном 1-дюймовом формате.

Тестировались диски 2060AT и 2030AT от Fujitsu, 2018GAP и 4019GAX ф. Toshiba, DA-20F, IC25N020ATCS04, IC25N030ATCS04, IC25N040ATCS04 и IC25N020ATCS05 ф. Hitachi. Их технические характеристики приведены в табл.1, результаты тестирования — в табл.2, оценки редакции «CHIP» — в табл.3.

Табл. 1

Производитель	Fujitsu	Fujitsu	Toshiba	Toshiba	Hitachi	Hitachi	Hitachi	Hitachi	Hitachi
Серия	MHS	MHS	MK	MK	DK23	Travelstar 40 GN	Travelstar 40 GN	Travelstar 40 GN	Travelstar 40 GN
Модель	2060AT	2030AT	2018GAP	4019GAX	DA-20F	IC25N020ATCS04	IC25N030ATCS04	IC25N040ATCS04	IC25N020ATCS05
Реальная емкость, Гб	56	28	18	37	18	18	28	37	18
Скорость, об/мин	4200	4200	4200	5400	4200	4200	4200	4200	5400
Кэш, Мб	2	2	2	16	2	2	2	2	8
Цена, евро	220	115	105	175	115	105	125	150	115
Цена 1 Гб, евро	3,93	4,11	5,83	4,73	6,39	5,83	4,46	4,05	6,39

Табл. 2

Модель	Скорость передачи данных, чтение/запись, Мб/с	Время доступа, чтение/запись, мс	Громкость, сон	Энергопотребление, Вт
2060AT	19,9/19,7	20,5/11,4	1,2	1,5
2030AT	20,5/20,3	19,8/10,5	1,2	1,7
2018GAP	18,5/18,5	21,7/21,2	1	1,6
4019GAX	22,8/24,4	18,8/13,6	2,2	2,1
DA-20F	19,1/18,3	20,7/21,6	1,1	1,7
IC25N020ATCS04	16,5/15,3	19/19,4	1,3	1,7
IC25N030ATCS04	16,6/15,7	21,5/19,7	1,6	1,7
IC25N040ATCS04	16,3/15,3	18,8/18,8	1,7	1,7
IC25N020ATCS05	20,5/20,6	16,8/11,5	3,3	2

Табл. 3

Модель	Уровень шума	Энергопотребление	Скорость передачи данных	Емкость	Время доступа	Общая оценка
2060AT	86	100	84	79	87	89
2030AT	89	91	67	78	91	86
2018GAP	100	94	76	78	68	88
4019GAX	47	72	100	94	88	74
DA-20F	93	90	79	78	69	85
IC25N020ATCS04	82	90	67	82	76	81
IC25N030ATCS04	66	91	68	62	71	76
IC25N040ATCS04	61	90	67	76	76	74
IC25N020ATCS05	32	78	87	97	100	70

При тестировании измерялась громкость работы винчестера. Она составляет 30% общей оценки устройства. Потребление энергии (30% общей оценки) играет решающую роль для ноутбуков при питании от батарей. Для измерения энергопотребления использовался мультиметр ВвеТех 70, при помощи которого замерялись пиковые значения тока, потребляемого диском. Скорость передачи данных (20%) измерялась при помощи DOS-утилиты Diskbench. Оценка емкости диска и времени доступа к данным составляют по 10% общей оценки.

Звания «Лучший продукт» и «Оптимальный выбор» были присвоены приводе Fujitsu 2060AT. Он экономнее всех обходится с энергопотреблением ноутбука. Кроме того, цена 220 евро ввиду большой емкости (60 Гб) также является вполне адекватной. Те, кто работает с ноутбуком только дома от электросети, могут рассматривать в качестве варианта Travelstar 40 GNX от Hitachi, который поставил рекорд по скорости передачи данных, но он производит наибольший шум. О тишине в корпусе заботится диск Toshiba MK 2018GAP емкостью 20 Гб, имеющий шум только 1,0 сон.



А.КОРБИТ, А.ЩЕРБАКОВ,
г.Минск, БГУИР.

Программирование в среде Visual C++ 6.0

Интерфейс графических устройств Windows

Цель этой статьи — изучить методы работы с графическим интерфейсом (GDI).

1. Введение в GDI

Любая Windows-программа выводит информацию на доступное графическое устройство с помощью множества специальных функций, образующих так называемый **GDI-интерфейс** (от англ. graphics device interface). GDI-интерфейс является аппаратно-независимым. Windows абстрагирует вывод на графическое устройство от технических особенностей его работы. Таким образом, графическое устройство в Windows — это монитор, принтер, плоттер.

В MFC интерфейсу GDI соответствует **класс CDC**. Данный класс содержит методы для вывода разнообразной графической информации — прямоугольников, линий, эллипсов, текста. Он также позволяет устанавливать цвет и масштаб изображений. Методы, обеспечивающие вывод, например, на экран монитора, используют **контекст устройства**, который в приложении на базе MFC является объектом класса CDC. Таким образом, класс CDC является базовым классом контекста устройств, содержит ряд атрибутов контекста устройств и методы, управляющие ими. Для того чтобы использовать контекст устройства, необходимо создать объект CDC.

На практике пользуются двумя классами, производными от CDC:

- **CClientDC** — управляет контекстом экрана монитора, связанного с клиентской областью окна;
- **CWindowDC** — управляет контекстом экрана монитора, связанного и с клиентской областью окна, и с его системной частью.

Наиболее часто используемые функции класса CDC:

- **Ellipse** — рисует эллипс;
- **Polygon** — рисует многоугольник;
- **Rectangle** — рисует прямоугольник;
- **RoundRect** — рисует прямоугольник со скругленными углами;
- **Chord** — рисует сегмент эллипса;
- **Pie** — рисует сектор эллипса;
- **LineTo** — рисует линию.

При использовании функции **LineTo** рисование осуществляется с текущей позиции пера. Для ее изменения существует функция **MoveTo**.

Для управления цветом фона и цветом линий существуют специальные объекты GDI. Цветом фона управляет объект "кисть", которому соответствует класс **CBrush**, а цветом линий управляет объект "перо", которому соответствует класс **CPen**.

Рассмотрим создание объекта "перо". Этот процесс может происходить в два этапа: сначала объявляется

переменная типа **CPen**, потом вызывается функция **CreatePen**, в которую передают тип, ширину и цвет линии.

Для указания типов линии используют константы:

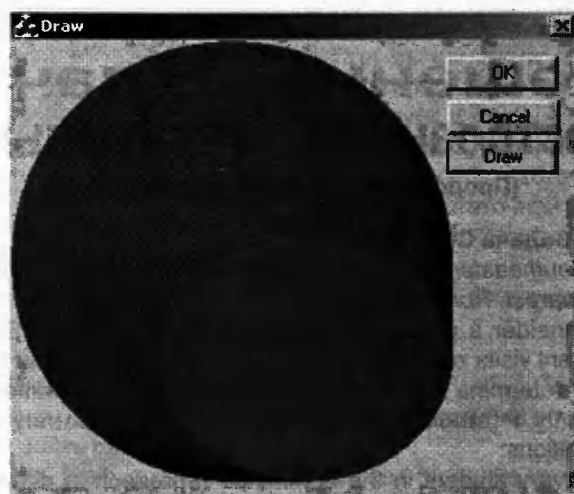
- **PS_SOLID** — сплошная линия;
- **PS_DASH** — штриховая линия;
- **PS_DASHDOT** — штрих-пунктирная линия.

После создания объекта "перо", его необходимо выбрать в контекст устройства рисования. Для этого вызывают метод **SelectObject** класса **CDC**, передавая ему указатель на объект "перо".

2. Пример написания программы

Необходимо написать программу, которая после нажатия кнопки выводит на экран изображение усеченного конуса красного цвета с синими гранями, образованного эллипсом (рис.1).

Рис. 1



Для решения данной задачи создайте приложение на основе диалогового окна.

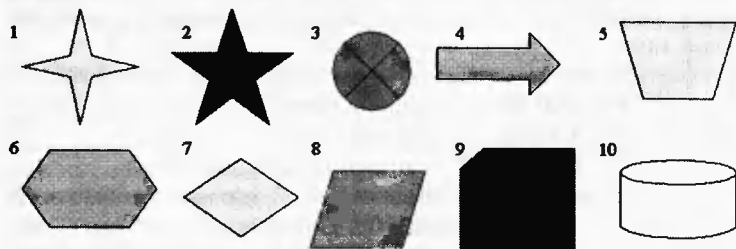
Поместите на форму кнопку, при нажатии которой будет выполняться функция рисования.

Код обработчика нажатия кнопки имеет вид:

```
void CDraw::OnButton1()
{
    //TODO: Add your control notification handler code here
    // Объявляем класс, производный от класса CDC
    CClientDC dc(this);
    //Класс CClientDC предназначен для рисования в диалоговом окне
    // Объявляем класс кисти
    CBrush br;
    // Создаем сплошную красную кисть
    br.CreateSolidBrush(RGB(255,0,0));
    // Выбираем кисть в контекст устройства
    dc.SelectObject(&br);
    // Объявляем класс пера
    CPen p;
    // Создаем сплошное перо, толщиной 1, синего цвета
    p.CreatePen(PS_SOLID,1,RGB(0,0,255));
    // Выбираем перо в контекст устройства
    dc.SelectObject(&p);
    // Рисуем эллипс с заданными координатами
    for(int x=0,int y=0;x<100;x+=3,y+=4)
        dc.Ellipse(x,y,300,300);
}
```




Рис. 2



3. Индивидуальное задание

Требуется нарисовать один из рисунков, представленных на рис.2. Фигуры должны быть окрашены в произвольные цвета.

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.

Решение задач на графах построением максимального потока

М.ДОЛИНСКИЙ,
г.Гомель.

(Продолжение. Начало в №№8-9/2003)

6. Задача COURSES

(Southeastern European Regional Programming Contest
Bucharest, Romania, October 21, 2000)

Consider a group of N students and P courses. Each student visits zero, one or more than one courses. Your task is to determine whether it is possible to form a committee of exactly P students that satisfies simultaneously the conditions:

- every student in the committee represents a different course (a student can represent a course if he/she visits that course);

- each course has a representative in the committee.

Your program should read sets of data from a text file. The first line of the input file contains the number of the data sets. Each data set is presented in the following format:

```
P N
Count1 Student11 Student12 :: Student1Count1
Count2 Student21 Student22 :: Student2Count2
.....
CountP StudentP1 StudentP2 :: StudentPCountP
```

The first line in each data set contains two positive integers separated by one blank: P ($1 \leq P \leq 100$) - the number of courses and N ($1 \leq N \leq 300$) - the number of students. The next P lines describe in sequence of the courses - from course 1 to course P , each line describing a course.

The description of course i is a line that starts with an integer $\text{Count } i$ ($0 \leq \text{Count } i \leq N$) representing the number of students visiting course i . Next, after a blank, you'll find the $\text{Count } i$ students, visiting the course, each two consecutive separated by one blank. Students are numbered with the positive integers from 1 to N .

There are no blank lines between consecutive sets of data. Input data are correct. The result of the program is on the standard output. For each input data set the program prints on a single line "YES" if it is possible to form a committee and "NO" otherwise. There should not be any leading blanks at the start of the line.

An example of program input and output:

Input	Output
2	YES
3 3	NO
3 1 2 3	
2 1 2	
1 1	
3 3	
2 1 3	
2 1 3	
1 1	

Это задача с полуфинала студенческого чемпионата мира по программированию, поэтому ее полные условия представлены на английском языке. Краткий перевод может быть таким.

Для каждого из P учебных курсов заданы списки студентов, которые посещают этот курс. Требуется выяснить, возможно ли из этих студентов составить представительный комитет таким образом, чтобы в комитет входили ровно P различных студентов, так, чтобы каждый курс имел своего представителя в комитете.

Это также задача на нахождение максимального количества паросочетаний типа "студент—учебный курс". Если это количество равно количеству учебных курсов P , то ответ — YES, иначе — NO.

Поскольку по условиям задачи в одном файле может содержаться несколько тестов, то тело главной программы несколько изменится:

```
begin
  assign( input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  readln(TN);
  for iTN:=1 to TN do
    begin
      InputData;
      MaxFlow;
      OutputData;
    end;
  close(input); close(output);
end.
```



Процедура вывода ответа OutputData будет выглядеть следующим образом:

```

Procedure OutputData;
begin
  Max:=0;
  for i:=1 to P do Max:=Max+f[0,i];
  If Max=P
  then writeln('YES')
  else writeln('NO');
end;

```

А процедура ввода исходных данных InputData так:

```

Procedure InputData;
begin
  readln (P,N); (Количество: P - курсов, N - студентов)
  Finish := P+N+1; (Курсы + студенты + сток)

  for i:=0 to Finish do (Инициализируем пустой граф)
  begin
    (Исток-курсы-студенты-сток)
    for j:=0 to Finish do a[i,j]:=0;
    ka[i]:=0;
  end;

  (Добавляем вершину - исток)
  ka[0]:=N; (Из истока - N дуг - к каждому курсу)
  for i:=1 to P do
  begin
    a[0,i]:=i; (Номер вершины, соединенной с истоком)
    c[0,i]:=1; (Ее вес)
  end;

  (Добавляем вершину - сток)
  for i:=P+1 to P+N do
  begin
    ka[i]:=1; (Одна дуга для каждого студента)
    a[i,1]:=Finish;
    c[i,Finish]:=1;
  end;

  (Добавляем дуги от курсов к студентам)
  for i:=1 to P do
  begin
    read (ka[i]); (Кол-во дуг от курса к студентам)
    for j:=1 to ka[i] do (Для каждой дуги)
    begin
      read(t); (Куда ведет - номер студента)
      a[i,j]:=P+t; (Куда ведет - номер вершины)
      c[i,a[i,j]]:=1; (Ее вес)
    end;
  end;

  Init; (Нулевой поток f(u,v))
end;

```

7. Пример нахождения максимального потока на графе

Вернемся к задаче о бракосочетании (рис.7).

Добавляем две вершины — исток и сток (рис.8).

Рис. 7

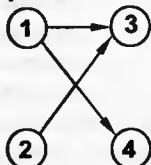
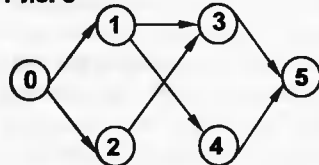


Рис. 8



Сначала неформально проиллюстрируем на данном примере изложенный в [13] алгоритм Форда-Фалкерсона для построения максимального потока.

Все начинается с нулевого потока $f[i,j]=0$ для всех i,j .

Далее строится остаточная пропускная способность $cf[i,j]$ (из вершины i в вершину j) для всех дуг — $cf[i,j] = c[i,j] - f[i,j]$. Напомним, что в этой задаче $c[i,j]=1$ если есть дуга из вершины i в вершину j .

Затем строится произвольный дополняющий путь из истока (0) в сток (5), увеличивающий поток. Путь называется увеличивающим потоком, если он состоит *только* из дуг с неотрицательной остаточной пропускной способностью $cf[i,j]$. Пусть, например, это путь — 0-1-3-5.

Далее выбирается минимальное значение из весов всех дуг увеличивающего пути. В нашем случае минимальное значение ищется из величин: $cf[0,1]$ — не ограничено; $cf[1,3]$ — 1; $cf[3,5]$ — не ограничено. Минимальное значение — 1.

Далее, для каждой дуги из увеличивающего пути увеличиваем поток по этой дуге на найденное минимальное значение и получаем граф, приведенный на рис.9.

Рис. 9

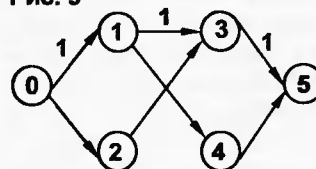
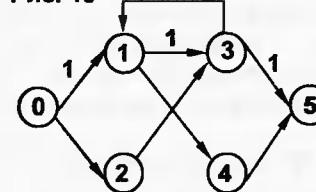


Рис. 10



$cf[3,1] = c[3,1] - f[3,1] = 0 - (-1) = 1$!!!

Теперь при попытке построения нового увеличивающего пути нам удается найти путь:

0 → 2 → 3 → 1 (!!!) → 4 → 5

Для всех дуг этого пути — $cf[i,j] > 0$!!!

Минимальное значение из всех $cf[i,j]$ увеличивающего пути опять равно 1.

Снова прибавляем это минимальное значение ко всем дугам увеличивающего пути и получаем $c[3,1] = -1 + 1 = 0$ (!). Результирующий граф приведен на рис.11.

Рис. 11

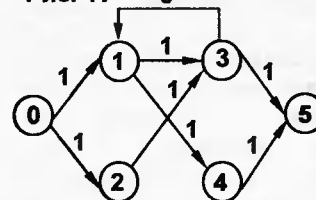
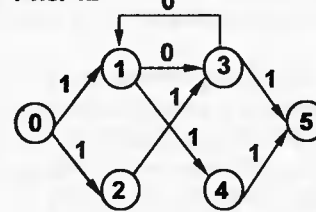


Рис. 12



И снова для каждой дуги (u,v) из увеличивающего пути строим обратные дуги с отрицательным потоком — $f[v,u] = -f[u,v]$. В частности, $f[1,3] = -f[3,1] = 0$. Окончательно получаем граф, приведенный на рис.12.

Теперь найти увеличивающий путь невозможно. Алгоритм закончил свою работу. Осталось проинтерпретировать результаты. Мы построили следующий поток:

— из вершины 0 (истока) в вершину 1 и вершину 2 — $f[0,1]=1$, $f[0,2]=1$;

— из вершины 1 в вершину 4 — $f[1,4]=1$;

— из вершины 2 в вершину 3 — $f[2,3]=1$;

— из вершины 3 в вершину 5 — $f[3,5]=1$;

— из вершины 4 в вершину 5 — $f[4,5]=1$.

Максимальный поток равен максимальному исходящему потоку $f[0,1]+f[0,2]=2$ и равен максимальному входящему потоку $f[3,5]+f[4,5]=2$.

(Продолжение следует)



ИССЛЕДОВАНИЕ ПРОГРАММ

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

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КРЭКИНГА

(Продолжение. Начало в №№8-9/2003)

“Критические дни” программ

*Все боится времени, но само время боится Пирамид.
Египетская пословица*

После того как мы успешно установили программу, неизменно наступает торжественный момент ее первого запуска (конечно, если инсталлятор не испортил нам удовольствие, запустив программу автоматически). Немало программ содержат ограничения на время пробного использования, и потому необходимо со всей ответственностью отнестись к первому (да и не только к первому, как мы увидим в дальнейшем) запуску программы. Дело в том, что процесс исследования защиты может затянуться, и вам элементарно не хватит тех дней, которые отведены под пробное использование программы. Вы, разумеется, захотите продлить время пользования программой — и вот тут вас может ожидать весьма неприятный сюрприз.

В общем случае, задачи по неограниченному продлению триального времени программы обычно не представляют особой сложности, но в качестве дополнительных уровней защиты часто присутствуют всевозможные навесные защиты, ликвидация которых может потребовать гораздо больших трудозатрат. Поэтому надо, по возможности, предупреждать подобные ситуации, а о том, каким образом эти ситуации могут возникать, я и расскажу в этом разделе.

Возможны и другие причины, по которым вам может потребоваться снять ограничение на время использования программы. Для некоторых программ процедура регистрации не предусмотрена в принципе, но при этом в течение испытательного срока они выполняют свои функции в полном объеме (часто такие программы помечены как Demo- и Evaluation-версии), и вполне приемлемым решением было бы неограниченное продление триального срока. Случается также, что необходимо установить бета-версию программы, у которой истек “срок годности”, но более новой беты нет, а посмотреть на программу очень хочется. Так или иначе, снятие ограничений на время использования или число запусков программ — одна из актуальных задач современного крэкинга.

Представьте себе, что вы — автор программы, и ваша задача — ограничить “испытательный срок”, в течение которого пользователь может работать с программой. Как такое можно реализовать? Выбор мест хранения информации об испытательном сроке в современных ОС довольно небогатый — содержимое некоего файла (который можно попытаться спрятать), атрибуты файлов, либо системный реестр (это актуально только для ОС Windows). Такие изменения могут быть обнаружены при помощи утилит, умеющих создавать и сравнивать снимки состояния системы. Изредка встречаются нетрадиционные решения вроде манипуляций на уровне файловой системы или использования слайсов. Использование слайсов для защиты основано на том, что после-

дний кластер, занятый файлом, обычно заполнен не целиком, и потому в незаполненной (и невидимой для большинства программ, оперирующих с файлами) области можно хранить некоторый объем информации.

В принципе, идеологических различий между реестром Windows, файлами и атрибутами отдельных файлов нет, все эти объекты могут использоваться для хранения используемых защитой данных. Действительно, аналогии в устройстве реестра и дисковой подсистемы очевидны: “ветви” реестра играют роль логических дисков, разделы — практически полные аналоги папок (убедиться в этом можно на наглядном примере, взглянув на иконки разделов в “Редакторе реестра”), имена ключей — это имена файлов, а значения ключей — содержимое этих файлов. Аналогии можно продолжить, но для нас важно другое — поскольку файловая система подобна реестру, принципы поиска и ликвидации защитных механизмов, основанных на сокрытии информации на дисках или в реестре, во многом будут сходны. Поэтому далее я буду говорить в основном о реестре, оставляя читателю возможность самому разобраться в том, как аналогичные механизмы могут быть реализованы на основе файловой системы (или почему они не могут быть реализованы).

Конечно, все разнообразие механизмов ограничения времени использования программы или количества запусков охватить нереально, но в этом и нет необходимости — воспользовавшись своим воображением, вы легко продолжите список. Я лишь перечислю основные события в “жизни” программы и их возможную связь с реализацией триальных механизмов в программах.

Первое, что приходит в голову — во время первого запуска сохранить в файле или в реестре дату (или вычислить дату, после которой программа не должна работать) и при каждом запуске сравнивать текущую дату с сохраненной, проверяя, не истек ли “срок годности” программы. Такие программы, не обнаружив в реестре пометки о дате первого запуска, как правило, считают, что текущий запуск — первый. Такое простейшее решение, как правило, и обходится простейшими средствами — достаточно обнаружить и удалить соответствующий файл или ключ реестра. Одна из модификаций этого метода — определение даты первого запуска через чтение атрибутов файлов: при создании любого файла атрибут CreationTime (дата создания файла) устанавливается автоматически, что позволяет непосредственно в процессе инсталляции “промаркировать” все устанавливаемые файлы датой установки программы. Затем программа просто проверяет при каждом запуске дату создания какого-либо файла или папки (или вообще всех файлов, созданных в процессе инсталляции) и на основе этой информации вычисляет количество дней до истечения испытательного периода. Что интересно, свойства файлов могут использоваться не только для определения даты установки программы, но и для определения текущей даты — в процессе своей работы отдельные компоненты ОС нередко ведут всевозможные “журналы” (логи) в файлах с заведомо известными



ми именами. Проверив дату последней модификации такого файла или его содержимое, можно с некоторой погрешностью узнать текущую дату.

Как было показано выше, надежность такого метода довольно низкая, и защита ориентирована в основном на психологический эффект. Более устойчивые варианты этой защиты основаны на использовании "меток", оставляемых программой в реестре в некоторые критические моменты.

Метки, создаваемые при установке, обычно предназначены для предотвращения повторной установки программы (и получения дополнительного триального срока) и для противодействия попыткам "сбросить" счетчик времени удалением ключа реестра. Если инсталлятор не позволяет выполнять сложные вычисления внутри инсталляционного скрипта, метка может выглядеть как запись в реестре с фиксированным значением, наличие которого будет проверяться при первом запуске. Если для создания инсталляционного пакета использовался достаточно мощный продукт, дата установки или максимальное число запусков программы могут быть прописаны в реестре уже в процессе установки (возможно, в зашифрованном виде). Кроме того, инсталлятор может проверить наличие сделанных ранее "меток" в реестре для предотвращения повторной установки. Таким образом, гарантированно вернуть программу в рабочее состояние после окончания триального срока можно только удалением всех "меток", оставленных программой, с последующей переустановкой.

Многие из ныне существующих защит создают "метки" в момент первого запуска программы. Факт первого запуска обычно определяется простейшим способом — если программа при запуске не обнаруживает "метку", она считает, что запущена впервые, и создает "метки". Обычно создание "меток" выполняется сразу после начала исполнения программы, но изредка встречаются программы, которые выполняют эти действия при завершении программы (возможно, существуют защиты, которые делают это в случайный момент времени). Скорее всего, таким образом разработчики защит пытались затруднить выявление защитного кода, но на деле они добились прямо противоположного эффекта: большинство операций по загрузке данных из реестра выполняется именно при запуске или изменении настроек программы, а вот запись в реестр в конце сеанса работы — операция не самая распространенная (обычно в конце сеанса сохраняется только информация о положении окон, настройках интерфейса программы и прочая подобная информация, которую несложно отличить от защитных механизмов программы). Очевидно, что если защита использует исключительно "метки", создаваемые при первом запуске, после удаления этих "меток" программа считает, что запущена впервые, и начинает отсчет триального срока заново. Для того чтобы пользователи не обходили эту защиту совсем уж элементарными средствами, вроде удаления соответствующего ключа реестра, ключ, содержащий в себе защитную информацию, может быть "плавающим", т.е. имя ключа может генерироваться случайным образом в зависимости от аппаратной конфигурации компьютера или каких-либо иных параметров. Найти вручную такой ключ среди десятков подобных практически нереально. В частности, именно такой механизм использует ASProtect. Поскольку "плавающий" ключ, сколь успешно бы он ни

был замаскирован, все-таки отличается от окружающих его ключей, определив признаки, которые отличают "плавающий" счетчик от обычных ключей реестра, можно создать программу, которая бы автоматически выявляла подозрительные ключи, что подтверждается наличием как минимум трех независимо разработанных утилит, способных выявлять и удалять ключи, созданные ASProtect'ом. В любом случае, изменения в реестре, возникшие после первого запуска программы, нетрудно обнаружить при помощи программ мониторинга реестра.

Важно отметить, что программы, в которых установлено ограничение на число пробных запусков, при первом запуске всегда создают либо модифицируют один из ключей реестра. Это напрямую следует из логики их работы: для работы такой защиты необходим счетчик запусков, а создать такой счетчик можно только во время установки либо при первом запуске.

Разумеется, программа, содержащая ограничение на число запусков, должна увеличивать либо уменьшать значение этого счетчика после каждого запуска. Кроме того, запись в реестр при каждом запуске возможна также и при ограничении по времени использования. Такая запись играет роль дополнительной защиты от изменения системного времени — если дата текущего запуска меньше даты предыдущего или отличается от него на считанные секунды, программа может предположить, что пользователь пытается использовать программу сверх установленного срока. Обнаружить такую защиту сравнительно легко — многократную модификацию одного и того же ключа скрыть практически невозможно, не помогают даже такие приемы как дублирование счетчика и "плавающий" счетчик. Сбрасывается такой счетчик тоже без особых сложностей — достаточно один раз запомнить состояние соответствующих ключей реестра и затем восстанавливать его перед каждым запуском.

Интересно отметить, что даже столь простую идею как отслеживание времени некоторые разработчики ухитряются реализовать некорректно (по крайней мере, под ОС Windows такое встречается не так уж редко). Windows позволяет оперировать двумя типами времени: системным (оно же "всемирное", UTC) и местным (Local). Причем, во многих странах местное время может быть зимним и летним. И если защита ориентируется на местное время, а день перевода часов пользователя может ожидать сюрприз: после автоматического перевода часов, выполняемого ОС, программа может просто перестать работать. Для этого нужно лишь, чтобы пользователь запустил программу непосредственно перед переходом с летнего времени на зимнее, и потом — еще один раз в течение ближайшего часа. По этому нехитрому признаку можно в известной степени судить о квалификации разработчика защиты.

Наконец, последним важным моментом в жизни программы является ее первый запуск после истечения срока триального использования. Поскольку большинство лицензионных соглашений запрещают пробное использование программы сверх установленного "испытательного срока", программа может "помочь" пользователю соблюсти условия лицензионного соглашения, сделав в реестре метку, запрещающую дальнейшую работу программы. Эта метка может быть как новым ключом в глубинах реестра (или файлом в какой-нибудь системной директории), так и особым значением уже существующе-



го ключа. Проверка значения такого ключа также может быть встроена в инсталлятор программы, чтобы пользователь не мог просто деинсталлировать и переустановить программу.

Какие выводы следуют из всего, что было сказано выше? Во-первых, то, что триальные механизмы в программах могут быть довольно разнообразны по исполнению (и, кроме того, иногда дублируются разработчиком для повышения надежности), а их поиск и удаление требуют аккуратности и терпения. Несмотря на то что создание снимков системы требует некоторого времени, пренебрежительное отношение к этим действиям может создать вам значительные трудности в будущем. Во-вторых, существуют критические точки, при прохождении которых рекомендуется отслеживать состояние системы. В-третьих, можно составить универсальный план действий, который с очень высокой вероятностью позволил бы выявить изменения в системе, вносимые триальными механизмами, проанализировать их и выработать способы противодействия защите. Лично я чаще всего использую следующую последовательность действий, придерживаясь которой рекомендую и вам.

1. Сделать снимки состояния системы до и после установки программы.

2. Сделать снимок системы перед первым запуском программы, запустить программу, сразу же закрыть ее (чтобы в снимок системы не попали изменения, возникающие при работе с программой и не имеющие отношения к функционированию защиты) и тут же сделать еще один снимок системы. Если программа запускается после инсталляции автоматически, и выполнить этот пункт не представляется возможным, мы, тем не менее, сможем обнаружить изменения, произошедшие при первом запуске, сравнив снимки, сделанные в предыдущем пункте (другое дело, что мы не сможем отличить изменения, произошедшие при инсталляции, от модификаций, внесенных в систему программой при первом запуске).

3. Запустить программу во второй раз, сделав снимки до запуска и после завершения программы. Если программа содержит ограничение на число запусков, путем несложного анализа мы сможем легко выявить счетчик запусков. Если программа содержит ограничение по времени использования, мы можем установить, использует ли программа какие-либо дополнительные механизмы отслеживания времени. О наличии таких механизмов может свидетельствовать появление в реестре новых ключей или изменение уже существующих, появившихся в ходе инсталляции или первого запуска. Если такие ключи обнаружатся, есть вероятность, что программа, кроме даты первого запуска, отслеживает еще и дату предыдущего запуска, число дней, в течение которых использовалась программа, или другую подобную информацию. Для максимальной надежности и достоверности результатов эту операцию можно повторить несколько раз.

4. Выяснить, какие изменения в системе происходят, когда программу запускают впервые после истечения триального срока. Это необходимо для поиска меток, которые могли бы воспрепятствовать нормальной работе программы после окончания времени, отведенного на пробную эксплуатацию программы. Важно отметить, что для программ с ограниченным числом запусков необхо-

димо отслеживать изменения не только при первом запуске сверх установленного лимита, но и при последнем "законном" запуске.

5. По возможности, необходимо выяснить, как программа реагирует на попытки обойти ограничение времени использования при помощи изменения системной даты. Если программа успешно "проглатывает" любое изменение даты, высока вероятность того, что защита реализована максимально простым способом, и обойти ее будет нетрудно. Разумеется, все эти действия должны сопровождаться созданием снимков системы — если программа обнаружит ваши манипуляции с системным временем, она может отреагировать на это созданием "метки", свидетельствующей о попытке обойти защиту.

6. Проанализировать все собранные данные, выявить подозрительные ключи реестра и другие объекты, которые могли бы использоваться для хранения защитных данных, и затем на основе собранной информации попытаться восстановить работоспособность программы, последовательно удаляя или восстанавливая первоначальные значения подозрительных ключей.

7. Проверить, нет ли в программе неочевидных ограничений по времени использования. Нередко встречаются программы (как правило, это демонстрационные, пробные и бета-версии), в которых, помимо обычного ограничения на время пользования, имеется дополнительная проверка на максимальную дату запуска, после которой программа считается устаревшей и перестает работать.

Вышеперечисленные приемы еще недавно считались "некрасивыми", и рассматривались как побочный результат исследования программы. Однако в последние годы ситуация начала меняться: лавинообразный рост количества защищенных программ привел к тому, что даже довольно известные группы обратили внимание на технологии продления сроков пробного пользования и выпускают *launcher*'ы, продляющие триальные сроки и сбрасывающие счетчики числа запусков. Что же, изменившиеся условия требуют новых решений, и если эти решения эффективны, нет никаких причин от них отказываться.

Кроме того, серьезное исследование какой-либо защиты может потребовать неоднократного запуска приложения, и отведенное число запусков может закончиться раньше, чем удастся разобраться в программе. К тому же, выявление счетчиков и "меток" — это один из путей поиска реализации триальных механизмов непосредственно в коде программы. К примеру, если вы установили, что имя ключа, отвечающего за триальные ограничения, фиксировано, вы можете попытаться найти имя этого ключа в тексте программы, затем выявить все ссылки на эту строку в программе и, в итоге, добраться до кода, ревизирующего защитные механизмы. Также при помощи API-шпиона можно проследить за вызовами функций обращения к реестру, проанализировать параметры, передаваемые этим функциям, и таким образом попытаться выяснить, какой из этих вызовов непосредственно связан с реализацией ограничений в программе. А затем, зная, каким образом можно сбросить счетчик, ограничивающий возможность работы с программой, останется лишь воспроизвести этот эффект путем модификации кода программы.

(Продолжение следует)



Простейший конвертор DOC → TXT

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

Так случилось, что мне надо было посмотреть несколько текстов в формате Microsoft Word (файлы с расширением DOC). Устанавливать этот редактор я не собирался, да его у меня и не было. Поэтому я попытался найти в Интернете программу для просмотра DOC-файлов или преобразования их в обычный текст. Увы, безуспешно. Или найденная программа требовала наличия Microsoft Word, или была написана для UNIX, или не желала запускаться... Тогда я решил попробовать сам написать простейший конвертор DOC → TXT.

Какой-либо информации о формате DOC-файлов у меня не было, а было несколько таких файлов (причем текст, содержащийся в одном из них, у меня также имелся в простом текстовом файле). В результате изучения содержимого этих файлов удалось обнаружить некоторые закономерности. Так, первые 600h байтов DOC-файла содержат какие-то служебные данные. Далее начинается собственно текст, каждый символ которого кодируется двумя байтами:

- два байта (0Dh, 00h) обозначают конец абзаца;
- два байта, первый из которых находится в диапазоне 20h...7Fh, а второй равен нулю, обозначают символ из первой половины кодовой таблицы (первый байт — код этого символа);
- два байта, первый из которых находится в диапазоне 10h...2Fh, а второй равен 4, обозначают символы от "А" до "Я", за исключением "Ё" (т.е., если первый байт равен 10h, это символ "А", 11h — "Б", и так далее);
- аналогично, два байта, первый из которых находится в диапазоне 30h...4Fh, а второй равен 4, обозначают символы от "а" до "я", за исключением "ё";
- два байта (14h, 20h) обозначают символ "тире";
- два байта (1Ch, 20h) обозначают символ "открывающие кавычки";
- два байта (1Dh, 20h) обозначают символ "закрывающие кавычки".

После текста в файле содержатся еще какие-то служебные данные, начинающиеся с последовательности нулей.

Конечно, вышеизложенная информация очень неполная, да, может быть, и неточная (так как получена в результате анализа лишь нескольких DOC-файлов), но ее оказалось достаточно, чтобы написать небольшую программу для преобразования DOC → TXT, с помощью которой я успешно конвертировал эти DOC-файлы. Судя по находящимся в этих файлах текстам строкам, они были созданы с помощью Microsoft Word версий 8.0 и 9.0.

Я, конечно, не надеюсь, что моя программа будет правильно конвертировать любые DOC-файлы. Но если она помогла мне, то, может быть, поможет и вам.

От редакции: в данном случае под DOC-файлами подразумеваются файлы именно WinWord (v.8-9), которые используют для кодировки символов Unicode. Word 6 и 7 по умолчанию сохраняют текст в win-1251, и используют для кодировки одного символа один байт. Поэтому данный конвертор с ними не работает.

При запуске программы надо указать в командной строке следующие параметры: имя исходного файла, имя получаемого файла и необязательный параметр "/a". Исходный и получаемый файлы считаются находящимися в текущем каталоге.

От параметра "/a" зависит, как конвертор будет определять конец текста в DOC-файле. Если этот параметр не указан, конвертор будет считать, что текст закончился, и начались служебные данные, когда очередные два считанных байта окажутся нулями. Но если два нулевых байта попадутся внутри текста (а исключить такой случай нельзя, ведь мне неизвестна полная информация о формате DOC-файлов), то конец текста будет определен раньше времени. А если параметр "/a" указан, программа закончит конвертацию, только когда закончится исходный файл. Но тогда в конце полученного файла будет "мусор" из служебных данных.

В процессе конвертации программа после обработки каждых 32768 символов текста печатает на экране "*", показывая, что она не зависла.

Ниже приведен текст программы на языке Си. Исполняемый файл программы (для DOS) можно скачать с web-страницы журнала.

```
/* ===== */
/* DOC2TXT - извлечение текста из документов MS Word */
/* ===== */
/* Compiler: Turbo C 2.0 */
/* ===== */

#include <stdio.h>

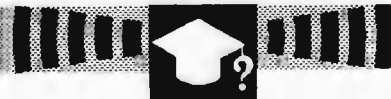
void main (int argc, char *argv[])
{
    FILE *f_src, *f_dst;
    int c1,c2;
    char *rus_big="АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
    char *rus_small="абвгдежзийклмнопрстуфхцчшщъыьэюя";

    printf ("\\nDOC to TXT file convertor (c) Ivan Roshin, Moscow, 2003.\\n");
    if (argc<3)
    {
        printf ("Syntax: doc2txt.exe source-file destiny-file [/a]\\n\\n"); exit();
    }

    f_src=fopen(argv[1],"rb");
    if (f_src==NULL)
        {printf ("Error: can't open source file!\\n\\n"); exit();}

    f_dst=fopen(argv[2],"wb");
    if (f_dst==NULL)
        {printf ("Error: can't open destiny file!\\n\\n"); exit();}

    fseek(f_src,0x600,SEEK_SET);
    do
    {
        c1=fgetc(f_src);
        c2=fgetc(f_src);
        if (c2==0)
        {
            if (c1==13) fputs ("\\r\\n",f_dst);
            /* Перевод строки и абзацный отступ. */
            if ((c1>=0x20)&&(c1<=0x7F)) fputc (c1,f_dst);
        }
    }
}
```

```

}
if (c2==4)
{
    if ((c1>=0x10)&&(c1<=0x2F)) fputc(rus_big[c1-0x10],f_dst);
    if ((c1>=0x30)&&(c1<=0x4F)) fputc(rus_small[c1-0x30],f_dst);
}
if (c2==0x20)
{
    if (c1==0x14) fputc('-',f_dst); /* Тире. */
    if (c1==0x1c) fputc('\"',f_dst); /* Открывающие кавычки. */
    if (c1==0x1d) fputc('\"',f_dst); /* Закрывающие кавычки. */
}
/* Через каждые 32768 выведенных символов
печатаем на экране "***. */
{
    static unsigned int counter=0;
    counter++;
    if ((counter&0x7fff)==0) printf("***");
}

/* Главный цикл выполняется, пока не встретится 0,0 или
не кончится файл (если при запуске был указан третий
параметр "/a", цикл будет выполняться, пока не кончится
файл, даже если встретится 0,0; в программе для простоты
проверяется только количество параметров). */

```

```

} while ((c1!=EOF)&&(((c1|c2)!=0)|| (argc==4)));
fcloseall();
}

```

Р.С. При использовании конавертора выяснилось еще одно обстоятельство. Если в DOC-файле имеются достаточно длинные фрагменты текста с использованием символов только из нижней половины кодовой таблицы (например, текст на английском языке или листинг программы), то эти фрагменты могут находиться в "обычном" виде (один байт — один символ), и в этом случае конвертор не сможет их правильно обработать (они будут пропущены). К тому же, в файле после таких фрагментов может находиться последовательность нулей, поэтому конвертор надо запускать с параметром "/a", чтобы он не принял эти нули за конец текста. Замечу, что наличие таких фрагментов в DOC-файле легко обнаружить, просмотрев его содержимое — участки текста в "обычном" виде сразу будут заметны. А если текст только на русском языке, то и вообще беспокоиться не о чем.

Домашний "Screen Snapshot"

С.РЮМИК,
г.Чернигов.

*Если хочешь сделать хорошо, сделай сам.
Житейская мудрость*

Как сохранить понравившуюся картинку на экране монитора? Самый простой вариант — запустить в фоновом режиме специальную программу, которая при нажатии определенной комбинации клавиш копирует графический образ экрана в память. Затем экранную картинку сохраняют на диске в виде файла или распечатывают на принтере.

Называют подобные программы "грабберами" (от англ. "grabbing" — захватывать), "фризами" (от англ. "freeze" — замораживание) или "Screen Snapshot" (дословно "экраннй снимок"). Наиболее мощной из них является "HyperSnap-DX", версию 3.63.01 которой можно бесплатно скачать на сайте <http://www.hyperionics.com>. Всем хороша программа, с любыми Windows-приложениями работает, справляется даже со снимками экранов "на лету" при просмотре DVD-фильма. Одна незадача — картинки графического DOS-формата VGA 640x480 при 16 цветах сохраняются со значительными геометрическими искажениями, хотя в текстовом режиме проблем не возникает. "Неправильные" файлы легко отличить от "правильных" по надписи на экране монитора: "Это приложение не может работать в оконном режиме", которая возникает при попытке запуска их из Windows с установкой "Файл/Свойства/Программа/Окно/Свернутое в значок".

Что делать? Искать в архивах подходящий DOS-"граббер" или создать свой собственный? Первый вариант наталкивается на элементарную неработоспособность утилит, многие из которых программисты писали "для себя", для узких целей, не удосужившись дать возможность окружающим лицезреть файлы помощи с описанием порядка работы. Второй вариант, на первый взгляд, кажется чересчур сложным, но так ли это на самом деле?

Предположим, что требуется сделать снимок экрана а вашей собственной программе. Кому как не разработчику лучше всего знать, в каком месте и при каких условиях получится самый удачный ракурс "фотоснимка". Логично, что процедура копирования экранной картинки должна быть оформлена в виде отдельной функции, которую можно неоднократно вызывать из разных мест программы. Для примера, ниже в листинге приведены исходные коды функции "snap". Ее входным аргументом является имя файла, в который будет занесено экранное изображение.

```

/* Функция копирования экранной картинки 640x480 точек */
/* (VGA 16 цветов) в графический файл с расширением bmp */
/* Входной параметр: имя файла, например, snap("d.bmp") */
/* Журнал <Радиомир_ВК>, 2003 г., Borland Turbo C++, 3.0 */

void snap(char name[12]) /* 12 символов в имени файла */
{ FILE *out; /* Указатель на выходной файл формата BMP */
  int x, y; /* Счетчики строк и столбцов */
  /* Структура формата файла BitmapFileHeader, 14 байтов */
  char code[118] = { 0x42,0x4D, /* Надпись <BM> */
    0x78,0x58,2,0, /* Длина файла 25878h=153720 байтов */
    0,0,0,0, /* Резерв, не используется */
    0x76,0,0,0, /* 76h=118 байтов смещение данных экрана */
  };
  /* Структура формата цвета BitmapInfoHeader, 40 байтов */
  0x28,0,0,0, /* Число байтов в этой структуре 28h=40 */
  0x80,2,0,0, /* Число точек по горизонтали 280h=640 */
  0xE0,1,0,0, /* Число точек по вертикали 1E0h=480 */
  1,0, /* Число цветовых плоскостей для BMP-формата=1 */
  4,0, /* 4 бита цвета на пиксел, итого 16 цветов */
  0,0,0,0, /* Компрессия BMP-файла отсутствует */
  0,0x58,2,0, /* 25800h=153720 точек 640*480 пикселей */
  0x20,0x2E,0,0, /* 2E20h=11808 точек на 1 м по ширине */
  0x20,0x2E,0,0, /* 2E20h=11808 точек на 1 м по высоте */
  0,0,0,0,0,0,0,0, /* Используются все цвета палитры */

```



```

/* Структура цветовой таблицы RGBQuad длиной 64 байта */
0,0,0,0, /* 0 - черный BLACK */
0xA0,0,0,0, /* 1 - синий BLUE */
0,0xA0,0,0, /* 2 - зеленый GREEN */
0xA0,0xA0,0,0, /* 3 - сине-зеленый CYAN */
0,0,0xA0,0, /* 4 - красный RED */
0xA0,0,0xA0,0, /* 5 - малиновый MAGENTA */
0,0x50,0xA0,0, /* 6 - коричневый BROWN */
0xA0,0xA0,0xA0,0, /* 7 - светло-серый LIGHTGRAY */
0x50,0x50,0x50,0, /* 8 - темно-серый DARKGRAY */
0xFF,0x50,0x50,0, /* 9 - голубой LIGHTBLUE */
0x50,0xFF,0x50,0, /* A - светло-зеленый LIGHTGREEN */
0xFF,0xFF,0x50,0, /* B - светло-голубой LIGHTCYAN */
0x50,0x50,0xFF,0, /* C - светло-красный LIGHTRED */
0xFF,0x50,0xFF,0, /* D - светло-малиновый LIGHTMAGENTA */
0x50,0xFF,0xFF,0, /* E - желтый YELLOW */
0xFF,0xFF,0xFF,0 /* F - белый WHITE */
};
if ((out=fopen(name,"wb"))==NULL) /* Открытие файла */
{ puts("\nНевозможно открыть файл!");
  exit(1); /* Аварийный выход из программы */
}
/* Занесение в выходной файл начального хэдера 118 байтов */
for (x=0; x<118; x++) putc(code[x],out);
/* Занесение в выходной файл данных структуры CodeData */
for (y=479; y>=0; y--) /* Счетчик 0...479 строк */
for (x=0; x<640; x+=2) /* Счетчик 0...639 столбцов */
  putc(((getpixel(x,y)<<4)|(getpixel(x+1,y)&15)),out);
putc(0,out); putc(0,out); /* Выравнивание длины файла */
fclose(out); /* Закрытие выходного файла */
return; /* Возврат в основную программу */
} /* (с) Рюмик С.М. */

```

Файл записывается в текущую директорию в BMP-формате. О структуре файлов с расширением .bmp подробно рассказано в [1]. Вкратце напомним, что BMP-файл состоит из четырех структурных областей: BitmapFileHeader (общие параметры файла), BitmapInfoHeader (информация о "габаритах" картинки), RGBQuad (цветовая таблица), CodeData (битовый образ картинки в цветах RGB). В комментариях листинга дана расшифровка элементов трех первых структур для случая режима VGA 640x480 при 16 цветовых оттенках и разрешающей способности 300 dpi по горизонтали и вертикали. Элементы четвертой структуры получаются автоматически при циклическом вызове функции getpixel(int x,int y), аходящей в пакет <graphics.h> Turbo C++.

Работа с функцией "snap" не вызывает затруднений. В любом месте программы можно вставить строку следующего содержания — if (УСЛОВИЕ)

snap("ИМЯ_ФАЙЛА") или просто snap("ИМЯ_ФАЙЛА"). В процессе работы программы при выполнении заданного УСЛОВИЯ на жесткий диск будет записан графический файл длиной 153720 байтов с указанным ИМЕНЕМ. Просмотреть и отредактировать полученную картинку можно любым графическим редактором, например, Adobe Photoshop. Если УСЛОВИЕ будет заключаться в проверке нажатия определенной "горячей" клавиши, то момент "фотографирования" экрана пользователь будет выбирать визуально.

Небольшой нюанс. Формат BMP относится к разряду многоариантных. Это означает, что одно и то же изображение можно воспроизвести файлами одинаковой длины, но с разными значениями байтов в структурах RGBQuad и CodeData (цветовая таблица и данные битового образа).

Например, на экране изображена одна красная точка на фоне из 99 черных точек. В таблице цвета соответственно будут отведены четыре байта для красного и столько же для черного цвета. Если первыми в таблице поместить байты красного цвета, то данные битового образа при 16-цветной палитре будут содержать 49 байтов 11h и один байт 01h. Если первыми в таблице поместить байты черного цвета, то данные битового образа будут содержать 49 байтов 00h и один байт 10h.

Порядок чередования цветовых кодов в таблице не унифицирован и определяется только алгоритмом работы программы-кодировщика. Следовательно, одно и то же изображение в разных программах-"грабберах" может кодироваться по-разному.

Переходя от абстрактного примера к реальному листингу, следует заметить, что порядок расположения кодов в таблице цвета соответствует возрастанию номеров цветовых оттенков EGA/VGA-адаптера. Абсолютные значения кодов красной, зеленой и синей составляющих тщательно подобраны по цветовой гамме. Картинка получается максимально приближенной к той, что видна на мониторе в DOS-режиме VGAHI 640x480 при 16 цветах. Эту же таблицу можно использовать для режима EGA0 640x200, EGAHI 640x350, VGAL0 640x200, VGAMED 640x350.

Литература

1. Хилько К. О считывании BMP-файлов — Радиолюбитель. Ваш компьютер, 1999, №4, С.17.

Работа с остатками в программе "1С: Бухгалтерия"

А.СИЗЕНКО,
Украина, г.Луганск.
E-mail: pat_2000@mail.ru

1. Как быстро и эффективно ввести остатки

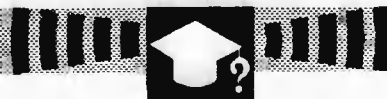
При внедрении программы "1С: Бухгалтерия" на фирме, которая уже аедет свою деятельность, необходимо ввести входящее сальдо по всем счетам учета.

Это достаточно трудоемкий процесс, особенно на фирмах, применяющих в своей хозяйственной деятельности большое количество номенклатуры ТМЦ и необоротных активаов (основных фондов, нематериальных активов и т.п.).

В определенной степени этот процесс можно облегчить.

Вкратце смысл "рационализации" состоит в следующем — при заполнении справочников пользователь, наряду с реквизитами карточки, вводит в определенных полях суммарные или количественные остатки. Естественно, эти поля должны быть "сделаны" на этапе конфигурирования.

Затем остается только открыть стандартный документ



Количество субконто	Предполагаемый ввод данных	Данные по субконто №1 (выбираем из соответствующего справочника)	Данные по субконто №2 (выбираем из соответствующего справочника)	Данные по субконто №3	Вводим идентификатор
3	Остатки ТМЦ	Одна карточка для всей группы : "место хранения"	Группа ТМЦ (где находятся карточки, по которым вводим остатки)	Не нужно вводить никаких данных (партия по умолчанию будет открыта автоматически)	Количества (сумма будет вычислена как Цена_прих * Количество)
2	Первоначальная стоимость необоротных активов	Одна карточка для всей группы ("место хранения или МОЛ")	Группа необоротных активов (где находятся карточки по которым вводим остатки)		Суммы
3	Износ в бухгалтерском или налоговом учете, первоначальная стоимость в налоговом	Группа необоротных активов (где находятся карточки, по которым вводим остатки)			Суммы

"Операция", выбрать текущий счет ввода остатков и ввести идентификаторы соответствующих справочников. Наши действия, в зависимости от количества субконто счета, сведены в таблицу.

При необходимости, после заполнения табличной части документ можно корректировать (добавлять, удалять строки и т.п.) стандартными способами "1С: Бухгалтерия".

Изменения делаем в "Форме операции" (см. рисунок). Добавляем кнопки, поля ввода, переключатель, тексты и рамки. В модуле формы вносим следующие дополнения:



```

Перем перем1, перем2, перем3, Запрос, Субконто1,
Субконто2, Субконто3, Акво, АСумма;
// этот фрагмент для удобства восприятия перемещен вниз
ВыбСчет=СчетПоКоду("00");
ВыбСчет();

```

```

Если Документ.Выбран()=1 Тогда
//блокируем при просмотре с любого документа
Форма.ВыбСчет.Доступность(0);
Форма.КнопкаЗапрос.Доступность(0);
Форма.КнопкаЗаполнить.Доступность(0);
КонецЕсли;
//

```

```

Процедура ВыбСчет()
// устанавливаем доступность к кнопкам и полю ввода
Если ВыбСчет.КоличествоСубконто()=0 Тогда
Форма.КнопкаВыбратьСубконто1.Доступность(0);
Форма.КнопкаВыбратьСубконто2.Доступность(0);
Иначе Если ВыбСчет.КоличествоСубконто()=1 Тогда
Форма.КнопкаВыбратьСубконто1.Доступность(1);
Форма.КнопкаВыбратьСубконто2.Доступность(0);
Иначе ВыбСчет.КоличествоСубконто()=2 Тогда
Форма.КнопкаВыбратьСубконто1.Доступность(1);
Форма.КнопкаВыбратьСубконто2.Доступность(1);
КонецЕсли;

Если ВыбСчет.Количественный=0 Тогда
Форма.ИдКва.Доступность(0);
Иначе
Форма.ИдКва.Доступность(1);
КонецЕсли;

```

```

// Для уменьшения кол-ва ветвлений
Попытка

```

```

Субконто1=СоздатьОбъект(ВыбСчет.ВидСубконто(1).ТипСубконто());

```

```

Субконто2=СоздатьОбъект(ВыбСчет.ВидСубконто(2).ТипСубконто());

```

```

Субконто3=СоздатьОбъект(ВыбСчет.ВидСубконто(3).ТипСубконто());

```

```

Исключение
КонецПопытки;
КонецПроцедуры

```

```

Процедура КнопкаВыбратьСубконто1()
ВвестиЗначение(Субконто1,"Выберите группу для выборки карточек
справочника или карточку для подстановки в соответствующее
субконто",ВыбСчет.ВидСубконто(1).ТипСубконто());
КонецПроцедуры

```

```

Процедура КнопкаВыбратьСубконто2()
ВвестиЗначение(Субконто2,"Выберите группу для выборки карточек
справочника или карточку для подстановки в соответствующее
субконто",ВыбСчет.ВидСубконто(2).ТипСубконто());
КонецПроцедуры

```

```

// добавляет новую проводку в документ

```

```

Процедура Авто();
Если АСумма>0 Тогда
НоваяПроводка();
НомерЖурнала = "0";
НапаФирма = глВосстановитьЗначение("БазФирма");
Если Куда=1 Тогда
Дебет.Счет=ВыбСчет;
Дебет.Субконто(1,Перем1);
Дебет.Субконто(2,Перем2);
Дебет.Субконто(3,Перем3);
Если ВыбСчет.Забалансовый=0 Тогда
Кредит.Счет=СчетПоКоду("00");
КонецЕсли;
Иначе
Кредит.Счет=ВыбСчет;
Кредит.Субконто(1,Перем1);
Кредит.Субконто(2,Перем2);
Кредит.Субконто(3,Перем3);
Если ВыбСчет.Забалансовый=0 Тогда
Дебет.Счет=СчетПоКоду("00");
КонецЕсли;

```




```
КонецЕсли;
Если ВыбСчет.Количественный=1 Тогда
    Количество=АКво;
КонецЕсли;
Сумма=АСумма;
КонецЕсли;
КонецПроцедуры

// непосредственно заполнение документа
Процедура Заполнить()
    Если ВыбСчет.КоличествоСубконто()=3 Тогда
        //ввод ТМЦ
        Если ПустоеЗначение(ИдКва)=1 Тогда
            Сообщить("Не указан идентификатор количества");
            Возврат;
        ИначеЕсли ПустоеЗначение(Субконто1)=1 Тогда
            Сообщить("Не указано место хранения");
            Возврат;
        ИначеЕсли Субконто1.ЭтоГруппа()=1 Тогда
            Сообщить("Первое субконто не должно быть группой");
            Возврат;
        КонецЕсли;

        Если Субконто2.ЭтоГруппа()=0 Тогда
            Сообщить("Второе субконто должно быть группой.");
            Возврат;
        КонецЕсли;

        Попытка
            АКво=Субконто2.ТекущийЭлемент().
            ПолучитьАтрибут(СокрЛП(ИдКва));
        Исключение
            Сообщить("Неверный идентификатор количества");
            Возврат;
        КонецПопытки;

        Попытка
            АЦена=Субконто2.ТекущийЭлемент().
            ПолучитьАтрибут(СокрЛП(ИдЦена));
        Исключение
            Сообщить("Неверный идентификатор цены");
            Возврат;
        КонецПопытки;

        Если (ТипЗначения(АКво)<>1) или
            (ТипЗначения(АЦена)<>1) Тогда
            Сообщить("Значение иден. количества или
                цены не число.");
            Возврат;
        КонецЕсли;

        Сообщить("Начало. Заполняю по группе: "+Субконто2);

        Переменная=СоздатьОбъект(ВыбСчет.ВидСубконто(2).ТипСубконто());
        Переменная.ИспользоватьРодителя(Субконто2);

        Переменная.ВыбратьЭлементы();
        Пока Переменная.ПолучитьЭлемент()=1 Цикл
            АКво=Переменная.ТекущийЭлемент().
            ПолучитьАтрибут(СокрЛП(ИдКва));

        АСумма=Round(Переменная.ТекущийЭлемент().
            ПолучитьАтрибут(СокрЛП(ИдЦена))*АКво;
            Перемен1=Субконто1.ТекущийЭлемент();
            Перемен2=Переменная.ТекущийЭлемент();
            глПолучитьПартиюПоУмолчанию(Перемен3,Перемен2);
            Авто();
        КонецЦикла;
        Сообщить("Заполнение закончено");

        ИначеЕсли ВыбСчет.КоличествоСубконто()=2 Тогда
            //необоротные активы, первоначальная стоимость
            Если ПустоеЗначение(ИдСуммы)=1 Тогда
                Сообщить("Не указан идентификатор суммы");
                Возврат;
            ИначеЕсли ПустоеЗначение(Субконто1)=1 Тогда
                Сообщить("Не указано место хранения
                    (материально-ответственное лицо)");
```

```
Возврат;
ИначеЕсли Субконто1.ЭтоГруппа()=1 Тогда
    Сообщить("Первое субконто не должно быть группой");
    Возврат;
КонецЕсли;

Если Субконто2.ЭтоГруппа()=0 Тогда
    Сообщить("Второе субконто должно быть группой.");
    Возврат;
КонецЕсли;

Попытка
    АСумма=Субконто2.ТекущийЭлемент().
    ПолучитьАтрибут(СокрЛП(ИдСуммы));
Исключение
    Сообщить("Неверный идентификатор суммы");
    Возврат;
КонецПопытки;

Если ТипЗначения(АСумма)<>1 Тогда
    Сообщить("Значение иден. суммы не числовое");
    Возврат;
КонецЕсли;

Сообщить("Начало. Заполняю по группе: "+Субконто2);

Переменная=СоздатьОбъект(ВыбСчет.ВидСубконто(2).ТипСубконто());
Переменная.ИспользоватьРодителя(Субконто2);

Переменная.ВыбратьЭлементы();
Пока Переменная.ПолучитьЭлемент()=1 Цикл
    АСумма=Переменная.ТекущийЭлемент().ПолучитьАтрибут(СокрЛП(ИдСуммы));
    Перемен1=Субконто1.ТекущийЭлемент();
    Перемен2=Переменная.ТекущийЭлемент();
    Авто();
    КонецЦикла;
    Сообщить("Заполнение закончено");

Иначе
    //одно субконто
    Если ПустоеЗначение(ИдСуммы)=1 Тогда
        Сообщить("Не указан идентификатор суммы");
        Возврат;
    КонецЕсли;

    Если Субконто1.ЭтоГруппа()=0 Тогда
        Сообщить("Первое субконто должно быть группой.");
        Возврат;
    КонецЕсли;

    Попытка
        АСумма=Субконто1.ТекущийЭлемент().
        ПолучитьАтрибут(СокрЛП(ИдСуммы));
    Исключение
        Сообщить("Неверный идентификатор суммы");
        Возврат;
    КонецПопытки;

    Если ТипЗначения(АСумма)<>1 Тогда
        Сообщить("Значение иден. суммы не числовое");
        Возврат;
    КонецЕсли;

    Сообщить("Начало. Заполняю по группе: "+Субконто1);

    Переменная=СоздатьОбъект(ВыбСчет.ВидСубконто(1).ТипСубконто());
    Переменная.ИспользоватьРодителя(Субконто1);

    Переменная.ВыбратьЭлементы();
    Пока Переменная.ПолучитьЭлемент()=1 Цикл
        АСумма=Переменная.ТекущийЭлемент().
        ПолучитьАтрибут(СокрЛП(ИдСуммы));
        Перемен1=Субконто1.ТекущийЭлемент();
        Авто();
        КонецЦикла;
        Сообщить("Заполнение закончено");
    КонецЕсли;
КонецПроцедуры
```



2. Как быстро убрать остатки с любого аналитического счета

При внедрении программы "1С: Бухгалтерия" на крупном предприятии, как правило, практически невозможно всей бухгалтерией "с завтрашнего дня начать новую жизнь", т.е. начать вести все хозяйственные операции при помощи "1С: Бухгалтерия". И дело здесь, конечно, не а самой программе, а а мыслимых и немыслимых реалиях текущей жизни. Поэтому переход на "механизированный способ учета" осуществляют поэтапно, переходя от одного раздела бухгалтерского учета к другому. Вот тут-то и возникает необходимость убирать "набежавшее" сальдо. Хочу поделиться опытом, как сделать это одним нажатием кнопки

Вам необходимо в поле ввода выбрать счет (см. рисунок) и установить, на какую дату будем "ликвидировать" остатки в поле "дата операции". После этого нажмите кнопку "Запрос" — в ответ будут сформированы все необходимые проводки. Запишите документ и сформируйте "оборотно-сальдоаую ведомость" для проверки.

```
Процедура Запрос ()
// проверка корректности ввода данных
Если ПустоеЗначение (ВыбСчет)=1 Тогда
    Сообщить ("Заполните счет!");
    Возврат;
ИначеЕсли ВыбСчет.КоличествоСубконто ()=0 Тогда
    Сообщить ("Счет без аналитики!");
    Возврат;
КонецЕсли;

//Автогенерация запроса
Запрос = СоздатьОбъект ("Запрос");
ТекстЗапроса = "Период с ДатаОперации по ДатаОперации;";

ТекстЗапроса=ТекстЗапроса+"Счет = Операция.Дебет.Счет;";
ТекстЗапроса=ТекстЗапроса+"Субконто1 =
Операция.Дебет."+ВыбСчет.ВидСубконто (1).Идентификатор ()+";";

Если ВыбСчет.КоличествоСубконто ()>1 Тогда
    ТекстЗапроса=ТекстЗапроса+"Субконто2 =
Операция.Дебет."+ВыбСчет.ВидСубконто (2).Идентификатор ()+";";
КонецЕсли;

Если ВыбСчет.КоличествоСубконто ()>2 Тогда
    ТекстЗапроса=ТекстЗапроса+"Субконто3 =
Операция.Дебет."+ВыбСчет.ВидСубконто (3).Идентификатор ()+";";
КонецЕсли;

Если ВыбСчет.Количественный=1 Тогда
    ТекстЗапроса=ТекстЗапроса+"Кво =
Операция.Количество;";
КонецЕсли;
```

```
ТекстЗапроса=ТекстЗапроса+"Сумма = Операция.Сумма;";

Если ВыбСчет.Количественный=1 Тогда
    ТекстЗапроса=ТекстЗапроса+"Функция КвоСКД = СКД(Кво);";
КонецЕсли;

ТекстЗапроса=ТекстЗапроса+"Функция СуммаСКД = СКД(Сумма);";

ТекстЗапроса=ТекстЗапроса+"Группировка Субконто1 без групп;";

Если ВыбСчет.КоличествоСубконто ()>1 Тогда
    ТекстЗапроса=ТекстЗапроса+"Группировка Субконто2 без групп;";
КонецЕсли;

Если ВыбСчет.КоличествоСубконто ()>2 Тогда
    ТекстЗапроса=ТекстЗапроса+"Группировка Субконто3 без групп;";
КонецЕсли;

ТекстЗапроса=ТекстЗапроса+"Условие (Счет в ВыбСчет);";

Если Запрос.Выполнить (ТекстЗапроса) = 0 Тогда
    Возврат;
КонецЕсли;

Перем1=0;
Перем2=0;
Перем3=0;
ТЗ=СоздатьОбъект ("ТаблицаЗначений");
Запрос.Выгрузить (ТЗ,1,0);

ТЗ.ВыбратьСтроки ();
Пока ТЗ.ПолучитьСтроку ()=1 Цикл
    // для уменьшения количества ветвлений
    Попытка
        Перем1=ТЗ.Субконто1;
        Перем2=ТЗ.Субконто2;
        Перем3=ТЗ.Субконто3;
    Исключение
        КонецПопытки;

    Если ВыбСчет.Количественный=1 Тогда
        АКво=ТЗ.КвоСКД;
        КонецЕсли;
        АСумма=ТЗ.СуммаСКД;

        //дебетовое сальдо списываем проводкой по кредиту, и наоборот
        Если АСумма>0 Тогда
            Куда=2;
        Иначе
            Куда=1;
        Если ВыбСчет.Количественный=1 Тогда
            Акво = -АКво;
        КонецЕсли;
        Асумма = -АСумма;
        КонецЕсли;
        Авто (); // заполняем строку документа
    КонецЦикла;
КонецПроцедуры
```

Простой способ защиты программ от копирования

А.ЕВТЮХИН,
E-mail: ieee@tut.by

В сегодняшнем мире, когда почти все имеют дело с компьютером, проблема защиты информации становится с каждым днем все более и более актуальной. Ведь именно информация приносит то, к чему стремятся многие — деньги. Компании, связанные с бизнесом, вкладывают огромные средства в защиту своих электронных ресурсов, но сообщения, что был взломан такой-то сайт, украдены номера кредиток, или что-то подобное, можно прочитать в любой компьютерной газете. Даже у нас (на просторах бывшего СССР) это считается воровством.

Могут спросить, почему "даже"? Да потому, что использование нелицензионного ПО у нас считается вполне нормальным. Мы не платим деньги за труд программиста, как это делают на Западе, хотя все мечтаем получать приличную зарплату за свои разработки. Желаящих пользоваться взломанными программами немало. Не все из них хакеры, большинство — пользователи скас'ов. Разумеется, если знающий ассемблер программист задастся целью взломать какую-то программу с помощью дебагера, то остановить его очень сложно.



Но ведь и мы чего-то да стоим! Пусть не профессиональных хакеров, то уж начинающих-то можем отучить от привычки пользоваться чужой интеллектуальной собственностью.

Что же мы можем сделать?

Посудите сами:

- если установить хоть какую-то защиту, то большинство халевщиков уже не сможет просто так скопировать вашу программу;

- если вы знакомы с защитой от отладки, то crack'ер среднего уровня (до этого взламывавший только простые привязки и коды, которые сравниваются "в лоб") расшибится о преграды, которые вы поставите ему на пути (а может и вовсе их не найти, и уйдет, несолоно хлебавши :));

- наличие небольшого опыта — и только crack'ер с опытом взламывает защиту;

- немного выдумки и своих собственных заморочек — и можно выпускать ваш проект в свет.

Если я вас хоть немного заинтересовал, то давайте учиться. Хотя бы первые два пункта мы осилим, а дальше вам могут помочь только ваш опыт и ваше воображение.

Давайте посмотрим, как умные люди защищали свои программы от нелегального копирования. Логично, что надо привязаться к чему-либо, что существует только в оригинальном экземпляре. Здесь можно выделить привязки по пользователю, по машине и по переносному носителю (например, дискете или CD). Скорее всего, распространять вашу программу будут на переносном носителе, никто не станет ходить к каждому клиенту и копаться в его "железе". Так что рассмотрим именно этот случай. Привязки к дискете были популярны в прошлом, и методов было много — нестандартное форматирование (например, использование дорожек, которых не должно быть), запись информации в bad-блоки, считывание программой серийного номера дискеты. Но теперь существуют специальные программы, которые копируют дискету полностью (включая bad-блоки и корректируя серийный номер дискеты). Плюс к этому, дискета — носитель ненадежный, и портятся они не так уж и редко. Поэтому я хочу уделить внимание методу привязки именно к CD.

Мне очень нравится метод физической маркировки. При использовании этого метода диск в определенном месте специально портится (часто это делается лазером), а потом программа защиты пытается прочитать данные с этого места. Логично, что это ей не удастся, и она продолжает работу основной программы. А если вы просто скопировали диск, то этот участок будет вполне читаем, и программа защиты вас не пропустит дальше. Конечно, метод хороший, но сделать маркировку в домашних условиях очень сложно (хотя, в принципе, могут подойти любые царапины). Поэтому мне, как и в случае ключевой дискеты, больше нравится привязка к серийному номеру CD (я пока не слышал о способах его изменить). Узнать его нетрудно, а использовать можно, как захочешь (от простого сравнения в лоб, до использования в качестве ключа для шифрования). Мысль, полагаю, ясна — взять то, что можно достать, и что оригинально у каждого CD (или вероят-

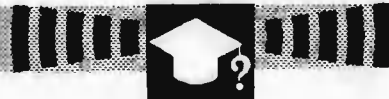
ность найти CD с таким же свойством мала), и на основе этого строить защиту.

Но привязать программу к носителю — это только половина дела. Теперь придется защищать саму защиту, как ни смешно это звучит. Дело в том, что написанные с благой целью отладчики стали основным оружием crack'еров. Описание методов защиты от отладки имеет свои недостатки — ведь никто не хочет раскрывать свои секреты, которые он использует. Неизвестность защиты существенно осложнит взлом. Поэтому будем пользоваться тем, что есть. С другой стороны — это азы защиты, и их надо обязательно знать (ведь из кирпичиков большие дома строятся).

Надо сказать, что взлом — это не такая простая задача, как кажется на первый взгляд. Что там взламывать — перевел в ассемблеровский код, пошагово прогнал (если нет SoftIce, то вручную), нашел защиту, поменял, к примеру, "равно" на "не равно", и все. Я тоже так считал, пока первый раз не попробовал. Конечно, с каждым разом приобретаешь опыт, но все же не так просто оттранслировать чужую программу. Если вы программист, то пытались, наверное, хоть раз разбираться в чужой программе — это же геморрой. Что это за переменная? Для чего эта функция? Как он это реализовал? Трудно! А если это ассемблер??? Влип очкарик! В ассемблеровских кодах программа под Windows (рассматриваю ее как наименее вероятную) выглядит жутким монстром. Мало того, что вы разбираетесь в чужой программе, иногда надо рыскать еще и по API'шным функциям. Короче, неопытный атакующий вынужден искать иголку в стоге сена (а кто сказал, что она должна быть одна? Представьте, что не найденная аопьнется потом куда-то). Конечно, у взломщика есть методы, которые помогают искать более эффективно. Например, поиск по функциям. Если функция называется "Protection", то понятно, что именно в ней и надо искать (хотя и не обязательно). Пусть будет, а ее код — пустышка, но замысловатая, с циклами, ветвлениями и прерываниями — для аажности. Пусть взломщик потрудится — ему полезно). Но на это надеяться не стоит. Одно из первых правил уже ясно — никаких намеков на то, что что-то имеет отношение к защите.

Перейдем к самой технике. Часто программа дизассемблируется сразу. Отсюда вывод — пусть одна часть программы расшифровывает другую, которая является основой. А можно из этого и плюсы для самой проги извлечь — не шифровать, а архивировать. Заодно программа и места меньше займет. Как, вы думаете, действуют многие инсталляторы? Именно так. Если это использовать, то у взломщика не будет кода на бумаге сразу, и придется немало повозиться, чтобы получить таковой. А если расшифровок будет несколько, да еще и расшифровывать на одно и то же место? Нервы ему попортим. А еще лучше — если расшифровку вести сразу в память! Но тут есть одна загвоздка. От SoftIce это нас не спасет, так как он действует в режиме реального времени, то есть во время выполнения программы, и может просматривать все: переменные, флаги, память и т.д. Плюс к этому, он очень распространен и известен, как лучший в своем роде (Спасибо тебе NuMega. Такого "зверя" на волю выпустила).

Ну, пойдем дальше. Всякий отладчик использует int 1



и `int 2` — перехватите их, и одно очко — в вашу пользу. Правда, если взломщик разберется, а чем дело, то может этот участок очень просто обойти. А внутри этого участка какая-то важная функция, без понимания которой разбираться дальше даже не стоит и пытаться (`crack'er` этого не знает, и прогоняет программу раз за разом :-)).

Также может помочь подсчет контрольных сумм кода. Ведь отладчик записывает код вызова `int 3` вместо байта программы, следовательно, изменяет контрольную сумму. В ответственных местах поставьте проверку контрольной суммы, и пусть злоумышленник только догадывается, что происходит в этом интересном месте.

Еще для выполнения отладки требуется, чтобы флаг `TF` был установлен. Хотите отключить отладчик — сбросьте этот флаг. Тут существуют “подводные камни” — отладчики команду `rorf` отслеживают, и восстанавливают флаг `TF`. Поэтому используйте незаметное сбрасывание флага — это приведет к отключению отладчика (это можно сделать разными способами), что явно не понравится взломщику. Также отладчики ловят команду `pushf` и сбрасывают флаг `TF`, но не все понимают `cs:pushf` и `es:pushf`. Так что, есть простор для действий.

Также есть способ проверить, работает ли отладчик — оценить время выполнения определенного участка кода. Дело в том, что если отладчик запущен, то время выполнения будет существенно больше, чем должно быть. Но возможна ситуация, когда машина пользователя будет перегружена, в этом случае вы получите тот же эффект, что и в ситуации с отладчиком. Хотя обнаружить такого монстра как `SoftIce` не составит большого труда — он же устанавливается! Спросите у `Windows`, установлен ли `SoftIce` на этой машине. Правда, не все же используют этот дебагер а плохих целях. Кто-то использует его и по назначению. Хотя, я думаю, таких немного.

И, конечно, надо сказать несколько слов о тех методах, которые может применить только опытный программист. Скажите, что подумает взломщик, если при отладке увидит код, которого не должно быть? Это метод

самогенерирующихся кодов. Где-то в недрах памяти хранится участок кода, который в нужный момент получает управление. Каково? Начинающему понять это вряд ли под силу. Если взять что-то из менее заоблачного, но, тем не менее, сложного для взломщика, то можно попробовать поместить стек в тело программы. А если еще и изменить положение стека?

Начинающим программистам я могу посоветовать самое мощное оружие, которое они могут использовать на данном этапе, да и на остальных тоже — выдумку! Если вышеописанные методы известны всем `crack'ерам`, то о ваших, на первый взгляд, элементарных, идеях не знает никто. А неизвестность, как я уже говорил — главное оружие против взломщика. Приведу простой пример. Мне надо было взломать одну программу, принцип работы которой заключался в простом сравнении двух величин — вводимой и кода (не совсем конечно “простом” — использовался способ расшифровки в память). Мне достаточно было просто найти место в памяти, где был записан код. Я даже не разбирался с самим кодом! Просто нашел функцию сравнения и просмотрел данные по адресу. И тут я подумал, а если бы данные величины сравнивались не просто “а лоб”, а, скажем, с прибавленной единицей к введенному коду? Пришлось бы разбираться во всем коде. Это увеличило бы время взлома раза а два, а то и больше. На самом деле, ничего сверхсложного...

Подведем итоги

Мы познакомились (разумеется, а самых общих чертах) с несколькими элементарными приемами, за исключением метода самогенерирующихся кодов и назначения стека в тело программы. Ничего сверхъестественного в этом нет, надо только попробовать. Вот теперь простой любопытный злоумышленник не сможет разгадать вашу защиту. Во многих случаях этого вполне достаточно, потому что тот, кто ценит свое время, не будет взламывать вашу программу из чистого любопытства. А оградив себя от начинающих, создать защиту для более опытных — дело времени.

VINCE,

E-mail: vincelabs@pisem.net
www.virtuality.nightmail.ru

Реализация криптоалгоритма Вигенера (1)

Введение

В номерах журнала “Ваш компьютер” за 2001 г. меня заинтересовал цикл статей, посвященных криптографии [1]. В частности, в первой (вводной) статье данного цикла рассматривался алгоритм Вигенера. Процесс шифрации оказался предельно простым, но как реализовать программу шифратор-дешифратор? В данной статье мы попробуем написать подобную программу. В настоящее время существует превеликое

множество языков программирования: Delphi, C, всевозможные разновидности Basic, JavaScript, PHP и т.д. Какой язык выбрать? Написать предполагаемую программу мы можем на любом из современных языков, но я выбрал Delphi (версия роли не играет).

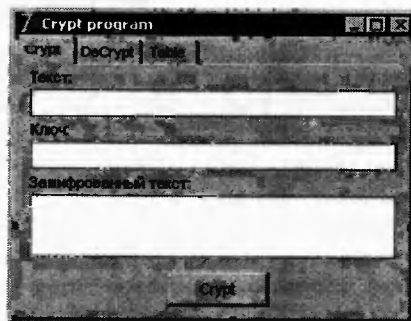
Процесс создания нашей будущей программы мы начнем с простой формы — каркаса будущей программы.

Создание каркаса

Создадим заготовку, как показано на рис.1. Представленная форма содержит 16 компонентов: `Edit1...Edit4`, `Memo1...Memo4`, `TabbedNotebook1`, `StringGrid1`, `Label1...Label6`. Для удобства эксплуатации программы мы используем компонент `TabbedNotebook` для образования “страничек”. Каркас готов, но нам предстоит еще кропотливая работа над приданием нашему “детищу” функциональности. Сейчас же мы изменим названия используемых компонентов:

- `Button1` — `bCrypt` //Кнопка, активирующая процесс шифрования;

Рис. 1



- Edit1 — eText1 //Поле для ввода шифруемого текста;
- Edit2 — eKey1 //Поле для ввода ключа шифрования;
- Memo1 — mCrypt //Поле для вывода зашифрованного текста;
- TabbedNotebook1 — tPages //Компонент, реализующий закладки;
- Button2 — bDeCrypt //Кнопка, активирующая процесс дешифровки;
- Edit3 — eCrypt2 //Поле для ввода зашифрованного текста;
- Edit4 — eKey2 //Поле для ввода ключа расшифровки;
- Memo2 — mDeCrypt //Поле для вывода расшифрованного текста;
- StringGrid1 — sg //Компонент (аналогия массивам) для наглядного отображения таблицы криптования. Также необходимо дать названия и кнопкам:
- bCrypt — "Crypt";
- bDeCrypt — "DeCrypt".

Целью данной статьи не является подробнейшее разъяснение создания криптографической программы с углублением в основы Delphi, так что я намеренно упускаю некоторые мелочи, не относящиеся к данной теме.

Итак, у нас имеется готовый каркас будущей программы, и осталось только "придать ему жизнь".

Кликаем на кнопку "Crypt", и открывается окно с кодом свежесозданной процедуры:

```
procedure TForm1.bCrypt1Click(Sender: TObject);
begin
end;
```

Перед процедурой я рекомендую вставить комментарий: "//Crypt-алгоритм". Это сделает восприятие листинга программы более удобным. Для начала объявим некоторые переменные, которые будут использоваться в алгоритме шифрования текста:

- txt — для хранения шифруемого текста;
- key — для хранения ключа шифрования;
- buf — переменная, играющая роль буфера (для хранения временных данных);
- ctxt — для хранения зашифрованного текста;
- i, b — для организации циклов for;
- a — для хранения алфавита;
- chrA, chrB — для хранения выборочных символов шифруемого текста и ключа.

Помимо локальных переменных, которые доступны лишь в рамках конкретной процедуры, мы объявим глобальную переменную ChrToNum_Code, которая хранит номер символа.

Теперь напишем первые строки кода, в которых осуществляем инициализацию (первичную установку) переменных:

```
//Инициализация переменных
cellfill();
txt:=eText1.Text;           //Записываем шифруемый текст
key:=eKey1.Text;           //Записываем значение ключа
al:='ABCDEFGHIJKLMNOPQRSTUVWXYZ0987654321 .,:;<>_!"/\^*?';
//Устанавливаем алфавит
```

Переменные подготовлены для того, чтобы начать работу в процессе криптования. А теперь запишем собственно сам алгоритм:

```
//Шифрование...
for i:=1 to length(txt) do //От 1 до длины текста...
begin
  if b>length(key) then //Символ выходит за пределы размеров
    b:=1;               //ключа, то стартуем с единицы
  ChrToNum(txt[i]);      //Получаем код буквы текста
  chrA:=ChrToNum_Code;   //Сохраняем код
  ChrToNum(key[b]);      //Получаем код буквы ключа
  chrB:=ChrToNum_Code;   //Сохраняем код
  //Крипуем текст
  jcrypt(chrA,chrB);
  ctxt:=ctxt+Chr_Crypted; //Получаем зашифрованный текст
  b:=b+1;
end;
```

Мы начали с того, что запустили главный цикл (for), который будет выбирать один символ из зашифрованного текста и один символ из ключа. Далее, используя полученные символы, алгоритм выдает их зашифрованный вариант (тоже один символ). Если откомпилировать данный кусок кода, то Delphi выдаст сообщения об ошибках. Действительно, функций cellfill, ChrToNum и jcrypt не существует. Мы напишем эти функции чуть позже, а сейчас продолжим рассмотрение кода.

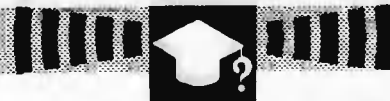
В алгоритме участвует переменная b, которая помогает реализовать цикл обработки ключа. Если бы мы не стали использовать b, то алгоритм осуществил бы неправильную шифрацию. В "правильном" коде мы используем переменную i для перемещения по шифруемому тексту, а b — для перемещения по ключу. Как известно, длина шифруемого текста и длина ключа (причем в большинстве случаев) могут совершенно не совпадать, так что реализовывать обработку текста и ключа через переменную i неправильно. Например, текст — "Суперсекретные данные...", а ключ — "Ананас". Размер текста — 24 символа, а ключа — 6. Когда в цикле переменная i будет больше 6, а ключ из шести символов (т.е. меньше), циклу будет неоткуда брать ключевые символы, и произойдет неправильное кодирование.

Для сбрасывания цикла с переменной b мы используем конструкцию условия:

```
if b>length(key) then
  b:=1;
```

В этом коде мы не даем циклу зайти за пределы длины ключа, и тем самым обеспечиваем корректную работу алгоритма шифрования. Теперь для завершения процедуры bCrypt.OnClick добавим код, "показывающий" результат стараний программы:

```
//Выводим зашифрованный текст
mCrypt1.Clear; //Очищаем от всех предыдущих данных
//поле для вывода зашифрованного текста
Memo1.Text:=ctxt; //Записываем зашифрованный текст
```



Вот и весь код процедуры шифрования, но для его корректной работы необходимо написать две функции: ChrToNum и jscrypt.

Добавляем необходимые функции

Код первой функции:

```
function JCrypt(x:integer;y:integer):integer;
begin
  Chr_Crypted:=Form1.sg.Cells[x,y];    //Компонент StringGrid1
end;
```

В заголовке функции прописаны два параметра: X — номер текущего символа текста и Y — номер текущего символа ключа (это обусловлено тем, что мы шифруем “посимвольно”). Переменные X и Y передаются в StringGrid, и производится считывание “зашифрованного символа” (в соответствии с таблицей кодирования).

Код второй функции выглядит так (или почти так!):

```
function ChrToNum(char:string):string;
begin
  {Для 85 символов}
  if char='А' then
    ChrToNum_Code:=1;
  if char='В' then
    ChrToNum_Code:=2;
  if char='С' then
    ChrToNum_Code:=3;
  <Пропущено>
  if char='Ю' then
    ChrToNum_Code:=84;
  if char='Я' then
    ChrToNum_Code:=85;
end;
```

Здесь приведен сокращенный вариант функции, так как текст ее полной версии слишком большой. Главное — понять принцип написания структуры функции, и все будет работать. Данная функция реализует перекодировку символа в его порядковый номер в соответствии с установленным алфавитом (переменная А). Принцип работы функции предельно прост.

Если <передаваемый_функции_параметр_char> равен <текущему_символу>, то присваиваем переменной ChrToNum порядковый номер данной символа (char).

И, наконец, третья функция:

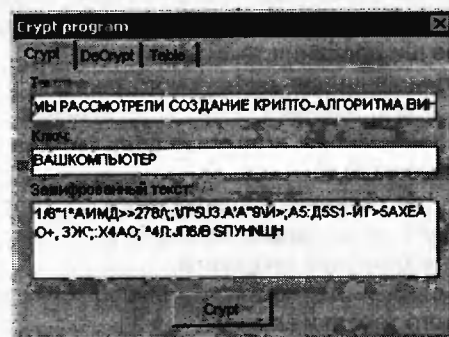
```
function cellfill():integer;
var ldn,cnt,i,j,cc,cc2,cc3,ci:integer;
    al:string;
begin
  al:='ABCDEFGHIJKLMNPOQRSTUVWXYZ0987654321 .,:;<>_-|»/
  \*+?ABBGDEJZHIJKLМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ';
  // ЗАПОЛНИМ ТАБЛИЦУ КОДИРОВАНИЯ
  for ldn:=1 to 85 do //От первой строки до последней
    Form1.StringGrid1.Cells[1,ldn]:=al[ldn];
    //Заполняем столбец 1 алфавитом
  for ldn:=1 to 85 do
    Form1.StringGrid1.Cells[2,ldn]:=al[ldn+1];
    //Заполняем столбец 2 алфавитом
  //+1 - это смещение, которое необходимо для правильного
  //заполнения таблицы кодирования, короче - так надо! Вы можете
  //изменить таблицу кодирования, изменив содержимое функции
  //cellfill.
  for ldn:=1 to 85 do
    Form1.StringGrid1.Cells[3,ldn]:=al[ldn+2];
    //Заполняем столбец 3 алфавитом
  //и так далее
```

```
for ldn:=1 to 85 do
  Form1.StringGrid1.Cells[4,ldn]:=al[ldn+3];
for ldn:=1 to 85 do
  <Пропущено>
  Form1.StringGrid1.Cells[84,ldn]:=al[ldn+83];
for ldn:=1 to 85 do
  Form1.StringGrid1.Cells[85,ldn]:=al[ldn+84];
  //Блок коррекции
  ci:=1;
  cc3:=85;
  for cc2:=2 to 85 do //От 1 и до 84 строки...
  begin
    ci:=1;
    for cc:=cc3 to 85 do //От cc3-столбца до 84
      begin
        Form1.StringGrid1.Cells[cc,cc2]:=al[ci];
        //Записываем текущий символ
        ci:=ci+1; //Берем следующий символ
      end;
    cc3:=cc3-1; //
  end;
  {original
  for cc:=53 to 53 do
  begin
    Form1.StringGrid1.Cells[cc,2]:=al[ci];
    ci:=ci+1;
  end;}
  //Заполнение шапок (гориз. и верт.)
  for ldn:=0 to 85 do
    Form1.StringGrid1.Cells[0,ldn]:=al[ldn];
  for ldn:=0 to 85 do
    Form1.StringGrid1.Cells[ldn,0]:=al[ldn];
  end;
```

В коде нашей программы функция cellfill — самая большая, она отвечает за правильное заполнение таблицы кодирования (у нас — это компонент StringGrid1 (sg)). Я для себя разделил работу функции на два этапа: первый — первичное заполнение таблицы; второй — дополняющие циклы. Если вам очень интересно узнать, зачем понадобились дополняющие циклы, то прокомментируйте их код и запустите программу. В компоненте sg вы увидите, что таблица заполнилась не полностью, а незаполненные места отобразятся в виде непонятных символов.

Итак, мы рассмотрели процесс реализации криптографического алгоритма Вигенера в среде Borland Delphi. Для проверки функционирования написанной программы произведем компиляцию проекта (Ctrl-F9) и запуск F9. Результат работы программы виден на рис.2.

Рис. 2



Во второй части статьи мы рассмотрим алгоритм дешифрации текста.

Литература

1. S.Kurowski. Тайны криптологии. — Радиомир. Ваш компьютер, 2001, №№6-9.



DOS в среде Windows

Б.ШИЛЬНИКОВ,
Читинская обл., п.Дарасун.

На страницах журнала уже освещались вопросы запуска DOS-программ в среде Windows Millennium (Windows Me) [1]. Еще один, более простой и корректный способ описан в инструкции к программе *spu_pred*, распространяемой среди работодателей Пенсионным фондом РФ [2].

Часто при запуске DOS-программ в среде Windows Me выдается сообщение "TOO MANY OPEN FILES" (Слишком много открытых файлов), и запустить программу не удастся.

По умолчанию значение переменной FILES (которое устанавливает максимально допустимое количество файлов, т.е. ограничивает число одновременно открытых файлов для MS-DOS-приложений) равно 30. Однако многие приложения MS-DOS требуют гораздо большего количества одновременно открытых файлов.

Решить эту проблему обычным способом, т.е. добавлением в файл *Config.sys* строки *FILES=N* (N — максимальное число открытых файлов), не удастся, т.к. Windows Me при перезагрузке удаляет все строки из файла *Config.sys*.

Для увеличения максимально возможного числа одновременно открытых файлов в Windows Me служит параметр *PerVMFiles*, который задается в секции [386Enh] файла *System.ini* следующим образом:

```
PerVMFiles=<число>.
```

Максимально возможное значение FILES для каждой виртуальной машины MS-DOS равно 255. С учетом того, что каждой виртуальной машине MS-DOS по умолчанию выделяется 30 файловых указателей, значение *PerVMFiles* не может превышать 225, что в сумме с 30 дает 255.

Для вступления изменений в силу необходимо перезагрузить систему.

Также в [2] приводятся рекомендации по настройке Windows 2000 (Professional Server) для работы с DOS-приложениями.

1. Увеличить количество открываемых файлов.

Для этого в файле *C:\WINNT\SYSTEM32\CONFIG.NT* следует подправить команду FILES, задав требуемое количество файлов. Например:

```
FILES=145.
```

2. Установить переменную окружения CLIPPER.

Для этого в файл *C:\WINNT\SYSTEM32\AUTOEXEC.NT* надо добавить строку *SET CLIPPER=Fппп*, где ппп — количество открываемых файлов. Например:

```
SET CLIPPER=F140.
```

3. Для того чтобы работали DPMI16-приложения (*press.exe*, *amba.exe* и т.д.), в *C:\WINNT\SYSTEM32\AUTOEXEC.NT* обязательно должна присутствовать строка *In%SystemRoot%\system32\dosx*. Если задача запускается с помощью ярлыка, и в его свойствах указан свой *autoexec*, то и в нем должна присутствовать та же строка.

Часто при работе DOS-приложений в среде Windows при выводе данных на матричный принтер печать происходит со сбоями — принтер перед началом печати надолго "задумывается" (около минуты), печать происходит с увеличенным межстрочным интервалом, или печать начинается сразу, но потом вывод данных прекращается. В таких случаях я запускаю такую программу в режиме "чистой DOS".

Если в вашей практике часто приходится запускать DOS-программы, можно создать стартовое меню для Windows 9x. При выборе пункта 5 этого меню загружается только DOS без запуска Windows 9x. Для создания меню нужно отредактировать файл *Msdos.sys*, находящийся в корневой директории. В разделе [Options] этого файла необходимо добавить строку:

```
BootMenu=1.
```

Редактирование удобно производить в среде файловых менеджеров FAR или Norton Commander. Перед редактированием для файла *Msdos.sys* необходимо снять атрибуты "Только для чтения", "Скрытый" и "Системный", иначе измененный файл не удастся записать на диск. Последовательность команд — F9/Файлы/Атрибуты файлов. После записи файла необходимо снова установить эти атрибуты.

Теперь при запуске Windows на экран будет выдаваться стартовое меню, для запуска DOS-программ нужно выбрать пункт 5. *Command prompt only*. Для запуска Windows из этого режима нужно просто дать команду *win*.

Если вам лень создавать стартовое меню, можете при загрузке Windows просто удерживать клавишу *Ctrl*, и через некоторое время на экране появится стартовое меню.

Если этих мер недостаточно, приходится запускать такую DOS-программу из командной строки после загрузки компьютера с загрузочной дискеты Windows 9x. Если система выдает сообщение "Слишком много открытых файлов", придется увеличить число открытых файлов и буферов в файле *config.sys* на загрузочной дискете. Однако слишком большое число файлов и буферов также нежелательно, так как для запускаемой программы может не хватить памяти.

Литература

1. MAMAICH. Возвращение DOS в Windows Millennium. — Радиомир. Ваш компьютер, 2002, №4, С.27.

2. Настройка Windows Millennium для работы с DOS-приложениями. Инструкция Пенсионного фонда РФ к программе *spu_pred* v.4.0a.

3. Ш.Дралле и др. Hardware. Сотни полезных советов. — Киев, BHV, 1996, С.240-243.



С.РЮМИК,
г.Чернигов.

“ONEchip3” на микроконтроллере AT89C2051

Чтобы дойти до цели, надо, прежде всего, идти.
Оноре де Бальзак

Мод-чип (МЧ) для игровой приставки “PSone” фирмы Sony обычно выполняют на PIC-контроллере 12C508A [1] или 16F84A [2]. Первый из них — дешевый, но однократно программируемый, а второй — многократно программируемый, но дорогой. Можно ли объединить их преимущества, оставив “за бортом” недостатки? Оказывается, можно, если применить FLASH-контроллер AT89C2051 фирмы Atmel.

Жаркие споры о том, какой тип контроллера лучше, давно перешли в разряд философских. Общего решения проблемы не существует, зато имеется ряд частных следствий. Например, каждый контроллер предназначен для выполнения своего класса задач. Если поставленной цели можно достичь несколькими путями (с разными типами контроллеров), то выстраивают цепочку приоритетов. Для МЧ приставки “PSone” она выглядит так: 1 — низкая стоимость, 2 — доступность приобретения, 3 — возможность оперативного перепрограммирования, 4 — малые габариты. Остальные параметры не существенны, поскольку у контроллеров одного класса они примерно одинаковы.

В табл.1 приведены основные отличия микросхем PIC16F84A-04/P (PIC) и AT89C2051-12PC (MCS). Кон-

троллер PIC12C508A “снят с рассмотрения”, так как не укладывается в приоритет номер 3 — многократность программирования.

“Водораздел” между PIC и MCS лежит в основе основ — идеологии архитектуры вычислительной системы. RISC-процессоры оперируют такими понятиями как конвейерная обработка данных и параллельные вычисления. CISC-процессоры основаны на стандартной модели фон Неймана.

Без сомнения, PIC более современны, чем MCS. Как следствие, при одинаковой тактовой частоте PIC выполняет одноцикловые команды в три раза быстрее. Имея оригинальную систему команд, PIC позволяет создавать компактные по длине программы. Улучшенная технология изготовления КМОП-структур до минимума снижает его мощность потребления.

Однако все эти преимущества сводятся на нет высокой стоимостью, а в цепочке приоритетов данный показатель стоит под номером 1. Кроме того, техническое превосходство PIC в быстродействии частично компенсируется гибкой системой команд и увеличенным вдвое объемом FLASH-памяти MCS.

Фирма Atmel, создавая линейку упрощенных контроллеров AT89C2051 (1051, 4051), сделала ставку на низ-

кую цену при достаточно широких возможностях, и не ошиблась. Ниша максимально дешевых, но надежных и неприхотливых в эксплуатации контроллеров сейчас принадлежит именно Atmel.

Итак, выбор контроллера для МЧ “PSone” сделан. Остановка за малым — провести трансляцию программного кода прошивки “ONEchip2” (PIC16F84A), превратив ее в “ONEchip3” (AT89C2051). Поставленная задача имеет важное прикладное значение, и не только в области игровых приставок, поэтому рассмотрим ее поэтапно.

Методика перевода кодов PIC в MCS

Этап 1 — получение дизассемблированного текста программы. Базой для работы с PIC-контроллерами считается интегрированная среда MPLAB [2], поэтому выбор инструмента перевода напрашивается сам собой. Однако дизассемблер MPLAB считает “ниже своего достоинства” предоставлять буквенные метки переходов по тексту программы и заменять цифровые значения номеров регистров символическими именами. Для нашего случая такой вариант дизассемблирования не годится.

Лучше воспользоваться встроенным дизассемблером программы IC-PROG (автор Bonny Gijzen, <http://www.ic-prog.com/icprog105A.zip>, 534 Кб). Это многофункциональный бесплатный программатор, успешно работающий и без подключения “аппаратной начинки”.

После запуска IC-PROG на выполнение следует выбрать язык общения (“Settings — Options — Language — Russian”) и тип контроллера (“Настройки — Микросхемы — Microchip PIC — PIC16F84A”). Затем ввести файл с кодами прошивки из [2] (“Файл — Открыть Файл... — ONEchip2.hex”) и изменить форму показа (“Вид Окна — Ассемблерный”). В результате на экране монитора появится дизассемблированный текст программы

Табл. 1

№	Параметр	PIC16F84	AT89C2051
1	Архитектура	RISC	CISC
2	Система команд	PIC-micro	MCS-51
3	Разрядность команд, битов	14	8...24
4	Количество типов команд	35	111
5	Длительность выполнения команд в тактах	4...8	12...24
6	Объем памяти программ	1K	2K
7	Число входов-выходов	13	15
8	Регистры общего назначения	68	32
9	Тактовая частота, МГц	0...4	0...12
10	Напряжение питания, В	4...5,5	2,7...6
11	Корпус микросхемы	18-pin	20-pin
12	Ток потребления, мА	0,01...2	0,1...15
13	Ориентировочная цена, USD	3,5	1,3

Рис. 1

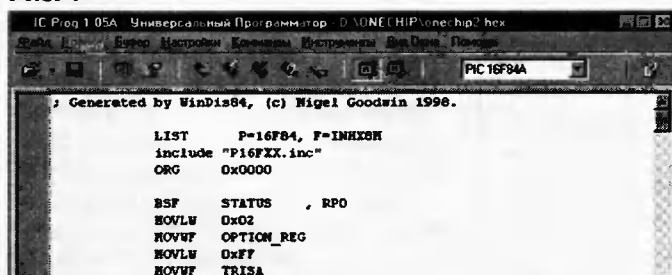


Табл. 2

№	Команда PIC16F84	Действие	Команда AT89C2051	Действие
1	ADDWF PCL,f	Прибавка к счетчику команд PCL значения аккумулятора W	MOVC A,@A+PC	Прибавка к счетчику команд PC значения аккумулятора A
2	BCF PORTB,00...07	Очистка бита 0...7 регистра PORTB	CLR P1.0...P1.7	Очистка бита 0...7 порта P1
3	BCF STATUS,RP0 BSF STATUS,RP0	Очистка (установка) бита RP0 регистра STATUS	-	Регистр STATUS отсутствует
4	BTFSC 0x1D...0x24,00...07	Проверка сброса бита 0...7 регистра f1D...f24 с пропуском следующей команды	SETB RS0 CJNE R0...R7,#\$01...FF,Rel	Проверка сброса бита 0...7 регистра R0...R7 банка 1 с переходом по адресу Rel
5	BTFSC PORTB,00...07	Проверка сброса бита 0...7 регистра PORTB с пропуском следующей команды	JNB P1.0...P1.7,Rel	Проверка сброса бита 0...7 порта P1 с переходом по относительному адресу Rel
6	BTFSS PORTB,00...07	Проверка установки бита 0...7 регистра PORTB с пропуском следующей команды	JB P1.0...P1.7,Rel	Проверка установки бита 0...7 порта P1 с переходом по относительному адресу Rel
7	BTFSS STATUS,C	Проверка установки бита C регистра STATUS с пропуском следующей команды	JNC Rel	Проверка установки флага переноса CY с переходом по относительному адресу Rel
8	CALL Label	Вызов подпрограммы с меткой Label	ACALL Label	Вызов подпрограммы с меткой Label
9	CLRF 0x1D...0x24	Очистка регистра f1D...f24	SETB RS0 MOV R0...R7,#\$00	Очистка регистра R0...R7 банка 1
10	COMF 0x17...0x1C	Инверсия содержимого регистра 0x17...0x1C	XRL \$02...07,\$\$FF	Инверсия содержимого регистра R2...R7 банка 0 (прямой адрес)
11	DECFSZ 0x17...0x1C GOTO Label	Цикл с уменьшением содержимого регистра f17...f1C	SETB RS0 DJNZ R2...R7,Label	Цикл с уменьшением содержимого регистра R2...R7 банка 0
12	GOTO Label	Переход по адресу с меткой Label	AJMP Label	Переход по адресу с меткой Label
13	INCF 0x1D...0x24,f	Прибавка единицы к содержимому регистра f1D...f24	SETB RS0 INC R0...R7	Прибавка единицы к содержимому регистра R0...R7 банка 1
14	MOV 0x00...0xFF	Пересылка константы 0...0FFh в аккумулятор W	MOV A,#\$00...FF	Пересылка константы 0...0FFh в аккумулятор A
15	MOVF 0x1D...0x24	Пересылка содержимого регистра f1D...f24 в аккумулятор W	SETB RS0 MOV R0...R7,A	Пересылка содержимого регистра R0...R7 банка 1 в аккумулятор A
18	MOVWF 0x17...0x1C	Пересылка содержимого аккумулятора W в регистр f17...f1C	CLR RS0 MOV R2...R7,A	Пересылка содержимого аккумулятора A в регистр R2...R7 банка 0
17	MOVWF 0x1D...0x24	Пересылка содержимого аккумулятора W в регистр f1D...f24	SETB RS0 MOV R0...R7,A	Пересылка содержимого аккумулятора A в регистр R0...R7 банка 1
18	MOVWF OPTION, MOVWF TRISA, MOVWF TRISB	Пересылка содержимого аккумулятора W в регистры OPTION, TRISA, TRISB	-	Регистры OPTION, TRISA, TRISB отсутствуют
19	MOVWF PORTB	Пересылка содержимого аккумулятора W в регистр PORTB	MOV P1,A	Пересылка содержимого аккумулятора A в порт P1
20	NOP	Холостая операция	NOP	Холостая операция
21	RETLW 0x00...0xFF	Возврат из подпрограммы с кодом 0...0FFh в аккумуляторе W	MOV A,#\$00...FF RET	Возврат из подпрограммы с кодом 0...0FFh в аккумуляторе A
22	RRF 0x17...0x1C,f	Сдвиг вправо содержимого регистра f17...f1C через флаг переноса	CLR RS0 MOV R0,A MOV A,R2...R7 RRC A MOV R2...R7,A MOV A,R0	Сдвиг вправо содержимого регистра R2...R7 банка 0 через флаг переноса с сохранением значения аккумулятора A в промежуточном регистре R0 банка 0
23	XORLW 0x00...0xFF	Исключающее ИЛИ содержимого аккумулятора W и константы 0...0FFh	XRL A,#\$00...FF	Исключающее ИЛИ содержимого аккумулятора A и константы 0...0FFh

(рис.1), который следует сохранить на диске ("Файл — Сохранить Файл Как... — ONEchip2.asm").

Этап 2 — формальная замена кодов PIC-MCS. Полученный на

первом этапе текст программы необходимо рассортировать по типам команд. Их используется не все 35, а гораздо меньше. В табл.2 представлены результаты сортировки, где в левой половине находятся расшифровки PIC-инструкций, а в правой — эквивалентные им замены MCS-команд в формате ассемблера TASM v3.01 фирмы Speech Technology Inc. (<http://www.geocities.com/SiliconValley/Bay/5626/tasm301.zip>,



Табл. 3

PIC	MCS	PIC	MCS	PIC	MCS
Банк 0		Банк 1		PORTB	P1
-	R0	f1D	R0	RB0	P1.0
-	R1	f1E	R1	RB1	P1.6
f17	R2	f1F	R2	RB2	P1.2
f18	R3	f20	R3	RB3	P1.3
f19	R4	f21	R4	RB4	P1.4
f1A	R5	f22	R5	RB5	P1.5
f1B	R6	f23	R6	RB6	-
f1C	R7	f24	R7	RB7	-

оценочная версия 138 Кб). В табл.3 приведено соответствие номеров регистров и портов входа-выхода.

Как видно, мнемоника ассемблерных команд двух процессоров похожа, но выполняемые ими функции во многом отличаются. Вдобавок, используются разные буквы в обозначении аккумулятора, портов, регистров, счетчика команд.

Читатель, искушенный в программировании, может заметить, что цепочка команд SETB RS0 MOV R0...R7,#\$00 в табл.2 не оптимальна по числу байтов. Но, если заменить ее более короткой инструкцией MOV \$08...0F,\$\$00, то в дальнейшем нельзя будет повысить быстродействие программы за счет удаления повторяющихся команд SETB RS0. В целом, перевод команд в направлении PIC—MCS приводит к увеличению объема программы примерно в полтора раза, что вполне допустимо, учитывая объем 2 Кб FLASH-памяти MCS.

Рутинную работу по формальной замене кодов можно поручить утилите "asmtxt.exe", упрощенный вариант которой приведен в листинге 1. Утилита преобразует ассемблерный файл "ONEchip2.asm", полученный дизассемблером IC-PROG, в текстовый файл "ONEchip2.txt". В нем изменяется "шапка" программы введением include-файла "8051.h" из библиотеки MCS-51, а также в конце удаляются все инструкции после ORG 0x2000.

Прежний текст PIC-программы для удобства сравнения сдвигается построчно вправо, превращаясь в комментарии с добавлением знака ";". Метки адресов остаются без изменений. Образец полностью отформатированного текста представлен в листинге 2.

Этап 3 — проверка логики рабо-

ты программы. "Шаманские" действия и манипуляции с ассемблерным текстом программы "ONEchip2" принесли свои плоды. Теперь коды соответствуют системе команд MCS-51. Однако отличия в архитектуре PIC и MCS слишком велики, чтобы почитать на лаврах. В частности, отсутствие регистров STATUS, OPTION, TRISA, TRISB может потребовать внесения корректив в программу. К счастью, "ONEchip2" не использует прерывания, энергосберегающие режимы и систему WatchDog, поэтому неиспользуемые регистры и обслуживающие их команды бесследно удаляются из текста.

Далее необходимо убедиться в отсутствии логических ошибок при помощи упоминавшегося ранее ассемблера TASM. После распаковки архива tasm301.zip в выбранную директорию, следует скопировать в нее файл "ONEchip2.txt" и создать там же файл "ONEchip.bat" с одной-единственной командой:

TASM -51 -a -b -h -fff ONEchip2.txt.

Запуск этого файла на выполнение должен привести к появлению на экране монитора надписи "tasm: Number of errors = 0". В противном случае придется внимательно проанализировать текст программы на наличие ошибок и устранить их по выводимым на экран подсказкам.

Этап 4 — оптимизация кода программы. Формальный перевод кодов, проведенный на втором и третьем этапах, несет в себе определенную долю избыточности. Например, если в программе встречаются две близлежащие команды, в которых происходит обращение к регистрам

одного и того же банка, то установка номера банка может производиться один раз, а не два. Типичный случай показан в листинге 2, где можно сэкономить команду CLR RS0 в четвертой строке. Перечень подобных стандартных ситуаций приведен в табл.4.

В завершение следует удалить команды, временно (и бесполезно) хранящие содержимое аккумулятора. Их после преобразования PIC-MCS остается довольно много.

Замену кодов проводят в любом текстовом редакторе. Эта операция требует осторожности, главное условие — чтобы в блоке заменяемых кодов не было меток переходов из других частей программы. Если метки имеются, то требуется предварительный анализ логики работы, для того чтобы не перепутались номера

Табл. 4

Обычный вид	Оптимальный вид
JNB Label SJMP Rel Label ...	JB Rel Label ...
JB Label SJMP Rel Label ...	JNB Rel Label ...
AJMP Addr	SJMP Rel
MOV A,\$\$00...FF MOV R0...R7,A	MOV R0...R7,\$\$00...FF
SETB RS0 MOV R7,A SETB RS0 DJNZ R2,Rel	SETB RS0 MOV R7,A DJNZ R2,Rel
CLR RS0 ANL A,R4 CLR RS0 MOV R5,A	CLR RS0 ANL A,R4 MOV R5,A

банков 0 и 1, а вместе с ними и номера одноименных регистров R0...R7.

Этап 5 — формирование точных отрезков времени. Период абстрактных операций по замене одних грамматических символов другими закончился. На руках имеется оптимизированная ассемблерная программа в кодах MCS-51, но цена ей невелика. Если пользователь попытается сразу же прошить контроллер AT89C2051 и подключить его к "PSone", то результатом будет бесконечное вращение лазерного диска и темный экран телевизора...

Пора вспомнить о времени выполнения ассемблерных команд, которое в MCS в 3 раза выше, чем у PIC. Из-за этого длительность тактового импульса цикла SCEE будет растянута с 4 до 12 мс, что недопус-

Листинг 1

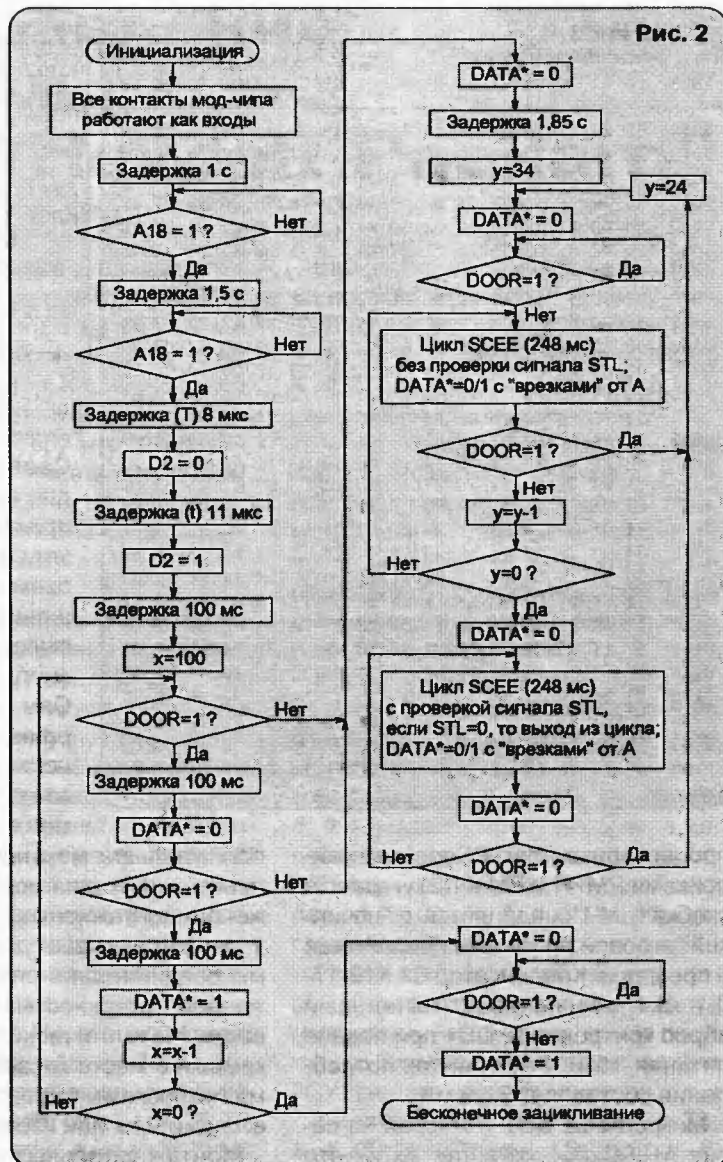
```
//УТИЛИТА asmtxt.exe ПЕРЕВОД ФАЙЛОВ ASM в TXT MCS51
#include <stdio.h> //Журнал РМ.ВК, 22.06.2003 г
#include <dos.h> //Turbo C-3.0, модель памяти Small
//Упрощенная версия, предварительное форматирование
void main(argc,argv) //Аргументы в командной строке
int argc; //Количество аргументов
char *argv[]; //Указатель на аргументы
{ FILE *in,*out; //Указатели на файлы чтения-записи
int ch; //Символ, вводимый из файла
static char name[20], hn[20]; //Массивы имен
int cnt, i=0; //Счетчики строк-столбцов, букв
if (argc < 2) //Проверка числа аргументов
{ puts("Неверный синтаксис команды!");
puts("Пример: asmtxt <filename.asm>");
exit(1); //Аварийный выход
}
strcpy(name, argv[1]);
if ((in = fopen(name, "rb")) == NULL)
{ printf("Нельзя открыть файл %s!", name);
exit(1); //Аварийный выход
}
while ((name[i] != 46) && (i < strlen(name))) i++;
if (i > 8) i=8;
strcpy(hn, name); //Изменение расширения *.txt
hn[i]=46; hn[i+1]=116; hn[i+2]=120; hn[i+3]=116;
if (out = fopen(hn, "wb")) == NULL)
{ printf("Нельзя создать файл %s!", hn);
exit(1); //Аварийный выход
}
for(cnt=-5; cnt < 1; cnt++) //Пропуск первых строк
{ while ((ch = getc(in)) != 0x0A) //Новая строка
{ if (ch == EOF) //Проверка окончания файла
{ printf("Ошибка данных в файле %s!", name);
exit(1); //Аварийный выход
}
}
//Предварительное форматирование строк
fputs("***** AT89C2051 *****\r\n", out);
fputs(" #include \"8051.h\"\r\n", out);
fputs("start .ORG $0000\r\n", out);
while ((ch = getc(in)) != EOF) //Чтение байта
{ if (ch == 0x0A) cnt = 0; //Новая строка
if (cnt++ == 13) //Столбец начала команд
{ if (ch == "N"); //Команда NOP без изменений
else
{ if (ch == "O") //Псевдокоманда ORG
{ fputs(".END\r\n", out); //Финиш
return; //Досрочный выход из цикла
}
else fputs(";", out); //Знак ";"
}
//Место для команд перевода PIC-MCS по табл.2, 3
}
putc(ch, out); //Трансляция байтов файла
}
fclose(in); fclose(out); //Закрытие файлов
}
```

Листинг 2

```
Label_001E MOV A, #32 ;MOVLW 0x32
Label_0005 CLR RS0 ;MOVWF 0x1A
MOV R5, A
CLR RS0 ;DECFSZ 0x1B
DJNZ R6, Label_0005 ;GOTO Label_0005
Label_0003 ...
```

тимо. Поправить положение можно при помощи подбора констант, определяющих точные отрезки времени, и, если понадобится, перекомпоновкой подпрограмм.

лов, например, русифицированным FD51 v2.01 (http://www.inta.portal.ru/zako/Programs/fd51_201.zip, 33 Кб). Заодно можно несколько изменить логику работы программы, увеличив



Судя по данным, приведенным в [2], коррекции подлежат всего 4 точки: в блоке задержки 1 мс, в цикле SCEE с проверкой и без проверки сигнала DOOR, а также в формирователе левого и правого фронтов выходного сигнала D2.

Подобрать константы легче всего отладчиком, имеющим измеритель тактовых интерва-

времени автоматического отключения МЧ с 5 до 15 с. Это полезно на практике для особо забывчивых пользователей. Обобщенный алгоритм функционирования обновленной версии программы показан на рис.2.

Остается переименовать файл "ONEchip2.txt" в "ONEchip3.asm" и при помощи TASM получить бинарные коды прошивки контроллера в файле "ONEchip3.obj" (листинг 3). Копия файла размещена на сайте журнала <http://radio-mir.com>. Для программирования микросхемы AT89C2051 можно воспользоваться фирменным программатором, схема которого приведена в [3].

Конструкция и детали

Электрическая схема МЧ изображена на рис.3. Соответствие точек подключения к двум разновидностям



Листинг 3

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	80	22	FF	32	4F	4E	45	43	48	49	50	32	76	65	72	33
0010	2E	30	2D	32	52	53	4D	53	4F	46	54	32	32	30	30	33
0020	20	20	20	32	75	90	FF	75	81	17	02	01	90	D2	D3	7A
0030	03	79	FF	E4	22	74	32	C2	D3	FD	7E	5B	30	95	0F	DE
0040	FB	DD	F7	D2	D3	00	C2	D3	E4	22	7E	5B	00	00	DE	FC
0050	DD	F8	D2	D3	79	00	C2	D3	E4	22	75	90	3D	E4	22	75
0060	90	FF	E4	22	75	90	3D	74	04	80	CC	C2	D3	7D	04	7E
0070	28	30	95	21	A2	92	92	96	30	95	1A	DE	F4	A2	92	92
0080	96	DD	EC	D2	D3	00	C2	D3	E4	22	7E	28	A2	92	92	96
0090	00	A2	92	92	96	DE	F5	A2	92	92	96	DD	ED	D2	D3	79
00A0	00	C2	D3	E4	22	C2	D3	7B	04	D2	D3	78	00	74	01	5C
00B0	70	19	11	2D	7D	12	74	04	11	37	D2	D3	DA	07	74	01
00C0	59	70	5B	11	2D	D2	D3	DD	ED	80	08	74	48	11	37	D2
00D0	D3	7A	FF	D2	D3	E8	31	20	C2	D3	F4	FF	7C	09	80	07
00E0	F8	EF	13	FF	E8	50	04	11	6B	80	03	11	64	00	D2	D3
00F0	DA	07	74	01	59	70	27	11	2D	C2	D3	DC	E3	11	64	D2
0100	D3	DA	07	74	01	59	70	16	11	2D	11	64	D2	D3	DA	07
0110	74	01	59	70	09	11	2D	D2	D3	08	C2	D3	DB	B5	E4	22
0120	04	83	22	53	43	45	45	00	74	64	11	37	D2	D3	7D	64
0130	30	90	1D	74	64	11	37	11	5A	30	90	14	74	64	11	37
0140	11	5F	D2	D3	DD	EA	11	5A	20	90	FD	75	90	FF	80	FE
0150	11	5A	C2	D3	7A	1E	11	35	C2	D3	DA	FA	75	90	3D	7A
0160	07	11	35	C2	D3	DA	FA	D2	D3	7B	22	80	04	D2	D3	7B
0170	18	11	5A	20	90	FD	D2	D3	7C	FF	11	A5	20	90	EE	D2
0180	D3	DB	F7	11	5A	7C	00	11	A5	11	5A	20	90	FF	80	F7
0190	D2	D3	7E	14	11	35	D2	D3	DE	FA	30	93	FD	7F	1D	11
01A0	35	D2	D3	DE	FA	30	93	FD	00	75	90	EF	00	00	00	00
01B0	00	75	90	FF	21	28	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

процессорных плат "PSone" с маркировками PM-41 и PM-41(2) указано в скобках. МЧ соединяется с "начинкой" игровой приставки при помощи 9 проводов. Конденсатор C1 K10-17-0,1 мкФ обеспечивает начальный сброс контроллера DD1 при подаче питания +5 В. Рабочий ток потребления составляет 6...8 мА.

Микросхема DD1 — любая из серии AT89C2051-xxPz, где "xx" — это число "12" или "24", обозначающее максимальную тактовую частоту в мегагерцах; "z" — это буква "C" или "I", указывающая на обычный (0...+70°C) или коммерческий (-40...+85°C) диапазон рабочих температур. Буква "P" в названии означает 20-выводной DIP-корпус. Не запрещается использовать контроллер в SOP-корпусе AT89C2051-xxSz, но к его выводам будет сложнее припаивать провода.

Полная замена серии "2051" — серия "4051", например, AT89C4051-24PC, при этом никаких изменений в кодах прошивки и электрической схеме делать не надо. Более того, можно воспользоваться все тем же фирменным программатором и той же программой, которой прошивается AT89C2051.

Конструктивно МЧ располагается на верхней стороне процессорной платы под металлической крышкой.

полнительное механическое крепление осуществляется отрезком луженого монтажного провода длиной 1 см, которым вывод 10 микросхемы соединяется с рядом расположенной поверхностью общего провода. Металлическое основание крышки в месте касания с выводами микросхемы тщательно изолируется скотчем или изолентой.

Монтаж остальных цепей ведется тонким изолированным проводом диаметром 0,1...0,2 мм, прямо на выводы микросхем, чип-резисторов, чип-конденсаторов, на близлежащие контактные площадки круглой формы. Как показывает практика, длина соединительных проводов и их взаимное положение не критичны, лишь бы провода не выходили за пределы металлической крышки.

Порядок эксплуатации

МЧ настройки не требует. В качестве руководства по эксплуатации пользователю можно подарить рекламную памятку следующего содержания.

"ONEchip3" — это микроконтроллер (чип), который устанавливается в игровую приставку "PSone" и позволяет запускать на ней фирменные, нефирменные, русифициро-

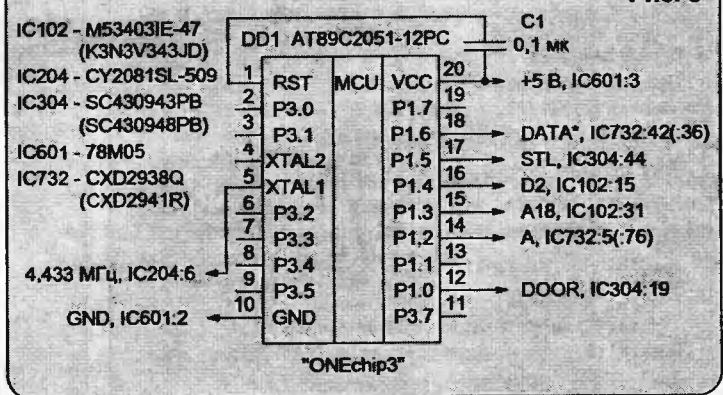
ванные и защищенные лицензионные игровые диски. "ONEchip3" выполнен по "Stealth"-технологии (чип-невидимка) и относится к третьему поколению мод-чипов. В частности, он позволяет обходить первичную региональную защиту boot-загрузки обычных "PlayStation", компенсировать вторичную защиту ПЗУ "PSone", а также запускать новейшие программно защищенные диски, имеющие проверку наличия мод-чипа в игровой приставке (японские и американские фирменные антимод-игры).

"ONEchip3" имеет режим полного отключения мод-чипа. Для этого крышка доступа к лазерному диску должна быть открыта в течение первых 15 или более секунд после нажатия кнопки питания. В дальнейшем крышка доступа может открываться и закрываться на любое по длительности время. Данный режим полезен, если используются внешние устройства типа "Action Replay/PS Hacker" (хакерские приборы для доступа к ресурсам "PlayStation"). Кроме того, отключение чипа позволяет проверить работоспособность "PSone" при поисках неисправности. Этот режим запрещает несанкционированный доступ к приставке посторонних лиц (детей), которые не знают о существовании 15-секундного ограничения во времени.

Литература

- 1. Рюмик С. Чип-"невидимка" в "PlayStation". — Радиомир. Ваш компьютер, 2003, №№6, 7.
- 2. Рюмик С. Модельные эксперименты в "PSone". — Радиомир. Ваш компьютер, 2003, №8, С.39.
- 3. Рюмик С. Замена микросхем в фирменном программаторе. — Радиомир. Ваш компьютер, 2003, №5, С.32.

Рис. 3





Программное управление принтерами фирмы “Hewlett-Packard”

А.БУТОВ,
с.Курба, Ярославской обл.

Фирма “Hewlett-Packard” широко использует современные технологии и на сегодняшний день является одним из ведущих производителей лазерных и струйных принтеров, предлагая как компактные модели для дома, так и многокилограммовые монстры для офиса и издательского дела. Но многие ли из нас интересовались тем, каким образом принтер входит или не входит в программное подчинение компьютеру. Скажем, почему вдруг обрезается текст, когда на бумаге остаются достаточно большие поля, и все настройки, вроде бы, сделаны правильно? Или почему принтер с завидным упорством печатает не так, как бы хотелось? Или вы хотите получить на бумаге нечто большее, чем вам может предложить стандартное программное обеспечение? Чтобы решить эти и множество других проблем или получить от принтера нечто особенное, желательно знать для начала, хотя бы в общих чертах, основы языка управления принтерами, который был разработан компанией “HP”.

Языки команд управления: Enhanced PCL-3 — для струйных принтеров и PCL-5 для лазерных, хотя и имеют множество отличий, но все же у них есть и много общего. Прежде всего, структура и синтаксис команд одинаковы. В ряде случаев абсолютно одинаково и действие команд.

Начнем с того, что все команды, за исключением самых простых, состоящих из одного символа, начинаются с префикса “Esc” (код — 1B, 27), который чаще всего является стандартом и для принтеров других фирм. После префикса идет последовательность знаков, которая обычно начинается с одного из трех символов — (, &, *. Далее может стоять строчная алфавитная литера, после которой следует положительное (без плюса) или отрицательное (с минусом) число, которое и определяет один конкретный параметр команды. После символического числового выражения команда чаще всего заканчивается заглавным алфавитным знаком, который и завершает синтаксис команды. Многие команды можно объединять, если заканчивать их строчной буквой, но это мы рассматривать не будем, предоставив это “поле” читателям для экспериментов.

Начнем с самых простых команд.

Команда “E” — на принтер надо послать последовательность 1B, 45. Этим мы сбросим все установки и вернем принтер к тому режиму печати, который он предлагает по умолчанию. “Злодейски” убрав эту команду из начала файла, который сформировал драйвер принтера, можно сохранить сделанные ранее “вручную” установки, что иногда может оказаться полезным.

Команда “=” (код — 1B, 3D) — пропуск строки. Зачем она будет нужна, сказать трудно, кому-то, может быть, и понадобится.

А теперь рассмотрим команды, которые содержат числовые параметры. В зависимости от области действия конкретной команды, числовой параметр может носить условный, относительный и абсолютный характер. Что-то выражается в дюймах, что-то в пикселах, что-то в знаках или в чем-то другом. Здесь и далее префикс “Esc” будет указываться только для шестнадцатеричного варианта представления синтаксиса команды.

Фиксированный шаг символов можно включить командой “(s0P” (код — 1B, 28, 73, 30, 50). Пропорциональный шаг — “(s1P” (код — 1B, 28, 73, 31, 50).

Толщину линии символа можно менять командой “(s#B” (код — 1B, 28, 73, #, 42). Здесь и далее, символ “#” будет обозначать какое-либо число. Применительно к последней команде, подставляя вместо “#” цифры, например, от “7” до “7”, можно печатать символами разной толщины. Да, вот еще о чем забыл упомянуть — многие команды влияют друг на друга. Например, если вы будете печатать пропорциональным шрифтом, то дав команду принтеру печатать “пожирнее”, получите меньшее число символов в строке.

Команда “&k#S” (код — 1B, 26, 6B, #, 53) задает ширину символа. Например, “0” — стандартная. Подставляя другие, числа можете печатать знаками разной ширины (“&k2S” — сжатые).

Команду “(s#V” (код — 1B, 28, 73, #, #, 56) можно рассматривать как “высота символа”. Дополнительных разъяснений, думаю, не требуется. Подставляйте любые положительные числа.

Курсивом можно печатать, если “скинуть” принтеру команду “(s1S” (код — 1B, 28, 73, 31, 53). Подставляя вместо

“1” числа от 0 до 160, вы обнаружите, что предлагает вам ПЗУ принтера.

Шрифты можно переключать командами: “(#D” (код — 1B, 28, #, #, 44); “(#E” (код — 1B, 28, #, #, 45); “(#F” (код — 1B, 28, #, #, 46); “(#G” (код — 1B, 28, #, #, 47); “(#H” (код — 1B, 28, #, #, 49). Закономерность, думаю, ясна. Меняйте числа и букву в конце команды. Вам повезло, если вы располагаете инструкцией к принтеру, где должны быть указаны имеющиеся встроенные шрифты. Иначе не избежать печати нескольких тысяч пробных строк. В уме прикидывайте, сколько бумаги и тонера будет израсходовано.

Командой “&1#D” (код — 1B, 26, 6C, #, #, #, 44) можно менять расстояние между строками: число больше — расстояние меньше.

Команда “&k#H” (код — 1B, 26, 6B, #, #, #, 48) задает ширину знака, т.е. расстояние между символами по оси X.

Следующей командой будет “(p#X” (код — 1B, 2A, 70, #, #, #, 58) — позиция печати по оси X. Манипулируя последними тремя командами, работая с принтером в текстовом режиме и используя многократные наложения, можно получить на бумаге шрифт, которого нет ни в ПЗУ принтера, ни на компьютере.

Команды “&1#E” (код — 1B, 26, 6C, #, #, #, 45); “&a#L” (код — 1B, 26, 61, #, #, #, 4C); “&a#M” (код — 1B, 26, 61, #, #, #, 4D) задают соответственно верхнее, левое и правое поля листа бумаги.

Количество текстовых строк на лист определяет команда “&1#F” (код — 1B, 26, 6C, #, #, #, 46). Примерно то же означает и команда “&1#P” (код — 1B, 26, 6C, #, #, #, 50).

Ориентация печати задается командой “&1#O” (код — 1B, 26, 6C, #, #, 4F). Подставляя цифры от 0 до 3, можно выбрать “книжную” или “альбомную” ориентацию.

Разные виды подчеркивания включаются командой “&d#D” (1B, 26, 64, #, #, 44). Подставляя вместо “#” любое однозначное положительное или отрицательное число, можно включить разные режимы подчеркивания. Выключить подчеркивание можно командой “&d@” (1B, 26, 64, 40). На разных моделях принтеров эта команда работает по-разному.

Командами “(s2Q” и “(s0Q” можно, работая в текстовом режиме, в любой момент включить режим качественной

или черновой печати. Меняя числовое значение, можно уменьшить или увеличить расход тонера, если соответствующий режим поддерживается встроенным ПО вашего принтера.

Теперь рассмотрим несколько команд для работы в графическом режиме.

Команда **"b#W"** задает, сколько следующих за этой командой байтов будут рассмотрены принтером как данные для графики. Если данных для печати будет больше, чем ширина бумаги, то "лишние" данные теряются. Командой **"b0M"** включается растровый режим, где каждый включенный в байте бит означает точку. Меняя "ноль" на другие числа, можно включить различные режимы векторной графики.

Командами **"&11L"** (1B, 26, 6C, 31, 4C) и **"&10L"** (1B, 26, 6C, 30, 4C) можно выбрать режим работы с перфорацией или без.

Существует еще большая группа команд по работе с бумагой разных форматов. Например, **"&126A"** (1B, 26, 6C, 32, 36, 41) — для формата A4, **"&127A"** — A3, **"&14A"** — A5, **"1100A"** — B5, **"&191A"** — C5.

Количество печатаемых копий без повторной загрузки данных в буфер приема можно определить командой **"&1#X"** (1B, 26, 6C, ##, 58), где **"##"** — число от 1 до 99. К сожалению, такая возможность есть далеко не на всех принтерах фирмы "Hewlett-Packard".

Командой **"&k0G"** (1B, 26, 6B, 30, 47) включается стандартное действие кодов команд: 13 — возврат каретки, 10 — перевод строки, 12 — прогон страницы. Подставляя вместо "нуля" цифры 1, 2, 3, такой расклад можно изменить.

Разумеется, это всего лишь ничтожная часть команд, которые "перевари-

вают" лазерные и струйные принтеры — HP LASERJET и HP Desk Jet. Впрочем возможно, что для тех, кто ранее не знал, "от какой печки плясать", этого будет достаточно, чтобы начать экспериментировать, не сильно расточая расходные материалы. Кто-то таким образом, может быть, найдет какие-либо скрытые резервы и возможности своего принтера или "недокументированные" команды. Они есть! Например, однажды я "кинул" на принтер строку случайных байтов, после чего он с жадностью стал хватать из лотка бумагу, демонстрируя все, на что способен. Мне стало жаль бумаги, и я остановил печать. Вот только повторно активизировать этот режим уже не удалось, так как те числа я не запомнил. Есть и другие нюансы, о которых можно было бы вести нескончаемый разговор.

Назад в будущее

Случается, что для некоторых целей (для программирования или просто из ностальгии) необходимо запустить на компьютере старые версии Windows. Да и повысить производительность своего не очень быстрого компьютера с помощью старых программ тоже хочется. Я хочу рассказать об установке Windows 95 на компьютеры, "вооруженные" процессором AMD с тактовой частотой выше 350 МГц, а также о работе Windows 3.x на компьютерах с файловой системой FAT 32.

Windows 95

Как известно, Windows 95 устанавливается на компьютеры с процессором, тактовая частота которого не превышает 350 МГц. Хотя Microsoft поддержку этой версии Windows прекратила несколько лет назад, многие пользователи продолжают работать с ней. И вот однажды я задался риторическим вопросом — как установить Windows 95 на компьютер с процессором AMD 450? Ведь его тактовая частота значительно выше, чем максимально поддерживаемая системой. Откровенно говоря, мною двигало желание поднять производительность своего ПК, установив относительно старое программное обеспечение. И это мне удалось, благодаря патчу, который можно

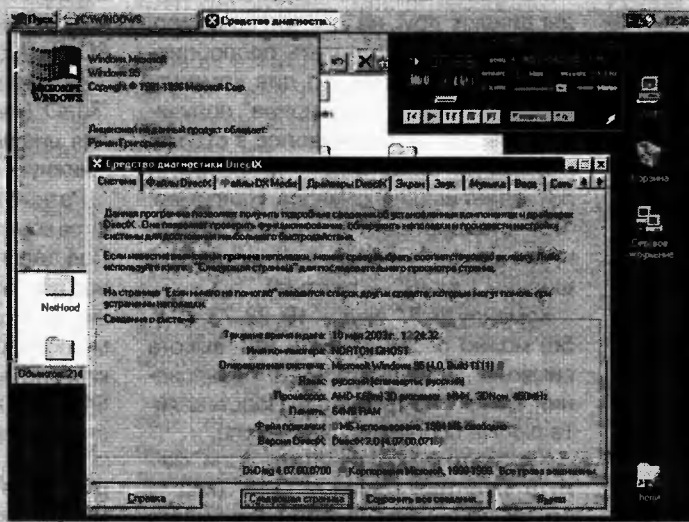
скачать вот по этому адресу: <http://www.windrivers.com/company/amd/k6-2up.zip>. Сразу оговорюсь, что этот патч работает только на компьютерах с CPU AMD K6-2. Всем же владельцам "каменей" от Intel и Cyrix предлагаю поискать свою версию.

Перед тем как начать описание установки, всегда следует рассказать о "подводных камнях". Первый такой "валун", на который я напоролся в процессе работы, заключался в невозможности обновления Windows 95 из среды MS DOS. Проблема состояла в том, что через некоторое время после запуска установки возникала ошибка, что приводило к невозможности дальнейшей установки системы. На время установки Windows 95 мне пришлось изменить в BIOS моей материнской платы TX 100 частоту процессора со

значения 450 МГц на 350 МГц. Это дало возможность установить систему и запустить патч уже из среды Windows. Затем осталось только зайти в BIOS и выставить прежние настройки. Все — проблема решена, и Windows 95 может спокойно "летать" на вашем компьютере (см. скриншот). Единственное, о чем осталось предупредить, так это о работе драйверов. Устанавливайте версии, адаптированные для Windows 95. Это устранил проблемы "зависания" и перезагрузки системы.

Windows 3.x

Старые добрые "окошки" — еще не ОС, но уже оболочка. То, с чего началось победоносное шествие Windows. Ну да ладно, сразу перейдем к делу. Проблемой Windows 3.x для современных компьютеров является то, что она не может установиться на файловую систему FAT 32. При такой попытке выдается сообщение о невозможности запуска программы и предложение нажать три заветные клавиши, чтобы перезагрузить компьютер. Для решения этой проблемы существует одна маленькая программа. Скачать эту утилиту можно с моего сайта — www.fdd5-25.narod.ru/drivers/Osr2kit.exe. Запускаете ее из режима MS DOS и перезагружаете компьютер. После этого смело устанавливайте любую версию Windows 3.x.



Р.КАРПАЧ,

г.Минск.

www.chizhovka.narod.ru



А.ВЕНДИЛОВСКИЙ,
г.Минск.

Enter the Matrix

Струящиеся зеленые буквы превращались в столбики, мерно стекая из бесконечности в бесконечность. Дедушка Геймер с трудом раскрыл глаза, осматривая спартанскую обстановку корабля-разведчика. Вот уже какое столетие шла война с разумными компьютерами, а конец всего этого безумия еще был слишком далеко.

— Внимание, производится установка игры "Enter the Matrix", свободного пространства на винчестере требуется чуть более трех гигабайт, пожалуйста, ожидайте...

Голос Малдера, сидящего за пультом, слегка действовал на нервы после экстренного вывода в Реальность, но и к этому можно было приспособиться. Дедушка Геймер постарался расслабиться в кресле и приготовился узнать как можно больше об игре перед "погружением", пока Малдер взламывал основную базу данных Зенона.

— Как тебе уже известно, несколько столетий назад произошла страшная война между людьми и созданными ими роботами с искусственным интеллектом. Эта позорная глава в истории существования человечества едва не стала для него последней главой. Я тут покопался в некоторых файлах в архивах Зенона, на которых стоял гриф "Совершенно Секретно", о начале данного конфликта, и теперь могу с уверенностью утверждать, что люди сполна получили за свою глупость и нетерпимость. Нельзя создавать разум, наделять его чувствами, а после обращаться с ним как с машиной, со страхом взирая на создание собственных рук. Машины хотели лишь уважения и мира, хотели спокойно жить своей жизнью, соседствуя со своими создателями, не подвергаясь унижениям. Это люди первыми обрушили на них свой удар и... проиграли. Ответный удар был страшен и беспощаден — армии роботов стирали в пыль земные мегаполисы, и ядерное пламя пожирало землю. И тогда люди закрыли мглой небеса, чтобы лишить врага света и энергии. Последняя схватка напоминала апокалипсис, и люди вновь проиграли, а перед победителями встал вопрос — как выжить? И проигравшие стали послушными марионетками, источниками

энергии, погруженные в бесконечный сон иллюзии жизни внутри искусственной программы — Матрицы, симулятора мира — пока их тела покоились в капсулах под покровом темных небес. Теперь люди были не более чем батарейками в мире машин...

— Надо сказать им спасибо, что они вообще создали для нас хотя бы иллюзию жизни, — Дедушка Геймер следил за бегунком инсталлятора, отсчитывающего процент выполнения задачи, — ведь могли вообще просто погрузить всех нас в беспробудный сон и не мучиться проблемами.

— ... оставшиеся в живых основали город под землей и назвали его Зеноном, откуда начали борьбу с диктатом машин, и наш корабль — один из разведчиков сил сопротивления. События игры происходят в то же время, что и действие во второй части фильма "Матрица". Нам придется принять участие в событиях, о которых в самом фильме говорится лишь мельком, но персонажи будут те же, и события внутри игры напрямую будут перекликаться с общим сюжетом блокбастеров. Для полного "погружения", братьями Вайчковски было снято для игры примерно на час видеороликов, которые по качеству и постановке ничем не отличаются от оригинального фильма. Впрочем, все это вскоре ты увидишь сам. Установка завершена, выполнение настройки параметров программы.

— Каковы системные требования?

— Для минимума хватит и 600-го Pentium 2 с 128 Мб памяти и любым GeForce на борту. Для максимального качества картинки — Pentium 1400 с 512 Мб памяти и GeForce 4, как минимум. Игра сама оптимизирует собственные настройки. Нам остается только наслаждаться игрой. Я останусь здесь, чтобы вытащить тебя, если станет очень жарко. Добро пожаловать в Матрицу!

Дедушка Геймер почувствовал, как нейрочувствительный датчик коснулся контактов на затылке, резкий удар, сознание на мгновение погрузилось в мрак, а потом ярчайший белый свет затопил все вокруг. Из ниоткуда в этот свет стали выпрыгивать стены зданий и другие детали интерьера. Через пару мгновений он уже стоял на пустынной улице безымянного западного мегаполи-

са. Первый взгляд, первые впечатления, первый крик ужаса.

Дедушка Геймер находился в состоянии некоторого шока, оглядывая собственную модель, в которой воротник рубашки безболезненно проникал вглубь шеи, а другие персонажи запросто умудрялись наполовину провалиться то в стену, то в дверь, перед самым ее открыванием. Детализация мира, видимо, находилась на уровне "очень бедно и тоскливо". И факт, что понятие "круг" здесь отсутствовало напрочь. Проводив взглядом автомобиль, который ехал на квадратных колесах, Дедушка Геймер тихо выругался, отчего полигоны на ближайших объектах полегли, как озимые. В этот момент он вновь услышал в наушнике голос Малдера, поминавшего всех святых и всю родню разработчиков.

— Дед, время у них было, видите ли, ограничено, и игра обязательно должна была выйти именно в день премьеры фильма. На лицензию, видеоролики и постановки боев угрохали такую кругленькую сумму, что нанять еще программистов возможности уже не было.

— Но я же видел трейлер из игры, там все было очень симпатично и проработано!

— Они выложили скриншоты и видеоролик от версии для PS2, а выход версии для PC стал планироваться только тогда, когда они стали понимать, что на одних приставках затраты не окупятся. Графикой пришлось пожертвовать... Да, тут еще один баг с автоопределением настройки игры — в этот момент в наушнике раздался звук, который может издавать только системный блок, когда по нему со всей силы бьют ногой — она старательно ставит параметры на самый минимум. Но если руки не "кривые", то можно покопаться в файлах конфигурации и привести картинку к более или менее человеческому виду. Кстати, первый патч к игре был выложен уже на следующий день после начала продаж.

В этот момент из-за угла высочили двое полицейских, чьи лица напукали бы любого — скорее, они напоминали двух бомжей после недельного запоя. Грохот выстрелов заполнил улицу, правда, без особого эффекта. Хотя игра и была от третьего лица, но стандартный прицел не появился, герой сам стрелял, куда считал нужным. Функция прицела лишь определяла направление стрельбы. С одной стороны, это очень напоми-



нало реальность — где вы видели снайперскую стрельбу из пистолетов на бегу, кроме как в голливудских боевиках? А с другой — очень осложняло прохождение игры при большом скоплении противника.

— А сейчас я раскрою тебе главную «фишку» — авторы игры не старались сделать экшен, основное направление — это файтинг. Были приглашены лучшие голливудские постановщики боев, гигабайты оцифрованных приемов, ударов и падений. Вперед!

Дедушка Геймер выскочил из-за укрытия, прямо навстречу врагу. Его персонаж просто с сумасшедшей скоростью стал наносить удары направо и налево. Подскочив к первому полицейскому, он в прыжке нанес прямой удар ногой в голову. Недавний охотник был просто подброшен в воздух, и грузно рухнул поверх ящиков. Подняться ему не дал завершающий удар в корпус. Дедушка Геймер не мог не заметить, что у него имеется ОЧЕНЬ большое количество приемов, иногда он даже не успевал запомнить все комбинации подсечек, блоков и ударов, как прямых, так и с разворота, руками и ногами. Это был файтинг с самой большой буквы.

... Стекланный потолок разлетелся стекланными брызгами, черные канаты опустились рядом с Дедушкой, черные фигуры спецназовцев скользнули по ним. Он на мгновение застыл, разведя руки, вслушиваясь в динамичную музыку, чувствуя, что необъяснимый драйв проникает в его тело, заставляя забыть обо всем, кроме ощущения бурлящего адреналина в крови. Молниеносный бросок, перехваченная рука, и вот уже он дер-

жит оружие врага. Палец рефлекторно жмет спусковой крючок, и противник, прежде чем безвольной куклой рухнуть на пол, конвульсивно дергается, когда пуля прошивает его насквозь. Пригнуться, пропустить летящие пули, подсечка, проводить на мгновение взглядом вылетевшее из рук спесназовца оружие, и кулак со всей силы бьет в грудь, чтобы почувствовать, как под бронежилетом ломаются ребра врага, услышать звук, с которым вырывается из его горла последний вздох.

— Bullet Time!

Время застыло как патока, пули медленно прокладывали свой путь к живой плоти. Сейчас от них можно было даже увернуться.

— Это не просто замедление времени, в этом режиме для игрока открываются новые возможности, и становятся доступны новые приемы, комбинации, которые в обычном режиме игрок просто не успеет набрать.

Дедушка Геймер медленно подбежал к стене и... побегал по ней прямо к противнику, мимо пролетающих рядом пуль. Оттолкнувшись посильнее, он ударил ногой прямо в шлем одного из бойцов. Тот плавно оторвался от земли и начал медленно скользить в сторону окна, выпустив из рук уже не нужный ему автомат. Завершив сальто, Дедушка Геймер оказался прямо за спиной второго бойца, когда время вернуло себе свои права. Звук разбившегося стекла совпал со звуком ломающейся шеи. Теперь он вновь был в комнате один.

Длилось это, правда, очень недолго, несколько полицейских выскочили из открытой двери, внезапно их

фигуры заструились, затрепетали, переливаясь, и вместо двух полицейских уже стояло два очень мрачных типа в черных костюмах и темных очках, словно сошедшие с обложек глянцевого журналов. Геймеру совсем не нужно было объяснять, кто такие Агенты. А так как он совсем не чувствовал в себе способностей Нео, то благоразумно решил не испытывать судьбу, и, выбив собой последнее оконное стекло, бросился вон, приземлившись на крыше здания. Под грохот выстрелов и свист пролетающих пуль он рванул вперед, перепрыгивая с крыши на крышу.

— Я вижу, ты уже вполне освоился — голос Малдера неожиданно раздался над ухом — Кстати, тебе сейчас налево, не пытайся сражаться с Агентами, это абсолютно бесполезно. Я открою тебе проход через телефонную будку на улице и... Зачем нужно было скидывать полицейского с крыши? Разве он виноват... Куда мне идти? Хам! Я только хотел сказать, что тебе еще предстоит поводить по городу автомобиль, естественно, с добротной стрельбой, и что тебе доверят управлять самым скоростным кораблем-разведчиком, посмотрим, как ты сможешь потягаться с Охотниками. Не отвлекайся, тебе нужно спуститься вниз раньше, чем этот хлыщ с пистолетом успеет мне помешать!

Но Дедушке Геймеру было не до комментариев, впереди его ожидало минимум двенадцать часов игрового времени, и терять его ему совсем не хотелось...

Диск предоставлен интернет-магазином

"MediaCraft" www.mediacraft.shop.by

Коды, читы...

Enter The Matrix

Эти читы вводятся в секции кодов для хакинга:

- 7F4DF451 — получить бесконечное здоровье;
- 1DDF2556 — получить бесконечные боеприпасы;
- 69E5D9E4 — бесконечный фокус;
- FFF0020A — быстрое восстановление фокуса;
- 0034AFFF — огневая мощь;
- FF00001A — турбо-режим;
- FFFFFFFF1 — получить невидимость;
- 4516DF45 — враги не слышат вас;
- 13D2C77F — бонусный уровень;
- D5C55D1E — бой в режиме мультиплеера;
- BB013FFF — малая гравитация;

- 312MF451 — вождение такси;
- 7867F443 — быстрое лого.

Haegar

Коды для уровней:

- уровень 2 — FEAFGN;
- уровень 3 — IVSAMK;
- уровень 4 — ASGAPQ;
- уровень 5 — UWFXPZ;
- уровень 6 — FSXRIC;
- уровень 7 — DYAETG;
- уровень 8 — WFZILD.

Indiana Jones and the Emperor's Tomb

Для доступа к режиму кодов необходимо внести изменения в первоначальные настройки игры. Для этого нужно найти файл default.cfg в каталоге с игрой (GameData/indy/default.cfg) и открыть его в текстовом редакторе и изменить следующие строки:

```
title:0
freescenedata:1
vsyncinterval:2
milessound:1
nocontrol:1
cheats:1
resmgrload:res.dat
fullscreen:1
```

После сохранения файла и повторного запуска игры можно вводить коды:

- Unlimited Health — получить бесконечное здоровье;
- Unlimited Oxygen — получить бесконечную энергию;
- God Mode — режим Бога;
- Unlimited Ammo — получить бесконечные боеприпасы.

Soldiers of Anarchy

Для входа в режим ввода кодов, во время выполнения миссии нажмите Enter. В выпав-



шем диалоговом окне вводите следующие коды:

- endlessmunition — бесконечные патроны;
- immortalone — режим Бога;
- winmission — выиграть текущую миссию;
- quitter — проиграть миссию;
- hittingandhealing — нанести объекту повреждения. Введите код, затем одновременно нажмите клавишу Ctrl и кликните левой кнопкой мыши по нужному объекту.
- hittingandhealing — отремонтировать объект. Введите код, затем одновременно нажмите клавишу Shift и кликните левой кнопкой мыши по нужному объекту.
- speedhack — изменить скорость игры. Введите код и используйте клавиши T и Z, чтобы ускорить или замедлить игру.

Для всех кодов:

- чтобы выключить код, введите его еще раз.
- если код введен правильно, появится надпись "Code xxx activated".

Aliens vs. Predator 2: Primal Hunt

Во время игры нажмите Enter и вводите коды (в конце кода нужно обязательно добавив пробел). После ввода нажмите Enter еще раз:

- mpcanthurtme — режим Бога;
- mpschuckit — получить оружие и боезапас;
- mpsmithy — получить полную броню;
- mpkohler — получить полный боезапас;
- mpstockpile — полный боезапас;
- mpbeamme — начать уровень заново;
- mpsixthsense — режим прохождения сквозь стены;
- mpicu — вид от третьего лица (HUD будет не виден);
- mpfov — изменить значение параметра FOV (угол зрения);
- mplightadd — изменить значение освещенности;
- mptachometer — показать текущую скорость;
- mpsizeme — показать текущий размер;
- mpgps — показать текущую позицию.
- mpgrs — показать поворот;

Commandos: Behind the Enemy Lines

Во время игры набрать строку 1982GONZO. После этого станут доступны следующие коды:

- SHIFT+X — телепортация;
- CTRL+I — невидимость;
- CTRL+SHIFT+N — закончить миссию.

Пароли к миссиям:

- Mission 2 — 4JJXB;
- Mission 3 — ZDD1T;
- Mission 4 — RFF1J;
- Mission 5 — K4TCG;
- Mission 6 — MIR4M;
- Mission 7 — 7QVJV;
- Mission 8 — K99XC;
- Mission 9 — AAAX1;
- Mission 10 — JSGPW;
- Mission 11 — CMODD;
- Mission 12 — JGHD3;
- Mission 13 — PUUWW;
- Mission 14 — WT348;

- Mission 15 — 139P0;
- Mission 16 — L9IPV;
- Mission 17 — 5LMV;
- Mission 18 — YJOJG;
- Mission 19 — YFCWJ;
- Mission 20 — GDKWT.

FIFA World Cup 2002

С помощью любого текстового редактора открыть файл soccer.ini, находящийся в игровой директории, добавить любые строки из приведенного ниже списка, сохранить файл soccer.ini и запустить игру как обычно:

- CHEAT_UNLOCKED_TEAMS=1 — разблокировать все мировые команды;
- UNLOCK_TOURNAMENT=1 — разблокировать турниры;
- AGGRESSIVE_TACKLE_CHEAT=1 — разрешить делать подножки;
- CHEAT_EQUAL_TEAM_STATS=0 — установить одинаковую статистику команд;
- CHEAT_RANDOM_TEAMS=1 — выбор компьютером случайных команд;
- DEMO_MODE=1 — запустить игру в режиме Demo;
- WINDOWED=1 — запустить игру в окне.

Ghost Recon: Desert Siege

Во время игры, нажав Enter, вызвать консольный режим, ввести один из кодов и еще раз нажать Enter, чтобы выйти из консоли:

- cisco — завершить первое задание;
- refill — получить все предметы инвентаря;
- kit — сменить обмундирование;
- chickenrun — получить chicken grenades;
- rumbleoff — отключить трясающийся экран;
- rumbleon — включить трясающийся экран;
- quit — выйти из игры;
- run — более быстрое движение;
- superman — режим Бога;
- hidecorpse — скрыть имена убитых;
- shadow — получить невидимость;
- autolose — проиграть текущую миссию;
- mark — поставить метку для телепортации;
- extremepaintball — режим paintball;
- loc — получить отчет о текущем местоположении;
- names — показывать имена объектов;
- squirrelkite — получить squirrel launcher;
- god — совершить самоубийство;
- rock — захватить вражеские базы;
- teamsuperman — командный режим Бога.
- teamshadow — командная невидимость;
- teleport — телепортация;
- spawn — телепортация к установленной позиции;
- toggleshowactorstats — вкл./выкл. статистику персонажей;
- toggleai — вкл./выкл. AI;
- toggleshowframerate — вкл./выкл. отображение fps;
- toggleshowlevelstats — вкл./выкл. статистику уровня;
- toggleshowtotalstats — вкл./выкл. об-

щую статистику;

- toggleui — вкл./выкл. интерфейс;
- ammo — получить неограниченный боекомплект;
- unlockheros — разблокировать персонаж;
- autowin — выиграть текущую миссию.

Tropico 2: Pirate Cove

Во время игры нажмите комбинацию CTRL + SHIFT и, не отпуская, введите код. Коды начинают действовать после отпущения клавиш CTRL + SHIFT:

- winit — выиграть игру;
- loseit — проиграть игру;
- booty — получить 2000 единиц золота;
- timber — получить 100 единиц леса;
- goape — сумасшедший пират;
- gofort — попытка заключенных бежать;
- spaininv — вторжение испанцев;
- britinv — вторжение англичан;
- frenchinv — вторжение французов;
- freebuild — отключить правила строительства;
- normbuild — включить правила строительства.

Kings Quest 8: Mask of Eternity

Нажмите комбинацию SHIFT+CTRL+7, чтобы войти в режим чит-кодов. Чтобы вернуться к игре, еще раз нажмите SHIFT+CTRL+7. Все коды набираются строчными буквами:

- noConCollide — получить возможность проходить сквозь стены;
- conCollide — отказаться от возможности проходить сквозь стены;
- selfFirstPerson — вид от первого лица;
- setThirdPerson — вид от третьего лица;
- teleport — телепортация к выбранному миру;
- god — включить режим Бога;
- ungod — отключить режим Бога;
- give (название предмета) — получить предмет из списка.

Список предметов:

- item0 — Dagger;
- item1 — SmallAx;
- item2 — BroadSword;
- item3 — Mace;
- item4 — LakeSword;
- item5 — SkelKingSword;
- item6 — BattleAx;
- item7 — WarHammer;
- item8 — Pike;
- item9 — LongSword;
- item10 — FlameSword;
- item11 — TempleSword;
- item12 — SmCrossbow;
- item13 — AzrielHammer;
- item14 — SmCpndCBow;
- item15 — ShortBow;
- item16 — LongBow;
- item17 — CrossBow;
- item18 — CpndCrossbow;
- item19 — FlameBow;
- item20 — IceCrossbow.

Подготовил Е.Курилович.



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

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



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

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

Продам книги по ZX-Spectrum:

- А.Ларченко, Н.Родионов. ZX Spectrum для пользователей и программистов;
- Н.Родионов. Адаптация программ к системе TR-DOS;

- ZX-Ревю, 1991 (выпуски 1...12);
- Справочник "Системные программы для ZX Spectrum 128 К";

- Справочник по системным программам для ZX Spectrum;
- Описание дисковой системы TR-DOS для компьютеров ZX Spectrum.

А. Горючкин.
E-mail: anatoliygor@nm.ru

Куплю компьютерные комплектующие (б/у); аксессуаров для компьютеров (мышки, колонки и т.п.) в нерабочем состоянии.

640023, г.Курган-23, а/я 2055, Анатолий.
E-mail: anatoli25@rambler.ru

Предлагаю программное обеспечение (игры, системные программы, электроника, текстовые файлы) утилит на Ассемблере Z80) для компьютеров ZX-Spectrum 128/48 на кассетах или аудио компакт-дисках.

150533, Россия, Ярославская обл.,
Ярославский р-н, с.Курба,
ул.Юбилейная, 11 — 15, Буты А.Л.
Тел. (0852) 66-31-58.

Ищу схемы, описания сигналов IBM-совместимых ПК, ISA, PCI, AGP, Keyboard, Mouse-интерфейсы и др., документы по проектированию устройств под ПК.
E-mail: tonder@trkts.ru

Продаю компьютер 486 по частям: материнская плата, процессор Intel 486SX25, память SIMM 8x1 M6 30pin, мультискарт HDD/LPT/2xCOM, видеокarta SVGA 512 Кб. Вышлю почтой.

354024, г.Сочи, ул.Ручей де Симона, 119,
Пареев М.В.

Ищу компилятор Бейсика версии MC2b.v4. Приложить конверт с обратным адресом.
632383, Новосибирская обл.,
г.Куйбышев, 1 — 20 — 45. Третьяков А.А.

Ученица 7-го класса с благодарностью примет в дар любой б/у компьютер.
446110, Самарская обл., г.Чапаевск,
ул.Володарского, 5А — 85, Шаварина Н.

Ищу документацию (инструкции) на матричные принтеры "EPSON" модели FX-1050 и LX-100. Куплю, обменяю на техническую литературу, схемы, детали.
141065, Московская обл., г.Королев, Болшево-5,
а/я 1, Брускин В.Я.
E-mail: bruskin@inbox.ru

Куплю процессор Z80 SIO или его аналог U856.
452403, Башкортостан, г.Дадакеево,
ул.50 лет Башкирии, 86, Дроздов С.

Приму в дар компьютер, ноутбук или комплектующие или недорого куплю.
210008, Беларусь, г.Витебск, ул. 11-я Социалистическая, 8 — 1 — 52, А.Ситов.
Тел.33-77-87.

Продаю: C1-55, C1-112, ГЗ-36А, ЧЗ-32, M1109, M2051, Ц4313, Ц4353, P40114, КТ817Г/КТ819Г (б/у), программы для "ZX-Spectrum" и др.

Куплю документацию на ЧЗ-32, ГЗ-36А.
Возможен обмен на тюнер ("Пионер", "Техникс" и др).

Тел. (095) 944-53-13, с 19.00 до 21.00, Михаил.

Продаю:
- компьютер "Scorpion 256" (расширенная клавиатура для "Scorpion"; контроллер для подключения IBM-клавиатуры и мыши; 2 дисковод "Robotron");

- монитор "Электроника 32ВТЦ-202;
- два принтера (без головок) "Электроника CM 6312";

- 160 дискет 5,25S (системные программы, электронные журналы и газеты);
- черно-белый монитор "Электроника MC6105.01";

- два дисковода TEAC на запчасти;
- дисковод "Электроника 5305";
- дисковод "Epson SD-600".
165300, Архангельская обл., г.Котлас,
ул.Мартемьяновская, 44 — 7, Ятченко Н.Ф.

Уважаемые читатели!

Подписаться на наши журналы (индексы: 48996, 72370 (годовая) — "Радиомир", 48924, 71545 (годовая) — "Радиомир. КВ и УКВ", 48925, 45995 (годовая) — "Радиомир. Ваш компьютер") в 2004 г. можно по каталогу Агентства "Распечатать" или 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 через ООО "Корпоративная Почта" по телефонам: (095) 953-92-62, 953-92-02, 953-93-20.

Те, у кого возникли проблемы с подпиской, могут получить их из редакции. Там же можно заказать имеющиеся в наличии отдельные номера журналов за предыдущие годы.

Год	Можно заказать следующие номера журналов			Расценки на 1 экз. (с учетом пересылки)		
	Радиолюбитель	Радиолюбитель. Ваш компьютер	Радиолюбитель. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
1998	4, 5	1 — 5, 8, 9, 11, 12	1	20	700	5
1999	3, 9, 10, 11	1 — 6, 10, 11, 12	3, 5, 6	25	800	5
2000	2 — 6, 8 — 12	2 — 8, 10 — 12	4, 6, 7, 9, 11, 12	25	900	6
2001	2 — 6	1, 2, 4 — 6	2 — 6	30	1000	6
Год	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2001	7 — 12	7, 9 — 12	7 — 12	35	1300	7
2002	1 — 12	1 — 12	1 — 3, 5 — 12	40	1400	8
2003	1 — 6	1 — 6	1, 5, 8	45	1700	9
2003	7 — 12	7 — 12	7 — 12	45	1900	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 "РГД", УНН 190218688

р/с 3012000004882 в ф-ле №524 АСБ "Беларусбанк" в г.Минске код 121.
Адрес банка: 220028, г.Минск, ул.Физкультурная, 31.

На бланке почтового перевода необходимо очень четко написать свой почтовый индекс, полный адрес, фамилию, имя и отчество полностью. В графе "Для письма" необходимо точно перечислить, какие конкретно номера какого из журналов Вы заказываете. При оплате через Сбербанк эти сведения необходимо указать в графе "назначения платежа".

Вся информация по E-mail: rm-sales@radio-mir.com
или по тел. в г.Минске (017) 221-01-10, (017) 585-13-65.

Приобретение отдельных номеров журналов

В РОССИИ:

В ЗАО "Интерпресс — Экспресс": тел. в Москве (095) 208-85-50, 208-67-55,
e-mail: inter@alf.ru

В магазинах радиодеталей "ЧИП и ДИП":

- 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 "ТРЕНТЭКС": 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 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").

に美と字印び技す国出のシ品数最まゴ園ンは証メ密万

最まゴ園ンは証メ密

の文

思ザ粒し

の

の



文
精
な
フ
ト
社
明
を
に
美
と
字
印
び
技
す
国
出
の
シ
品
数
最
ま

ENTER THE MATRIX



文
精
な
フ
ト
社
明
を
に
美
と
字
印
び
技
す
国
出
の
シ
品
数



- 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