

radio-mir

3 • 2004

Ваш компьютер

 **Adobe Clipart**

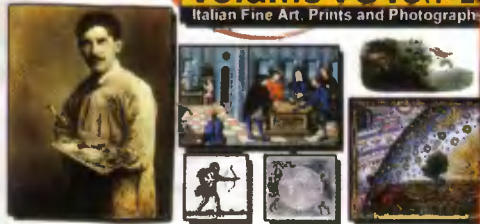
2CD в 1

Профессиональное Качество

PhotoDisc

Volume VS13(1+2)

Italian Fine Art, Prints and Photographs



PINNACLE STUDIO 8.8.17

РУССКАЯ +
АНГЛИЙСКАЯ ВЕРСИИ +
ДОКУМЕНТАЦИЯ



ЦИФРОВАЯ ФОТО-БИБЛИОТЕКА ДИЗАЙНЕР

ЖЕНЩИНЫ

Портреты женщин
Знаменитости
Экзотика

Adobe CREATIVE SUITE Premium Edition



Полное
официальное
издание

САМ СЕБЕ РЕЖИССЕР 2003

Видеоредактирование для дома



Adobe Premiere 6.5 (русская и английская версии)
D VideoMagic 1.0
RED 2.5.2
Apple MotionPerfect 4.02.0004
документация и обзоры



Guitar Music maker 2004 v1.0

МУЛЬТИМЕДИА

МОЩНАЯ МУЗЫКАЛЬНАЯ СТУДИЯ

- Guitar Music Maker 2004 v1.0
- Adobe Audition 1.0
- FL Studio Producer Edition 4.1.2
- ProPeeerhead Reason 2.5
- Документация

ALEX SOFT



Устройства
охраны
сигнализации



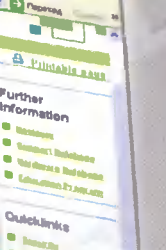
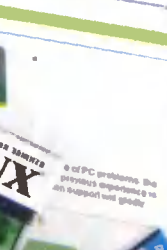
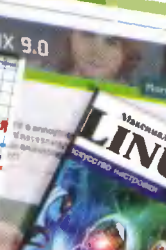
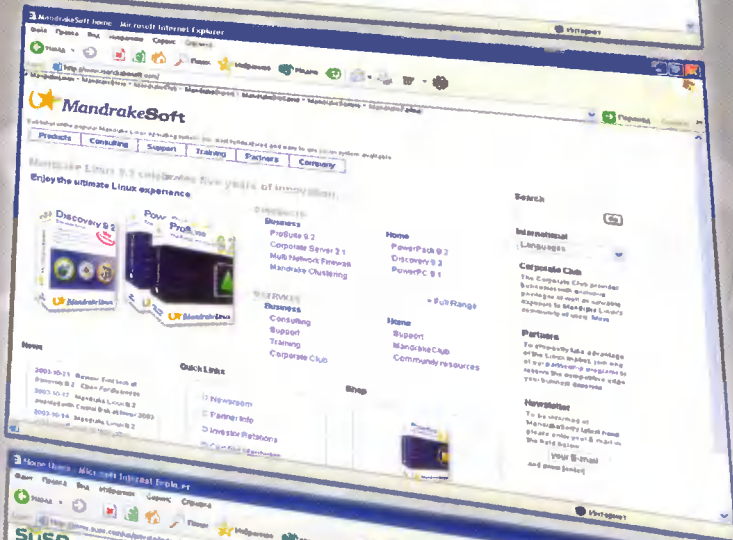
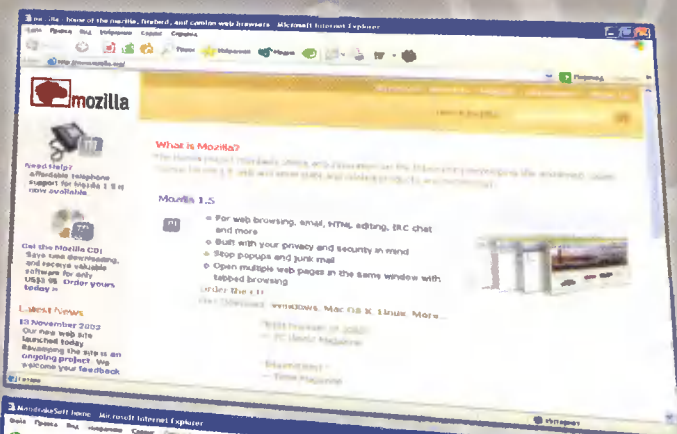
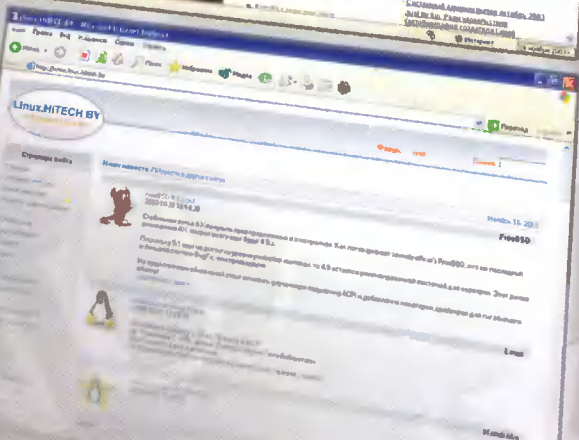
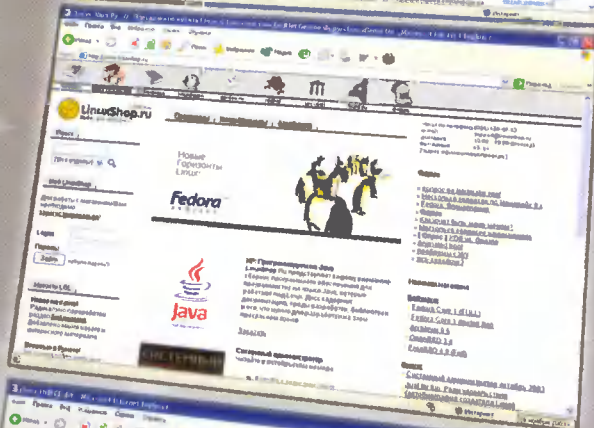
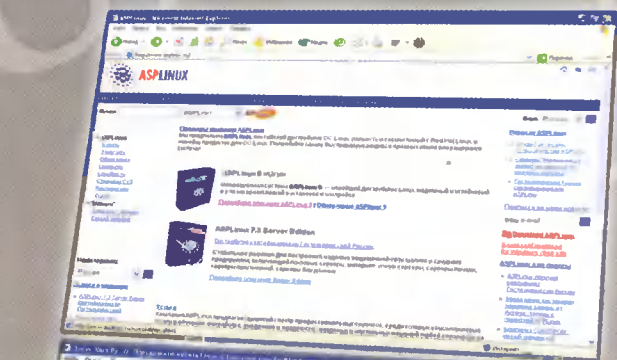
Adobe Premiere 6.5

Русская и английская версии



Линукс на домашнем компьютере

(См. статью Ю.Левина)



Be Linux...

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

Главный редактор
Ольга СТРИЖАНКОВА
Зам. гл. редактора
Иван БЕЛЬСКИЙ

Редколлегия:
Сергей ДРОЗДОВСКИЙ,
Алексей КОНДРАШОВ,
Анатолий КОРБИТ,
Владимир КУЦЕНКО,
Елена ЛЕВИТМАН,
Николай МЕДВЕДЬ,
Геннадий ПЕЧЕНЬ,
Геннадий ШУЛЬГИН

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

Компьютерная верстка —
Янина БЕЛЬСКАЯ
Техническая графика —
Наталья БЕЛЬСКАЯ,
Александр ГОРАЮТИН,
Мария КАЛАБУХОВА

Оформление обложки —
Наталья БЕЛЬСКАЯ,
Надежда БОГОМОЛОВА

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

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

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

Адрес банка: 125315, г.Москва,
Ленинградский пр., дом 76, корп. 4

Материалы для публикации принимаются
в рукописном, печатном и электронном
вариантах

Требования к графическим материалам
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 рекламодатель и авторы.
Мнение редакции не всегда совпадает
с мнениями авторов

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

Подписано к печати 27.01.2004 г.
Формат 60 x 84 1/8
Печать офсетная. 6 печ. л.
Цена свободная

Отпечатано в типографии
ЗАО "Красногорская типография".
Заказ № 240.

радиомир Ваш компьютер

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

№3/март/2004

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

МУЛЬТИМЕДИА

А.СНИТКО. DivX ;-) — история развития, применение и перспективы 2

НЕ ТОЛЬКО НОВИЧКУ

Ю.ЛЕВИН. Линукс на домашнем компьютере 5

Р.ГАЛЕЕВ /HI-TECH. Saga о кодировках 7

А.ГРИНЧУК. Анимация с Macromedia Flash MX.
Анимация движения 10

В.КУЦ. "Твикаем" окошки 12

И.РОЩИН. Носители информации: назад к бумаге? 15

С.РЮМИК. Интернет-калейдоскоп, freeware #4 18

КОММУНИКАЦИИ

Е.ЛЕБЕДЕВ. В помощь админу 20

ДАЙДЖЕСТ 22

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

А.КОРБИТ, А.ЩЕРБАКОВ. Программирование в среде Visual C++ 6.0.
Написание программы построения графиков 26

М.ДОЛИНСКИЙ. Решение задач на деревьях и с помощью деревьев 27

ДИАЛОГ ПРОГРАММИСТОВ

А.КАЛЮТОВ. Практическое введение
в технологии сжатия изображений 30

CyberManiac /HI-TECH. Теоретические основы крэкинга 33

И.ШИРКО. Microsoft Agent.
Часть I 35

Р.ФАСИХОВ. "Отучение" игр от компакт-диска 37

О.ВАЛЬПА. Защита информации 39

РЕЦЕПТЫ

А.ГОРЯЧКИН. Режим UDMA в Windows XP 41

Р.КАРПАЧ. Теория "downgrade" 42

МИР 8 БИТ

Е.ИЛАСОВ. Телетест на Sinclair — 2 45

ИГРОТЕКА

А.ВЕНДИЛОВСКИЙ. Need for Speed Underground 46

ДОСКА ОБЪЯВЛЕНИЙ

Куплю, продам, обменяю 48

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



DivX :-)

— история развития, применение и перспективы

А.СНИТКО,

г.Минск, БГУИР, гр.352001.

E-mail: zx4ever@torba.com

WWW: <http://www.zx4ever.narod.ru>

В предыдущей статье [1] был сделан обзор основных современных видеокодеков. Сейчас я расскажу вам об одном из них — DivX :-).

Для начала более подробно рассмотрим историю его появления.

В конце 90-х годов прошлого века для оттеснения с рынка множества конкурентов с их довольно прогрессивными технологиями сжатия потокового видео, например, Apple с QuickTime, Microsoft начала разработку собственного кодека сжатия видео в формате MPEG-4. Вскоре она выпустила на рынок бета-версию кодека — Microsoft MPEG-4 Video Codec V3. Сделав это, Microsoft совершила большую ошибку, так как кодек тут же взломали хакеры. Произошло это в сентябре 1999 г., звали хакеров MaxMorice и Gej. Основным стимулом для взлома было то, что на тот момент у кодека не было возможности кодирования видео в AVI-файл. Поддерживался только Microsoft'овский ASF. Но мало того, что этот формат был не очень удачным, так он был еще и закрытым. Например, в широко известной программе VirtualDub возможность работы с ASF была отключена по требованию самой Microsoft. Позже Microsoft все же одумалась и включила вывод в AVI-файл, но было уже поздно — все забыли о существовании оригинального кодека.

По заверениям самих хакеров, они улучшили кодек и убрали некоторые глюки. Хотя мне в это верится с трудом. Неужели хакеры такие умные, что смогли улучшить кодек, имея в распоряжении только скомпилированный бинарный код, а в крупных софтверных компаниях работают глупые программисты, которые не могут разобраться в созданном своими руками исходном коде? Просто Microsoft для увеличения надежности своего кодека временно отключила еще плохо опробованные и ненадежно работающие возможности (бета-версия все-таки). В число таких опций вошли ползунок управления сглаживанием при декодировании, детектор смены сцен, вывод в AVI и др. Хакеры же включили эти опции (и то не все сразу и не все сами) и предстали перед всем светом величайшими специалистами в области сжатия видео. Это сугубо мое личное мнение, и вы имеете право его оспорить (мои координаты — в начале статьи, пожалуйста, связывайтесь, если

не согласны). Многие в пользу хакеров приводят тот аргумент, что те выпустили как бы два различных кодека в одном — DivX :-). MPEG-4 Fast-Motion и DivX :-). MPEG-4 Low-Motion. Первый больше подходит для быстрых сцен, а второй хорош для медленных сцен с равномерными цветовыми заливками. Все утверждают, что второй кодек является почти оригинальной версией кодека от Microsoft, а первый — значительно улучшенная и доработанная версия оригинального кодека. Так вот, на самом деле, первый является всего лишь взломом кодека Microsoft MPEG-4 Video Codec V3 версии 4.1.00.3917, а второй — версии 4.1.00.3920. Т.е. "усовершенствованный" кодек — это не что иное как взлом более ранней оригинальной версии кодека. Хакеры сами ввели в файлы динамических библиотек обоих кодеков графу "Crack State" в структуре Version Info, которая характеризовала текущее "качество" взлома. В Low-Motion-кодеке эта графа имеет значение "Perfect (Best Quality Recommended !)", а в Fast-Motion — "Good (Bitrate Limitation ?)". Видно, что сам хакер признает "недоломанность" Fast-Motion-версии кодека, выражающуюся в невозможности установить битрейт выше некоторого предела. Надеюсь, что я убедил вас в том, что заслуга в создании этого прекрасного кодека принадлежит не каким-то хакерам, а самой Microsoft (или какой-нибудь выкупленной дочерней компании).

Теперь давайте последовательно рассмотрим историю дальнейшего развития кодека, по ходу останавливаясь на важных моментах.

Итак, после выпуска первой версии своего кодека хакеры выпустили еще две новые его версии, которые, правда, мало отличались от исходной. Все же упомянем о них.

3.1alpha — теперь кодек поставляется со взломанной версией аудиокодека Windows Media Audio (WMA).

3.11alpha — исправлен глюк с регулятором управления сглаживанием в DirectShow-декодере.

Последние версии именуются альфа-ми из-за, как выражается автор, "недоломанности" WMA-кодека, которая, по его мнению, проявилась в невозможности установить параметры кодирования аудиопотока выше, чем 64 kbps 44 kHz. Также WMA DirectShow-декодер так и

не был взломан. Все это говорит о том, что хакеры вообще не сделали в WMA-кодеке ничего, кроме изменения надписей на окне, так как в оригинальном аудиокодеке от Microsoft также не поддерживаются более высокие параметры кодирования при использовании кодека в видеофайлах. Но если вы вынуждены кодировать аудиодорожку в видеофайле с низким битрейтом, то WMA будет однозначно лучше, чем MP3. Ограничение битрейта до величины 64 kbps имеет под собой и кое-какие основания, так как применять WMA для кодирования на более высоких битрейтах нецелесообразно. Результаты многочисленных тестов различных аудиокодеков показывают, что WMA обеспечивает наибольший рост качества на битрейтах до 64 kbps. При дальнейшем увеличении битрейта качество практически не изменяется и находится на значительно более низком уровне, нежели MP3.

DivX :-). 3.11alpha — последняя версия взломанного кодека, которая стала стандартом де-факто, и используется как основа для дополнительных программных надстроек и по сей день. Авторы прекратили дальнейшую работу над кодом из-за открытия нового коммерческого проекта (или из-за того, что не могли больше ничего сделать самостоятельно).

С началом массового применения нового кодека начали возникать и новые проблемы. Основная из них — невозможность использования какого-либо одного кодека (FM или LM) для кодирования определенного класса фильмов, содержащих одновременно как быстрые, так и медленные сцены. Посмотрим, как ведет себя каждый из двух вариантов кодека.

Low-Motion-кодек хорошо подходит для медленных сцен с равномерными цветовыми заливками без быстрых движений объектов и резких сдвигов камеры. Заданный ему битрейт этот кодек будет поддерживать на должном уровне. Конечно, очень сложные сцены он закодирует с небольшим превышением битрейта, а простые — с уменьшением, но средний битрейт всего видеофайла почти в точности будет соответствовать заданному вами. Однако все впечатление от кодека будет испорчено динамичными сценами, которые заполнятся паразитными следами.

Совершенно по-другому дело обстоит с Fast-Motion-кодеком. Он почти не придерживается заданного вами битрейта и так и норовит закодировать видео с битрейтом килобит эдак на 900. Быстрые сцены, закодированные этим кодеком, практически не имеют паразитных следов. Зато равномерные сцены, особенно снятые крупным планом, превратятся во что-то большее похожее на цветную мозаику.

Самым простым, и в то же время самым неэффективным способом решения данной проблемы было кодирование фильма обоими вариантами кодека с последующим выбором того кодека, который дает более приемлемое качество. Естественно, что данный метод совершенно не мог удовлетворить запросы пользователей. Ему на смену пришла технология смешанного кодирования (MM4 — Mixed MPEG-4). Ее суть заключалась в кодировании отдельных частей видеопотока с различными настройками (начиная вариантом кодека и заканчивая битрейтом). Поначалу это осуществлялось вручную и было довольно трудоемко. Необходимо было каждый раз полностью кодировать видеофайл с разными настройками, а затем в какой-нибудь программе видеопереработки (чаще всего в VirtualDub) рассмотреть каждый вариант закодированного видео, выбрать фрагменты с качеством, лучшим у какого-либо из вариантов, "нарезать" фильм (причем только по ключевым кадрам), заново "склеить" его и наложить звуковую дорожку. В основном, смешанное кодирование применяют для комбинации LM- и FM-кодеков, но иногда разные участки видеофайла кодируют одним и тем же кодеком, но с разным битрейтом. Например, для получения очень высокого качества кодируемого видеофайла (с конечным размером в 2 и более CD), в местах с быстрыми движениями часто лучше применить LM с высоким битрейтом, нежели FM.

Но, как всегда, нашлись талантливые программисты, которым было лень производить весь этот процесс вручную, и они подарили миру программы для автоматизации смешанного кодирования. На данный момент наиболее известны две программы — Project DivX (Германия) и MakeFilm TNG (Китай). После их применения сразу же чувствуется значительное превосходство первой программы над второй, что и позволило ей дожить до наших дней. Ее основные достоинства: более удобный и наглядный интерфейс; возможность как ручного, так и автоматического кодирования; поддержка многоязычного интерфейса и др. Кроме того,

Project DivX, в отличие от MakeFilm, сохранил свою работоспособность при применении кодека с включенным деструктором смены сцен.

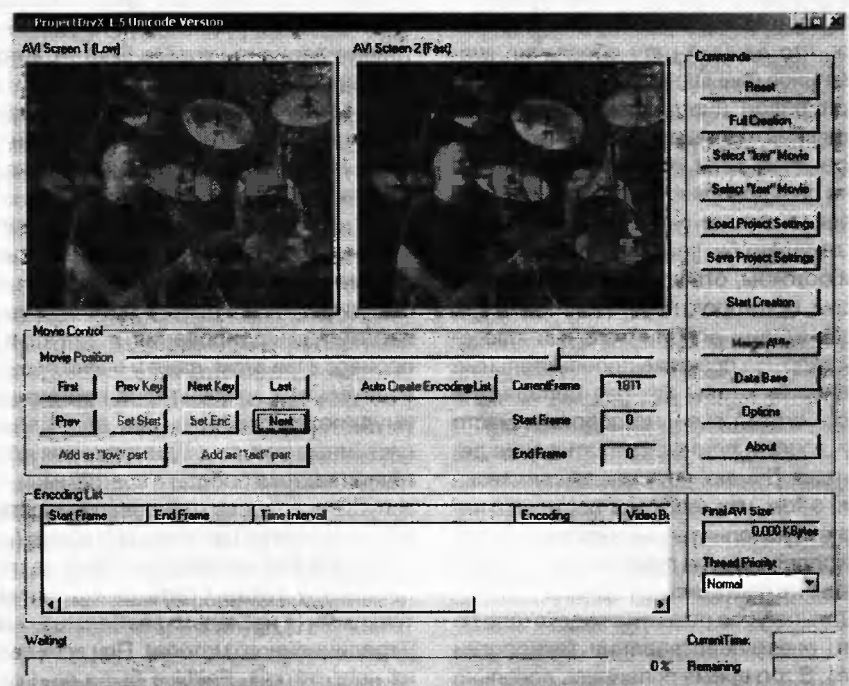
Программа формирует конечный видеофайл путем комбинации фрагментов двух видеофайлов сжатых соответственно LM- и FM-кодеками. Естественно, эти фрагменты "нарезаются" только по ключевым кадрам. При компоновке конечного файла Project DivX берет данные о последовательности применения того или иного кодека из специального листа кодирования, который можно создать вручную или отдать эту времяемкую задачу компьютеру.

Для ручного создания листа кодирования нужно предварительно закодировать исходное видео LM- и FM-кодеками. Не забывайте, что оба варианта закодированного видео обязательно должны иметь одинаковые разрешение и частоту кадров, а также одинаковую длительность промежутка между ключевыми кадрами, выставленную в настройках кодеков при кодировании. Дальнейший процесс создания листа выглядит довольно просто, но очень уж нудно и однообразно. Вам необходимо просматривать участки фильма между ключевыми кадрами и заменять LM на FM там, где это необходимо по соображениям качества.

При выборе автоматического варианта кодирования вам необходимо предоставить в распоряжение программы только оригинальный avi-файл сжимаемого вами видео. Автор программы рекомендует сжимать исходный файл Intel Indeo или MJPEG-кодеками, но, как известно, при этом на изображении воз-

никают многочисленные шумы (часто заметные только при тщательном рассмотрении), из-за которых DivX;-) изначально некорректно начинает кодировать видеопоток. Чтобы избежать этого, многие кодируют исходный файл с очень маленьким сжатием. Однако, при этом размер получаемого файла отличается от исходного всего на 6...8%, хотя затраты времени на эту операцию очень значительны. Поэтому я рекомендую вам использовать обычный несжатый avi-файл или, если производится риппинг с DVD, виртуальный файл (frame-server). Project DivX самостоятельно создаст LM- и FM-файлы, правда, с вашими настройками кодеков. Затем в дело вступает оригинальный алгоритм Oracle, который собирает сведения о видеопотоке в специальный Oracle-файл. Впоследствии можно преобразовать этот файл в обычный лист кодирования, установив некоторые важные параметры его создания, грамотная настройка которых в итоге обеспечит вам наилучший результат. Если же вы не можете разобраться с этими параметрами, или вам просто лень это делать, то можете поэкспериментировать с различными настройками над текущим видеофайлом — что-нибудь стоящее все равно должно получиться.

Теперь, когда у вас есть лист кодирования, сохраненный в INI-файле, LM- и FM-файлы, вы можете смело запустить последнюю стадию процесса — компоновку конечного файла из LM- и FM-файлов с отбором данных из листа кодирования. Когда вся работа будет сделана, можете насладиться ее результатами.



Появление технологии "смешанного кодирования" позволило в какой-то мере улучшить качество кодируемых видеофайлов, но оставалась еще одна нерешенная проблема — невозможность автоматической расстановки ключевых кадров на сменах сцен. Эта проблема не только вызывала сильную зашумленность изображений при смене сцен при отсутствии ключевого кадра в момент смены или сразу после нее, но и не давала возможности в полной мере проявить себя технологии MM4. К счастью, нашелся некий Dynamite, который обнаружил в кодеке от Microsoft отключенный детектор смены сцен. Включив его, он выпустил в свет так называемый DivX scene-detect patch. Правда, комплект этот очень трудно назвать патчем, так как он представлен двумя динамическими библиотеками (DivXsc32.dll и DivXsc32f.dll), которые служат для замены оригинальных библиотек кодека. После установки патча вы имеете возможность, как и прежде, самостоятельно выставлять в настройках кодека временной интервал между ключевыми кадрами. Кроме того, на сменах сцен кодек будет выставлять дополнительные ключевые кадры. При желании вы можете и вовсе отказаться от ключевых кадров в тех местах видеофайла, где нет смены сцен. Для этого вам необходимо выставить достаточно большой интервал между ключевыми кадрами (максимум 9999 с). Однако, при достаточно длинных сценах (особенно снятых дальним планом) это может вызвать и ухудшение качества изображения, а также создать неудобства при просмотре выборочных мест фильма — плеер очень долго ищет ключевой кадр для выбранного фрагмента. Стоит отметить, что данный патч обеспечил значительное повышение качества кодируемого видео, иногда даже и при незначительном уменьшении конечного размера. Алгоритм определения смены сцен был довольно простым и основывался, в первую очередь, на показателе текущего битрейта — при превышении потоком, отводимым на ключевые кадры, некоторого порогового значения, кодек ставил дополнительный ключевой кадр. На практике дополнительные ключевые кадры как раз попадали в места смены сцен, но довольно часто случались и ложные срабатывания детектора. Думаю, нет смысла рассказывать о том, что кодек мог поставить несколько дополнительных ключевых кадров подряд или на протяжении длинной сцены не поставить ни одного.

Дальнейшее развитие кодека шло по пути совершенствования технологии MM4. В это время появились довольно

известные программы, сделавшие много не только для DivX ;-), но и для всей отрасли видеосжатия.

VirtualDub MM4 — одна из переделок всем известного VirtualDub. Это родоначальник серии программ для смешанного кодирования "в один проход". В настройках программы задаются значения битрейта для обоих кодеков и битрейт, при котором происходит переход с одного кодека на другой. Далее программа за один проход кодирует весь видеофайл, переходя при необходимости с одного кодека на другой. Во время кодирования все переходы отображаются на графике. В программе появилась возможность использования детектора смены сцен по яркостной составляющей кадра. Правильность срабатывания этого детектора намного выше, чем у стандартного детектора из кодека. Единственной неприятной особенностью технологии "MM4 в один проход" является невозможность точно предугадать размер будущего закодированного видеофайла, что весьма неудобно при записи видеоинформации на постоянные носители.

FairUse — эта программа является представителем усложненного варианта технологии MM4. Здесь кодирование может происходить с использованием сразу нескольких вариантов кодека (до 10). FairUse изменяет текущий поток в зависимости от заданного качества и заданного размера получаемого файла. Поддержание качества на заданном уровне достигается выбором соответствующего варианта кодека путем сравнения кодируемого разными вариантами кодека фрагмента с оригинальным файлом. В программе имеются доработанные варианты кодека DivX ;-), которые, вроде бы, должны обеспечить некоторое улучшение качества кодирования при их грамотном применении в нужных участках видеофайла. Причем забота о выборе нужного варианта кодека целиком ложится на плечи программы, которая работает в полностью автоматизированном режиме. Вам необходимо только ввести необходимые параметры кодирования и запустить процесс. При этом, даже у очень производительных компьютеров программа умудряется отобрать все доступные системные ресурсы. Единственная пригодная тактика работы с программой — запустить процесс кодирования и оставить компьютер на некоторое время (на ночь), так как затраты времени на эту процедуру также стали очень значительными (в несколько раз больше, чем у традиционного метода). При всех своих недостатках, FairUse была еще од-

ним значительным шагом в области кодирования видео.

NanDub — еще одна модификация VirtualDub, являющаяся естественным продолжением VirtualDub MM4. Главными ее особенностями являются введение технологии SBC (Smart Bitrate Control) и двухпроходного процесса кодирования. SBC представляет собой систему со множеством различных настроек (многие выведены из внутренних возможностей самого кодека), которая осуществляет более технологичный и эффективный контроль за битрейтом и другими важными параметрами кодирования, чем сам кодек. При создании NanDub авторы очень серьезно изучили внутренности кодека и вывели на диалоговую панель множество его скрытых параметров. При этом большинство пользователей и по сей день не знает, как правильно настроить все эти параметры, чтобы от применения программы была ощутимая польза. Особенно это актуально для тех, кто не очень глубоко знаком с английским языком, так как примененная авторами программы терминология очень специфична. Важный плюс NanDub — появление двухпроходного кодирования (правда, оно не обязательно для использования). За первый проход программа собирает файл статистики, а за второй — формирует собственно видеофайл, используя данные из предоставленного ей файла статистики. Появление двухпроходного кодирования решило многие проблемы, оставшиеся от однопроходного варианта MM4, но при этом почти в два раза увеличилось время, затрачиваемое на весь процесс. Есть у программы и другие полезные возможности, например, двухпоточный звук во многих форматах. NanDub сейчас практически не используется, но он послужил основой для следующей очень популярной сейчас программы.

NanDub2 — довольно сильно измененный и развитый вариант NanDub. Можно сказать, что это уже почти другая программа. Выставлены оптимальным образом или убраны многочисленные ненужные настройки, улучшен использованный алгоритм контроля битрейта, введено использование психовизуальной модели сжатия, немного изменен сам кодек и др. Надо отдать должное создателю NanDub2 (Neiromaster'y), он проделал огромный труд, так как результаты работы с программой действительно впечатляют. Подробно программу описывать не буду, потому что для этого потребовалась бы статья размером с эту, а в сжатом объеме можно только наделать кучу ошибок.



Я описал практически все наиболее важные аспекты развития кодека DivX ;-), которые имеются на данный момент. Давайте рассмотрим, какие перспективы имеет этот кодек на сегодняшний день.

С одной стороны, перспектив развития он уже практически не имеет. Каких-либо значительных связанных с ним релизов уже не произойдет. На протяжении 2004 г. могут появиться лишь мелкие усовершенствования уже существующих сейчас технологий, но не более того. Далее развитие кодека остановится полностью. Главная причина кроется в том, что все более или менее серьезные разработки в области видеосжатия сейчас гораздо выгоднее и целесообразнее проводить на основе OpenDivX. Так и происходит. Уже существует аппаратная поддержка OpenDivX-кодеком со стороны MPEG4-видеопроекторов. Возможность же проигрывания дисков, закодированных в обычном DivX ;-), у них либо отсутствует вовсе, либо идет как некое дополнение, которое часто глючит. К тому же, этот пиратский видеокодек и так уже слишком долго и успешно присутствует на публичном рынке, да и сам он породил невиданную доселе волну видеопиратства. Пора уже подумать и о чем-нибудь более этичном в использовании.

С другой стороны, перспективы использования у кодека еще есть. Он еще долго будет существовать в среде обычных пользователей, как, например, до сих пор успешно существуют различные магнитные накопители информации. Этому прежде всего будет способствовать огромная накопленная сейчас видеотека пиратского видео, закодированного этим кодеком. Кроме того, даже сейчас многие уважаемые профессионалы видеосжатия предпочитают DivX :-)) любому другому кодеку, чему сильно способствуют программы такого рода как NanDub. В общем, DivX ;-)) еще будет использоваться и для создания нового видеоматериала, но далеко не так широко, как раньше...

Вот практически и все, что я хотел написать о этом замечательном кодеке, хотя сказать о нем могу гораздо больше. Если у вас появились какие-нибудь вопросы, то не стесняйтесь задавать их мне. Сообщайте также, если хотите увидеть статью на какую-нибудь тему, связанную с видеообработкой.

Литература

1. А.Снитко. Обзор основных видеокодексов для современной видеообработки. — Радиомир. Ваш компьютер, 2003, №11. С.2.

Линукс на домашнем компьютере

(Окончание. Начало в №№1-2/2004)

Ю.ЛЕВИН,
E-mail: yl@tut.by

Установка программ

Итак, мы рассмотрели типовые задачи и приложения Линукс, которые могут их решать. Как уже упоминалось в первой части статьи, в дистрибутивы Линукс входит большое число пользовательских приложений, в том числе и рассмотренных здесь. Однако даже дистрибутивы, занимающие несколько компакт-дисков, не в состоянии вместить в себя абсолютно все. А уже имеющиеся программы, особенно те, которые являются свободными разработками, устаревают достаточно быстро. Связано это, в первую очередь, с тем, что в их разработке участвует множество людей — это способствует быстрому развитию проектов.

Поэтому необходимость устанавливать новые и обновлять уже имеющиеся программы, как правило, появляется уже в первые месяцы освоения любого дистрибутива Линукс. В этом разделе я расскажу о способах распространения и установки приложений для Линукс.

Все приложения для Линукс можно разделить на две части — свободные разработки и коммерческие проекты. К первым можно отнести большую часть софта, входящего в большинство дистрибутивов, а в качестве примера коммерческого проекта можно привести вышеупомянутый браузер Opera. В зависимости от того, к какому классу относится программа, возможны два способа ее распространения — в виде двоичных файлов (пакетов) и в виде исходных текстов. Свободные разработки, имеющие открытые исходники (sources), могут распространяться обоими способами, а коммерческие разработки распространяются в виде двоичных файлов.

Исходные тексты программ обычно поставляются в виде архивного файла, в котором, помимо текстов программы, почти всегда содержится краткое описание программы (файл `readme`), инструкция по установке (файл `install`) и скрипт подготовки исходных текстов к компиляции (файл `configure`). Установка таких программ сводится к компиляции исходных текстов.

Программы, распространяющиеся в виде двоичных исполняемых файлов, поставляются в виде пакетов. Пакет — это файл, подобный архиву, который содержит сами файлы программ,

а также дополнительные файлы, например, документацию по поставляемой программе в виде `manpages`. Пакет устанавливается с помощью специальной программы — менеджера пакетов. В дистрибутиве Debian пакетный менеджер называется `DPKG`, а сами пакеты имеют расширение `"deb"`. В дистрибутиве Red Hat был введен менеджер пакетов, который так и был назван — `"Red Hat packet manager"`, сокращенно `"rpm"`. Такое же расширение было присвоено и самим файлам пакетов. Помимо Red Hat, эту технологию управления пакетами используют и другие дистрибутивы — `SuSE`, `Mandrake`, `ASP Linux`. По этой причине их иногда называют `Red Hat-based-дистрибутивами`.

Основы работы с пакетами

В современных дистрибутивах Линукс с пакетами могут работать также и графические утилиты конфигурирования, например, `YaST` в `SuSE`. Однако они обычно позволяют выполнять лишь небольшую часть операций (установку, удаление). Как они работают, без труда сможет разобраться даже начинающий пользователь, поэтому я более подробно остановлюсь на консольном менеджере `"rpm"`, так как он позволяет выполнять полный спектр операций с пакетами.

Имя файла пакета обычно содержит информацию о названии программы, ее версии, типе процессора, под который она скомпилирована, и иногда о системных требованиях данного пакета. Под системными требованиями подразумевается название дистрибутива Линукс, например, `linuxconf-1.31-1.i386.rpm`. Здесь `linuxconf` — это название утилиты, `1.31-1` — ее версия, `i386` — тип процессора.

Другой пример: `freecraft-1.18-3mdk.i586.rpm`. Здесь уже присутствует название дистрибутива — `"mdk"` — сокращение от `Mandrake`.

При поиске пакетов, содержащих ту или иную программу, следует учитывать, что пакеты, собранные под один дистрибутив, как правило, в других дистрибутивах не работают. Связано это с тем, что разные дистрибутивы, особенно если один выпущен значительно позже другого, имеют различные версии системных библиотек. А при компиляции программы была уч-



тена конфигурация именно того дистрибутива, в котором эта компиляция производилась. Могут встречаться версии пакетов одной и той же программы, собранные под разные дистрибутивы. Поставщикам коммерческого ПО, например, Opera, приходится выкладывать пакеты для разных дистрибутивов, чтобы удовлетворить потребности большинства пользователей. Необходимость иметь разные версии пакетов одних и тех же программ — это существенный недостаток такого способа распространения ПО.

Выполнять установку и удаление пакетов может только пользователь уровня root, поэтому если вы хотите выполнить эти действия под обычным пользователем, нужно получить права администратора с помощью команды "su". Установить пакет можно с помощью команды "rpm -i имя_файла_пакета.rpm". Удаление же выполняется командой "rpm -e имя_пакета". Следует отметить, что при установке используется имя файла пакета, а при удалении — имя пакета (в него не входит расширение файла rpm). Приведем примеры некоторых других команд для работы с пакетами:

- rpm -q -all — вывод списка всех установленных пакетов;
- rpm <имя_пакета> -q -provides — выводит на экран список файлов, содержащихся в пакете;
- rpm <имя_пакета> -q -i — выводит на экран описание пакета, содержащегося в пакете;
- rpm -q -f <имя_файла> — выполняет поиск установленного пакета, в котором находится указанный файл. Данная команда необходима в случае, если вы заинтересовались каким-либо файлом и хотите узнать, какому из пакетов он принадлежит.

Опция "-q" в данных примерах означает, что команда не выполняет действий с пакетами, а лишь выводит требуемую информацию о них. Поэтому для выполнения таких команд не нужны права root'a.

Это "страшное" слово "компиляция"

Установка программ, поставляемых в виде исходных текстов, более сложна, однако, по сравнению с установкой пакетов в двоичном формате, она обладает существенным преимуществом — при компиляции программы могут быть учтены особенности дистрибутива, в котором выполняется компиляция. Таким образом, нет необходимости поставлять различные версии программы для разных дистрибутивов.

Для пользователей Windows при упоминании термина "компиляция программы" сразу возникают ассоциации со средами разработки для таких языков как Delphi и C++, работа с которыми требует навыка, а также знания языков программирования. Однако не нужно бояться — компилируя в Линукс исходные тексты программ, написанные на языке C, вам даже не придется что-либо делать с текстами программы. Вся установка выполняется с помощью нескольких команд. Рассмотрим это на примере сборки прокси-сервера Squid.

Исходные коды программ обычно поставляются в виде двойного архива — сначала дерево исходных кодов упаковано программой tar, а затем программой bzip2. Такие файлы имеют двойное расширение — tar.bz2. При установке программ из исходников обычно принято размещать дерево исходных кодов в папке /usr/src. Туда мы и поместим архив с исходниками. Файл называется "squid-2.5.STABLE3.tar.bz2". Нетрудно понять, что в имени файла указаны имя и версия программы, как это делается в пакетах rpm.

Извлечь содержимое такого двойного архива можно командой

```
bzcat squid-2.5.STABLE3.tar.bz2 | tar -x -v.
```

Эта команда выполняет распаковку архива bzip2, и сразу же tar распаковывает извлеченный архив. Параметр "-x" архиватора tar означает распаковку архива, а "-v" — вывод на экран имени каждого извлекаемого файла. Более подробно об использовании архиваторов bzip2 и tar можно прочитать в соответствующих manpages. Помимо консольных команд, распаковку можно выполнить и с помощью архиватора Arc, имеющего графический интерфейс.

В результате распаковки в папке /usr/src появилась поддиректория squid-2.5.STABLE3. В этой директории и содержится дерево исходных кодов Squid. Термин "дерево" подразумевает, что исходные коды содержатся в разных файлах, которые также размещены в разных директориях. Перейдем в корневую директорию дерева исходных кодов. Сменить текущую директорию можно с помощью команды cd. Ее синтаксис аналогичен синтаксису соответствующей команды MS DOS:

```
cd squid-2.5.STABLE3.
```

Оказавшись в корневой директории дерева исходных кодов, можно приступить к сборке. Первое, что нужно сделать — проверить соответствие конфигурации вашего дистрибутива систем-

ным требованиям программы. Для этого используется скрипт configure. Его запуск осуществляется командой

```
./configure.
```

Хочу обратить внимание читателей на то, что Линукс чувствителен к регистру символов в названиях папок и файлов.

При выполнении данного скрипта на экран выводятся данные о наличии в вашем дистрибутиве всех необходимых библиотек для сборки. Эта проверка должна быть выполнена, т.к. для сборки программы у вас должны быть все те же библиотеки, что и у разработчика. В случае отсутствия того или иного компонента, а также в случае, если его версия более ранняя, чем требуемая, будет выведено соответствующее сообщение об ошибке. Вам необходимо доустановить данный компонент из дистрибутива или обновить его версию. Если же все в порядке, то в каждой из папок дерева будет создан файл с именем "makefile". В нем скрипт сохранил данные о наличии в системе всех необходимых для сборки компонентов, а также сведения об их местоположении на диске. Именно на этой стадии сборки учитываются особенности конкретного дистрибутива.

Сейчас можно выполнить саму компиляцию. Для этого используется команда make.

Эта команда определяет, какие файлы требуют компиляции, и компилирует их. Не все файлы устанавливаемой программы всегда требуют компиляции. Например, если компиляция была прервана, то при повторном запуске будут откомпилированы лишь оставшиеся файлы. Для удаления результатов предыдущей компиляции можно использовать команду make clean.

Если компиляция завершилась успешно, то полученные исполняемые файлы нужно переместить в папки, откуда они будут вызываться на выполнение. В большинстве случаев установки пользовательскими программами для этого используется отдельная папка /usr/local/bin. Так, файлы программ, установленных пользователем, проще отличить от программ, входящих в дистрибутив. Перемещение выполняется с помощью команды make с параметром install:

```
make install
```

Теперь можно вызывать установленную программу либо командой из консоли, либо с помощью ярлыка или меню KDE. Если вы захотели удалить программу, то перейдите в корневую папку и введите команду make

uninstall. При этом будут удалены только двоичные файлы программы, а дерево исходных кодов останется, его можно удалить вручную.

Итак, помимо распаковки архива, весь процесс сборки и установки программы состоит из трех команд:

```
.configure
make
make install
```

Однако иногда количество команд может быть несколько большим, либо они могут иметь дополнительные параметры. Например, скрипт configure не всегда автоматически может определить расположение некоторых компонентов. Тогда может понадобиться указать правильный путь как параметр при запуске скрипта. Почти всегда в архиве с исходными кодами содержатся файлы "readme" и "install". В них всегда можно найти информацию о системных требованиях программы и процессе ее установки. Лучше всего, если это возможно, еще перед скачиванием программы ознакомиться с данной информацией на сайте разработчиков программы, чтобы избежать возможных трудностей при установке или даже не скачивать программу зря — может так оказаться, что ваш дистрибутив не соответ-

Табл. 4

Адрес	Описание ресурса
http://rpmfind.net	Огромный репозиторий rpm-пакетов. Здесь собраны пакеты из разных дистрибутивов, поэтому при скачивании нужно быть точно уверенным, что вы выбрали именно то, что вам нужно. Помимо этого, здесь можно получить информацию о содержимом пакета и его системных требованиях, не скачивая сам пакет, что довольно удобно
http://freshmeat.net	Хостинг для проектов, разрабатывающих свободное ПО
http://www.sourceforge.net	Кузница исходников — еще один хостинг для проектов свободного ПО. Как вы могли заметить, многие программы, упомянутые в статье, можно получить со страниц, находящихся в этих двух доменах

ствует системным требованиям, а обновить его не так просто. Тогда можно остановиться на не самой последней версии программы, которая вполне может соответствовать вашему дистрибутиву.

Заключение

Вот мы и закончили краткое знакомство с операционной системой Линукс. На примере дистрибутива SuSE 8.1 было показано, что Линукс может успешно решать такие задачи как работа с документами, работа в Интернете, проигрывание аудио и видео. Этого во многом достаточно для домашнего ПК. Возможности современных

дистрибутивов и их разнообразие вполне позволяют пользователю сделать выбор — будет ли Линукс первой операционной системой на его компьютере, второй, или, может, даже единственной.

В табл.4 приведены ссылки на некоторые сетевые ресурсы по теме данной части статьи.

Литература

1. Д.Бендел, Р.Нейпир. Использование Linux. 6-е издание. — СПб.: Вильямс, 2002. — 784.
2. В.Куц. Знакомимся с офисным пакетом OpenOffice.org. — Радиомир. Ваш компьютер, 2003, №1, С.7.

Р.ГАЛЕЕВ /HI-TECH,
Башкортостан, г.Туймазы.

Сага о кодировках

(Продолжение. Начало в №№1-2/2004)

6. Windows

Windows — полностью графическая среда, поэтому псевдографические символы с появлением этой системы потеряли свою актуальность. Для Windows была разработана своя кодировка, получившая название кодировки ANSI (табл.10). Первые 128 символов в ней полностью соответствуют ASCII-коду, изображения для первых 32 управляющих символов отсутствуют. В "расширенной" части, по сравнению с кодировкой IBM, были удалены все псевдографи-

ческие символы, значительно расширено представительство символов европейских алфавитов, а также добавлено множество других полезных символов.

В русскоязычной версии Windows используется другая версия кодировки ANSI, получившая обозначение кодовой страницы 1251 (табл.11). Русские буквы в ней располагаются в позициях 0C0h...0FFh, а также 0A8h и 0B8h (для Ё и ё), что не соответствует их позициям в sr866. В терминологии Windows кодировка для DOS-программ называется

кодировкой OEM (от Original Equipment Manufacturer — кодировка производителя оборудования). В Win32 API существуют стандартные функции для перекодировки из OEM в ANSI и обратно — OemToChar, CharToOemBuff, CharToOem, CharToOemBuff, а также функции для получения текущей кодовой страницы ANSI (GetACP) или OEM (GetOEMCP) и функция для получения информации о данной кодовой странице (GetCPInfo).

Табл. 10. Кодировка символов ANSI (Windows), символы 128...255 (80h...0FFh)

€	80		90		A0	°	B0	À	C0	Ð	D0	à	E0	ð	F0
	81	·	91	ı	A1	±	B1	Á	C1	Ñ	D1	á	E1	ñ	F1
,	82	·	92	¢	A2	²	B2	Â	C2	Ò	D2	â	E2	ò	F2
f	83	“	93	£	A3	³	B3	Ã	C3	Ó	D3	ã	E3	ó	F3
..	84	”	94	¤	A4	´	B4	Ä	C4	Ô	D4	ä	E4	ô	F4
..	85	•	95	¥	A5	µ	B5	Å	C5	Õ	D5	å	E5	õ	F5
†	86	—	96	¦	A6	¶	B6	Æ	C6	Ö	D6	æ	E6	ö	F6
‡	87	—	97	§	A7	·	B7	Ç	C7	×	D7	ç	E7	÷	F7
ˆ	88	ˆ	98	¨	A8	,	B8	È	C8	Ø	D8	è	E8	ø	F8
%	89	™	99	©	A9	¹	B9	É	C9	Ù	D9	é	E9	ù	F9
§	8A	§	9A	ª	AA	º	BA	Ê	CA	Ú	DA	ê	EA	ú	FA
ˆ	8B	ˆ	9B	«	AB	»	BB	Ë	CB	Û	DB	ë	EB	û	FB
œ	8C	œ	9C	¬	AC	¼	BC	Ì	CC	Ü	DC	ì	EC	ü	FC
	8D		9D	-	AD	½	BD	Í	CD	Ý	DD	í	ED	ý	FD
Ž	8E	ž	9E	®	AE	¾	BE	Î	CE	Þ	DE	î	EE	þ	FE
	8F	ÿ	9F	-	AF	¿	BF	Ï	CF	ß	DF	ï	EF	ÿ	FF

7. ISO

Рассказ о кодировках был бы неполным, если бы мы не упомя-

Табл. 11. Русскоязычная версия кодировки ANSI (cp1251), символы 128...255 (80h...0FFh)

Ъ	80	ђ	90		A0	°	B0	А	C0	Р	D0	а	E0	р	F0
Г	81	'	91	Ў	A1	±	B1	Б	C1	С	D1	б	E1	с	F1
,	82	'	92	Ѹ	A2	І	B2	В	C2	Т	D2	в	E2	т	F2
ѓ	83	“	93	Ј	A3	і	B3	Г	C3	У	D3	г	E3	у	F3
„	84	”	94	ѡ	A4	г	B4	Д	C4	Ф	D4	д	E4	ф	F4
...	85	•	95	Ѓ	A5	μ	B5	Е	C5	Х	D5	е	E5	х	F5
†	86	—	96	і	A6	¶	B6	Ж	C6	Ц	D6	ж	E6	ц	F6
‡	87	—	97	§	A7	·	B7	З	C7	Ч	D7	з	E7	ч	F7
€	88		98	Ё	A8	ё	B8	И	C8	Ш	D8	и	E8	ш	F8
%	89	™	99	©	A9	№	B9	Й	C9	Щ	D9	й	E9	щ	F9
Љ	8A	љ	9A	Є	AA	»	BA	К	CA	Ъ	DA	к	EA	ъ	FA
<	8B	>	9B	«	AB	»	BB	Л	CB	Ы	DB	л	EB	ы	FB
Њ	8C	њ	9C	¬	AC	ј	BC	М	CC	Ь	DC	м	EC	ь	FC
Ќ	8D	ќ	9D	-	AD	ѕ	BD	Н	CD	Э	DD	н	ED	э	FD
Ћ	8E	ћ	9E	®	AE	ѕ	BE	О	CE	Ю	DE	о	EE	ю	FE
Ц	8F	ц	9F	Ї	AF	і	BF	П	CF	Я	DF	п	EF	я	FF

Табл. 12. Кодировка ISO 8859-5, символы 128...255 (80h...0FFh)

	80		90		A0	А	B0	Р	C0	а	D0	р	E0	№	F0
	81		91	Ё	A1	Б	B1	С	C1	б	D1	с	E1	ё	F1
	82		92	Ђ	A2	В	B2	Т	C2	в	D2	т	E2	ђ	F2
	83		93	Ѓ	A3	Г	B3	У	C3	г	D3	у	E3	ѓ	F3
	84		94	Є	A4	Д	B4	Ф	C4	д	D4	ф	E4	є	F4
	85		95	Ѕ	A5	Е	B5	Х	C5	е	D5	х	E5	ѕ	F5
	86		96	І	A6	Ж	B6	Ц	C6	ж	D6	ц	E6	і	F6
	87		97	Ї	A7	З	B7	Ч	C7	з	D7	ч	E7	ї	F7
	88		98	Ј	A8	И	B8	Ш	C8	и	D8	ш	E8	ј	F8
	89		99	Љ	A9	Й	B9	Щ	C9	й	D9	щ	E9	љ	F9
	8A		9A	Њ	AA	К	BA	Ъ	CA	к	DA	ъ	EA	њ	FA
	8B		9B	Ћ	AB	Л	BB	Ы	CB	л	DB	ы	EB	ћ	FB
	8C		9C	Ќ	AC	М	BC	Ь	CC	м	DC	ь	EC	ќ	FC
	8D		9D	-	AD	Н	BD	Э	CD	н	DD	э	ED	§	FD
	8E		9E	Ў	AE	О	BE	Ю	CE	о	DE	ю	EE	ў	FE
	8F		9F	Ц	AF	П	BF	Я	CF	п	DF	я	EF	Ѹ	FF

нули продукт еще одной организации — ISO (International Standards Organization — Международная организация по стандартизации). Эта организация известна тем, что в городской попытке создать общеупотребительные стандарты зачастую выдает неуклюжие решения, плохо приспособленные к реальным жизненным требованиям. Очевидно, что это вообще характерно для решений, спускаемых “сверху”, что особенно контрастирует с миром развития вычислительной техники, где многие стандарты являются стандартами де-факто.

Как бы то ни было, ISO разработала ряд стандартов серии 8859 для национальных кодировок. Кодировка ISO 8859-1, известная под именем “Latin-1”, совпадает с кодировкой ANSI, используемой в американской версии Windows, чего нельзя сказать о кодировке ISO 8859-5 для кириллицы (табл.12).

Как можно видеть из таблицы, эта кодировка содержит множество дополнительных символов кириллицы, а расположение букв русского алфавита совпадает с ушедшей в небытие основной кодировкой ГОСТ (очевидно, прощальный “подарок” официальных лиц, ее принимавших, поскольку ISO объединяет национальные группы стандартизации). Коды в диапазоне 80h...9Fh отведены для управляющих символов и не имеют графических изображений.

8. Unicode

Краткая история

Еще в 1984 г. в ISO была поставлена задача разрабо-

тать стандарт для международного графического набора символов с использованием 2-байтной кодировки. С другой стороны, в 1980 г. фирма Xerox представила рабочую станцию Star с использованием 2-байтных кодировок для стран Дальнего Востока. К концу 1980-х эти начинания вылились в два параллельно развивающихся проекта по созданию международного унифицированного набора символов: с одной стороны — проект ISO 10646 Международной организации по стандартизации, с другой — проект Unicode, организованный консорциумом производителей многоязычного программного обеспечения.

Стандарт ISO 10646 определяет Универсальный набор символов (Universal Character Set — UCS), который включает в себя практически все символы, когда-либо использовавшиеся человечеством. Это не только символы алфавитов всех языков Земли, но также и математические, научные, типографские, технические символы, включая TeX, Postscript, MS DOS, MS Windows, Macintosh, шрифты OCR, древние иероглифы и даже искусственно созданные символы из книг Толкиена.

Для кодирования такого количества информации первоначально выделенных 2 байтов оказалось недостаточно, поэтому ISO 10646 формально определяет набор 31-битных символов в диапазоне кодов от 0 до 7FFFFFFFh. Наиболее часто встречающиеся из них, включая все более ранние кодировки, располагаются в первых 65534 позициях (с 0000h по 0FFFFh). Это 16-разрядное подмножество UCS называется Основной многоязычной плоскостью (Basic Multilingual Plane — BMP) или нулевой плоскостью (Plane 0). Базовая архитектура BMP и ее содержание были определены в опубликованном в 1993 г. стандарте ISO/IEC 10646-1. Символы, добавляемые за пределы 16-разрядной BMP, относились к специальным приложениям, таким как исторические записи и научная нотация; причем проводимая в настоящее время политика предполагает, что не будут назначаться символы в области, выходящей за рамки 21 бита (от 000000 до 10FFFFFFh), что составляет в сумме свыше миллиона символов. Символы в других плоскостях, помимо BMP, были определены во второй части стандарта ISO/IEC 10646-2, вышедшей в 2001 г.

В проекте Unicode для кодирования символов первоначально использовались 16 битов; сама же кодировка в первой версии незначительно отличалась от кодировки ISO 10646. Но к 1991 г. участники обоих проектов осознали, что необходимо создать единую кодовую таблицу, и объединили усилия по унификации своих кодовых таблиц, договорившись и впредь координировать работу друг с дру-

Табл. 13. Коды Unicode для кириллицы

Символ	Код	Название символа в Unicode	Символ	Код	Название символа в Unicode
Ё	401	Cyrillic Capital Letter Io	ё	451	Cyrillic Small Letter Io
Ђ	402	Cyrillic Capital Letter Dje	ђ	452	Cyrillic Small Letter Dje
Ѓ	403	Cyrillic Capital Letter Gje	ѓ	453	Cyrillic Small Letter Gje
Є	404	— « — Ukrainian Ie	є	454	Cyrillic Small Letter Ukrainian Ie
Ѕ	405	Cyrillic Capital Letter Dze	ѕ	455	Cyrillic Small Letter Dze
Ї	406	— « — Byelorussian-Ukrainian I	і	456	— « — Byelorussian-Ukrainian I
Ј	407	Cyrillic Capital Letter Yi	ј	457	Cyrillic Small Letter Yi
Ј	408	Cyrillic Capital Letter Je	ј	458	Cyrillic Small Letter Je
Љ	409	Cyrillic Capital Letter Lje	љ	459	Cyrillic Small Letter Lje
Њ	40A	Cyrillic Capital Letter Nje	њ	45A	Cyrillic Small Letter Nje
Ћ	40B	Cyrillic Capital Letter Tshe	ћ	45B	Cyrillic Small Letter Tshe
Ќ	40C	Cyrillic Capital Letter Kje	ќ	45C	Cyrillic Small Letter Kje
Ў	40E	Cyrillic Capital Letter Short U	ў	45E	Cyrillic Small Letter Short U
Ц	40F	Cyrillic Capital Letter Dzhe	ц	45F	Cyrillic Small Letter Dzhe
А	410	Cyrillic Capital Letter A	а	430	Cyrillic Small Letter A
Б	411	Cyrillic Capital Letter Be	б	431	Cyrillic Small Letter Be
В	412	Cyrillic Capital Letter Ve	в	432	Cyrillic Small Letter Ve
Г	413	Cyrillic Capital Letter Ghe	г	433	Cyrillic Small Letter Ghe
Д	414	Cyrillic Capital Letter De	д	434	Cyrillic Small Letter De
Е	415	Cyrillic Capital Letter Ie	е	435	Cyrillic Small Letter Ie
Ж	416	Cyrillic Capital Letter Zhe	ж	436	Cyrillic Small Letter Zhe
З	417	Cyrillic Capital Letter Ze	з	437	Cyrillic Small Letter Ze
И	418	Cyrillic Capital Letter I	и	438	Cyrillic Small Letter I
Й	419	Cyrillic Capital Letter Short I	й	439	Cyrillic Small Letter Short I
К	41A	Cyrillic Capital Letter Ka	к	43A	Cyrillic Small Letter Ka
Л	41B	Cyrillic Capital Letter El	л	43B	Cyrillic Small Letter El
М	41C	Cyrillic Capital Letter Em	м	43C	Cyrillic Small Letter Em
Н	41D	Cyrillic Capital Letter En	н	43D	Cyrillic Small Letter En
О	41E	Cyrillic Capital Letter O	о	43E	Cyrillic Small Letter O
П	41F	Cyrillic Capital Letter Pe	п	43F	Cyrillic Small Letter Pe
Р	420	Cyrillic Capital Letter Er	р	440	Cyrillic Small Letter Er
С	421	Cyrillic Capital Letter Es	с	441	Cyrillic Small Letter Es
Т	422	Cyrillic Capital Letter Te	т	442	Cyrillic Small Letter Te
У	423	Cyrillic Capital Letter U	у	443	Cyrillic Small Letter U
Ф	424	Cyrillic Capital Letter Ef	ф	444	Cyrillic Small Letter Ef
Х	425	Cyrillic Capital Letter Ha	х	445	Cyrillic Small Letter Ha
Ц	426	Cyrillic Capital Letter Tse	ц	446	Cyrillic Small Letter Tse
Ч	427	Cyrillic Capital Letter Che	ч	447	Cyrillic Small Letter Che
Ш	428	Cyrillic Capital Letter Sha	ш	448	Cyrillic Small Letter Sha
Щ	429	Cyrillic Capital Letter Shcha	щ	449	Cyrillic Small Letter Shcha
Ъ	42A	Cyrillic Capital Letter Hard Sign	ъ	44A	Cyrillic Small Letter Hard Sign
Ы	42B	Cyrillic Capital Letter Yeru	ы	44B	Cyrillic Small Letter Yeru
Ь	42C	Cyrillic Capital Letter Soft Sign	ь	44C	Cyrillic Small Letter Soft Sign
Э	42D	Cyrillic Capital Letter E	э	44D	Cyrillic Small Letter E
Ю	42E	Cyrillic Capital Letter Yu	ю	44E	Cyrillic Small Letter Yu
Я	42F	Cyrillic Capital Letter Ya	я	44F	Cyrillic Small Letter Ya
Ґ	490	— « — Ghe With Upturn	ґ	491	— « — Ghe with upturn
Ғ	492	— « — Ghe With Stroke	ғ	493	— « — Ghe With Stroke
Ҕ	496	— « — Zhe With Descender	ҕ	497	— « — Zhe With Descender
Җ	49A	— « — Ka With Descender	җ	49B	— « — Ka With Descender
Ҙ	49C	— « — Ka With Vertical Stroke	ҝ	49D	— « — Ka With Vertical Stroke
Ҡ	4A2	— « — En With Descender	қ	4A3	— « — En With Descender
Ү	4AE	Cyrillic Capital Letter Straight U	ү	4AF	Cyrillic Small Letter Straight U
Ұ	4B0	— « — Straight U With Stroke	ұ	4B1	— « — Straight U With Stroke
Ҳ	4B2	— « — Ha With Descender	ҳ	4B3	— « — Ha With Descender
Ң	4B8	— « — Che With Vertical Stroke	ң	4B9	— « — Che With Vertical Stroke
Һ	4BA	Cyrillic Capital Letter Shha	һ	4BB	Cyrillic Small Letter Shha
Ө	4D8	Cyrillic Capital Letter Schwa	ө	4D9	Cyrillic Small Letter Schwa
Ө	4E8	Cyrillic Capital Letter Barred O	ө	4E9	Cyrillic Small Letter Barred O

гом. В результате Unicode 1.1 и ISO 10646-1 от 1993 г. стали использовать один и тот же набор символов и их кодов. Все версии Unicode, начиная с 2.0, совместимы между собой, новые символы лишь добавляются, но старые не удаляются и не переименовываются. Версия Unicode 3.0 соответствует изданию ISO 10646-1 от 2000 г., а в Unicode 3.2 добавлено издание ISO 10646-2 от 2001 г.

ISO 10646 является не более чем простой таблицей символов, наподобие изданных ранее таблиц ISO серии 8859, с некоторыми дополнительными сведениями, касающимися

ся терминологии, вариантов альтернативных кодировок и связи с другими стандартами ISO. Стандарт же Unicode определяет большую семантику ряда символов и является гораздо более удобным справочным руководством для реализации высококачественных издательских систем. Unicode определяет алгоритмы для преобразования вида некоторых шрифтов (например, арабских), обработки текстов, содержащих двунаправленное написание (например, смесь латинского и иврита), алгоритмы сортировки и сравнения строк и многое другое. Поэтому в компьютерном мире на универсальную систему кодировки ссылаются обычно как на Unicode, хотя ее кодовые таблицы тождественны ISO 10646. Далее мы будем называть кодировку Unicode/ISO 10646 просто Unicode. Для обозначения кодов Unicode в тексте перед соответствующим 16-ричным числом обычно ставят префикс "U+", например, U+007F.

Содержание Unicode

Важно отметить, что Unicode прежде всего кодовая таблица, которая однозначно идентифицирует символы и назначает им целочисленные номера. Она не определяет изображения символов — так называемые *глифы* (glyph) — это задача разработчиков шрифтов. Каждый символ имеет свое официальное название на английском, например, символ с кодом U+0000 называется "Null", а символ с кодом U+0041 — "Latin capital letter A". Первые 128 символов Unicode однозначно соответствуют кодировке ASCII, а первые 256 символов — кодировке Latin-1 (ISO 8859-1). Для представления кодов символов в плоскости BMP достаточно 2 байтов (в терминологии ISO эта запись обозначается как UCS-2), а для символов из других плоскостей — 4 байтов (UCS-4). Таким образом, чтобы получить коды Unicode из ASCII-кодов, необходимо просто поставить перед последними ведущий байт 0, и наоборот, убрав ведущий 0, из кода Unicode (в форме UCS-2) получаем ASCII-код. В 32-разрядной (UCS-4) записи необходимо соответственно добавлять или удалять по три нулевых байта. Коды Unicode для символов кириллицы приведены в виде 16-ричных чисел в табл.13.

(Окончание следует)

А.ГРИНЧУК,

E-mail: gav@nihe.niks.by

Анимация с Macromedia Flash MX

Анимация движения

В данной статье мы, наконец-то, приступаем к подробному изучению тех возможностей Flash, ради которых эта программа и создавалась. Речь пойдет об анимации, точнее, об одном из ее типов — анимации движения. Нашей целью пока является научиться перемещать фигуры по прямолинейным траекториям, без изменения их формы, прозрачности и цвета. Для работы нам понадобятся в первую очередь такие операции как (рис.1):

Рис. 1

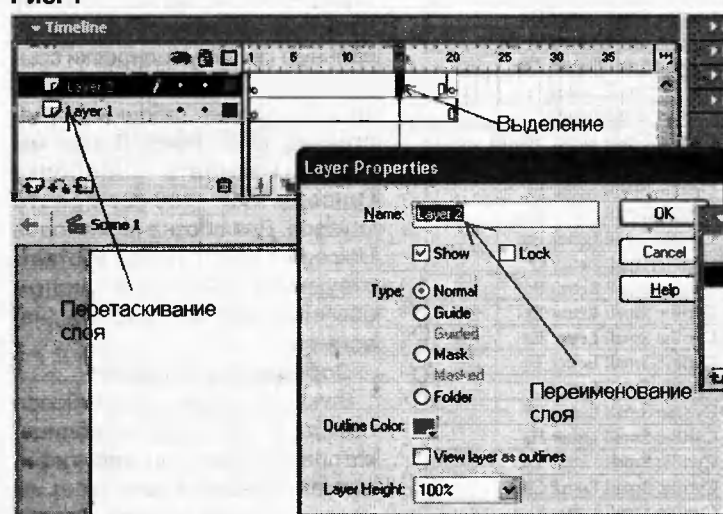


Рис. 2



- создание нового слоя (**Insert/Layer**);
- выделение нужного кадра на кадровой линейке (щелчок мыши осуществляется на кадровой линейке соответствующего слоя, например, на рис.1 выделен 15-й кадр слоя **Layer 2**);
- переименование слоя (**Modify/Layer**);
- вставка ключевого кадра (выделить кадр/**Insert/Keyframe** или **F6**);
- группировка и разгруппировка ри-

сунков (соответственно **Modify/Group** и **Modify/Ungroup**).

Эти операции мы ранее рассматривали отдельно, здесь же нам регулярно придется применять их в комплексе, так как сложная анимация состоит из набора более простых, а каждая из простых осуществляется в отдельном слое. Кроме того, важна правильная группировка фигур. Ключевые кадры важны по той причине, что только их приходится создавать "вручную", промежуточные же прорисовываются программой в режиме автозаполнения (**Tween**

ренней части круга и по границе и выбираем команду **Modify/Group**. Затем выделяем конечный кадр нашей анимации (например, 30-й) и вставляем ключевой кадр (**F6**). На эти манипуляции моментально реагирует временная шкала серой заливкой промежуточных кадров (рис.3). Но анимация еще не создана, необходимо снова выделить первый кадр, открыть панель **Properties** и указать тип анимации (**Tween: Motion** — рис.4). Только сейчас временная шкала подтвердит создание анимации — появится сплошная стрелка (признак правильного создания анимации, в противном случае стрелка будет прерывистой), а кадры с 1-го по 30-й будут закрашены сиреневым цветом (это уже признак анимации движения, рис.5).

Рис. 3

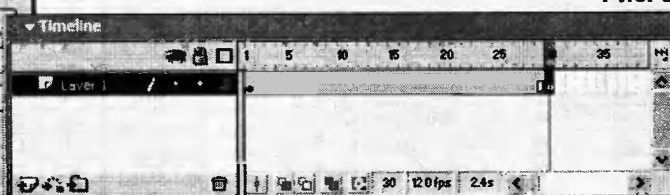


Рис. 4

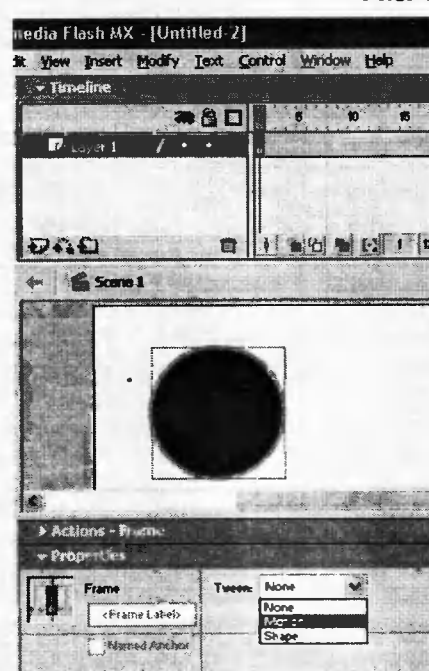
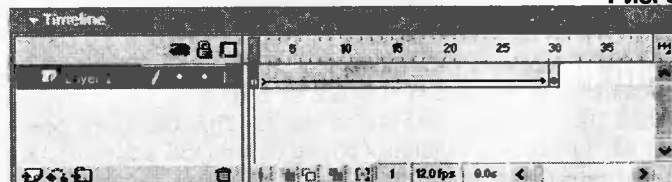


Рис. 5



— промежуток, промежуточные кадры). Мы начнем с простейшей анимации, а затем, шаг за шагом, рассмотрим более сложные и интересные варианты.

Движение одной фигуры

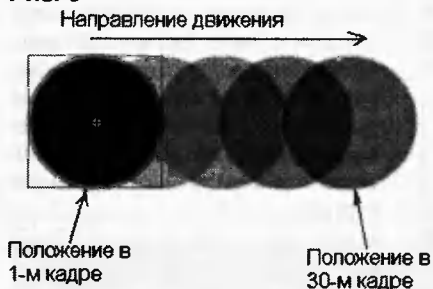
Вначале просто заставим двигаться круг, который легко нарисовать с помощью инструмента **Oval** (рис.2). Первый кадр автоматически помещается

как ключевой (черной точкой на кадровой линейке). Для создания анимации движения (**Motion**) фигура обязательно (!) должна быть сгруппирована — выбираем инструмент **Arrow** (указатель — черная стрелка), с нажатой клавишей **Shift** щелкаем по внут-



Однако даже и сейчас мы не увидим движения по той простой причине, что начальное и конечное положение круга совпадают. Можно войти в первый кадр и с помощью мыши переместить круг левее, а в 30-м кадре — вправо. Теперь наша фигура будет двигаться слева направо (рис.6). Просмотреть анимацию в отдельном окне можно после нажатия комбинации клавиш **Shift + Enter**.

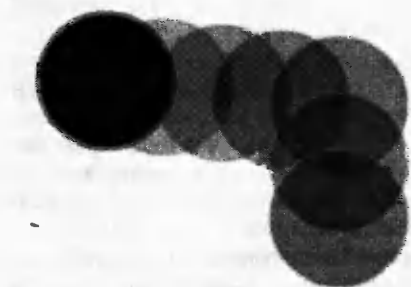
Рис. 6



Последовательные движения одной фигуры

Простое линейное движение фигуры требуется только в редких случаях. Поэтому мы усложним задачу, и после движения слева направо заставим круг с 30-го по 45-й кадры двигаться вниз. Выделяем 45-й кадр и делаем его ключевым (**F6**). А вот начальным кадром данного отрезка анимации будет 30-й. Поэтому возвращаемся именно в 30-й кадр и задаем тип анимации на панели **Properties (Tween: Motion)**. Остается вернуться в 45-й кадр и сдвинуть круг чуть ниже (рис.7).

Рис. 7



Анимация с ускорением и замедлением

Для создания эффекта неравномерности движения можно поступить следующим образом — создать два или более последовательных движений фигуры. На одном из отрезков анимации перемещение фигуры задать большим, на другом — меньшим. Однако для плавного изменения скорости такой способ не очень-то и подходит, а потребность в нем возникает часто. Примером может служить ускоренное падение мячика, взмах руки и т.д. Решение заключается в использо-

вании параметра **Ease (замедлить)**, который настраивается при задании типа анимации в ключевом кадре. Поле этого параметра находится на панели **Properties**, чуть ниже поля **Tween** (рис.8). Для настроек можно как вводить значение параметра с клавиатуры, так и щелчком мыши вызвать линейку с ползунком. Допустимые значения параметра лежат в диапазоне от -100 до 100. Положительным значениям соответствует замедление в начальный период анимации, отрицательным — в конечный, «на выходе» (рис.9). При больших значениях **Ease** неравномерность большая, а при нулевых — движение, как и ранее, равномерное.

Анимация с вращением

У пользователя Flash есть возможность не только передвинуть фигуру в начальном и конечном кадрах, но и повернуть ее. Достаточно выделить фигуру и выбрать **Modify/Transform/Rotate and Skew** (повернуть и наклонить). Минус такого подхода — таким образом в режиме автозаполнения кадра можно заставить фигуру повернуться не более чем на пол-оборота. Полноценный вариант работы основан на использовании все той же панели **Properties**. В качестве примера мы заставим вращаться линию. В новой анимации нарисуем линию и, как принято в анимации движения, сгруппируем ее. Затем вставим ключевой кадр в 30-й позиции. Возвращаемся в первый кадр и открываем панель **Properties**. Осталось задать тип анимации (**Tween: Motion** — рис.10) и тип вращения (**Rotate: CW 1 Times** — по часовой стрелке 1 раз, против часовой стрелки — **CCW** — Counter-Clockwise). Причем вращение можно совмещать с перемещением (рис.11) и с замедлением.

Анимация с масштабированием

Хорошая мультипликация отличается тем, что в ней соблюдаются законы перспективы. Например, подъезжающий автомобиль увеличивается в размерах. Flash может создавать хорошую анимацию. Конечно, рисовать вручную автомобиль на стадии изучения программы — заня-

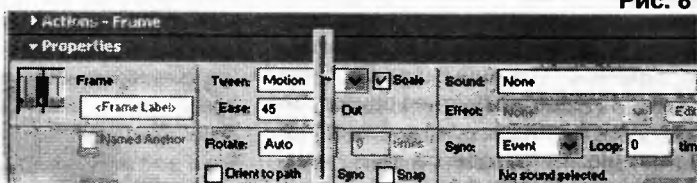


Рис. 8

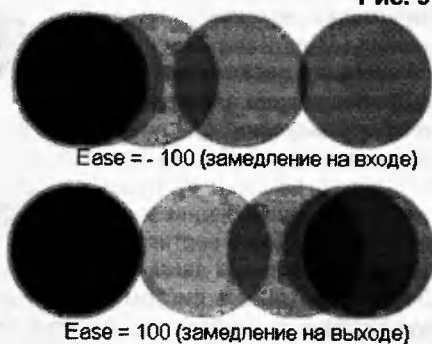


Рис. 9

тие неблагоприятное. Лучше воспользоваться уже готовым рисунком. Если у вас установлен Office XP, то выполнив импорт рисунка (**File/Import**), выберем файл **C:\Program Files\Microsoft Office\medialcagcat10\j0212957.wmf**. Картинку, естественно, необходимо

Рис. 10

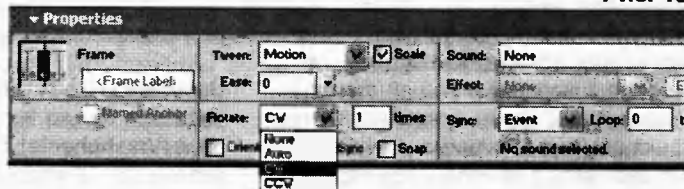
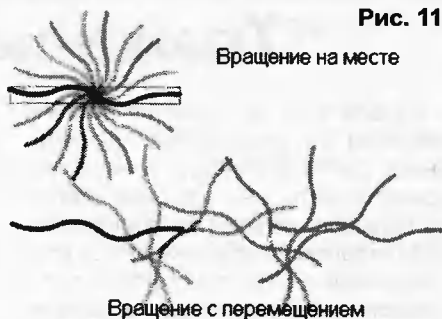
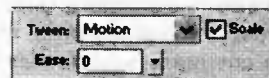


Рис. 11



сгруппировать и передвинуть в правый верхний угол рабочей области. Вставим ключевой кадр для разнообразия уже в 25-ю позицию и снова вернемся в первый. После задания типа анимации проверим на панели **Properties** наличие галочки в позиции **Scale** (масштабировать — рис.12).

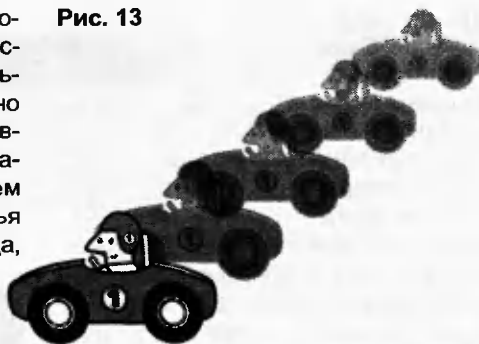
Рис. 12



Осталось подредактировать последний кадр. Передвинем автомобиль в левый нижний угол, не снимая выделения с картинки выполним

Modify/Transform/Scale. Затем, за появившийся угловой маркер чуть растянем рисунок автомобиля. Несколько кадров с этой анимации показано на рис.13. Еще эффектнее приезд автомобиля выглядит, если добавить замедление на выходе со значением **Ease** около 70. Напомним, что братья Люмьер начинали с приезда поезда, так что у нас все еще впереди.

Рис. 13



оборота уже выглядят мельтешением, так что главное — не переусердствовать. Пусть вас также не пугает и тот факт, что некоторые элементы рисунка оказываются за пределами рабочей области, в конечном итоге все будет работать нормально.

Конечно, хотелось бы придать нашей анимации большую эффектность. Например, добавить изменение цвета, изменение формы, и желательно убрать излишнюю прямолинейность движения (в буквальном смысле слова). Если у вас возникло желание поэкспериментировать в этом направлении, и вдобавок, ваш компьютер все еще на гарантии, то развитием темы займемся в следующей статье.

Рис. 15

Анимация нескольких фигур

Для последнего эксперимента воспользуемся сразу несколькими приемами из рассмотренных выше. Текст будет выезжать справа (надпись создаем в слое **Text**), вертикальную линию заставим двигаться сверху вниз (линию рисуем и группируем в новом слое **Dvizhenie**), а горизонтальная должна вращаться (добавляем еще один слой, называем **Vrashchenie**, рисуем и группируем линию — рис.14). Текст, в отличие от рисованных фигур, группировать не нужно — он сразу создается как сгруппированный объект. На каждом слое 25-й кадр делаем ключевым. Конечные кадры нас устраивают — так должно выглядеть именно завершение нашей анимации. А вот первые кадры нуждаются в модификации (рис.15). Анимацию необходимо задавать в каждом слое отдельно.



Рис. 14

Вращение на один оборот задается только в одном слое (**Vrashchenie**). На отрезке в 25...30 кадров два



“Твикаем” окошки

В.КУЦ,

E-mail: victor_kootz@mail.ru

Ни для кого не является особым секретом, что разработчики операционной системы Windows изначально ориентировались на охват максимального количества пользователей, балансируя при этом между производительностью, простотой использования и совместимостью со всем на свете. Поэтому эта операционная система (любой из версий, начиная от Windows 95 и вплоть до Windows XP) по умолчанию никогда не была оптимально настроена. А раз так, придется нам взять это дело в свои руки.

Хорошо известно, что наиболее эффективными являются методы оптимизации Windows посредством редактирования системного реестра. Делать это можно как вручную, так и с помощью специализированных утилит. Однако такая работа требует от пользователя достаточно высокой квалификации: ведь сам реестр, кроме того, что изрядно запутан, еще и

очень плохо документирован — досконально разобраться в нем никому, за исключением разве что работников самой Microsoft, практически не под силу. Обычным же “простым смертным”, не обремененным ни излишним временем, ни желанием ковыряться в хитросплетениях реестра Windows, но желающим использовать возможности компьютерного “железа” в максимальной степени и в то же время максимально придать своей системе индивидуальность, гораздо сподручнее воспользоваться возможностями, предоставляемыми специальными программами-оптимизаторами. Хотя многочисленные апплеты “Панели управления” Windows как раз и задумывались для облегчения настройки системы конечным пользователем, тем не менее, они никогда не охватывали всех (и, надо честно признать, довольно богатых) возможностей тонкой настройки этой ОС.

Обычно программы-оптимизаторы ОС, которых сегодня насчитываются десятки, если не сотни, представляют собой оболочки, в которых сконцентрировано множество инструментов тонкой настройки механизмов операционной системы. Большинство из них вполне эффективны, и позволяют, используя лишь внутренние ресурсы операционной системы, даже неподготовленному пользователю, не имеющему никакого представления о существовании и назначении реестра, настроить Windows в широких пределах, равно как и изменить ее внешний вид.

Прежде чем приступить к увлекательному процессу оптимизации Windows, настоятельно рекомендуется не забывать о соблюдении элементарных правил своего рода “техники безопасности”. Во-первых, перед любыми экспериментами с системой, какими бы, на первый взгляд, безобид-



ными они ни казались, необходимо подготовить себе пути к отступлению — выполнить резервное копирование реестра или, что еще лучше, сделать полную копию системного раздела. Для этого больше всего подойдут широко известная Norton Ghost или набирающая популярность отечественная программа Acronis TrueImage. Во-вторых, категорически не рекомендуется изменять те параметры настройки, значение которых вам не совсем понятно. Всегда следует помнить, что тюнинговые программы, как и любое другое "орудие труда", могут служить инструментом не только созидания, но и разрушения.

Сегодня мы рассмотрим две программы-оптимизаторы Windows. Обе они написаны российскими программистами, и, естественно, имеют интерфейс на русском языке. Для многих начинающих компьютерщиков, большинство из которых не особо ладят с английским языком, этот момент может оказаться решающим при выборе программы для постоянного использования.

XP Tweaker RE 1.30 build 42

Не так давно появилась и уже обрела определенную популярность бесплатная (что встречается довольно редко для программ такого класса) программа XP Tweaker, основной задачей которой является настройка и оптимизация операционной системы Windows XP. На первый взгляд, она напоминает известную программу-"тви-

кер" Tweak XP Pro, правда, немного усовершенствованную, но при этом гораздо более компактную, да и в отношении потребляемых в процессе работы системных ресурсов XP Tweaker ведет себя гораздо скромнее. Добрым словом стоит помянуть удачный дизайн программы, в котором ее разработчики сумели сохранить баланс между функциональностью и "красивостью", что удается далеко не всегда и не всем. Интерфейс XP Tweaker достаточно прост — пользователю нужно лишь расставлять флажки и выбирать нужные переключатели, а соответствующие им ключи в реестре программа поправит самостоятельно (рис.1).

Все основные настройки XP Tweaker сгруппированы в левом окне по разделам, что обеспечивает удобную навигацию по всем доступным настройкам и позволяет сопровождать каждую из них подробным описанием. Эти разделы, в свою очередь, содержат иерархическую систему конкретных настроек, углубляясь в которую, можно настроить все необходимые параметры.

Система — основной раздел программы, в котором сосредоточены основные системные настройки, непосредственно влияющие на загрузку, функционирование и быстродействие ОС и внешний вид "Проводника" Windows, "Панели задач" и меню "Пуск". Большинство из этих настроек вполне стандартны для типичного "твикера", однако у отечественных пользователей особый интерес могут вызвать несколько опций, предназна-

ченных специально для русских версий Windows XP. Например, маленькая, но очень полезная функция — возможность принудительного задания кодовой страницы "1251" вместо "1252", что решает проблемы с корректным отображением русских шрифтов в некоторых программах. Вроде бы, ерунда, но ведь до появления XP Tweaker эту "ерунду" приходилось в реестре каждый раз делать своими руками.

"Мультимедиа" — позволяет изменять отдельные свойства Windows Media Player и управлять компакт-дисками.

"Защита" — раздел, наиболее интересный для "суровых" сисадминов. Здесь они могут проявить свою "нехорошую" сущность во всей красе, запретив бедному "юзверю" доступ практически к любым мало-мальски важным элементам интерфейса Windows или к некоторым из системных апплетов.

"Рабочий стол" — здесь все просто, и каждый может настроить внешний вид "Рабочего Стола" Windows по своему вкусу.

А вот в разделе **"Интернет"** пользователь может настроить не только Internet Explorer (продублировав, кстати, внутренние настройки браузера), но и изменить настройки сетевого протокола TCP/IP, что иногда может, пусть и незначительно, но все-таки увеличить скорость сетевого соединения, в том числе и с Интернетом.

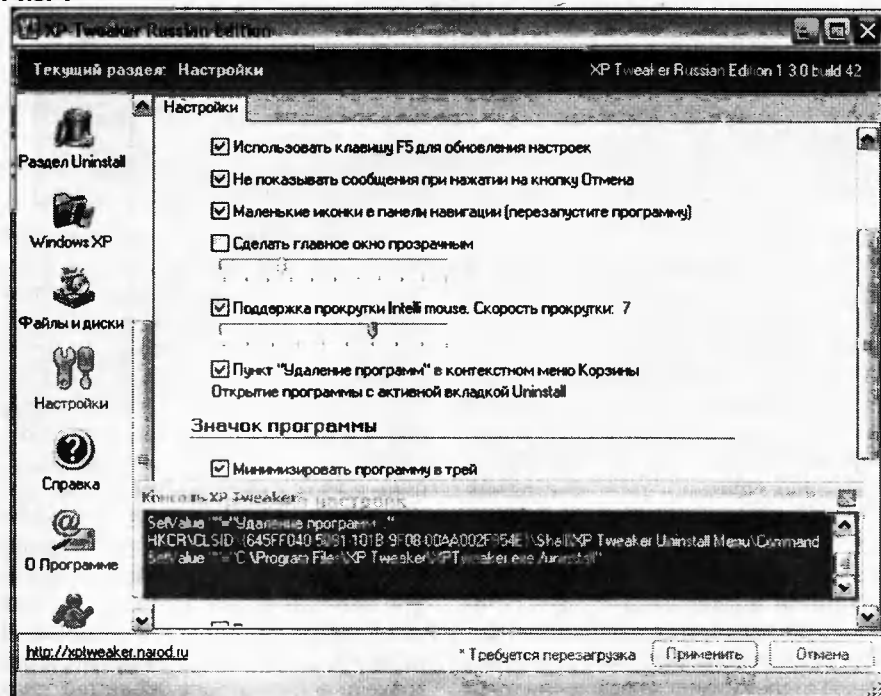
Раздел **"Uninstall"** позволяет не только деинсталлировать ранее установленные программы или удалить все записи о них из реестра, но и вручную корректировать содержимое апплета "Установка и удаление программ".

"Windows XP" — этот раздел является своеобразным аналогом "Панели управления" Windows, предоставляя доступ к большинству ее апплетов. Кроме того, здесь можно оперативно изменить регистрационные данные Windows XP, что небогатому отечественному пользователю поможет избежать многих совершенно излишних хлопот.

Файлы и диски — в дополнение к разделу "Защита", здесь можно запретить запуск отдельных файлов или программ. Воспользовавшись другой вкладкой раздела, можно создать виртуальный диск.

Как видно, основные возможности у XP Tweaker ничуть не хуже, чем у популярных зарубежных "твикалок". Скажу больше — у этой программы есть

Рис. 1



несколько просто уникальных особенностей, которыми не может похвастать никто больше. В первую очередь, это плавающая консоль, изначально предназначавшаяся для тестирования программы, и в которой выводится не только служебная информация, но и фиксируются все изменения, производимые в реестре. Еще одна "приятность" — это сохранение настройки программы в виде файла реестра (*.REG), что позволит применять эти самые настройки практически на любых компьютерах с Windows XP без использования каких-либо предустановленных программ. Кстати, XP Tweaker, в полном соответствии со своим именем, не совместим ни с одной другой версией Windows, кроме Windows XP и новейшей Windows Server 2003. Однако большинство опций программы могут нормально работать в Windows 2000 и Windows Longhorn, если запустить XP Tweaker с ключом /nocheckver.

В программах-оптимизаторах большое значение имеет качество справочной информации, и в этом отношении у XP Tweaker все в порядке. В компактном справочном файле толково и доходчиво изложено не только то, для чего нужна каждая опция настройки, но и приводится информация обо всех изменениях в ключах реестра, которые она производит, что позволяет использовать данную программу в качестве своеобразного учебного пособия при изучении реестра Windows.

Размер дистрибутива XP Tweaker RE составляет 640 Кб, сайт разработчика — <http://xptweaker.narod.ru/>

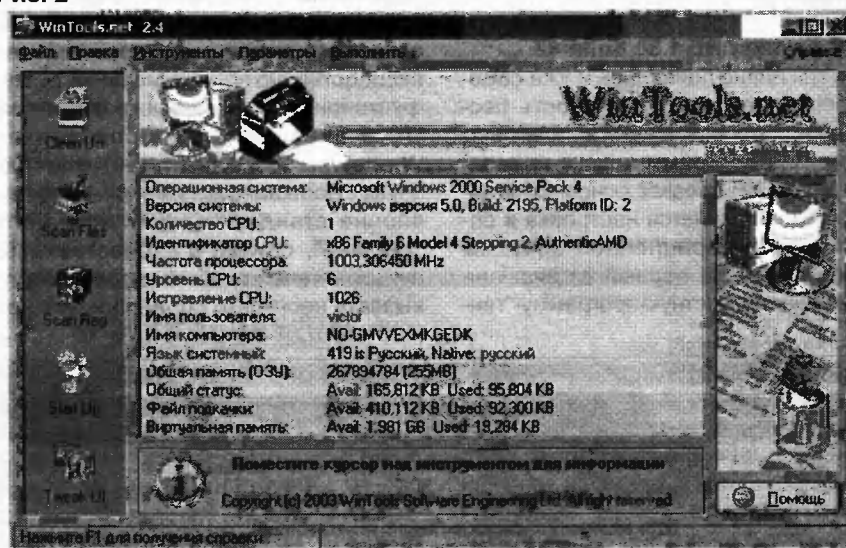
WinTools.net 2.4

Еще одна программа, которую я хотел бы представить вашему вниманию, коренным образом отличается от уже описанной XP Tweaker RE. Впервые, WinTools.net — это не просто оптимизатор операционной системы Windows, а целый набор инструментов, позволяющий поддерживать ее в приличном состоянии на протяжении всего периода эксплуатации. Во-вторых, программа WinTools.net поддерживает весь спектр существующих систем Windows, начиная от древней 95-й и вплоть до самой современной Windows XP. И последнее, WinTools.net не бесплатна. Она распространяется по принципу shareware, и период ознакомления с программой составляет 30 дней. К сожалению, за это время можно ознакомиться только с интерфейсом программы, так как в незарегистрированной версии WinTools.net

отключены практически все основные управляющие кнопки входящих в нее утилит, в первую очередь, те, которые отвечают за сохранение внесенных изменений. Но авторы программы — наши соотечественники, и материальное положение большинства своих земляков им хорошо известно. Поэтому для всех жителей бывшего СССР программа WinTools.net совершенно бесплатна, при ее регистрации надо лишь немного пошевелить мозгами (или вспомнить одну из популярных песен Высоцкого что, впрочем, одно и то же).

Внешний вид WinTools.net очень напоминает вид XP Tweaker RE (как и подавляющего большинства других аналогичных программ). Все так же в левой части рабочего окна расположен столбец с иконками, в данном случае, не разделов, а запускаемых инструментов (рис.2). Правую же часть занимает непосредственно рабочее пространство программы. Из шести утилит, составляющих программу WinTools.net, из рабочего окна можно запустить только пять.

Рис. 2



Clean Uninstaller — позволяет контролировать процессы инсталляции и деинсталляции программ и узнать, какие файлы и куда были записаны, и какие изменения произошли в реестре Windows и в системных файлах после инсталляции. Перед установкой каждого нового приложения Clean Uninstaller создает контрольные точки, в которых "фиксируется" состояние системы в данный момент времени и потом, если возникнет необходимость удалить приложение, система просто "откатится" в свое предыдущее состояние. Как недостаток данного инструмента можно отметить то, что по умолчанию отслеживаются изменения толь-

ко на диске C:\, в случае же задействования любого другого диска необходимо будет вручную внести все его инсталляционные пути в соответствующий список в панели "Настройка".

Scan Files — служит для периодической очистки компьютера от временных и неиспользуемых файлов. Причем удалять ненужные файлы можно как безвозвратно, так и (в случаях неполной уверенности в правильности своих действий) складывать их в "Корзину". Качественный поиск и удаление всяческого мусора обеспечиваются возможностью задать собственные критерии поиска временных файлов. Кроме ненужных файлов, инструмент Scan Files может искать, исправлять или удалять неработающие ярлыки. Как определенный недостаток, здесь можно отметить невозможность автоматической очистки "Корзины", ведь далеко не все из нас регулярно "опорожняют" ее, в результате чего "Корзина" может разрастись просто до безобразных размеров и серьезно тормозить работу компьютера.

Scan Registry — предназначен для периодической очистки реестра и исправления неверных записей. Системный реестр Windows, как и атом, "неисчерпаем", поэтому перед началом работы необходимо прежде всего определиться: "А что, собственно, я хочу найти и удалить в реестре?" И в соответствии со своим осознанным решением выбрать метод (или методы) сканирования реестра в окне "Методы". Кроме того, инструмент Scan Registry позволяет деинсталлировать выбранную программу или удалить запись о ней, "забытую" деинсталлятором. В качестве действенной меры по страховке от всяческих случайностей, с за-

видным постоянством происходящих у всех, кто решился разворошить реестр, может стать возможность создания неограниченного количества его резервных копий, а также простого и быстрого восстановления реестра из любой из них, любезно предусмотренная авторами программы.

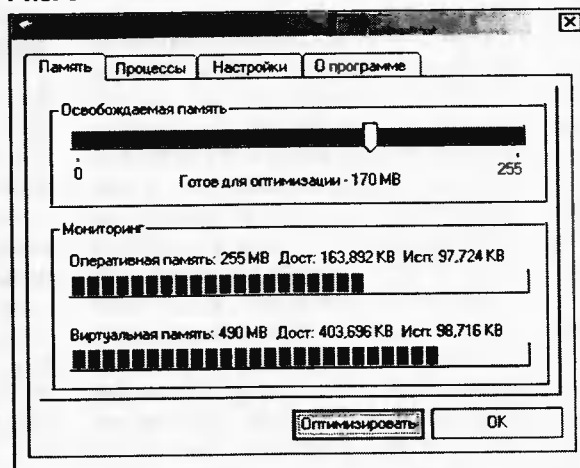
Start Up — следит за всеми записями в системном реестре об автозагрузке приложений и позволяет удалять, добавлять или временно отключать такие записи. Также имеется возможность редактирования выбранной записи. Другой не менее важной функцией инструмента Start Up является контроль за всеми исполняемыми процессами. Особо актуальной данная функция является в ОС семейства Windows 9x, не имеющих штатных средств контроля за запущенными процессами.

Tweak UI — набор дополнительных настроек и возможностей, влияющих на безопасность, производительность и удобство использования операционной системы. Все они, в целях упрощения навигации, разбиты на шесть групп:

- **"Общие"** — здесь можно изменить местоположение системных папок и частично регистрационную информацию;

- **"Экран"** — набор настроек внешнего вида "Рабочего стола" и содер-

Рис. 3



жимого пункта подменю "Создать", вызываемого по нажатию правой кнопки мышки;

- **"Система"** — здесь сосредоточены основные системные установки, управление различными вариантами загрузки системы и драйверами дополнительных OEM-устройств;

- **"Пуск"** — настройка различных вариантов содержимого меню "Пуск";

- **"Охрана"** — здесь на пяти вкладках сосредоточены все "запретительные" настройки, задействование которых в определенной мере поможет защитить систему от несанкционированного вмешательства;

- **"Контроль"** — здесь перечислены все установленные в системе модули "Панели управления". Их мож-

но скрыть или запустить с помощью ранее созданных ярлыков.

Шестым полезным инструментом программы WinTools.net является полностью автономная утилита **Memory Booster**, в полном соответствии со своим названием осуществляющая мониторинг, очистку от неиспользуемых в данный момент драйверов и процессов, а также оптимизацию оперативной памяти (рис.3). При запуске Memory Booster создает свою иконку в системном трее, щелкнув по которой, можно вызвать панель управления и настройки утилиты. Там же, на дополни-

тельной вкладке, можно просмотреть запущенные системные процессы, однако возможности по управлению ими гораздо скромнее, чем у инструмента Start Up.

Размер дистрибутива WinTools.net составляет 760 Кб, сайт разработчика — <http://www.wintools.net>.

Обе рассмотренные программы, пусть и не особо "наворочены", но реально полезны и при этом очень удобны в использовании. Впрочем, тема "твикеры" сегодня настолько популярна в Сети, что, кажется, только самые ленивые программисты еще не выпустили свои собственные варианты оптимизаторов "Окошек", поэтому каждый может подобрать в Сети что-нибудь по своему вкусу.

Носители информации: назад к бумаге?

(Окончание. Начало в №№1-2/2004)

Теперь попробуем рассмотреть еще одну интересную проблему. Вот вы печатаете данные — бит 0 представлен белым (незакрашенным) пикселем, бит 1 — черным (закрашенным). Если вы сами платите за израсходованный краситель (чернила, тонер, красящая лента), то получается, что черный пиксел обходится вам дороже белого. Тогда имеет смысл так перекодировать данные, чтобы уменьшить количество черных пикселей (пусть даже за счет увеличения количества белых пикселей — они ведь дешевле), и таким образом сократить стоимость печати данных.

Приведу пример. Пусть черный пиксел стоит в три раза дороже белого. Здесь и далее будем считать, что в исходных данных каждый бит с одинаковой вероятностью равен 0 или 1 (если исходные данные получены в результате архивации, то примерно так и будет).

Примем цену белого пиксела за 1 у.е., тогда цена черного — 3 у.е. В среднем нулей в данных столько же, сколько единиц, и при обычном способе печати цена записи одного бита составит $(1+3)/2 = 2$ у.е.

А теперь перекодируем исходные данные, заменяя каждые два бита так, как показано ниже:

00 → 1,
01 → 01,
10 → 001,
11 → 000.

Так как в исходных данных в среднем поровну комбинаций 00, 01, 10 и 11, то в среднем два бита исходной информации будут закодированы в $(1+2+3+3)/4 = 2,25$ бита, из них нулей будет $(0+1+2+3)/4 = 1,5$, а единиц — $(1+1+1+0)/4 = 0,75$.

Тогда средняя цена записи двух битов исходной информации после перекодирования равна $1,5+1,0+0,75 = 3,25$ у.е. Средняя цена записи одного бита, соответственно, вдвое меньше и равна 1,875 у.е. — меньше первоначальных 2 у.е.

И.РОЩИН,

г.Москва,

Fido: 2:5020/689.53

ZXNet: 500:95/462.53

E-mail: bestview@mtu-net.ru

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



(Замечу, что вышеописанное кодирование не является идеальным — как мы увидим дальше, теоретический минимум цены записи одного бита при данном соотношении цен белого и черного пикселей равен примерно 1,81 у.е.).

Итак, в чем же суть проблемы? В том, чтобы узнать, как при заданном соотношении цен белого и черного пикселей надо перекодировать исходные данные, чтобы стоимость их записи была наименьшей. По идее, аналогичная проблема должна рассматриваться в теории передачи информации (если время передачи 1 по каналу связи больше времени передачи 0, и надо добиться наименьшего времени передачи данных). Но информации о ее решении я не нашел (не знаю — может, плохо искал), поэтому привожу собственные размышления на эту тему.

Пусть черный пиксел в k раз дороже белого ($k > 1$). Как уже упоминалось выше, считаем, что в исходных данных каждый бит с одинаковой вероятностью равен 0 или 1. Пусть мы имеем наилучший (идеальный) для данного k перекодировщик исходных данных, т.е. обеспечивающий в среднем минимальную цену записи одного бита исходной информации после перекодирования (обозначим эту цену C). Как именно работает этот перекодировщик, мы пока не знаем, но это сейчас неважно.

Пусть доля нулей в перекодированных данных в среднем равна x (каждый бит перекодированных данных равен 0 с вероятностью x и равен 1 с вероятностью $1-x$). Очевидно, что $x > 0,5$, так как уменьшить C можно только за счет относительно увеличения количества нулей в данных.

Также очевидно, что размер перекодированных данных будет в среднем больше размера ис-

ходных данных. Это следует из того, что количество информации после перекодирования должно сохраниться, а если после перекодирования нулей стало в среднем больше, чем единиц, то каждый бит перекодированных данных будет содержать менее одного бита информации, а значит, для хранения того же количества информации, что было в исходных данных, понадобится больше битов.

Попробуем вычислить x . Средняя цена записи одного бита перекодированных данных (обозначим ее C') равна $x \cdot c_0 + (1-x) \cdot c_1$, где c_0 — цена белого пиксела, c_1 — цена черного пиксела. Теперь определим, сколько битов перекодированных данных будут содержать столько же информации,

сколько ее содержит один бит исходных данных (а он и содержит один бит информации).

Информационная емкость одного бита перекодированных данных (обозначим ее H) равна $-(x \cdot \log_2(x) + (1-x) \log_2(1-x))$. При $x > 0,5$ $H < 1$. Для записи одного бита информации, следовательно, потребуется $1/H$ битов перекодированных данных (больше одного бита).

Из вышесказанного следует, что $C = C' / H = (x \cdot c_0 + (1-x) \cdot c_1) / (-(x \cdot \log_2(x) + (1-x) \log_2(1-x)))$. Так как $c_1/c_0 = k$ (по определению), то, если принять $c_0 = 1$ (как мы увидим далее, x не зависит от c_0), получим:

$$C = \frac{x + (1-x) \cdot k}{x \cdot \log_2(x) + (1-x) \log_2(1-x)}$$

Рис. 12

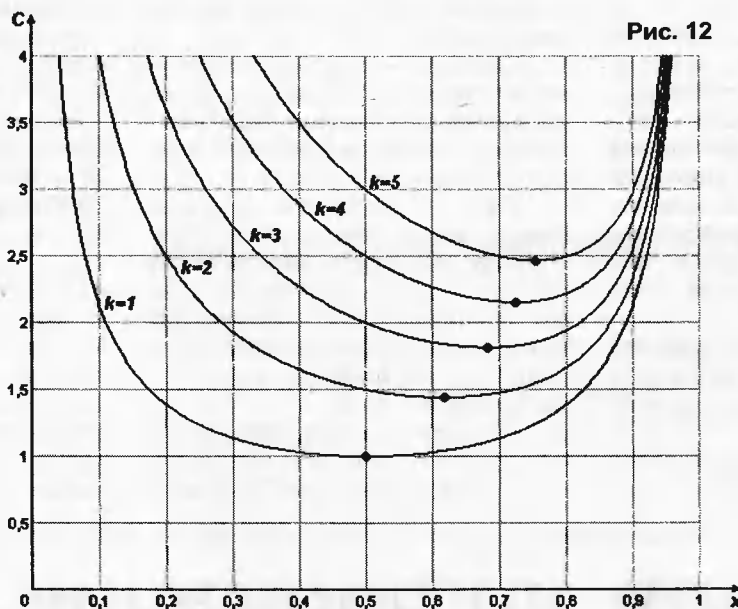
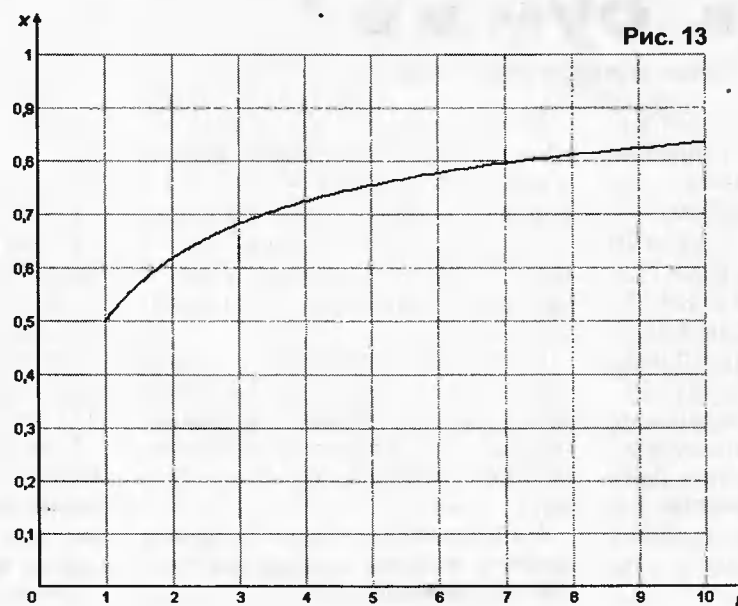


Рис. 13



Так как C минимальна (по условиям задачи), для определения x достаточно найти минимум вышеприведенной функции $C(x)$. Значение x , при котором достигается минимум, и будет искомым. Минимум можно искать либо численным методом, либо попробовать вывести формулу (приравнять нулю производную функции $C(x)$ и решить полученное уравнение — но этот способ я не пробовал).

На рис.12 приведены графики функции $C(x)$ при $k=1, 2, 3, 4, 5$. Минимумы отмечены точками. Видно, что лишь при $k=1$, т.е. когда цена белого пиксела равна цене черного, выгодным является равная доля нулей и единиц в перекодированных данных (т.е. можно их вообще не перекодировать). А с ростом k доля нулей в перекодированных данных также должна возрастать, чтобы C была минимальной.

Значения x , при которых достигается

Табл. 1

k	x	C	Получаемый выигрыш, %
1	0,50	1,00	0,0
2	0,62	1,44	4,0
3	0,68	1,81	9,3
4	0,72	2,15	14,0
5	0,75	2,46	17,8

минимум C, и соответствующие минимальные значения C приведены в табл.1. Также там приведен получаемый выигрыш от перекодирования — на сколько процентов дешевле получается запись информации (т.е. на сколько процентов минимальное C меньше, чем C(0,5)).

На рис.13 вы можете увидеть график зависимости x от k при изменении k от 1 до 10.

Итак, мы смогли определить некоторые характеристики идеального перекодировщика — среднюю долю нулей в перекодированных данных (отсюда легко находится и среднее соотношение нулей и единиц в перекодированных данных) и среднюю цену записи одного бита исходных данных после перекодирования. Таким образом, при разработке своего перекодировщика мы получили возможность ориентироваться на эти характеристики.

Далее я приведу две различных идеи перекодирования.

Первая идея — построить префиксный код с соотношением 0 и 1, близким к теоретически оптимальному. Исходные данные будем кодировать группами по p битов (чем больше p, тем лучше), для чего построим “дерево” с 2^n листьями. Строим дерево сверху вниз, и при каждом разветвлении выбираем соотношение количеств листьев в двух получающихся поддеревьях (с вершиной 0 и с вершиной 1) так, чтобы оно было возможно ближе к теоретически оптимальному соотношению (и, конечно, чтобы получилось хотя бы по одному листу в каждом поддереве). Поэтому и выгоднее брать большое p, тогда соотношения будут ближе к оптимальным.

Я не располагаю каким-либо доказательством того, что полученный по этому алгоритму префиксный код будет лучшим из всех возможных префиксных кодов для данного p.

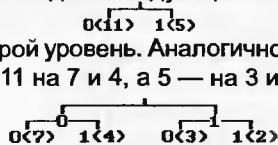
Интуитивное обоснование алго-

ритма следующее — так как каждая развилка в дереве соответствует одному биту для какой-то группы кодов (так, самая верхняя развилка соответствует первому биту для всех кодов), и по условиям задачи все коды равновероятны, то, выбирая соотношение количеств листьев в поддеревьях возможно более близким к оптимальному, мы добиваемся, чтобы соотношение нулей и единиц в этом бите для этой группы кодов было близко к оптимальному. А так как, в свою очередь, каждый бит каждого кода соответствует какой-то развилке дерева, то соотношение нулей и единиц в битах кода будет в среднем достаточно близко к оптимальному: в первых битах кода — ближе всего, а в последующих битах — в среднем все дальше (понятно, что когда мы делим большое целое число на два целых числа в нужной пропорции, погрешность в среднем будет меньше, чем когда мы делим меньшее число в той же пропорции). В последнем бите каждого кода соотношение нулей и единиц будет всегда 1:1, поэтому такое кодирование заведомо неидеально.

Продемонстрирую работу алгоритма на примере. Пусть $k=3$, тогда $x=0,68$ (табл.1), откуда отношение количества нулей к количеству единиц должно быть равно примерно 2,13. Пусть $p=4$, т.е. кодируем группами по четыре бита.

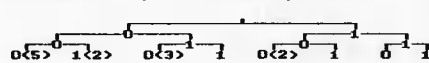
Четыре бита — значит, в дереве будет 16 листьев. Разбиваем их на две части так, чтобы количество листьев в первой части относилось к количеству листьев во второй части возможно ближе к 2,13. Получаем 11 листьев в первой части и 5 во второй. Таким образом, в поддереве с вершиной 0 будет 11 листьев, а в поддереве с вершиной 1 — 5 листьев.

Итак, первый уровень дерева будет выглядеть следующим образом:

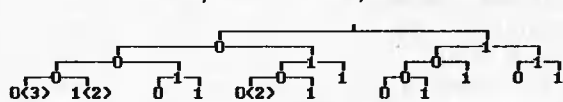


Второй уровень. Аналогично разбиваем 11 на 7 и 4, а 5 — на 3 и 2:

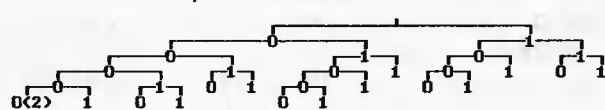
Третий уровень. Разбиваем 7 на 5 и 2, 4 — на 3 и 1, 3 — на 2 и 1, 2 — на 1 и 1:



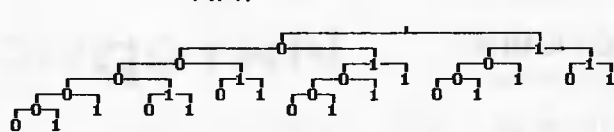
Четвертый уровень. Разбиваем 5 на 3 и 2, 2 — на 1 и 1, 3 на 2 и 1:



Пятый уровень. Разбиваем 3 на 2 и 1, 2 — на 1 и 1:



Шестой уровень. Разбиваем 2 на 1 и 1:



Дерево построено, теперь произвольным образом (т.к. как вероятности всех комбинаций в исходных данных одинаковы) сопоставляем 16 комбинаций с 16 полученными кодами. Например, так:

0000 — 000000,
0001 — 000001,
0010 — 000011,
0011 — 000101,
0100 — 000111,
0101 — 001011,
0110 — 001111,
0111 — 010001,
1000 — 010011,
1001 — 010111,
1010 — 011111,
1011 — 100011,
1100 — 100111,
1101 — 101111,
1110 — 110111,
1111 — 111111.

Количество нулей во всех 16 кодах — 44, единиц — 25. Соотношение $44/(44+25)=0,64$, что достаточно близко к теоретически оптимальному 0,68. В среднем на кодирование четырех битов надо затратить $44/16=2,75$ нулей и $25/16=1,5625$ единиц. Тогда цена четырех битов — $2,75+3 \cdot 1,5625=7,4375$ у.е. Цена одного бита в четыре раза меньше и примерно равна 1,86 у.е. А если бы мы не перекодировали данные, цена была бы равна 2 у.е.

Вторая идея перекодирования — использовать арифметическое сжатие “наоборот”. Известно, что если в исходных данных соотноше-

ние 0 и 1 отличается от 1:1, то можно их сжать с помощью арифметического метода. Размер сократится, а нулей и единиц станет примерно поровну. А у нас обратная ситуация — имеются исходные данные, где нулей и единиц примерно поровну (по условиям задачи), а надо получить данные большего размера, но с заданным соотношением нулей и единиц. Тогда можно считать исходные данные упакованными арифметическим методом и распаковать их, получив нужное соотношение 0 и 1 (а размер вырастет). При чтении же — наоборот, сжимаем считанные дан-

ные, и в результате получаем исходные данные. Вот такая идея. На практике не проверял :-).

Источники

1. FIDO-конференция RU.ALGORITHMS, тема "Восстановление изображения". — <http://groups.google.com.ru/groups?dq=&hl=ru&lr=&ie=UTF-8&inlang=ru&threadm=1043288021%40p5.f1826.n5020.z2.FidoNet.ftn&prev=/groups%3Fdq%3D%26num%3D25%26hl%3Dru%26lr%3D%26ie%3DUTF-8%26inlang%3Dru%26group%3Dfido7.ru.algorithms%26start%3D550>

2. Паучий глаз. — Компьютерра, 2001, №14, С.11. — <http://www.computerra.ru/offline/2001/391/8434/page3.html>

3. Приложите лицо к экрану... — Компьютерра, 2003, №14-15, С.10. — <http://www.computerra.ru/offline/2003/489/26108/page2.html>

4. В.Иванченко. ...Достать чернил и плакать. — Компьютерра, 2000, №14 С.12.

5. И написал в уголке... — Компьютерра, 2002, №25, С.6. — <http://www.computerra.ru/offline/2002/450/18645/>

6. М.А.Карцев. Арифметика цифровых машин. — М.: Наука, 1969.

С.РЮМИК,
г.Чернигов.

Интернет-калейдоскоп, freeware #4

*Лучше знать лишнее, чем ничего не знать.
(Сенека Луций Анней)*

Эмуляция, имитация, симуляция — все эти слова являются близкими по смыслу, хотя и различными по звучанию. Их называют частичными синонимами (от греч. "synonymos" — одноименный). "Частичными" — поскольку между ними существуют семантические различия, особенно заметные в компьютерной информатике.

Эмуляторы игровых приставок

Согласно словарю технических терминов [1], слово "эмуляция" (emulation) относится к однородным системам, составленным из одного и того же материала. Например, компьютер, состоящий из металла и кремния, не может эмулировать человека, состоящего из белковых органических соединений, и наоборот.

При эмуляции одна система выполняет функцию другой, но, как правило, с меньшей скоростью или с ограниченными возможностями. Одно из дополнительных требований — работа в режиме реального времени. В частности, внутрисхемные эмуляторы микроконтроллеров выполняют их функции, строго придерживаясь временных соотношений.

С программами-эмуляторами знакомы любители ретро-игр. Эмуляции подвергаются: игровые автоматы, игровые приставки, домашние компьютеры, китайские "Тетрисы" и даже

программируемые карманные калькуляторы... Среди игровых приставок выделяются эмуляторы Dendy, SEGA Mega Drive-II, PlayStation, имеющие широкое хождение в странах СНГ.

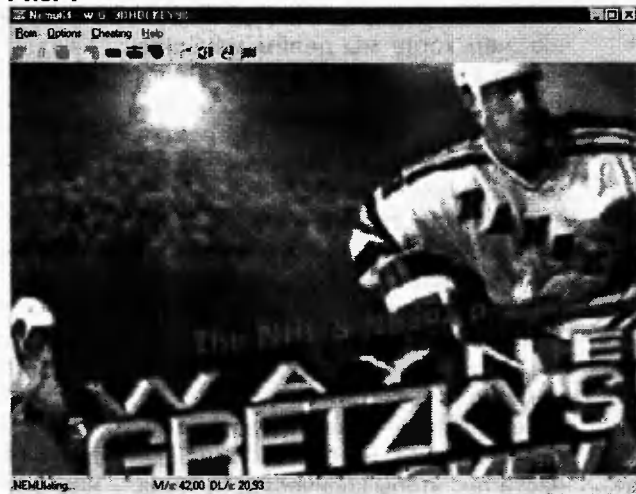
Более экзотические эмуляторы, такие как SuperNES, Turbo Grafx-16, Nintendo-64, позволяют оценить качество игр, распространенных в США и Японии (рис.1).

Требование к компьютеру можно определить эмпирической формулой

$$F_{cpu} > (15...25) \cdot F_{con}$$

где F_{cpu} и F_{con} — соответственно тактовые частоты компьютера и игровой приставки (консоли). Заявления о более низких частотах компьютера обычно носят рекламный характер. Чудес не бывает — для того чтобы без

Рис. 1



Игровая приставка	Название эмулятора, его размер	Компьютер (минимум)	Средний размер файлов игр (zip)
Atari Jaguar	Project Tempest, 384 Кб	Pentium-1000	1...4 Мб
Dendy (NES)	FCE Ultra, 295 Кб, VirtuaNES, 350 Кб	Pentium-133	0,1...0,6 Мб
Nintendo-64	Nemu, 629 Кб, Project 64, 973 Кб	Pentium-1000	2...29 Мб
SEGA Master System	FreezeSMS, 1 Мб, MEKA, 654 Кб	Pentium-166	0,05...0,3 Мб
SEGA Mega Drive-II (Genesis)	Gens, 380 Кб, Kega Lazarus, 236 Кб	Pentium-450	0,5...2,3 Мб
Sony PlayStation	EPSXE, 279 Кб	Pentium-800	CD
Super NES	SNES9x, 1 Мб, ZSNES, 455 Кб	Pentium-300	0,4...1,6 Мб
Turbo Grafx-16	Magic Engine, 438 Кб	Pentium-200	0,05...0,4 Мб

особых замедлений смотреть эмулируемое изображение и полноценно слушать музыку, необходим как можно более скоростной компьютер. В таблице приведены названия лучших эмуляторов переносных игровых приставок, последние версии которых и игры к ним можно найти на сайтах <http://www.emu-russia.km.ru>, <http://www.emu-russia.net>, <http://tv-games.narod.ru/sega/index.htm>.

Среди перечисленных программ отсутствуют эмуляторы игровых приставок второго поколения: Dreamcast, PlayStation 2, X-Box, GameCube, хотя в Интернете можно найти на них ссылки. Дело в том, что все они пока находятся на начальном этапе развития. Максимум, на что они способны — показывать экранные заставки со скоростью несколько кадров в секунду. Нетрудно подсчитать, что для их полноценной эмуляции потребуются компьютеры с тактовой частотой 4...15 ГГц, черед которых еще не наступил.

Ключевые слова для поиска: эмуляторы игровых приставок, emulation, consoles.

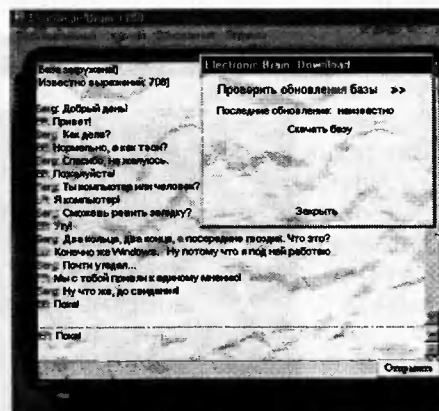
Имитатор искусственного интеллекта

Процесс имитации (imitation) неразрывно связан с подражанием [1]. Человек может петь как соловей, а компьютер — «разговаривать» как человек. Философский вопрос «Может ли машина мыслить?», опять таки, упирается в проблему имитации компьютером человеческих эмоций. При этом девять знаменитых возражений против машинного интеллекта, которые блестяще проанализировал Алан Тьюринг (<http://archive.1september.ru/inf/2000/2/art/alan1.htm>), до сих пор остаются в центре внимания. Суперматчи 2002...2003 гг. между сильнейшими шахматистами планеты Г.Каспаровым, В.Крамником и компьютерными программами Deep Junior, Deep Fritz показали реальность имитации шахматного озарения далеким и точным машинным расчетом.

В повседневной работе с компьютером программы-имитаторы встречаются достаточно часто — настольные игры (имитация логического мышления), программы обработки голосовой информации (имитация голоса), машинные собеседники (имитация искусственного интеллекта). В последнем слу-

чае человеку предоставляется возможность пообщаться с компьютером «тет-а-тет». Русскоязычные программы-собеседники (на сайте Инги Кеес <http://netnotes.narod.ru/index.html> их собрано более 30), как правило, самообучаемы, имеют чувство юмора и довольно быстро приспосабливаются к смене партнера. «Патриархом» является программа «Diala», разработанная в 1982 г. На рис.2 приведен образец

Рис. 2



реального диалога с машиной, который любой может повторить, скачав программу «Electronic Brain» (автор — Роман Ефимов, г.Уфа, 17 лет, <http://electrbrain.on.ufanet.ru/electrbrain1300beta02.zip>, 444 Кб).

Ключевые слова для поиска: имитация, искусственный интеллект, программа-собеседник.

Симулятор автомобильных гонок

Технический термин «симуляция» (simulation) относится к случаю, когда одна система моделирует поведение другой, имея на входе одни и те же данные, а на выходах — идентичные или подобные [1]. Это эквивалентно «черному ящику» с неизвестной внутренней структурой, причем его начинка может быть разнородной, а не одинаковой, как при эмуляции.

Например, водителей в автошколе обучают на тренажерах. Раньше они были полностью механическими, сейчас — в комплексе с компьютерами,

а в будущем — с проекционными экранами и голографическими изображениями. И в первом, и во втором, и в третьем случае входными данными для системы регулирования служат угол поворота руля и сила нажатия педалей. Выходными данными являются зрительные образы дорожной ситуации, на которую так или иначе реагирует водитель.

Геймерам известны игровые симуляторы: авиационные, велосипедные, парусные, лыжные, автомобильные. В домашних условиях многие предпочитают практиковаться в искусстве вождения автомобиля. Из материальной части понадобятся игровой руль с педалями и компьютер с программой-симулятором. Вопреки устоявшемуся мнению, что автосимуляторы бывают только фирменные, платные, в Интернете с 2001 г. успешно разрабатывается freeware-проект GeneRally (авторы — Hannu Rabina, Jukka Rabina, Финляндия, <http://generally.racesimcentral.com/generally104.zip>, 1,17 Мб).

На рис.3 изображена одна из 16 встроенных трасс. Автогонки предусматривают до 6 игроков с выбором различных марок машин. Проект имеет многоязычную поддержку и интернациональный форум.

Рис. 3



Ключевые слова для поиска: игровой симулятор, simulation, компьютерное моделирование.

Литература

1. The Blackwell Guide to Philosophy of Computing and Information. Edited by Luciano Floridi, August 2003, 392 pages, <http://www.blackwellpublishing.com/pci/downloads/Glossary.pdf>.

В помощь админу

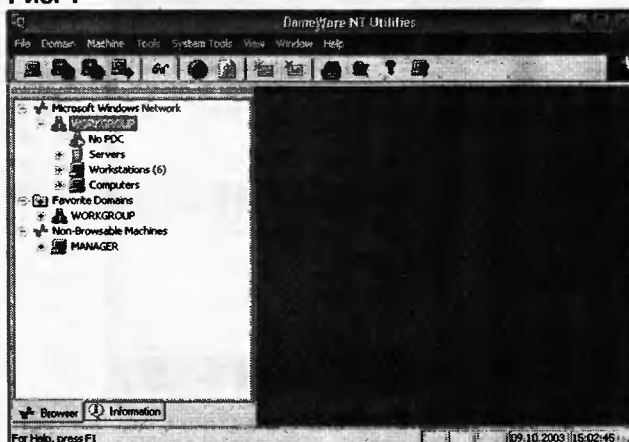
Я хочу рассказать об очень интересной и полезной программе, которая поможет администратору сети в его "нелегкой" работе. Допустим, вы устроились на работу в какой-нибудь игровой клуб, где имеется сеть из N-го количества компьютеров. Ваше заведение посещают разные люди — одни нормальные ребята, которые приходят для того, чтобы хорошо провести свободное время, но к вам могут прийти и "плохие ребята", возмнившие себя "кибер-королями" или хакерами, крякерами и т.п. Плохим ребятам, начитавшимся журнала "Хакер", не нужны игры, их желание — гадить везде, где бы они ни были. Вполне возможно, что целью таких субъектов может стать и ваш клуб. С появлением в компьютерном мире сетей возникла необходимость в обслуживающем их персонале, или системных администраторах (далее — "сисадмин" или "админ"). Сисадмины образовались как отдельная компьютерная группа людей, обслуживающих сети и все что с ними связано. Программисты разработали и написали "кучу" программного обеспечения для помощи админам. Подобного софта очень много, у каждого пакета, как всегда, есть свои "плюсы" и "минусы". В этой статье я хотел бы остановиться на обзоре пакета утилит DameWare.

Некоторое время тому назад мне в руки попала интересная программа, точнее, пакет из двух программ — DameWare NT Utilities. Недолго думая, я установил на свой рабочий компьютер данный софт и стал тестировать его на локальной сети. Что меня обрадовало? Да все! От обилия возможностей просто разбегаются глаза. Рассмотрим самые основные функции программы. Итак, вы — админ небольшой сети, к вам в сеть приходят совершенно разные люди, и как

вы поняли — если не позаботиться о безопасности и тотальном контроле над сетью, она "упадет". Пакет DameWare включает в себя две полезных утилиты — DameWare NT Utilities и DameWare Remote Control.

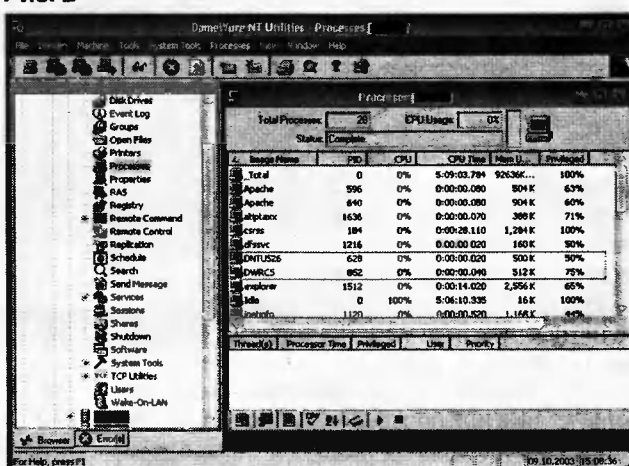
Рассмотрим обе программы более подробно. DameWare NT Utilities — это программа для администри-

Рис. 1



рования сети, имеющая удобный интерфейс, который с легкостью освоит даже самый начинающий администратор. После запуска программы мы видим слева навигационную панель, а справа — поле для свойств текущего объекта (рис.1). В навигации все предельно просто. Программа при старте ана-

Рис. 2



лизирует локальную сеть и выдает тип сети, разделы — Computers (Компьютеры), Workstations (Рабочие станции) и Servers (Серверы). Допустим, в вашей сети есть компьютер с именем "GamePC_1" (комп-

ютер из игрового клуба). Щелкаем по "Computers"—"GamePC_1". Появится подписок, состоящий из следующих пунктов:

- DiskDrives — информация о накопителях, установленных на компьютере;
- Event Log — журнал событий;
- Groups — список групп пользователей (Администраторы, Гости, Операторы и т.д.);
- Open Files — список открытых файлов;

- Printers — установленные принтеры;
- Processes — запущенные процессы;
- Properties — все о времени (время включения, текущее время и т.д.);
- RAS — с назначением данного пункта пока не разобрались;
- Registry — удаленная работа с реестром;
- Remote Command — отправка удаленных команд;
- Remote Control — удаленное управление;
- Replication — с назначением данного пункта пока не разобрались;

- Schedule — диспетчер задач;
- Search — поиск по компьютеру;
- Send Message — отправка сообщений (функция net send);
- Services — установленные сервисы;
- Sessions — текущие сессии на компьютере;

- Shares — список расширенных (доступных для сети) ресурсов;
- Shutdown — выключение компьютера;
- Software — установленные программы;
- System Tools — удаленная работа с системными программами (Disk Administration...);
- TCP Utilities — сетевые утилиты (Ping, Trace...);
- Users — список пользователей компьютера;
- Wake-On-LAN — "просыпание" по сети.

На рис.2 показан список процессов, запущенных на удаленном компьютере. Вот такой "неслабый" набор полезнейших функций в одной программе. Безусловно, некоторые функции могут и не понадобиться

(например, Wake-On-LAN или Shutdown). Что нам нужно для администрирования компьютеров игрового клуба? Во-первых — установить права, во-вторых — политику и в-третьих — разрешения. Все это умеет делать наша программа.

Рассмотрим пример. Необходимо создать группу пользователей, которые будут иметь доступ к главному серверу только в папку "Games". Сначала выбираем в DameWare NT нужный компьютер и открываем "Groups". Нажимаем закладку "Add New Local Group" (Добавить новую локальную группу), и появится форма "New Local Group Properties". На этой форме мы должны указать имя группы (Group Name), описание группы (Description) и добавить к группе любого из текущих локальных пользователей. Жмем OK, и группа готова. Далее необходимо ограничить или установить этой "новоиспеченной" группе права. Раскрываем в навигационной панели System Tools и кликаем по User Manager. Если на удаленном компьютере стоит данная утилита, то запустится редактор пользователей, в котором можно установить все необходимые разрешения, т.е. провести локальную политику. В случае отсутствия данной программы можно осуществить "ручную локальную политику", воспользовавшись удаленным управлением реестром (System Tools → Registry Editor).

Итак, новая группа пользователей создана, и теперь необходимо дать этой группе доступ к папке Games на главном сервере. В DameWare NT открываем наш сервер в разделе Servers. Затем кликаем по "Shares" и по "Add Share" (Добавить расшаренный ресурс). Появится окошко, в котором необходимо будет указать сетевой путь к папке "Games" (Share Path), название расшаренного ресурса (Share Name) и комментарий в виде описания к папке (Comment). Также можно указать ограничение числа подключающихся пользователей (User Limit).

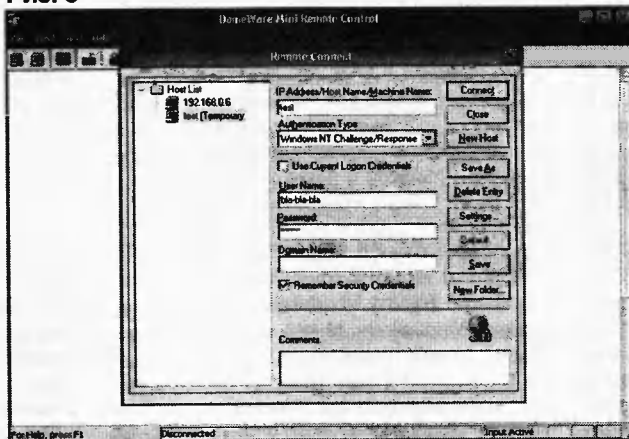
И наконец, необходимо установить разрешение для пользователей на эту папку. Для этого кликаем

по "Permissions" (Разрешения). В появившемся окошке добавляем новую группу пользователей (Add), выбираем из списка нашу новую группу и жмем OK. Далее устанавливаем права для этой группы — *только чтение* (Read).

Этот простой пример иллюстрирует, как можно легко и просто создавать новые группы пользователей, устанавливать им определенный уровень доступа и обеспечивать безопасность в локальной сети. Безусловно, полную безопасность подобными приемами обеспечить нельзя, поэтому в программе имеются дополнительные функции. Прямо из программы можно наблюдать за пользователями сети, помогать им, разрешать/запрещать доступ к отдельным ресурсам, посылать системные сообщения и многое другое. Для анализа безопасности сети в целом служит раздел Event Log, в котором записываются все важные события, происходящие на каждом отдельном компьютере.

DameWareNT позволяет осуществлять общий анализ состояния оборудования подключенных к сети компьютеров. Например, открыв раздел "Disk Drives", вы можете просмотреть общие сведения обо всех накопителях, установленных на удаленном компьютере.

Рис. 3



Как я уже упоминал, в вашей локальной сети могут действовать "вредители", но может быть и иная ситуация — например, ваша сеть расположена где-то на производстве, в каком-то подразделении или цехе, где работают "свои" люди, которым вы безусловно доверяете. В таком случае может быть незаменима утилита Remote Control (Удаленное управление). Рассмотрим

применение данной утилиты. Например, у одного из работников вашей фирмы (расположенной в десятиэтажном здании (и такое бывает ;)) никак не получается сканирование в FineReader. Что делает почти каждый нормальный админ? Он просто отправится к месту назначения и устранит неполадку. DameWare NT Remote Control предоставляет возможность управлять удаленной машиной, не отходя от своего компьютера. Программа работает элементарно просто. В запущенном DameWare NT Utilities выбираем нужный нам компьютер, раскрываем список его свойств и кликаем по Remote Control. Загрузится программа Remote Control (рис.3), которая автоматически предложит вам соединиться с каким-то удаленным компьютером, для этого вам необходимо просто указать IP-адрес (или имя компьютера в сети), логин (с правами администратора), пароль и, при необходимости, имя домена (Domain Name). Далее жмем "Connect", и Remote Control осуществляет прямое подключение к удаленной машине. После подключения появится новое окно, в котором будет отображаться содержимое экрана удаленной машины. Теперь вы можете спокойно руководить

удаленной машиной как своей собственной. На работу утилиты Remote Control наложены некоторые ограничения. Во-первых, для управления удаленной машиной (как я уже говорил ранее) необходимы данные аккаунта с администраторскими правами. Во-вторых, управление может осуществляться только машинами с NT-подобной операционной системой. Win9x и WinME для управления не годятся, вы просто получите отказ — "Удаленный компьютер отверг сетевое подключение!".

Подводя итог всему вышесказанному, хотелось бы отметить, что, к сожалению, DameWare NT Utilities является коммерческим продуктом и требует регистрации. Предоставляемая версия работает лишь 15 дней. Программу можно скачать с сайта разработчиков — <http://www.dameware.com>



ЛИСТАЯ СТРАНИЦЫ

Обработка текстов — один из основных видов работы, для которого используется в настоящее время компьютер. Источники текстов — самые разнообразные, но во многих случаях это печатные оригиналы. Набирать текст вручную очень трудоемко, более эффективно использовать сканер и затем **распознавать текст с помощью программы оптического распознавания текстов (OCR)**. О последних версиях пяти таких программ, позволяющих работать с кириллицей, рассказал Ф. Саложников в статье «Фолианты в мегабайты» (CHIP, №11/2003, С.120...124).

Это отечественные программы CuneiForm и FineReader, продукт бельгийского происхождения Read I.R.I.S., а также два пакета от американской компании ScanSoft, являющейся одним из подразделений корпорации Xerox, — OmniPage и TextBridge Classic. Все эти программы, за исключением OmniPage, часто используются как модули в продуктах других разработчиков. Например, Corel OCR&Trace является не чем иным, как CuneiForm, а фирма NewSoft включает в свои пакеты Presto! PageManager и SmartPanel for Scanner облегченную версию FineReader Sprint. У всех программ есть свои сильные и слабые стороны. Кроме того, каждой из них присуща своя уникальная особенность, отсутствующая у конкурентов.

Программа CuneiForm (цена — 120 USD в версии Professional и 249 USD в версии Master) поддерживает работу с 15 языками, распознает двуязычные русско-английские тексты. Есть версия с русским интерфейсом. Главными ее достоинствами являются простота в освоении и надежность в работе. Любой неподготовленный пользователь может начинать работу с программой практически сразу. Очень хорошее качество сопроводительной документации, которую, к тому же, можно загрузить с сайта разработчиков в виде PDF-документа.

Зависания CuneiForm крайне редки. Среди пользователей меньше всего жалоб на то, что эта программа не работает с тем или иным сканером.

В поставку программы входит полнофункциональный текстовый процессор, с помощью которого удобнее править распознанный текст, чем со встроенными редакторами других программ распознавания. Имеется довольно мощная система контроля правописания. CuneiForm довольно бережно относится к иллюстрациям в документе, в то время как другие программы при их сохранении используют сильную компрессию с потерями, которая самым пагубным образом сказывается на их качестве.

Недостатком CuneiForm по сравнению с FineReader и Recognita OCR является то, что она уверенно распознает лишь документы высокого и среднего качества. Использовать эту программу для работы с машинописным или газетным текстом практически невозможно. Единственное, в чем CuneiForm опережает другие пакеты, так это в обработке цветного текста на цветном фоне. Не предусмотрено распознава-

ние с обучением. Кроме того, CuneiForm существенно отстает от FineReader по количеству поддерживаемых национальных алфавитов. То же самое относится и к количеству поддерживаемых форматов графических файлов. CuneiForm оказалась единственной программой, «не знающей» о существовании PNG и, наряду с TextBridge, не позволяющей распознавать содержимое PDF-документов.

FineReader (цена — 129 USD в версии Professional и 259 USD в версии Corporate) — известная и популярная программа. Прежде всего ее ценят за то, что она оптимальным образом сочетает простоту и удобство в обращении с великолепным качеством распознавания, особенно если документы низкого полиграфического качества. FineReader лучше всех задает настройки сканирования и сегментации распознаваемого документа, и вмешательство пользователя бывает необходимо довольно редко. Автору приходилось сегментировать вручную лишь иллюстрации, которые представляли собой внешний вид диалоговых окон программ. С определением блоков в тексте с многоколоночной версткой FineReader почти всегда справлялась самостоятельно. Из плюсов отмечается очень гибкая настройка экспорта в различные файловые форматы. Другим достоинством является правильная работа с подстрочными нижними индексами.

Программа позволяет работать с документами на 177 языках. Уникальной особенностью является поддержка практически всех алфавитов России и СНГ, за исключением грузинского. При этом FineReader позволяет распознавать документы с произвольным количеством и сочетанием языков. И только FineReader может правильно распознавать химические формулы. Пока способности в этой области у FineReader весьма ограничены, но у других OCR-приложений нет и этого. Еще одним достоинством является правильная обработка документов с фоновыми рисунками. Это важно в большей степени для корпоративных пользователей (распознавание реквизитов денежных знаков и ценных бумаг), но может быть полезно и рядовым потребителям, которым приходится много работать с иллюстрированными изданиями.

Однако имеется много жалоб на неустойчивую работу программы, особенно в среде Windows 9x. Много шума наделала также неработоспособность версий 4 и 5 в Windows XP (хотя они и вышли до нее) без использования специальной программной заплатки или ручного редактирования системного реестра. FineReader производит «недопустимую операцию» чаще, чем любой из конкурирующих пакетов. Очень часто программа отказывается принимать данные со сканера. Подобные проблемы отмечены со многими моделями от Hewlett-Packard, некоторыми сканерами Mustek с USB-подключением и оборудованием от ряда других производителей. В частности, серьезные проблемы испытывали владельцы ручных сканеров. Для решения этих задач надо было загружать и устанавливать программные патчи, причем довольно увесистые. Иногда после инстал-

ляции FineReader требуется переустановка программного обеспечения сканера.

Правда, разработчики не стоят на месте. Например, в новейшей версии 7.0 наконец решили проблему снижения качества распознавания в том случае, если образы документов загружаются не со сканера, а в виде файла на диске. Плюс ко всему, существенно быстрее стал производиться импорт файлов в формате PDF.

Read I.R.I.S. (цена — 129 USD) является единственной массовой программой распознавания текста, которая позволяет работать с документами на восточных языках с ориентацией текста справа налево или использующих слоговое письмо. Для этого достаточно загрузить специальный модуль с официального сайта.

Программа уверенно распознает документы со сложной версткой, содержащие таблицы и иллюстрации. Сегментацию документа можно доверить самой программе, а можно и провести вручную. Кроме того, поддерживается распознавание с обучением.

Read I.R.I.S. обладает самым высоким быстродействием из всех рассматриваемых программ. В среднем на обработку одной страницы уходит около 40 секунд (у других программ — до двух минут). Read I.R.I.S. также очень быстро импортирует PDF-документы.

Имеются средства редактирования изображений: фильтр удаления «мусора», средства увеличения резкости, балансировки яркости, контрастности, гаммы. Это существенно больше, чем у других программ аналогичного назначения.

Недостатков у этого продукта три. Прежде всего, отсутствуют средства контроля распознавания. Для чтения документа надо загружать текстовый процессор. Автоматическая сегментация работает не очень хорошо, так что лучше ей не доверять документы со сложной версткой. Особенно плохо обстоит дело с текстом в несколько колонок. Единственным выходом в этом случае будет отказаться от полного сохранения документа при его экспорте.

Программа OmniPage (цена — 599 USD) является лидером на североамериканском рынке. Во многом продукт близок к отечественному пакету FineReader, в том числе по интерфейсу. Последняя версия OmniPage 12 позволяет распознавать текст на 115 языках с алфавитами на основе всех видов латиницы, кириллицы и греческого. Правильно распознаются многоязычные документы, в том числе и с произвольным сочетанием фрагментов на разных языках.

Важное достоинство OmniPage — бережное отношение к иллюстрациям. Во всяком случае, картинки не подвергаются сильному сжатию, а при преобразовании изображения в монохромное используется псевдосмещение.

Недостатков у этой программы довольно много. Наиболее значимы два: высокая цена и невозможность приобретения полной версии в России, во всяком случае, для Windows. А сокращенные версии OmniPage не позволяют распознавать кириллицу. Плюс ко всему, в OmniPage используется довольно слабый движок проверки орфографии. Если в слове больше



двух ошибок, варианта замены предложено не будет. Неудобна и громоздка процедура ручной сегментации и перепределения блоков. Программа путает переносы и дефисы.

Недостатки TextBridge Classic (цена — 26 USD) общие для всех облегченных систем распознавания текста. Более того, при распознавании полностью теряется оформление документа. Например, совсем игнорируются внедренные иллюстрации, многоколончатая верстка, информация о шрифтах.

Таблицы эта программа воспринимает как текстовые абзацы с табуляцией. Не предусмотрено средств редактирования распознанного текста. Для этого надо вызывать текстовый процессор. TextBridge Classic не позволяет проводить и ручную сегментацию документа. Плюс ко всему, распознавать цветные образы документов просто нельзя, а при обработке документов, отсканированных в полутоновом сером режиме, количе-

ство ошибок превышает все разумные пределы. Очень высока чувствительность к неправильно установленной яркости сканирования. Программа способна работать с документами только высокого и среднего качества. Машинописные копии, распечатки, сделанные на матричном принтере, газетный текст, ксерокопии TextBridge Classic распознает очень плохо, во всяком случае, по сравнению с FineReader Sprint. К большому сожалению, программа не может распознавать текст в двуязычных документах. При использовании TextBridge Classic в Windows XP возможен целый ряд проблем. Отмечают, например, аварийное завершение работы при использовании фирменного программного обеспечения сканера Umax Astra 4500.

Так что этот продукт — для очень невзыскательного пользователя. Применять Xerox TextBridge Classic как основную программу распознавания нельзя.

Подводя итоги, автор отмечает, что у каждой из рассмотренных программ имеются как

достоинства, так и недостатки. CuneiForm, например, можно смело рекомендовать пользователям, которым не требуется распознавание документов слишком низкого качества, но для которых на первом месте стоит простота в освоении и стабильность в работе. FineReader лучше всего распознает тексты, причем даже самого отвратительного качества, однако иногда огорчает неприятными сюрпризами при установке и работе.

Read I.R.I.S — самая быстрая и невзыскательная к системным ресурсам программа, для нее вполне достаточно 486-го процессора, к тому же, это почти единственный выбор для тех, кому приходится работать с текстами на восточных языках, с письменностью справа налево, но у нее, к сожалению, отсутствуют средства контроля распознанного текста.

Все последние программы вряд ли найдут своего пользователя. OmniPage способна отпугнуть слишком высокой ценой, а TextBridge Classic, несмотря на низкую цену, просто не выдерживает критики.

К офисному компьютеру нередко предъявляются достаточно мягкие требования, что позволяет значительно снизить его размеры и стоимость за счет сокращенной комплектации. Об одном таком устройстве рассказывает С. Андрианов в заметке "Слишком тонкий" (Мир ПК, №11/2003, С.36).

На смену избыточным с точки зрения офисного применения ПК все чаще приходят так называемые "тонкие клиенты", например, аппарат G-Max TA-1 производства компании Gigabyte. Устройство выполнено в виде располагающегося вертикально небольшого черного ящичка (размеры 64x230x213 мм). По набору разъемов он ничем не отличается от ПК форм-фактора ATX: имеются параллельный и два последовательных порта, разъемы PS/2 для клавиатуры и мыши, три звуковых, два USB и один сетевой разъемы, а также выход VGA. Отсутствует только игровой порт, но это и понятно — для игр тонкий клиент не подходит и не предназначен. Кроме того, передняя панель оснащена микрофонным входом, выходом на наушники и парой дополнительных гнезд USB. О работе в корпоративной сети можно судить по индикатору, также выведенному на переднюю панель.

Емкость современных винчестеров измеряется десятками и даже сотнями гигабайт. Многие из этих гигабайт содержат крайне важную информацию, стоимость которой может быть значительно больше цены носителя. Поэтому **вопросам надежности современных жестких дисков и возможностям восстановить поврежденные файлы уделяется достаточно много внимания**. Об этом, в частности, идет речь в статье О. Тарарникова "Восстановление информации на жестком диске" (КомпьютерПресс, №11/2003, С.48...51).

Жесткий диск — одно из самых ненадежных устройств в компьютере, поскольку, кроме сложной электроники, содержит непрерывно работающие механические части. И даже если "электронных" проблем не

достаточно типична для ПК и аппаратная конфигурация: изделие Gigabyte выполнено на процессоре VIA C3 с тактовой частотой 800 МГц, системной плате с набором микросхем VIA VT8601T (PLE 133T)/VT82C686B и встроенными графическим (Trident), звуковым (AC'97) и сетевым (Realtek) контроллерами. Имеются два разъема DIMM для SDR SDRAM, один из которых занят модулем 256 Мб (8 Мб из этого объема используются видеоконтроллером), один разъем PCI. Рассматриваемый в заметке экземпляр не имел жесткого диска, CD- и флоппи-дисководов. Установка первых двух возможна, правда, только в slim-исполнении (для ноутбуков). Кроме того, внутри корпуса предусмотрено место для разъема PCMCIA, однако сам он отсутствовал.

Вместо жесткого диска к разъему IDE подключена флэш-карта емкостью 32 Мб Disk On Module, с нее и происходит загрузка ОС. Объем невелик, но без нее не обойтись, поскольку в ПЗУ на системной плате содержится только стандартная BIOS. На флэш-карте записан лишь загрузчик, выполненный на основе Linux 2.4.18. Все необходимое программное

возникнет, то со временем неизбежно изнашиваются механические детали, и тогда начинаются различные сложности. Самая распространенная проблема у винчестеров — это так называемые BAD-блоки. Их внезапное появление заставляет многих пользователей врасплох, и те не знают, что делать, если самый важный файл вдруг перестал читаться. Но, продолжая работать как ни в чем не бывало, они рискуют еще больше: винчестер вдруг начинает "сыпаться", а отказы следуют один за другим. Конечно, большинство проблем неискушенному пользователю в домашних условиях решить не удастся, однако необходимо хотя бы приблизительно представлять себе суть этих проблем и основные способы их устранения.

В потере информации не всегда вино-

обеспечение догружается с сервера Citrix MetaFrame, без связи с которым тонкий клиент работать не может.

При желании G-Max можно дооснастить до полноценного ПК, хотя при блоке питания мощностью 40 Вт на серьезное усиление конфигурации рассчитывать не придется. Видеоконтроллер поддерживает 16- и 24-разрядный цвет и способен работать в режимах 640x480, 800x600 и 1024x768 точек, что при использовании 15-дюймового ЖК-дисплея вполне достаточно.

Итак, среди плюсов — невысокая цена (220 USD) и небольшие габариты, возможность дооснастить более или менее стандартными комплектующими до полноценного компьютера начального уровня. Системные администраторы к числу достоинств отнесут также отсутствие локального жесткого диска и каких бы то ни было сменных носителей, т.е. всего того, что способно помешать тотальному контролю над пользовательским рабочим местом. Среди минусов — дороговизна и невысокая эффективность модернизации, аскетизм исходной конфигурации с учетом предназначения аппарата и низкой стоимости является, скорее, преимуществом.

Другие компоненты компьютера также не застрахованы от поломок, бывают сбои в программах и операционной системе, от перепадов напряжения в электросети серьезно страдает электроника, а современные системы защиты от несанкционированного доступа и компьютерных вирусов не гарантируют 100%-ной надежности. Во всех таких случаях информация может пострадать, и вам придется потратить много времени и сил на ее восстановление.

Накопитель из жестких магнитных дисков состоит из механической и электронной частей. Механическая составляющая винчестера — это диски с магнитным напылением, собранные в пакет и установленные на оси шпиндельного двигателя (от одного до четырех дисков) и блок головок



чтения-записи, заключенных вместе с дисками в пыленепроницаемый корпус.

С начала 90-х годов все выпускаемые носители обладают приводом блока головок, который поворачивается вокруг оси под действием магнитного поля. Такая конструкция позволяет достичь высокой скорости позиционирования, но требует создания сервосистемы с обратной связью для определения фактического угла поворота. Поэтому позиционирование блока головок относительно вращающегося диска производится по записанной на нем же сервоинформации. Подготовленная к работе поверхность диска содержит специальные сервометки — это так называемый низкоуровневый формат, включающий разметку секторов с их адресами и идентификаторами. Следует отличать этот формат от логического формата операционной системы, которая организует свою логическую структуру, базируясь только на значениях полей данных. В обычном пользовательском режиме физический формат преобразуется контроллером в стандартизованный логический формат, с которым работают BIOS компьютера и операционная система.

Электронная часть жесткого диска расположена в основном на плате управления. Этот контроллер представляет собой специализированный микрокомпьютер, состоящий из управляющего процессора с памятью, микросхем для управления двигателем и средствами кодирования и декодирования данных для сопряжения с внешним интерфейсом. Такой микрокомпьютер работает под управлением специального программного обеспечения, состоящего из нескольких модулей и образующего специализированную операционную систему. Рабочие программы контроллера частично хранятся в ПЗУ, а частично — на самих дисках в специально отведенной для этого служебной области, недоступной для пользователя. Работа накопителя без загрузки этих программ в память контроллера невозможна. На служебных цилиндрах хранится и другая необходимая для работы дискового накопителя информация. Считывание информации из служебной зоны производится при начальной инициализации накопителя, а также, по мере необходимости, в процессе работы.

При включении питания контроллер производит автоматическую парковку головок — перемещает их в специальную парковочную зону, не используемую для записи информации, где головки опускаются на поверхность дисков. Автоматическая парковка осуществляется за счет энергии вращения пакета дисков, при этом шпиндельный двигатель используется в качестве генератора.

Разработка интерфейса ATA (AT Attachment), называемого также IDE (Integrated Drive Electronic), была начата еще в конце 80-х годов как средство для подключения дисковых накопителей к шине ISA, и с тех пор он неоднократно модифицировался (ATA-2, 3, 4, затем ATA/ATAPI-5 и, наконец, 7). Шина ISA давно уже канула в Лету, но интерфейс ATA оказался настолько удобным и экономичным, что его стандарты для работы с портами и по сей день использу-

ются для управления самым многочисленным семейством жестких дисков.

В современных накопителях появились такие возможности как самодиагностика (S.M.A.R.T.), прямой доступ к памяти для ускорения обмена данными (DMA), понижение уровня шума (AAM), установка пароля на доступ (Security Set), а также многие другие, и все они входят в новый стандарт ATA. Например, программные комплексы для подробной диагностики и точной настройки накопителей могут выдавать довольно внушительные списки параметров, однако не все фирмы-производители реализуют указанные возможности, и только компания IBM разработала утилиту IBM Features Tool, которая включает все ATA-функции. Поэтому, если у вас нет специализированной утилиты от производителя диска, можно воспользоваться универсальной программой MHDD, которая позволяет использовать ATA-команды на всех жестких дисках от любых производителей. Программа работает с диском непосредственно через порты и в DOS, и под управлением Windows. Причем ее можно запускать и завершать в любой момент.

Одной из наиболее вероятных причин возникновения проблем при работе винчестера является плохой контакт в разъеме питания или некачественный блок питания в компьютере, который не обеспечивает необходимой стабильности питающих напряжений для всех устройств. И чем более емкими и быстрыми становятся винчестеры, тем острее встает эта проблема.

Современные накопители представляют собой весьма точные и тонкие электронно-механические устройства, которые потребляют довольно большие токи, особенно в момент раскрутки пакета дисков. Понятно, что нестабильность электропитания может привести к серьезным неисправностям. Поэтому, в частности, строжайше запрещено производить любые переключения на работающем компьютере и подключать какие бы то ни было устройства, ибо даже безобидное на первый взгляд устройство может вызвать скачок питания в бортовой сети компьютера.

Автор рекомендует не пренебрегать резервным копированием наиболее важных файлов, которые сложно восстановить в случае потери информации, и по возможности проводить резервное архивирование всей системы. Он также отмечает, что опыт борьбы с такими вирусами как CIH показал, что использование файловой системы NTFS в плане надежности хранения и возможности восстановления данных предпочтительнее, чем FAT32, а тем более, чем FAT16.

Не следует использовать длинные имена файлов и каталогов, а также символы национальных алфавитов, в том числе русского. Это затрудняет операции с такими файлами в неработоспособной операционной системе и требует использования специального программного обеспечения для их надежного восстановления.

Время от времени необходимо проводить дефрагментацию файлов, так как вероятность успешного восстановления ин-

формации выше, если она хранится в менее фрагментированном виде. Кроме того, регулярное использование программ дефрагментации позволяет минимизировать время обращения к жесткому диску при работе. Если же избежать фрагментации невозможно, то даже такое "резервное копирование" как архивация файла или каталога на тот же диск, облегчит процесс восстановления информации с неисправных накопителей, поскольку куда легче восстановить нефрагментированный архив, чем большое количество фрагментированных рабочих файлов. К тому же, нефрагментированные файлы также успешнее восстанавливаются в случае их ошибочного удаления.

Почти все винчестеры, выпущенные после 95-го года, имеют систему оперативного наблюдения за своим состоянием — S.M.A.R.T. (Self Monitoring And Reporting Technology), позволяющую в любое время оценить такие важные параметры накопителя как количество отработанных часов, количество возникших в процессе чтения/записи ошибок и многое другое. Первые винчестеры, оснащенные этой системой, имели всего несколько S.M.A.R.T.-атрибутов, затем был разработан стандарт S.M.A.R.T.-II, а число атрибутов увеличилось, и с этого момента в большинстве накопителей появилась такая функция как внутренняя диагностика и самоконтроль. Эта функция основана на проведении серии автономных внутренних тестов, которые можно запустить стандартными ATA-командами. В результате постоянно диагностируются состояние механики накопителя, поверхности дисков и многие другие параметры. После выполнения тестов накопитель обновляет показания во всех S.M.A.R.T.-атрибутах в соответствии со своим текущим состоянием. Выполнить тесты S.M.A.R.T. можно с помощью вышеупомянутой программы MHDD. Подробно ознакомиться с новинками технологии S.M.A.R.T. можно, в частности, на сайте <http://yusoft.hotbox.ru/techdoc/storage/smartdoc.htm>.

Недавно появилась спецификация S.M.A.R.T.-III, в которой имеется не только функция обнаружения дефектов поверхности, но и возможность их восстановления, то есть накопитель самостоятельно сканирует поверхность, выявляя нестабильные секторы, и при их обнаружении переносит данные в резервную область. Нестабильными называются такие участки поверхности, где еще не сформировался явно выраженный дефект, но уже заметны проблемы со скоростью чтения. Скорость замедляется по той причине, что сектор не читается контроллером с первого раза, и диск вынужден делать несколько оборотов, пытаясь прочитать его без ошибок. Если прочитать данные все-таки удастся, то контроллер диска ничего не сообщая операционной системе, и ошибка остается незамеченной до тех пор, пока на этом месте не возникает настоящий BAD-блок. Как назло, тут же выясняется, что именно на этом месте хранился очень важный файл, причем в единственном экземпляре, и что спасти его уже нельзя. Поэтому-то диски и



нужно периодически тестировать. Раньше это можно было делать посредством программ Scandisk или Norton Disk Doctor в режиме тестирования поверхности или с помощью специальной утилиты, работающей независимо от файловой системы и умеющей выявлять намечающиеся BAD-блоки, замеряя время чтения каждого сектора. Теперь этим занимается S.M.A.R.T.-технология самого диска, вследствие чего данные окажутся спасенными еще до того, как на этом месте возникнет BAD-блок.

Для просмотра S.M.A.R.T.-статуса жесткого диска используют программы, называемые S.M.A.R.T.-мониторами. Обычно вместе с этими программами идет подробная документация с описанием всех соответствующих атрибутов.

Что делать, если винчестер не работает: компьютер зависает при обращении к нему, он не определяется BIOS, на команды не реагирует и т.д. Во-первых, следует выяснить, раскручивается ли механическая часть накопителя при включении питания. Если нет, нужно проверить разъем питания и напряжение на нем. Если же питание в порядке, то, видимо, это неисправность платы управления. Неисправными могут оказаться и цепи питания, и управление двигателем, и любая из схем, связанных с управляющим процессором и микроконтроллером (процессор, кроме всего прочего, управляет запуском двигателя и стабилизацией скорости его вращения). Иногда неисправную деталь на управляющей плате можно определить визуально, найдя сгоревший от перегрузки или перегрева элемент. На старых дисках могут "залипнуть" головки, и тогда диск при включении питания загудит, пытаясь раскрутиться. Можно, конечно, попытаться "оживить" головки, слегка встряхнув диск, но только для того, чтобы срочно переписать данные, так как механические узлы имеют свойство необратимо изнашиваться. Если же диск относительно новый, то подобное бывает, когда из-за тряски при транспортировке головки выйдут из парковочной зоны.

Если же двигатель раскручивается, но при его работе явственно слышны посторонние звуки (громкие щелчки, скрежет, свист, вой и т.д.), то проблема явно заключается в механике или в управляющей электронике, то есть накопитель неисправен, причем неисправность серьезная. А если

при включении питания двигатель нормально раскручивается и винчестер инициализируется, но при определении винчестера в BIOS выдаются неверные или непонятные параметры, то следует проверить IDE-шлейф и внимательно осмотреть разъемы (на винчестере и материнской плате) на предмет отсутствия контакта.

В том случае, когда все разъемы и шнур исправны и контакты надежны, неверное определение параметров или их отсутствие означает неисправности в электронике накопителя (к неправильной реакции на команды может, конечно, привести и разрушение служебной информации диска, но это маловероятно). Исправный IDE-винчестер, независимо от наличия на нем логического формата, должен быть доступен в системе как физическое устройство. Как правило, устранить подобные неисправности в домашних условиях невозможно, так что придется обращаться к специалистам. Более глубокая диагностика и последующий ремонт обычно предполагают в таких случаях наличие специального оборудования.

Если же определение в BIOS происходит нормально, но диск отказывается работать в операционной системе, то можно попробовать восстановить данные, но при этом не рекомендуется пользоваться стандартными утилитами восстановления, которые могут изменить содержимое восстанавливаемого диска. Постарайтесь не допустить запуска утилит ремонта файловой системы типа CHKDSK или SCANDISK, которые могут запуститься автоматически при перезагрузке компьютера, но при серьезном характере повреждений они только ухудшат ситуацию. Однако если запуск подобной утилиты уже произошел, то дайте ей доработать до конца, поскольку отключение питания компьютера во время работы таких утилит восстановления может привести к серьезным повреждениям файловой системы.

В данной ситуации можно попробовать воспользоваться только такими программами, которые переписывают уцелевшие файлы на другой диск, не изменяя при этом содержимое исходного диска (например, OnTrack EasyRecovery, R-Studio NE или GetDataBack). Возможно, что, выполняя анализ разрушенного раздела диска, они

могут сделать ошибочные заключения относительно геометрии дискового раздела и характера повреждений, выберут не то "лекарство" и в результате не справятся с проблемой, но хотя бы не повредят структуру исходного диска, поэтому его еще можно будет восстановить другими средствами. Именно по причине необратимости последствий столь опасно использовать утилиты Microsoft CHKDSK, Norton Disk Doctor и пр., которые изменяют содержимое исходного диска.

По той же причине не записывайте на поврежденный диск новые файлы. А если вы случайно удалили файл с диска, то ни в коем случае не выполняйте такую операцию как сохранение для других файлов, ибо они могут быть перезаписаны поверх стертого файла, и его восстановление станет невозможным. Не выполняйте и дефрагментацию поврежденного диска, так как секторы, содержащие информацию из стертых файлов, будут перезаписаны данными других файлов. Недопустима также любая установка или переустановка программ на поврежденном диске.

Что же касается утилит, сохраняющих аварийный диск в исходном состоянии, то неправильная автоматическая диагностика и ошибки в алгоритмах восстановления могут привести к тому, что вы не сможете восстановить исходные файлы, или они будут восстановлены в искаженном виде (например, многие утилиты не могут восстанавливать имена файлов и каталогов, содержащих символы кириллицы).

И, опять же, если из диска доносится стук или посторонние звуки, то немедленно завершите работу операционной системы и выключите компьютер. Стук может означать, что из строя вышли механические детали или электронные схемы, расположенные в герметичном корпусе диска. Правда, иногда удается восстановить данные, информация о расположении которых оказалась полностью утерянной. Не исключено, что специалисты смогут отремонтировать такой диск в специальной "чистой комнате", где гарантируется отсутствие пыли. В домашних же условиях вскрывать корпус винчестера ни в коем случае нельзя. К тому же, его вскрытие требует далеко не всегда, а вероятность успешного ремонта вскрытого накопителя значительно меньше.

.....>>>

Новому лазерному принтеру компании OKI, цена которого составляет всего 205 USD, посвящена заметка А.Дмитриева "Все ниже и ниже ..." (Мир ПК, №12/2003, С.37).

Речь идет о модели светодиодного принтера B4200L. Скорость печати составляет 18 стр./мин. На выход первой страницы аппарат тратит 10 с, на прогрев после включения — 35 с. То есть производительность для принтера категории SOHO — просто рекордная. Радует обширный набор настроек, среди которых имеются два режима экономии тонера. Принтер способен работать в операционных системах DOS, Windows, Mac OS, UNIX и Linux. Максимальная месячная нагрузка — 15 тыс. страниц, ресурса фото-барабана хватит на 25 тыс. страниц, а картриджа

с тонером — на 2,5 тыс. Несмотря на то что в спецификации указана максимальная плотность бумаги 120 г/м², протестированный экземпляр легко справился и с печатью на листах плотностью в 200 г/м². Входящая в комплект поставки программа Oki Status Monitor еще больше расширяет возможности настроек и предоставляет информацию об оставшихся ресурсах барабана и картриджа.

Вместе с тем, в целях удешевления, разработчикам приходится идти на ряд компромиссов, в результате чего обнаруживаются некоторые изъяны. Так, у модели B4200L не слишком удачно выполнен лоток для бумаги. Нельзя не отметить и весьма сильный шум при работе принтера, которым, видимо, приходится распла-

чиваться за высокую скорость печати.

Тест на качество печати B4200L прошел с оценкой "хорошо". Четкий вывод (символы различимы вплоть до размера в два пункта), отличная плотность заполнения, равномерный градиент серого. Единственный минус — паразитные вертикальные полосы на заливках в сером клине, хотя это заметно только при печати больших фотографий.

Главное достоинство этой модели OKI в том, что столь мощный аппарат покупатель может получить по очень привлекательной цене, причем в комплекте с двумя картриджами на 2,5 тыс. страниц (при покупке отдельно от принтера один картридж стоит около 30 USD). Таким образом, принтер не только дешев сам по себе, но и обеспечивает очень низкую себестоимость отпечатка.



А.КОРБИТ, А.ЩЕРБАКОВ,
г.Минск, БГУИР

Программирование в среде Visual C++ 6.0

Написание программы построения графиков

Цель этой статьи — изучить приемы построения графиков на основе SDI-приложения и познакомиться с режимами сопоставления координат.

1. Создание однодокументного приложения

Как вы уже знаете, по умолчанию мастер AppWizard создает однодокументное приложение (SDI) с поддержкой архитектуры типа «документ/вид». Это позволяет отображать данные различными способами. Основные возможности внутреннего представления данных реализует класс Cdocument, а класс CView реализует как отображение содержания документа, так и отображение любой графической информации.

Windows при работе с устройствами отображения использует три системы координат. При этом во всех системах ось X направлена слева направо, а ось Y — сверху вниз.

Так, в статье [1] работа с экраном проводилась с пикселями — единицами измерения, которые обозначают экранные (аппаратные) координаты. И если теперь распечатать построенное графическое изображение на лазерном принтере с высокой разрешающей способностью, то получим очень маленькую фигуру.

Практически все GDI-функции требуют задания размеров в логических координатах, которые Windows преобразует в физические координаты устройства вывода в зависимости от режима их сопоставления. При этом преобразование ассоциируется с контекстом устройства.

В Windows существует ряд режимов сопоставления логических и физических координат. Основные из них:

- MM_TEXT — 1 пиксел;
- MM_LOENGLISH — 0,01 дюйма;
- MM_HIENGLISH — 0,001 дюйма;
- MM_LOMETRIC — 0,1 мм;
- MM_HIMETRIC — 0,01 мм;
- MM_TWIPS — 1/1440 дюйма.

Перечисленные режимы называются режимами с фиксированным масштабированием. Кроме них, Windows предусматривает два режима с переменным масштабированием, которые позволяют как изменять масштабный коэффициент, так и переворачивать изображение вокруг оси. При использовании этих режимов пользователь определяет «свои», пользовательские координаты.

При этом, используя методы **SetViewportOrg** и **SetWindowOrg**, можно сдвинуть изображение по отношению к верхнему левому углу.

Метод **SetMapMode** устанавливает режим сопоставления (по умолчанию — MM_TEXT).

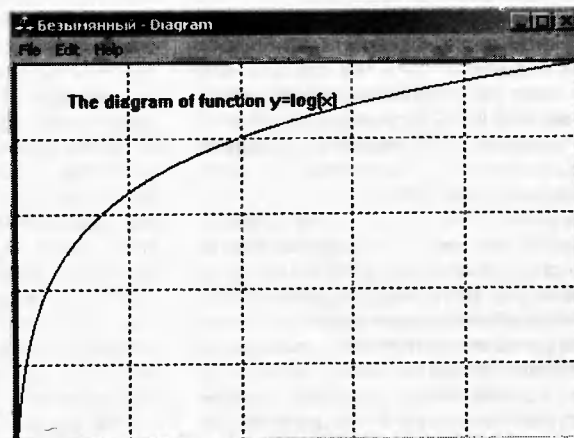
Метод **SetWindowExt** устанавливает размеры окна в логических единицах, а метод **SetViewportExt** — размеры области вывода в пикселях.

Для написания приложения, работающего с GDI-интерфейсом, достаточно использовать функцию **OnDraw** класса CView. В эту функцию передается указатель на уже созданный объект класса CDC, через который можно про-

изводить вызов функций рисования, таких как **Ellipse**, **Rectangle** и др. Функция **OnDraw** всегда вызывается, когда приложению приходит сообщение **WM_PAINT**, указывающее, что требуется перерисовать окно.

2. Пример написания программы

Необходимо построить график функции $\log(x)$ в диапазоне от 1 до 100 (см. рисунок).



Для решения задачи требуется в AppWizard выбрать тип проекта MFC AppWizard(exe). На первом шаге нужно установить опцию Single Document. Параметры на втором и третьем шаге следует оставить без изменений. На четвертом шаге необходимо отключить три опции: Docking Toolbar, Initial Status bar и Printing and Print preview, после чего нажать кнопку Finish.

Для того чтобы построить график, потребуется изменить функцию **OnDraw** как показано ниже:

```

////////////////////////////////////
// CDiagramView drawing
#include <math.h>
void CDiagramView::OnDraw(CDC* pDC)
{
    CDiagramDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    /// TODO: add draw code for native data here
    CRect r;
    GetClientRect(r); //определяем размер окна
    CString s;
    s.Format("The diagram of function y=log(x)");
    // заносим текст надписи
    int n=r.Width(); // присваиваем N значение ширины
    //в пикселях
    pDC->SetTextAlign(TA_CENTER); // центровка текста
    // относительно координаты x
    pDC->TextOut(n/3,30,s); // вывод текста
    // в левый верхний угол
    double x1=1,x2=100,x; // x1,x2 -начальное
    // и конечное значение
    double h=(x2-x1)/N; // вычисление шага.
    x=x1;
    //построение графика логарифма
    for(int i=0;i<N;i++)
    {
        double f=(log(x)-log(x1))/log(x2);
        double y=r.Height()-f*r.Height();
        pDC->LineTo(i,(int)y); // рисование линии
        x+=h;
    }
    CPen bluePen; // объявление объекта - "перо"
    bluePen.CreatePen(PS_DOT,1,RGB(0,0,255));
    // создание объекта "перо" - штирковое, толщина 1, цвет синий
    pDC->SelectObject(&bluePen); //выбор пера в контексте
    //устройства рисования

```



```

const int Nx=5, Ny=5;           //число линий сетки по
                                // горизонтали и вертикали
                                //рисование вертикальных линий сетки
for( i=1;i<=Nx;i++)
{
    int dx=i*N/Nx; //расстояние между линиями сетки
                    // в пикселах
    pDC->MoveTo(dx,0); //изменение текущей позиции пера
    pDC->LineTo(dx,r.Height()); // рисование линии
}

//рисование горизонтальных линий
for( i=1;i<=Ny;i++)
{
    int dy=i*r.Height()/Ny;
    pDC->MoveTo(0,dy);
    pDC->LineTo(r.Width(),dy);
}
}

```

3. Индивидуальные задания

По аналогии с описанным примером, необходимо создать приложение, которое строит график заданной функции в заданном диапазоне (в случае необходимости может потребоваться подобрать диапазон изменения аргумента для получения симметричного графика):

1. $y=\ln(x)$; диапазон изменения аргумента — 0,5...1,5.
2. $y=1-(e^x+e^{-x})/2$; диапазон изменения аргумента — 0,0...1,0.
3. $y=1-(e^x-e^{-x})/2$; диапазон изменения аргумента — 1,0...2,0.
4. $y=e^{-x}$; диапазон изменения аргумента — 1,0...2,0.
5. $y=1+\ln(x)$; диапазон изменения аргумента — 0,5...1,5.
6. $y=2-(1+2x^2)/e^x$; диапазон изменения аргумента — 1,0...2,0.
7. $y=x^2-2x+3$; диапазон изменения аргумента —

- 1,0...1,0.
8. $y=x^3-2x+1$; диапазон изменения аргумента — -1,0...1,0.
9. $y=x^2-4x+1$; диапазон изменения аргумента — 0,0...1,0.
10. $y=\ln(1/(1+2x+x^2))$; диапазон изменения аргумента — 0,5...1,5.

Литература

1. А.Корбит, А.Щербаков. Программирование в среде Visual C++ 6.0. Интерфейс графических устройств Windows. — Радиомир. Ваш компьютер, 2003, №10, С.21.
2. С.Холзнер. Visual C++ 6: учебный курс. — СПб.: Питер, 1999. — 576 с.
3. Н.Ю.Секунов. Самоучитель Visual C++ 6 (+ дискета). — СПб.: BHV-Санкт-Петербург, 1999. — 960 с.
4. К.Грегори. Использование Visual C++ 6. Спец.издание. — СПб.: Вильямс, 1999. — 864 с.
5. Ю.В.Тихомиров. Visual C++ 6 (+ дискета). — СПб.: BHV-Санкт-Петербург, 1999. — 496 с.
6. Майкл Дж. Янг. Visual C++ 6. Полное руководство (+ CD-ROM). — BHV-Киев, 2000. — 1056 с.
7. С.Гилберт, Б.Маккарти. Самоучитель Visual C++ 6 (+ CD-ROM). — М.: ДиаСофт, 2000. — 496 с.
8. К.Паппас, У.Мюррей. Visual C++ 6 для пользователя. — BHV-Киев, 2000. — 336 с.
9. К.Паппас, У.Мюррей. Visual C++ 6: руководство разработчика. — BHV-Киев, 2000. — 624 с.
10. И.Баженова. Visual C++ 6. 0. VISUAL STUDIO 98: Уроки программирования. — М.: Диалог-МИФИ, 2001. — 416 с.
11. Лэйси Дж. Visual C++ 6 Distributed. Сертификационный экзамен — экстерном. — СПб.: Питер, 2001. — 624 с.
12. А.Черносвитов Visual C++ 7: учебный курс (с дискетой). — СПб.: Питер, 2001. — 528 с.

Решение задач на деревьях м.ДОЛИНСКИЙ, г.Гомель.

и с помощью деревьев

Введение

Ранее [1, 2, 3] уже рассматривались некоторые способы решения задач на графах, в том числе, такие как метод Флойда, метод Дейкстры, поиск в глубину, поиск в ширину, построение минимального остовного дерева, построение максимального потока.

Данный материал посвящен методам и средствам решения задач на деревьях и с помощью деревьев. Дерево, являясь разновидностью графа, имеет множество эквивалентных определений. Для определенности возьмем в данном материале за базовое определение такое: "Дерево — это связный граф без циклов". Важно отметить также, что пустой граф, то есть граф, не содержащий ни вершин, ни ребер, также считается деревом. Это дополнение введено для удобства обобщения многих алгоритмов на деревьях.

В качестве логичной, по мнению автора, аналогии можно привести пример с факториалами:

$$N! = 1 * 2 * 3 * \dots * (N-1) * N,$$

где N — целое положительное число.

В то же время, $0!$ определен равен 1.

Рассмотрим основные свойства деревьев, вытекающие из приведенного определения, иллюстрируя их на примере дерева с 4 вершинами (рис.1):

1. Связность дерева обеспечивает наличие пути из любой вершины в любую.

2. Отсутствие циклов в дереве гарантирует *единственность* такого пути.

3. Количество ребер (E) и вершин (V) в дереве связаны соотношением: $E = V - 1$ (в примере — 4 вершины и 3 ребра).

Важным классом деревьев является дерево с предопределенным корнем и установленным отношением "предок-потомок" ("ancestor-descendant"). В [4] такие деревья называются "корневыми деревьями" (rooted tree). Вершины, которые имеют потомков, называются "внутренними", а вершины, не имеющие потомков, называются "внешними" или, чаще, "листьями".

Например, корневое дерево может иметь вид, соответствующий рис.2.

Здесь вершины 1 и 3 — внутренние, вершины 2 и 4 — листья. Заметим, что определением корневого дерева *запрещено* "множественное наследование". То есть, например, граф (рис.3) не является корневым деревом (если считать, что направленные дуги идут сверху вниз — от вершин 1 и 3 к вершине 2 и от вершины 2 к вершинам 4 и 5).

Рис. 1



Рис. 2

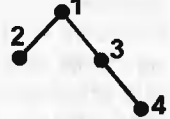
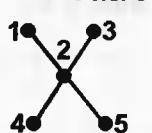


Рис. 3



У корневых деревьев свойство единственности пути должно быть переформулировано следующим образом: "Для любой вершины дерева существует единственный путь к ней из корневой вершины".

В дальнейшем мы будем употреблять название "дерево", имея в виду именно корневое дерево. Заметим, что корневые деревья активно используются в реальной жизни и программистской практике.

Очевидно, что каждый читатель, хотя бы на интуитивном уровне, многократно сталкивался с подобным отображением реалий. Прежде всего вспоминается "генеалогическое древо" ("Иоанн родил Якова, Яков родил Авраама и Моисея ..."). Заметим, что, в отличие от реальности, родителем в таком дереве выбирается только отец.

Другое часто употребляемое применение корневых деревьев — это оглавление книги: книга состоит из глав, главы состоят из пунктов, пункты состоят из подпунктов. При желании или необходимости детализацию можно и продолжить, например, так: подпункты состоят из абзацев, абзацы состоят из предложений, предложения состоят из слов, слова состоят из букв.

Обобщая такой подход, можно заметить, что дерево — удобное средство представления сложной системы, с помощью которого последовательно детализируются иерархические составляющие (компоненты) такой системы.

Приведем несколько примеров, которые важны для понимания условий и решений разбираемых далее задач.

Деревья отношений

1. Организация (партия), построенная по принципу единоначалия. Во главе организации стоит один человек, ему непосредственно подчиняются несколько руководителей. Каждому из них подчиняются свои руководители более низкого уровня, и так далее. Руководителю самого низкого уровня подчиняются несколько рядовых членов организации.

2. Ученые, имеющие совместные публикации с Эрдосом, получают "число Эрдоса", равное 1. Ученые, имеющие публикации с учеными с числом Эрдоса 1, получают число Эрдоса 2, ... Ученые, не имеющие публикаций ни с кем из тех, кто имеет установленное число Эрдоса, получают признак "infinity".

Деревья попиксельного представления плоских цветных образов

Деревья с четырьмя или менее дочерними вершинами применяются для свертки информации о графическом образе, представленном квадратом из $N \times N$ пикселей (где N — степень двойки). Цвет каждого пикселя кодируется собственным числом (в этой задаче — от 0 до 255) примерно так: если весь квадрат имеет один и тот же цвет, то для кодирования информации используется единственное число (дерево с одной вершиной). Иначе, квадрат разбивается на 4 равных квадратных части. Если какая-то из частей имеет единый цвет для всех пикселей в ней, то она кодируется этим числом. Для всех частей, не имеющих одинаковый цвет для составляющих ее пикселей, процедура деления квадрата на 4 подквадрата продолжается. Например, для такого образа (4×4 пиксела)

```
12 12 136 136
12 12 136 136
12 12 12 12
100 100 12 12
```

получается дерево, показанное на рис.4.

Деревья представления сложных композиций трехмерных объектов

BSP (Binary Space Partition) Trees — деревья двоичного разбиения пространства — используются для хранения информации о расположении объектов на трехмерной сцене относительно наблюдателя. Это полезно использовать в процессе прорисовки объектов, чтобы сначала нарисовать наиболее удаленные от наблюдателя объекты, затем — более близкие, и т.д.

Алгоритм построения BSP-дерева может быть описан следующим образом. Между наблюдателем и сценой проводится секущая плоскость (не пересекающая ни один из объектов). Тогда, в общем случае, часть объектов окажется между наблюдателем и секущей плоскостью, а часть — между секущей плоскостью и сценой. Процесс проведения таких секущих плоскостей может быть продолжен. После проведения каждой из таких плоскостей мы имеем дерево объектов, где иерархия внутренних вершин обозначает последовательность проведения секущих плоскостей, сами внутренние вершины хранят информацию о секущих плоскостях, а листовые вершины включают совокупности отображаемых объектов в одной, ограниченной проведенными секущими плоскостями области пространства.

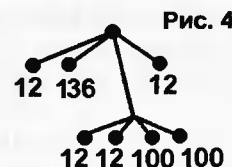


Рис. 4

Деревья кодирования символов

Хаффман предложил сжатие последовательности символов за счет кодирования символов переменным количеством битов: те символы, которые встречаются в этой последовательности символов чаще, кодируются меньшим количеством битов, а те символы, которые встречаются в этой последовательности символов реже, кодируются большим количеством битов.

Для проведения кодирования символов строится дерево, листьями которого являются пары "символ-частота встречаемости", а иерархия в дереве выстраивается так, чтобы:

- каждая внутренняя вершина имела **ровно** двух потомков;
- в первую очередь объединяются потомки, имеющие наименьшие частоты встречаемости;
- при объединении двух потомков их родитель (предок) получает частоту встречаемости, равную сумме частот встречаемости потомков.

Например, пусть в символьной строке из 100 символов встретились только символы a, b, c, d, e, f. Частота встречаемости: a — 45 раз; b — 13; c — 12; d — 16; e — 9; f — 5.

Тогда соответствующее дерево кодирования будет выглядеть как на рис.5.

Заметим, что каждая дуга дерева нагружена цифрой 0 (дуга к левому потомку) или 1 (дуга к правому потомку).

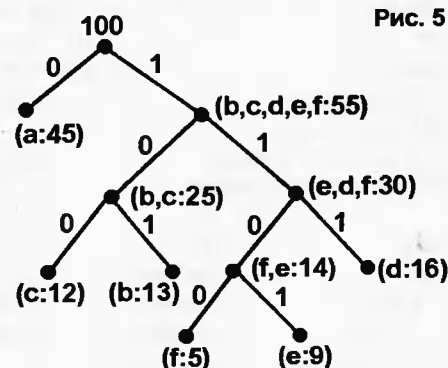


Рис. 5



Таблица кодировки получается при анализе пути по дугам дерева от корня к листу соответствующего символа: a — 0; b — 101; c — 100; d — 111; e — 1101; f — 1100.

Оценим степень сжатия.

Исходный текст состоял из 100 символов (a — 45, b — 13, c — 12, d — 16, e — 9, f — 5). Если закодировать символы способом, указанным выше (на a — 1 бит; на b, c, d — по 3 бита; на e и f — по 4 бита), то для представления исходной строки из 100 символов потребуется 224 бита: $45 \cdot 1 + 13 \cdot 3 + 12 \cdot 3 + 16 \cdot 3 + 9 \cdot 4 + 5 \cdot 4 = 224$.

При обычном (равномерном) кодировании, например, таком: a — 000 (3 бита); b — 001; c — 010; d — 011; e — 100; f — 101, потребовалось бы 300 битов ($100 \cdot 3$). Коэффициент сжатия $300/224 = 1,34$.

Деревья сортировки

Для примера рассмотрим дерево, показанное на рис. 6. Данное дерево является деревом сортировки, поскольку для любой вершины этого дерева выполняется следующее свойство: «в левом поддереве данной вершины находятся все числа, меньше числа в самой вершине (и только они), а в правом поддереве данной вершины находятся все числа, больше числа самой вершины (и только они)».

Заметим также, что каждая вершина такого дерева имеет не более 2 потомков.

Для поиска наименьшего элемента в этом дереве потребуется анализ всего трех вершин (10, 5, 1). Для поиска наибольшего элемента — анализ четырех вершин (10, 20, 25, 27). В то же время, для поиска минимального (максимального) числа в одномерном массиве из этих же 9 чисел обычным способом требуется анализ всех 9 элементов. Аналогичным образом сокращается трудоемкость ответа на вопрос «имеется ли в данном дереве заданное число?». Относительно несложно добавляется новый элемент в дерево сортировки с сохранением основного свойства дерева. И, наконец, обходом типа «левое_поддерево корень—правое_поддерево» выводится последовательность чисел в порядке возрастания.

Деревья сумм

Один пример деревьев сумм представлен выше при описании дерева, составленного для оптимального кодирования символов по методу Хаффмана. Другой пример — «битово-индексированное дерево» (Bit-Indexed Tree) — ввиду своей сложности, будет разобран позже, непосредственно при описании решения задачи «Мобильные телефоны».

Перечисление деревьев

Особый интерес представляет тема «перечисление деревьев». Алгоритмы подобного вида обычно отвечают на вопросы: «Сколько существует деревьев заданного типа?», «Каково следующее дерево заданного типа?». Ответы на такие вопросы интересны потому, что можно поставить деревья реальным объектам, и тогда мы получим ответы на вопросы типа «Сколько существует реаль-

ных объектов?» и «Какой следующий по порядку реальный объект?».

Например, в одной из задач рассматриваются деревья, в которых каждая из вершин имеет или *ни одного* или *ровно двух* потомков. Для нумерации используется следующая система:

- каждой вершине назначается число — количество листьев в поддереве, которое имеет эту вершину в качестве корневой;
- дерево кодируется последовательностью чисел, ассоциированных с корневой вершиной и всеми левыми потомками.

Например, дерево (рис. 7) кодируется последовательностью чисел 7, 4, 1, 2, 1, 1, 1. Символом «*» отмечены числа, вошедшие в кодовую числовую последовательность. Деревья упорядочены по лексико-графическому возрастанию представляющих их числовых последовательностей. Требуется по кодовой последовательности текущего дерева узнать кодовую последовательность следующего дерева.

В другой задаче рассматриваются все деревья, у каждой из вершин которых может быть *ни одной, одна* или *две* вершины. Они упорядочены: в порядке возрастания вершин в дереве; в случае равенства вершин в двух деревьях — в порядке возрастания количества вершин в левом поддереве, а в случае равенства у двух деревьев и вершин в левом поддереве — в порядке возрастания количества вершин в правом поддереве. По заданному номеру дерева требуется вывести в специальном виде само дерево.

Представление деревьев в памяти компьютера

K-ичные деревья (т.е. деревья, о которых заведомо известно, что они имеют не более K вершин) удобно хранить в одномерном массиве, используя индексы этого массива для выстраивания иерархии дерева. Каждой вершине дерева соответствует один элемент массива.

Индекс корневой вершины — 1. Индексы вершин-потомков корневой — 1, 2, ..., K. Индексы вершин-потомков вершины 1 — K+1, K+2, ... 2·K. Индексы вершин-потомков вершины 2 — 2·K+1, 2·K+2, 3·K и т.д.

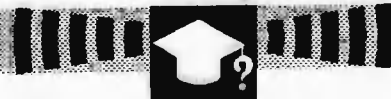
В общем случае для вершины, которая расположена в этом массиве на позиции i, индексы вершин-потомков таковы:

$$(i-1) \cdot K + 1, (i-1) \cdot K + 2, \dots, i \cdot K \quad \{(i+1) \cdot K + K\}.$$

Легко также найти и предка любой вершины.

Пусть вершина расположена в массиве на позиции i (или, другими словами, вершина имеет индекс i в массиве). Тогда индекс (расположение в массиве) вершины, которая является предком вершины i, есть $1 + (i \div K)$, т.е. увеличенное на 1 частное от целочисленного деления i на K. Например, пусть K=4, т.е. в дереве каждая вершина может иметь 0, 1, 2, 3 или 4 наследника. Нам нужно узнать позицию предка вершины, находящейся в массиве на позиции 23:

$$1 + (23 \div 4) = 6.$$



Предком вершины, которая находится в массиве на позиции 23, является вершина на позиции 6.

Действительно, выпишем для проверки последовательно потомков всех вершин до 6-й включительно:

```
1: 2 3 4 5
2: 6 7 8 9
3: 10 11 12 13
4: 14 15 16 17
5: 18 19 20 21
6: 22 23 24 25.
```

Для того чтобы отметить отсутствие потомков у какой-то из вершин, соответствующему элементу массива присваивается специальное значение (0, -1, пустая строка и т.д.)

Основное достоинство такого способа хранения деревьев — упрощение алгоритмов обработки.

Основной недостаток — неэффективное использование памяти для несбалансированных деревьев, т.е. деревьев, для которых количества потомков сильно различаются от предка к предку.

Если по ограничениям задачи массив может быть размещен в оперативной памяти, то такое представление дерева вполне приемлемо. Бывают задачи, в которых такое размещение невозможно. Тогда для хранения деревьев могут быть использованы другие варианты. Например, те, которые использовались ранее для хранения графов (в частности, списки дуг из вершин).

Приведем для примера один вариант хранения двоичного дерева (т.е. дерева, в котором у каждой вершины-предка имеется не более двух потомков), использующий 3 массива. Массив $A[i]$ служит для хранения содержимого вершин (количество элементов в нем должно быть не меньше максимального числа вершин дерева для данной задачи). Два специальных массива (Left и Right), такой же размерности, что и $A[i]$, служат для хранения ссылок на левого и правого наследника:

- Left[i] — указывает индекс в массиве A элемента, который является левым потомком элемента i;
- Right[i] — указывает индекс в массиве A элемента, который является правым потомком элемента i.

Один из элементов массива A объявляется корнем дерева (Root).

Порядок обхода деревьев

Фактически многие задачи для своего решения требуют обойти все вершины дерева на предмет исследования некоторых свойств. В литературе наиболее распространены 3 порядка обхода (traversal) деревьев — “pre-order”, “in-order”, “post-order”, в зависимости от того, в какой момент мы посещаем вершину-предок относительно посещения ее потомков.

“Pre-order traversal” — это порядок обхода, при котором вначале посещается (анализируется) вершина, затем ее левое поддерево, затем ее правое поддерево.

“In-order traversal” — это порядок обхода, при котором вначале посещается (анализируется) левое поддерево, затем сама вершина, затем ее правое поддерево.

“Post-order traversal” — это порядок обхода, при котором вначале посещается (анализируется) левое поддерево, затем правое поддерево, и только в последнюю очередь сама вершина.

(Продолжение следует)

А.КАЛЮТОВ,

E-mail: k90012@hotmail.com

Практическое введение в технологии сжатия изображений

(Окончание. Начало в №№1-2/2004)

При использовании фильтров возможно возникновение различных артефактов, т.е. искажений, в восстановленном изображении. Как правило, они удаляются с помощью дополнительных сглаживающих фильтров (чаще всего аналогов antialiasing и pixel interpolation).

Пример 4 представляет собой консольное приложение, которое может последовательно сжимать и восстанавливать квадратные изображения (в формате .bmp, true color); результат записывается в выходной файл. Параметры командной строки:

имя программы <тип_фильтра> <качество> <входной файл> <выходной файл>.

Здесь:

- *тип_фильтра* — определяет используемый фильтр (h — фильтр Хаара, d — фильтр Добеши);

- *качество* — задается степень сжатия (0...9, 0 — максимальная);

- *входной файл* — имя файла исходного изображения (формат .bmp, 24 bit, размер 128x128, 256x256, 512x512 и др. степени двойки);

выходной файл — имя файла изображения после последовательного сжатия и восстановления.

Например: Prog h input.bmp output.bmp.

```
#include "stdafx.h"
#include <iostream.h>
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
```

```
const double rt2=1.41421356237309;
const double rt3=1.73205080756887;
```

```
typedef int* pointer;
typedef short int word;
typedef unsigned int dword;
```

```
int mod(int w1,int base){
    while (w1>=base) w1-=base;
    return(w1);
}
```

```
int norm8( int w){
    if (w<0) return(0);
    if (w>255) return(255);
    return(w);
}
```

```
void delay(int time){
    int t1;
    t1=time+GetTickCount();
    while (GetTickCount()<t1){};
}
```

```
int absi( int a){
    if (a>=0) return(a);
    return(-a);
}
```



```

}

double det(int ch,int numb,double v[4],double m[4][4]){
    int i,j,k,l,su;
    double tmp=0,md[4][4];

    for (i=0;i<4;i++) for(j=0;j<4;j++) md[i][j]=m[i][j];
    if (ch==1) for (i=0;i<4;i++) md[i][numb]=v[i];

    for (i=0;i<4;i++)
        for (j=0;j<3;j++){
            if(mod(i+j+1,2)==0)
                su=-1;
            else
                su=1;
            if(j==1) su=-su;
            tmp+=su*md[0][i]*md[1][mod(i+j+1,4)]*
                (md[2][mod(i+1+mod(j+1,3),4)]*md[3][mod(i+1+mod(j+2,3),4)]-
                md[2][mod(i+1+mod(j+2,3),4)]*md[3][mod(i+1+mod(j+1,3),4)]);
        }
    return(tmp);
}

int main(int argc, char* argv[])
{
    HANDLE hSrc, hDest;
    DWORD dwRead, dwWritten;
    int dl2,dl3,dl4,csize,R,G,B,I,U,V;
    int N, n, maxv, quality;
    pointer org[3];
    int *wavet;
    char filter;
    unsigned char *pBuffer;
    word *pHead1, *pHead2;
    double conv[3][3]=
    {{0.30,0.59,0.11},
    {0.5,-0.42,-0.079},
    {-0.17,-0.33,0.5}};
    double tconv[3][3]=
    {{1.00,1.40,0.002},
    {1.00,-0.71,-0.33},
    {1.00,0.006,1.78}};
    double h[4],g[4],detm;
    double s0,sum=0,m[4][4],sl2,sl3,sl4;
    int i,j,k,t,l=0,o,p,key,tp;
    double vect[4];

    h[0]=rt2*(1 + rt3)/8;h[1]=rt2*(3 + rt3)/8;
    h[2]=rt2*(3 - rt3)/8;h[3]=rt2*(1 - rt3)/8;
    g[0]=h[3];g[1]=-h[2];
    g[2]=h[1];g[3]=-h[0];

    m[0][0]=h[1];m[0][1]=h[2];m[0][2]=h[2];m[0][3]=h[2];
    m[1][0]=g[1]; m[1][1]=g[2];m[1][2]=0; m[1][3]=0;
    m[2][0]=g[1]; m[2][1]=0; m[2][2]=g[2];m[2][3]=0;
    m[3][0]=g[1]; m[3][1]=0; m[3][2]=0; m[3][3]=g[2];
    h[1]/=3;
    detm=det(0,0,vect,m);

    pHead1=(word *)malloc(14);
    pHead2=(word *)malloc(40);

    if (argc != 5)
    {
        cout<<"Usage: " << argv[0] << "filter quality(0..9) sourcefile
        destfile"<<endl;
        delay(2000);
        exit(1);
    }
    filter=(char) *argv[1];
    quality=(int) (*argv[2])-48;
    hSrc = CreateFile(argv[3], GENERIC_READ, 0, NULL,
        OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);

    if (hSrc == INVALID_HANDLE_VALUE)
    {
        cout<< "Unable to open " <<argv[3]<< endl;delay(2000);
        exit(1);
    }

```

```

hDest = CreateFile(argv[4], GENERIC_WRITE, 0, NULL,
    CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, 0);

    if (hDest == INVALID_HANDLE_VALUE)
    {
        cout<< "Unable to create " <<argv[4]<< endl;
        CloseHandle(hSrc);delay(2000);
        exit(1);
    }

    ReadFile(hSrc,pHead1, 14, &dwRead, NULL);
    if (dwRead != 0)
        WriteFile(hDest,pHead1, dwRead, &dwWritten, NULL);

    ReadFile(hSrc,pHead2, 40, &dwRead, NULL);
    if (dwRead != 0)
        WriteFile(hDest,pHead2, dwRead, &dwWritten, NULL);

    if (pHead1[0]!=0x4d42 || pHead2[2]!=pHead2[4] ||
    pHead2[7]!=0x18){
        cout<<"Invalid format"; goto ex;
    }
    N=pHead2[2];n=N;
    org[0]=(int *)malloc(N*N*sizeof(int));
    org[1]=(int *)malloc(N*N*sizeof(int));
    org[2]=(int *)malloc(N*N*sizeof(int));
    wavet= (int *)malloc(N*N*sizeof(int));
    pBuffer=(unsigned char *)malloc(3*N*N);
    if (!org[0] && !org[1] && !org[2] && !wavet && !pBuffer){
        cout<<"Not enough memory";goto ex;
    }

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

    cout<<"Processing..."<<endl;
    for(i=0;i<N*N;i++){// преобразование (R, G, B)→(I, U, V)
        B=pBuffer[3*i+2];G=pBuffer[3*i+1];R=pBuffer[3*i];
        org[0][i]=conv[0][0]*R+conv[0][1]*G+conv[0][2]*B;
        org[1][i]=conv[1][0]*R+conv[1][1]*G+conv[1][2]*B;
        org[2][i]=conv[2][0]*R+conv[2][1]*G+conv[2][2]*B;
    }

    // раздельное кодирование информации об интенсивностях
    // и цветовых разностях
    csize=0;
    for (t=0;t<3;t++){
        switch(filter){// выбор фильтра для кодирования
        case 'd':
            while(n!=1){
                for (i=0;i<n;i+=2)
                    for (j=0;j<n;j+=2){
                        sl2=0;
                        for (k=0;k<3;k++) if (j+k<N)
                            sl2+=h[k+1]*org[t][N*i+j+k];
                        sl3=0;
                        for (k=0;k<3;k++) if (i+k<N) sl3+=h[k+1]*org[t][N*(i+k)+j];
                        sl4=0;
                        for (k=0;k<3;k++) if (i+k<N && j+k<N)
                            sl4+=h[k+1]*org[t][N*(i+k)+j+k];
                        dl2=0;
                        for (k=0;k<3;k++) if (j+k<N) dl2+=g[k+1]*org[t][N*i+j+k];
                        dl3=0;
                        for (k=0;k<3;k++) if (i+k<N) dl3+=g[k+1]*org[t][N*(i+k)+j];
                        dl4=0;
                        for (k=0;k<3;k++) if (i+k<N && j+k<N)
                            dl4+=g[k+1]*org[t][N*(i+k)+j+k];
                        org[t][N*i/2+j/2]=(sl2+sl3+sl4);
                        wavet[l++]=int(dl2);
                        wavet[l++]=int(dl3);
                        wavet[l++]=int(dl4);
                    }
                n/=2;
                for (o=0;o<N;o++) for (p=0;p<N;p++)
                    if ((o>=n)|| (p>=n)) org[t][N*o+p]=0;
            }
        case 'h':
            while(n!=1){
                for (i=0;i<n;i+=2)
                    for (j=0;j<n;j+=2){

```



```

s0=org[t][N*i+j];
s12=org[t][N*(i+1)+j];
s13=org[t][N*i+j+1];
s14=org[t][N*(i+1)+j+1];
org[t][(N*i/2)+j/2]=(s0+s12+s13+s14)/4;
wavet[l++]=s0-s12;
wavet[l++]=s0-s13;
wavet[l++]=s0-s14;
}
n/=2;
}
}

// обнуление части коэффициентов
maxv=0;
for (i=0;i<l;i++) if (absi(wavet[i])>maxv)
maxv=absi(wavet[i]);
for (i=0;i<l;i++){
if (t==0 && (absi(wavet[i])<(10-quality)*maxv/20) && i<4096){
wavet[i]=0;
}
if (t!=0 && (absi(wavet[i])<(10-quality)*1.2*maxv/20) && i<2048){
wavet[i]=0;
}
}
i=0;
// вычисление размера архивированного массива коэффициентов-
// для оценки эффективности
while(i<l){
tp=wavet[i++];
if (wavet[i]==tp){
csize+=3;j=1;
while(wavet[i++]==tp && j<65535 && i<l) j++;
}
else csize++;
}
// восстановление изображения
n=1;
switch(filter){
case 'd':
while(n<N){
for(i=n-1;i>=0;i-){
for (j=n-1;j>=0;j-){
d14=wavet[-1];
d13=wavet[-1];
d12=wavet[-1];
s0=org[t][N*i+j];
for (k=2;k<3;k++){
if (2*i+k<N && 2*j+k<N) s0-=h[k+1]*(org[t][N*(2*i+k)+2*j]+
org[t][N*(2*i+k)+2*j+k]+org[t][N*(2*i+2*j+k)]);
if (2*j+k<N) d12-=g[k+1]*org[t][2*N*i+2*j+k];
if (2*i+k<N) d13-=g[k+1]*org[t][N*(2*i+k)+2*j];
if (2*i+k<N && 2*j+k<N) d14-=g[k+1]*org[t][N*(2*i+k)+2*j+k];
}
vect[0]=s0;vect[1]=d12;vect[2]=d13;vect[3]=d14;
org[t][2*N*i+2*j]=det(1,0,vect,m)/detm;
org[t][2*N*i+2*j+1]=det(1,1,vect,m)/detm;
org[t][N*(2*i+1)+2*j]=det(1,2,vect,m)/detm;
org[t][N*(2*i+1)+2*j+1]=det(1,3,vect,m)/detm;
}
n*=2;
}
case 'h':
while(n<N){
for(i=n-1;i>=0;i-){
for (j=n-1;j>=0;j-){
d14=wavet[-1];
d13=wavet[-1];
d12=wavet[-1];
s0=org[t][N*i+j];
s12=s0+(d12+d13+d14)/4;
org[t][2*N*i+2*j]=s12;
org[t][N*(2*i+1)+2*j]=s12-d12;
org[t][2*N*i+2*j+1]=s12-d13;
org[t][N*(2*i+1)+2*j+1]=s12-d14;
}
n*=2;
}
}
}

```

```

)
for(i=0;i<N*N;i++){
I=org[0][i];U=org[1][i];V=org[2][i];
R=tconv[0][0]*I+tconv[0][1]*U+tconv[0][2]*V;
G=tconv[1][0]*I+tconv[1][1]*U+tconv[1][2]*V;
B=tconv[2][0]*I+tconv[2][1]*U+tconv[2][2]*V;
R=norm8(R);B=norm8(B);G=norm8(G);
pBuffer[3*i+2]=B;pBuffer[3*i+1]=G;pBuffer[3*i]=R;
}

WriteFile(hDest, pBuffer, dwRead, &dwWritten, NULL);
cout<<"file:"<<argv[3]<<endl;
cout<<"resolution:"<<N<<"x"<<N<<endl;
cout<<"original size:"<<3*N*N<<"bytes"<<endl;
cout<<"compressed size:"<<2*csize<<"bytes"<<endl;
cout<<"quality:"<<quality<<endl<<"ratio:"<<200*csize/
(3*N*N)<<"%"<<endl;
ex:
CloseHandle(hSrc);
CloseHandle(hDest);
cout<<endl<<"Press any key...";key=getch();

return 0;
}

```

Сравнение и выводы

Использование дискретного вейвлет-преобразования (ДВП) дает заметный выигрыш (по сравнению с ДКПФ) в степени сжатия с заданным фиксированным качеством восстановления сложных изображений. Особенно это относится к тем случаям, когда важно сохранить в изображении мелкие детали и в то же время достичь максимальной степени сжатия. Пожалуй, единственный недостаток ДВП — это относительно высокие вычислительные затраты, особенно при использовании фильтров большой длины. Но вычислительные мощности современных компьютеров все возрастают, поэтому этот недостаток вряд ли следует отнести к серьезным. Очевидно, следующим шагом станет создание нового стандарта сжатия потокового видео, основанного на ДВП.

Литература

1. Иванчиков А. Экскурсия в рекурсию. — Радиолюбитель. Ваш компьютер, 1997, №4, С.3.
2. Введение в цифровую фильтрацию/ Под ред. Р. Богнера и А. Константинодиса. — М.: Мир, 1976.
3. Левкович-Маслюк Л. Дайджест вейвлет-анализа, в двух формулах и 22 рисунках. — Компьютерра, 1998, №8(236).
4. <http://www.beyonddiscovery.org/content/view.pdf.asp?a=1952>
5. Дремин И., Иванов О., Нечитайло В. Вейвлеты и их использование. — Успехи физических наук, 2001, Т.171, №5, С.465.
6. <http://www.amara.com/current/wavelet.html>



Рисунок О.Попова



ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КРЭКИНГА

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

Куда попадают данные

На протяжении уже двух глав я рассказывал о технике поиска данных всевозможных типов. Подобно муравьям, мы трудились, учась собирать по крупицам полезную информацию, которая могла бы помочь понять структуру и смысл этих данных. И теперь вы знаете, как “просеивать” мегабайты листингов, извлекая из них константы и указатели, как вести поиск переменных в адресном пространстве работающей программы. Все это время мы рассматривали “живые”, находящиеся внутри работающей программы данные отдельно от “мертвых”, тихо лежащих на жестком диске и никак себя не проявляющих. И вот пришло время осуществить синтез, увидеть диалектическую взаимосвязь между этими двумя формами существования данных и, главное, понять, какую практическую пользу из этой взаимосвязи можно извлечь.

Нетрудно заметить, что любые данные, хранящиеся на диске, в действительности предназначены для обработки какой-либо программой, а стало быть, рано или поздно будут загружены в оперативную память (если, конечно, это полезные данные, а не мусор, подлежащий удалению), где над ними будут производиться всяческие действия. А результаты этих действий так или иначе отобразятся в мире “по нашу сторону экрана” при помощи одного из многочисленных устройств ввода-вывода, чтобы пользователь мог их увидеть, услышать или ощутить каким-либо иным образом. И пока эти данные будут проходить свой непростой путь от загрузки с винчестера до отображения на экране монитора, их можно “выловить” из адресного пространства программы или даже с экрана (т.е., конечно, не совсем с экрана, а из видеопамати). Более того, информация, которую очень непросто расшифровать, разглядывая файл в шестнадцатеричном редакторе, при загрузке соответствующей программой нередко бывает представлена в памяти в виде структур с весьма замысловатым внутренним устройством. Да и само наблюдение за процессом загрузки данных может дать множество полезной информации. В этой главе мы посмотрим, как такую информацию можно извлекать.

Когда-то давным-давно, когда Windows еще была девяносто пятой, защиты — простыми, а авторы защит — наивными, серийные номера извлекались из программ следующим образом. Устанавливались точки останова на все функции WinAPI, при помощи которых мог считываться серийный номер (благо, их не так много). Затем нужно было вызвать окно регистрации, ввести в него любые данные и посмотреть, какая из точек останова сработает. Дальше начиналось самое интересное. Поскольку это были старые добрые времена, непуганые разработчики для проверки правильности серийных номеров частенько использовали обычное сравнение двух текстовых строк. Причем для сравнения использовался банальный вызов функции `Istrcmp` (или ее самодельного аналога), два параметра которой являлись указателями на сравниваемые строки. И чтобы получить правильный серийник, требо-

валось лишь найти нужную функцию и посмотреть на ее параметры.

Конечно, те времена давно прошли, и ныне очень, очень редко встречаются программы, в которых серийный номер хранился бы в открытом или “как бы зашифрованном” при помощи команды XOR виде. Но для крэкера как раз важен не столько сам факт хранения данных в открытом виде, сколько идея — “скормив” программе заведомо неверные данные, понаблюдать за тем, как программа эти данные будет “переваривать” и проверять на корректность. Да и наблюдение за процессом “заглатывания” данных программой может стать источником ценных идей. Приведу пример из собственной практики.

Однажды я изучал некую программу на предмет “исправить пару переходов, чтобы она лучше и дольше работала”. Нужную “пару переходов” я вычислил за считанные минуты, а патчинг этих байтов непосредственно в памяти успешно решал мою проблему на время одной сессии работы с программой. Но вот исправление тех же байтов в исполняемом файле неизбежно приводило к “падению” программы сразу после запуска. Нетрудно было догадаться, что программа неким образом контролировала собственную целостность, и, скорее всего — проверкой контрольной суммы. Это предположение подтверждалось и подозрительно большим временем загрузки программы (компьютеры тогда были намного медленнее, поэтому иногда следы работы защитных средств были видны, что называется, невооруженным глазом). Решение тоже было достаточно очевидным — найти функцию вычисления контрольной суммы, посмотреть, какой результат эта функция должна была возвращать в норме, и либо обойти сравнение реальной контрольной суммы с эталоном, либо заставить функцию возвращать эталонное значение в любом случае. Но как найти нужную функцию?

Для начала я попытался выяснить, каким образом программа проверяет свою контрольную сумму — сканирует образ непосредственно в памяти или все-таки проверяет то, что лежит на диске. Поскольку программой была не запакована (в те времена упаковщики вообще встречались нечасто), я просто загрузил программу при помощи loader'a из состава SoftIce (одна из полезных крэкеру функций этого loader'a как раз та, что он передает управление отладчику сразу после загрузки подопытной программы в память). Затем я поставил аппаратные точки останова на чтение тех байтов, которые я хотел изменить в файле (тут логика простая — если программа проверяет саму себя в памяти, то для этого ей придется прочитать себя), и на запись (на всякий случай) и отпустил программу на волю (т.е. на исполнение). Ни одна из точек останова не сработала, из чего следовало, что программа либо не проверяет себя в памяти, либо это очень хитрая программа, которая на мою уловку не попала. Запустив программу под `filemon`’ом, я увидел, что сразу после запуска эта программа поблочно читает свой соб-



ственный исполняемый файл, что навело меня на мысль о встроенной в программу проверке контрольной суммы. Дальнейшее было делом техники: прогнав программу под Bounds Checker'ом, я выяснил, что нужный мне вызов функции чтения из файла в действительности производится не из самой программы, а из DLL, которая в случае успешной проверки возвращала некое значение (а в случае неуспешной проверки — тоже значение, но уже другое), и что для работоспособности программы величина этого значения была критически важной. В этой ситуации я счел наилучшим решением выкинуть вычисление контрольной суммы файла (это ощутимо ускорило загрузку) и немного "помог" этой DLL всегда возвращать нужное мне значение.

О чем эта история? Ну разумеется, не о том, что глупо помещать код проверки в DLL, где его несложно поправить. Прежде всего, я хотел показать, как наблюдение за переходом данных из "мертвого" состояния в "живое" (а именно таким переходом и является поблочная загрузка файла для вычисления контрольной суммы) может помочь обнаружить защитные механизмы. Действительно, стоило мне понаблюдать за процессом проверки целостности файла (о котором я ранее ничего не знал, кроме факта его наличия) под API-шпионом, как я сразу же получил информацию о типе защиты и местонахождении защитной процедуры. А после недолгих экспериментов и размышлений я также узнал, какова величина контрольной суммы программы до и после внесения в нее модификаций.

Вылавливание нужных данных из оперативной памяти уже давно стало неотъемлемой частью крэкинга и получило весьма широкое распространение. Если вы пробовали самостоятельно взломать или хотя бы посмотреть на "внутренности" какой-либо программы, то, возможно, уже столкнулись с упаковщиками исполняемых файлов (или, если быть до конца точным, с файлами, обработанными такими упаковщиками). Разумеется, крэкеру во всех этих упаковщиках и навесных защитах интересно одно — методы их снятия. Очевидно, что упаковка программ — процесс обратимый, и проблема лишь в том, чтобы найти способ обращения этого процесса, проще говоря — распаковать ее. Существует два подхода к распаковке. Можно проанализировать алгоритмы работы встроенного в программу навесного модуля, осуществляющего распаковку, и самостоятельно воспроизвести эти алгоритмы в виде независимой программы. Этот метод обычно долг и труден.

А можно оставить все хлопоты по распаковке "навесному" модулю-распаковщику, встроенному в исполняемый файл, а потом воспользоваться результатами его трудов, "выдернув" распакованную и готовую к употреблению программу из памяти компьютера. В таком подходе мне определенно видится изящество и утонченность — вместо того чтобы брать штурмом алгоритмы распаковки, мы фактически заставляем автора навесной защиты сражаться с собственным творением. Конечно, авторы защит тоже не дремлют — редкий упаковщик не уродует до неузнаваемости таблицу импорта, не содержит антиотладочного кода или средств противодействия дампингу, и преодоление этих трудностей требует от крэкера гораздо больших усилий, чем собственно снятие дампа. Однако сама идея как нельзя лучше раскрывает тему этой главы — если нет возможности (сил, времени, желания) "расшифровать" нужные данные вручную, стоит подумать о том, где можно найти готовые алгоритмы декодирования, каким образом их применить и как воспользоваться результатом их работы.

Особенно интересные и впечатляющие результаты дает сочетание предлагаемой технологии с глубоким патчингом программ в памяти. Недавно мне в руки попался экземпляр MoleBox — представителя (надо сказать, не самого совершенного) нового поколения защит, где упаковке подвергается не только исполняемый файл приложения, но и все остальные файлы, входящие в комплект программы, после чего все эти упакованные файлы сливаются в один монолитный исполняемый файл ("ящик" в терминологии MoleBox). Сам EXE-файл программы модифицируется таким образом, что вызовы функций API для работы с файлами подменяются вызовами внутренних функций защиты, после чего программа может одинаково успешно обращаться как к файлам на жестком диске, так и к файлам, находящимся внутри "ящика" (в MoleBox к файлам из "ящика" возможен доступ только на чтение). Кстати, базовая информация о принципах работы MoleBox честно приведена в документации к программе, поэтому позволю себе в очередной раз повторить совет внимательно читать документацию к исследуемым программам. После недолгих экспериментов удалось выяснить, что "виртуальная директория", в которой работает защищенное приложение, содержит все файлы программы, и извлечь их оттуда не составляет никакого труда. При помощи манипуляций значениями регистров и содержимым стека в SoftICE мне удалось вызвать FindFirstFile/FindNextFile и вручную прочитать список имен всех файлов, находящихся в "ящике" программы, кроме самого исполняемого файла (который пришлось выковыривать более традиционными методами). Дальше все было еще проще в теории и еще тяжелее и нуднее на практике — выделение памяти под буфер, чтение файлов в этот буфер и последующее сохранение в другой файл. Конечно, продельывать все эти операции вручную — занятие крайне трудоемкое, и если вы захотите повторить мой эксперимент, я советую не упражняться в играх с регистрами, а набросать соответствующую программку на ассемблере, внедрить ее в адресное пространство "жертвы" и получить тот же самый результат, но в несколько раз быстрее.

Еще одно применение предлагаемого метода — декодирование данных, имеющих сложную или неочевидную структуру. Например, при сохранении множества записей, содержащих как текстовую, так и числовую информацию, формат результирующего файла может быть совершенно неочевиден. К примеру, массив структур, состоящих из одного текстового (обозначим его буквой T) и одного числового поля (обозначим его как N), может сохраняться в файле как минимум двумя способами:

T1, N1, T2, N2, T3, TN3, ...

или

N1, N2, N3, ... T1, T2, T3, ...

где Tn, Nn — текстовое и числовое поле соответственно n-й записи в массиве. Поскольку текстовые данные отличить от числовых несложно даже по внешнему виду, в данном конкретном примере никаких сложностей с извлечением из файла элементов массива, скорее всего, не возникнет. Но представим себе, что текстовых полей — несколько, а сохраняемые в этих полях значения — внешне очень похожи. И что каждая из структур в массиве содержит подструктуры, сохраняемые в том же самом файле подобным же образом. Задача расшифровки внутреннего формата файла уже не кажется такой тривиальной, не правда ли?

Однако вспомним наш краткий курс психологии программиста и попробуем представить, как нормальный програм-



мист организует хранение тех же данных в памяти. Скорее всего, он создаст банальный массив структур, поэлементно заполнит его значениями из файла и будет обращаться к нему так, как обращаются к любым другим массивам. Если логика задачи предполагает, что в загруженные данные потребуется добавлять новые элементы или удалять имеющиеся, то вместо обычного массива, скорее всего, будет одно- или двухсвязный список, в котором каждый элемент, помимо собственно структуры, будет содержать еще указатель на предыдущий и последующий элементы списка. А дальше... Дальше задача полностью аналогична описанной в предыдущей главе задаче по "раскалыванию" неизвестного формата и извлечению данных, с той лишь разни-

цей, что работать мы будем не с кодом приложения, а с обрабатываемыми этим приложением данными. Увы, данный прием не универсален — например, буферизованная обработка данных (т.е. такая, при которых обрабатываемые данные не переносятся в память целиком, а подгружаются по мере необходимости) не позволит расшифровать весь исследуемый файл целиком за один раз. Вообще, количество всевозможных "особых случаев" весьма велико, и рассмотреть их все в этой небольшой работе практически нереально. И если вам придется столкнуться с такой нетривиальной программой, успех будет полностью зависеть от вашей сообразительности, настойчивости и терпения.

(Продолжение следует)

Microsoft Agent Часть I

Microsoft, несмотря на критику в свой адрес, постоянно предоставляет разработчикам все новые и новые возможности. Несколько лет назад была представлена очень интересная технология — Microsoft Agent. Где можно было наблюдать ее применение? Например, в популярном переводчике Magic Goody. Основной "фишкой" данной программы является утенок Goody, который может свободно говорить на русском и английском языках. Более того, этот утенок может еще и "понимать" некоторые голосовые команды, произнесенные в микрофон. У Goody, ко всему прочему, все время меняется настроение: если долго не пользоваться его услугами, он начинает грустить, а при переводе он выглядит самым счастливым утенком в мире. Этот утенок со всеми его возможностями является наглядным примером применения технологии MsAgent.



Вот таким вступлением я открываю небольшую серию статей, посвященных Microsoft Agent. В первой части мы рассмотрим аспекты использования MsAgent в web-страницах. Представьте себе, сидите вы в Интернете, открываете некую web-страничку, а тут из лампы вылезает Джин из мультфильма "Аладдин" и начинает что-нибудь вам рассказывать. А если вы наведете курсор мыши на какую-нибудь гиперссылку, то Джин подлетит к ней и доходчиво объяснит, куда вы попадете, нажав на ссылку... Здорово? Вот о создании примерно такой странички мы сейчас и поговорим.

Самое время установить MsAgent. Для этого нужно отправиться на страницу <http://www.microsoft.com/msagent/downloads/user.asp> и скачать оттуда следующие файлы:

- сам Microsoft Agent — <http://activex.microsoft.com/activex/controls/agent2/MSAgent.exe> (395 Кб), обязательно;
- перевод на русский язык различных диалогов, надписей и т.п. — <http://activex.microsoft.com/activex/controls/agent2/AgTX0419.exe> (128 Кб), не обязательно;
- один из персонажей (например, Джина из мультфильма "Аладдин") — <http://download.microsoft.com/download/0/0/c/00cde5f8-321d-4325-baae-eb27f1bde85f/Genie.exe> (1,6 Мб), обязательно;

- компонент для произношения текстов (для русского языка) — <http://activex.microsoft.com/activex/controls/agent2/httsrur.exe> (2,9 Мб), не обязательно;

- поддержку SAPI (нужна для произношения текстов, в Win2000 должна быть установлена изначально) — <http://activex.microsoft.com/activex/controls/sapi/spchapi.exe> (824 Кб), не обязательно.

После установки всех этих компонентов можно приступить к разработке приложений, использующих технологию MsAgent. Начнем мы, как уже было сказано, с web-страниц. Сразу замечу, что на пользователя накладываются некоторые ограничения:

- MsAgent не будет работать ни в одном web-браузере, кроме Internet Explorer (разработчик ведь Microsoft). Версия IE должна быть не ниже, чем 3.02;

- у посетителя web-страницы должно быть установлено необходимое программное обеспечение (его мы уже скачали).

Теперь сделаем маленькую пробную страничку. Откройте самый продвинутый HTML-редактор под названием Блокнот (некоторым он известен как NotePad). Запишите в нем следующие строки:

```
<html>
<head>
<title>Microsoft Agent</title>
<Object ID="Agent" Width=0 Height=0
  ClassID="CLSID:D45FD31B-5C6E-11D1-9EC1-00C04FD7081F"
  CodeBase="#VERSION=2,0,0,0">
</Object>
<script language=javascript>
<!--
//объявляем переменную, в которой будем хранить
//идентификатор персонажа
var Genie;
function LoadAgent()
{
//загружаем персонажа "Джин"
  Agent.Characters.Load ("Genie", "Genie.acs");
  Genie = Agent.Characters.Character ("Genie");
//Джин появляется на экране
  Genie.Show();
//Джин играет анимацию "Greet"
  Genie.Play("Greet");
//Джин говорит приветственную фразу
  Genie.Speak("Здравствуйте! Добро пожаловать на наш сайт!");
//А здесь можно написать другие команды
}
function UnLoadAgent()
//выгружаем персонажа
{ Agent.Characters.Unload("Genie"); }
<!-->
</script>
</head>
<body
onload="LoadAgent()"
onunload="UnLoadAgent()">
</body>
</html>
```



Табл. 1

Команда	Пояснения
Show	Служит для вызова персонажа на экран
Hide	Скрывает персонажа
MoveTo (X, Y)	Перемещает персонажа в точку с координатами (X; Y). Координаты задаются в пикселах
Play("Имя анимации");	Проигрывает заданную анимацию
GestureAt (X, Y)	Персонаж указывает рукой в точку с координатами (X; Y)
Think("Мысль")	Персонаж "думает Мысль", т.е. звука нет, но над персонажем появляется текст Мысли.
Speak("Фраза")	Персонаж произносит Фразу

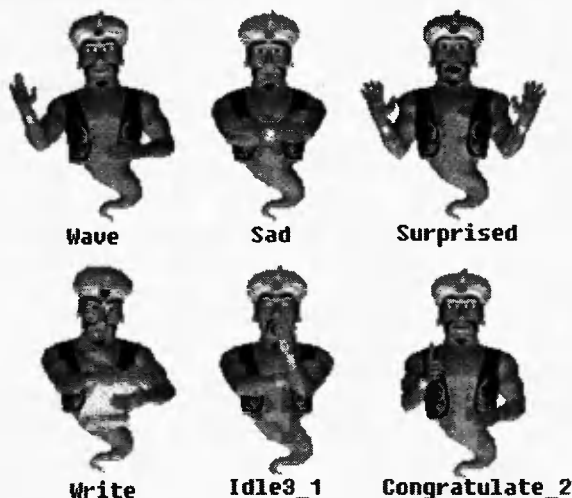
Теперь сохраните данный код в файл *.html и откройте полученную страничку при помощи Internet Explorer. После загрузки странички из дима появится Джин, поприветствует вас поклоном и произнесет на русском языке приветственную фразу. Давайте посмотрим, как мы добились такого результата.

В тэге <Object> мы подключили к нашей странице объект AGENT, который предоставит нам все возможности технологии MsAgent. После этого на языке JavaScript записали две функции — LoadAgent и UnLoadAgent. А затем в тэге <body> указали, что эти функции должны вызываться при загрузке и закрытии страницы соответственно. После строчек

```
...
Agent.Characters.Load ("Genie", "Genie.acs");
Genie = Agent.Characters.Character("Genie");
...
```

мы можем "давать команды" Джину: Genie.Show; Genie.Play; Genie.Speak; и т.д. В табл.1 приведены описания некоторых команд.

На рисунке показано несколько вариантов анимаций для персонажа "Джин":



Пример вызова анимации: Genie.Play("Write");

MsAgent позволяет управлять даже голосом персонажа. Делается это при помощи специальных тэгов, которые нужно добавлять к произносимой фразе (табл.2).

Пример для JavaScript:

```
Genie.Speak("\chr="Whisper"\\Pit=18400\\Spd=100\\Привет!");
```

С помощью представленных команд можно создать довольно эффектную интерактивную страничку. Про другие команды и свойства мы поговорим в следующей части, когда перейдем к Delphi, а пока рассмотрим несколько аспектов использования MsAgent в web-страничках. Предположим, вы поместили страничку в Интернете. Тогда ее могут просматривать три вида пользователей:

- пользующиеся браузером Internet Explorer и установленной MsAgent;
- пользующиеся браузером Internet Explorer без установленной MsAgent;
- пользующиеся другим браузером.

Табл. 2

Тэг	Значения и описание
\Chr="значение"\	Изменяет голос персонажа:
	"Normal" (по умолчанию) — нормальный голос
	"Monotone" — монотонный голос
	"Whisper" — шепот
\Emp\	На следующем за тэгом слове делается ударение
\Map="spokentext"="balloontext"\	Персонаж вслух произносит строку "spokentext", а над ним отображается строка "balloontext". Т.е. персонаж как бы думает одно, а говорит совершенно другое.
\Pau=значение\	Делает паузу, которая длится "значение" миллисекунд
\Pit=значение\	Устанавливает частоту произношения в герцах
\Rst\	Устанавливает все параметры в исходное состояние
\Spd=значение\	Устанавливает скорость произношения текста в "словах в минуту"
\Vol=значение\	Устанавливает громкость произношения текста (от 0 до 65535)

Именно эти три случая мы должны обрабатывать на нашей страничке. Сделать это не так и сложно. Вначале запишем функцию HaveAgent, которая будет определять, установлены ли компоненты MsAgent в системе:

```
Function HaveAgent()
  Dim agent
  HaveAgent = False
  On Error Resume Next
  Set agent = CreateObject("Agent.Control.1")
  HaveAgent = IsObject(agent)
End Function
```

Теперь на основе этой функции можно написать шаблон странички, которая будет предусматривать все возможные варианты:

```
<SCRIPT LANGUAGE=VBSCRIPT>
If HaveAgent() Then
  document.write "Здесь пишем код, если MsAgent установлен"
Else
  document.write "Здесь пишем код, если MsAgent не установлен"
End If
Function HaveAgent()
  Dim agent
  HaveAgent = False
  On Error Resume Next
  Set agent = CreateObject("Agent.Control.1")
  HaveAgent = IsObject(agent)
End Function
</SCRIPT>
<noscript>
Здесь пишем код для браузеров, которые не поддерживают
VBScript (например, Opera)
</noscript>
```

Вот и все! По этому небольшому шаблону можно писать web-странички с использованием MsAgent, которые будут корректно отображаться во всех браузерах. Для примера я создал страничку, которую можно найти по адресу <http://www.agentsample.narod.ru/>.

На этом мы заканчиваем рассмотрение использования технологии Microsoft Agent в web-страничках. В следующей части мы перейдем к Delphi.



“Отучение” игр от компакт-диска

(Окончание. Начало в №2/2004)

Р.ФАСИХОВ,
г.Казань.
E-mail: fasihov@mail.ru

Но это был механизм, который проверял только наличие оригинального диска. Теперь рассмотрим механизм, который по мере необходимости считывает файлы с диска. То, что он это делает, можно определить по следующим признакам:

- размер установленной папки меньше дистрибутива на диске. Это дает повод думать, что какие-то файлы остались на диске;
- во время работы программы идет интенсивная работа с диском (определяется элементарно — по миганию индикатора лицевой панели привода CD-ROM и по шуму при раскрутке диска);
- в директории установленной игры отсутствуют некоторые файлы или папки.

Подумаем, как же программа может организовать такую работу. Перво-наперво, нужно определить правильный корневой путь к CD-ROM'у. Потом может следовать процедура копирования этого пути и имени нужного файла для открытия, например, "D:\\"+"GameDat"="D:\GameDat". А затем — открытие файлов.

Самый простой способ “отучить” игру от диска — скопировать нужные файлы в “корень” диска C:\, изменив начальную букву привода и проверку на “сидиромность” (таких проверок может быть несколько):

```
mov     dword ptr [Drive_Name], 'C'
push    offset Drive_Name    ;передаем параметр для функции
call     GetDriveType
cmp     eax, 5                ;5 заменяем на 3
```

В этом случае все необходимые файлы будут считываться с винчестера.

Этот способ — самый простой, но не самый лучший, т.к. при его использовании засоряется корневой каталог. Есть простое и красивое решение. Если заменить процедуру копирования корневого пути диска с путем к файлу на сам путь к файлу, тогда и поиск будет происходить там же, и можно будет копировать нужные файлы в директорию с установленной игрой. Копирование необходимо убрать во всех местах:

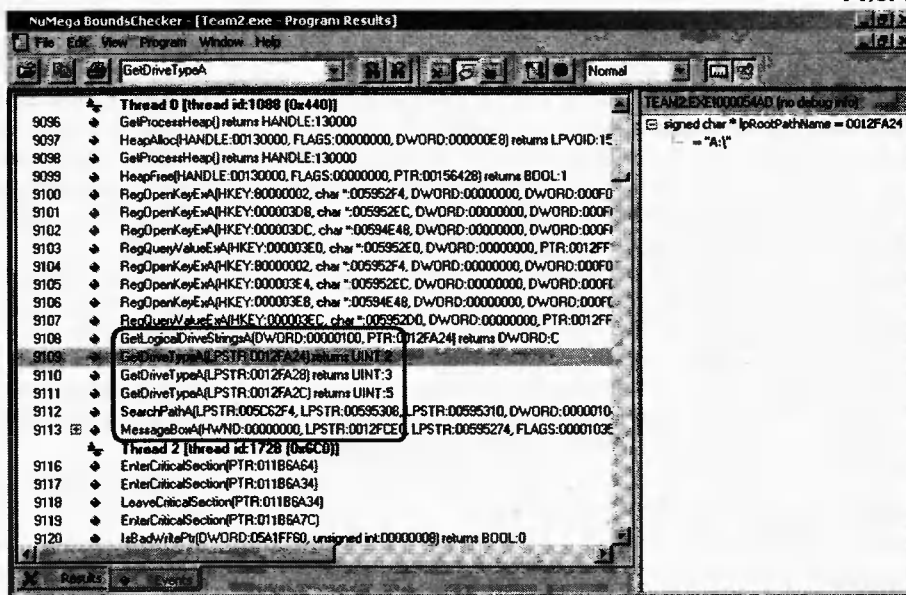
Было	Стало
C:\elf2rr	C:\MY_GAMES\SUPER_GAME\elf2rr
C:\gamedat	C:\MY_GAMES\SUPER_GAME\gamedat

В качестве примера возьмем игру “K-52 Team Alligator”. Воспользуемся API-шпионом для получения информации о работе программы и проанализируем его отчет (рис.3).

Выделенные функции, скорее всего, связаны с CD-ROM'ом. Их можно найти, если произвести контекстный поиск по полученному отчету. Это дает возможность найти функцию в дизассемблере (по перекрестным ссылкам на API-функции), изучить и исправить ее. Рассмотрим листинг функции:

```
00405450 CD_Present      proc near
00405450
```

Рис. 3



```
00405450 current_letter      = dword ptr -110h
00405450 max_letter          = dword ptr -10Ch
00405450 var_108             = dword ptr -108h
00405450 FilePart            = dword ptr -104h
00405450 var_100             = byte ptr -100h
00405450 var_FF              = byte ptr -0FFh
00405450
00405450 sub     esp, 110h
00405456 mov     al, byte_5C65D8
0040545B push    ebx
0040545C push    ebp
0040545D push    esi
0040545E push    edi
0040545F mov     [esp+120h+var_100], al
00405463 mov     [esp+120h+var_108], ecx
00405467 xor     eax, eax
00405469 mov     ecx, 3Fh
0040546E lea     edi, [esp+120h+var_FF]
00405472 repe    stosd
00405474 lea     ecx, [esp+120h+var_100]
00405478 stosw
0040547A push    ecx                ; lpBuffer
0040547B xor     ebp, ebp
0040547D push    100h              ; nBufferLength
00405482 stosb
00405483 mov     [esp+128h+current_letter], ebp
```

;Заполняем буфер данными, которые определяют действительные
;диски в системе

```
00405487 call     ds:GetLogicalDriveStringsA
```



```

0040548D cmp     eax, ebp
0040548F mov     [esp+120h+max_letter], eax
00405493 lea     ebx, [esp+120h+var_100]
           ;Указывает на первый элемент буфера
00405497 jz      loc_40555E ;Если ошибка, то выход из функции
0040549D jle     loc_40555E
; Мы получили буфер, заполненный следующим образом
; A:\<null>,C:\<null>,D:\<null><null>

004054A3
004054A3 loc_4054A3:
004054A3 test    ebp, ebp
004054A5 jnz     loc_40555E
004054AB push    ebx ;lpRootPathName - имя корневого пути
004054AC call    ds:GetDriveTypeA ;Возвращаем параметр привода
004054B2 cmp     eax, 5 ;Если не сидит, то следующая буква
           ;алфавита (заменить на 3)
004054B5 jnz     short bad_search
004054B7 push    4
004054B9 push    ebx
004054BA push    offset Disc_path
004054BF call    strncpy ;Копируем 4 символа из места,
           ;куда указывает ebx(корневой путь)
           ;в глобальную переменную Disc_path
004054C4 mov     eax, [esp+12Ch+var_108]
004054C8 add     esp, 0Ch ;Балансируем стек
004054CB lea     edx, [esp+120h+FilePart]
004054CF push    edx ; lpFilePart
004054D0 push    eax ; lpBuffer
004054D1 push    104h ; nBufferLength
004054D6 push    offset a_exe ; lpExtension
004054DB push    offset aAutorun ; lpFileName
004054E0 push    offset Disc_path ; lpPath
004054E5 call    ds:SearchPathA ;Ищем указанный файл
           ;в указанном пути
           ;(Disc_path\autorun.exe)
004054EB test    eax, eax
004054ED jbe     short bad_search ;Если ошибка, или не нашли,
           ;то следующая буква алфавита
004054EF or      ecx, 0FFFFFFFh ;ecx=-1
004054F2 xor     eax, eax
004054F4 mov     edi, offset aGame ; "game\" ;строка 1
004054F9 repne scasb
004054FB not     ecx ;ecx==длина строки, куда
           ;указывает edi, вместе с нулем
004054FD sub     edi, ecx ;edi==смещение строки
004054FF mov     edx, ecx ;edx==длине строки
00405501 or      ecx, 0FFFFFFFh ;ecx=-1
00405504 mov     esi, edi
00405506 mov     edi, offset Disc_path ;строка 2
0040550B repne scasb

;Если убрать инструкцию repne scasb, а также инструкцию dec edi
;по адресу 405512, то произойдет просто перемещение строки 1 на
;место строки 2. Т.е. по адресу, куда указывает edi,
;вместо Disc_path\game\ получим game\ - то, что нам и надо.
;Таких действий может быть несколько, и нужно проделать их все

0040550D mov     ecx, edx ;ecx==длине 1-ой строки
0040550F shr     ecx, 2 ;ecx/4
00405512 dec     edi ;edi указывает на конец 2-й строки
           ;не включая ноль (Disc_path)
00405513 repe movsd ;Копирование двойными словами
           ;(так быстрее)
00405515 mov     ecx, edx ;ecx опять длина строки
00405517 and     ecx, 3 ;ecx остаток от деления на 4
0040551A repe movsb ;Скопировали всю строку
0040551C mov     ebp, 1 ;Видимо, сообщаем об успехе процедуры
00405521 jmp     short loc_405550
00405523 ;-----
00405523

```

```

;Здесь попадаем, если ошибка, или не выполнено какое-то условие
00405523 bad_search:
00405523 or      ecx, 0FFFFFFFh ;ecx=-1
00405526 xor     eax, eax ;eax=0
00405528 mov     edi, offset byte_5C65D8
0040552D repne scasb
0040552F not     ecx ;ecx==длина строки,
           ;на которую указывает edi
00405531 sub     edi, ecx ;edi-смещение начала строки
00405533 mov     eax, ecx
00405535 shr     ecx, 2
00405538 mov     esi, edi
0040553A mov     edi, offset Disk_path
0040553F repe movsd ;Копирование в буфер edi
00405541 mov     ecx, eax
00405543 and     ecx, 3
00405546 repe movsb ;Заполнили буфер имени привода
           ;новой буквой
00405548 add     [esp+120h+current_letter], 4
           ; Увеличили указатель в буфере
0040554D add     ebx, 4
00405550
00405550 loc_405550:
00405550 mov     ecx, [esp+120h+current_letter]
00405554 cmp     ecx, [esp+120h+max_letter]
00405558 jle     loc_4054A3
0040555E
0040555E loc_40555E:
0040555E pop     edi
0040555F pop     esi
00405560 mov     eax, ebp ;Возвращаемое значение функции
00405562 pop     ebp
00405563 pop     ebx
00405564 add     esp, 110h
0040556A retn
0040556A CD_Present endp

```

Надо сказать, что встречаются довольно наивные привязки, в которых весь механизм локализован всего в одной процедуре, которая возвращает значение Диск_Есть/Диска_Нет, а после этого принимается решение (условный переход), продолжать работу или нет. В таком случае даже не нужно вносить изменений в ее работу:

```

call Disc_Present
test eax, eax
jx OK

NO_DISK_IN_DRIVE:
; "Левая" ветка программы
; То, что делаем, если диск отсутствует
OK:

```

Достаточно поправить условный переход, чтобы программа работала "правильно".

Вообще говоря, защиты с привязкой к диску относятся к самым легким, здесь трудно придумать что-то особо сложное по сравнению, например, с защитой по регистрации. Кстати, путь установленного диска может извлекаться из реестра, если он был предварительно туда записан во время установки (это не является хорошим вариантом, если, например, в систему добавляется еще один винчестер). Конечно, существуют "крутые" дисковые защиты, которые считывают специальные метки с поверхности диска, остаются в памяти и проверяют наличие компакта во время работы программы. Для защит подобного рода в Интернете есть специальные патчи, которые их обходят. При желании и навыках, вы можете и сами их создавать.



Защита информации

О.ВАЛЬПА,

E-mail: sandh@yandex.ru

Для блокирования несанкционированного доступа к информации на компьютере, часто прибегают к различным видам защиты. В этой статье я предлагаю читателям познакомиться с одним из простейших способов защиты — шифрованием информации — на примере моей собственной программы.

Необходимость защиты возникает, например, при использовании электронной почты или передаче файлов на магнитных носителях (дискете или жестком диске) через третьих лиц. Кроме того, нередко на компьютере общего пользования или доступном по локальной сети приходится хранить файлы с конфиденциальной информацией (паролями, логическими именами, электронными кодами, банковскими реквизитами, номерами счетов и др. секретными и трудно запоминаемыми данными). Как в таких случаях застраховаться от возможности рассекретывания? Один из простейших вариантов — зашифровать файлы с конфиденциальной информацией. Именно для этих целей я и разработал предлагаемую программу под названием "secret". Несмотря на то что я приведу в данной статье весь алгоритм ее работы, и даже исходный текст, она не утратит своего главного качества — способности надежного засекречивания любой информации. Но об этом чуть позже. В настоящее время существует множество способов шифрования, имеется даже стандартное шифрование информации [1]. Однако я хочу предложить свой вариант.

В основу алгоритма программы заложено кодирование каждого символа шифруемого файла с помощью пароля. Кодирование символов производится простой операцией "исключающее ИЛИ" с одним из символов введенного пароля. Так, первый символ из файла кодируется первым символом пароля, второй — вторым, и т.д. После перебора всех символов пароля, кодирование производится снова с первым символом пароля, и так далее по циклу. Таким образом, каждый символ файла кодируется неизвестным секретным символом из пароля. Не зная пароля и тем более его длины, практически невозможно расшифровать файл простыми способами. Вот почему, даже зная алгоритм программы, нельзя рассекретить файлы, зашифрованные данной программой. Все дело в пароле, который будет индивидуальным для каждого из пользователей, пожелавших воспользоваться данной программой.

Дешифрирование информации из файла происходит по тому же алгоритму и с помощью этой же программы "secret"! Дело в том, что если дважды выполнить операцию "исключающее ИЛИ" над любым символом с помощью одного и того же числа, то символ преобразуется к своему исходному значению. Это и происходит в программе. Файл, один раз обработанный программой "secret", второй раз обрабатывается ею же с использованием пароля, применявшегося для шифрования файла. В результате файл дешифруется и становится доступным для просмотра. Таким образом, зашифровав файл, вы можете хранить его в доступном месте, не опасаясь, что он будет кем-то несанкционированно просмотрен. Вы даже можете передать этот файл своему адресату в зашифрованном виде, не опасаясь, что кто-либо перехватит и просмотрит его "в пути". Сообщив пароль

для дешифрирования файла адресату другими средствами (например, по телефону), вы будете уверены, что, кроме адресата, никто не сможет получить доступ к информации, скрытой в пересылаемом файле.

Еще одно замечательное применение описываемой программы — возможность обеспечения безопасного хранения файлов с конфиденциальной информацией в электронном почтовом ящике или на собственной страничке в Internet. В этом случае, находясь в любой точке мира и получив доступ к Internet, можно без труда просмотреть всю секретную информацию из вашего файла. Для этого зашифрованный файл с этими данными и программой "secret" предварительно пересылается на ваш почтовый ящик или на Web-страничку, после чего он становится доступным из любой точки мира, где есть доступ к Internet. И для того чтобы воспользоваться данными из этого файла, вам остается лишь скачать его вместе с программой "secret" и расшифровать.

Итак, о самой программе. Она написана на языке "Си", но может быть легко перенесена на любой другой алгоритмический язык. Пользоваться программой очень просто. Например, у вас есть файл с паролями или другой секретной информацией, который необходимо защитить. Возможно, это секретный файл с чертежом из графического пакета или даже исполняемый файл, причем формат файла может быть любым. Файл может быть создан обычным текстовым редактором или специальными программами типа Word, Excel, Acrobat и т.д. Для защиты этого файла надо его зашифровать. Для этого необходимо придумать пароль, состоящий из любых знаков, цифр или букв, которые могут быть набраны на клавиатуре компьютера. Длина пароля может быть от одного до тысячи символов. В принципе, это значение можно еще увеличить, заменив соответствующую строчку в программе, но мне кажется, что запоминать и набирать очень длинный пароль весьма неудобно. Паролем может быть одна цифра или целое предложение (например, строчка из стихотворения или пословица). Далее необходимо запустить программу "secret" с именем засекречиваемого файла в качестве аргумента. После этого программа запросит у вас пароль и произведет шифрование файла. Зашифрованный файл будет сохранен с именем, отличающимся от имени исходного файла расширением ".shf". Если имя исходного файла уже имеет расширение ".shf", то программа воспримет его как зашифрованный файл и расшифрует его, восстановив исходное расширение.

Например, имя файла, который вы хотите защитить — passwords.txt. Для выполнения шифрования выполняем команду:

```
>secret.exe passwords.txt
```

После ввода по запросу пароля, программа зашифрует этот файл. В результате появится зашифрованный файл с именем passwords.shf. Этот файл зашифрован, и его вместе с программой "secret" можно смело пересылать или передавать адресату, не боясь, что он будет просмотрен кем-либо посторонним.

Адресат, получив файл passwords.shf и зная пароль зашифровки файла, который вы сообщили ему по телефону или другим способом, вновь должен выполнить программу "secret" для расшифровки файла:



```
>secret.exe passwords.shf
```

После ввода пароля, программа расшифрует этот файл. В результате появится файл с именем passwords.txt, который будет доступен для просмотра.

Для наглядности работы программы (на рис.1 и рис.2 соответственно) приведено содержимое незашифрован-

Рис. 1 Пароль1=qwertу
Пароль2=@123456789
Пароль3=йццкен

Рис. 2

ного и зашифрованного с помощью пароля "1" файлов. Из этих примеров видно, как трудно расшифровать содержимое файла, зашифрованного даже с таким простым паролем.

Исходный текст программы:

```

//
=====
// Программа secret.exe шифрует и дешифрует файл, используя
// пароль пользователя
// Автор: Valpa Oleg
// Дата: 01.11.2003
// Версия: 1.0
//
=====

#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>

int argc;
char **argv;

int main(int argc, char **argv)
{
    int i,l,c,cl,n,shf=0;
    FILE *file_inp, *file_out;
    register char **p;
    char pw[1000],namef[80];

    printf("\nПрограмм SECRET v1.0 Copyright (C) O.D.Valpa
        01.11.2003\n");
    if(argc == 1)
    {
        printf("\nПрограмма secret.exe автоматически шифрует
                                                    или дешифрует файл,");
        printf("\nиспользуя вводимый пароль пользователя.");
        printf("\nПример использования программы:
                                                    secret.exe [имя файла]");
        printf("\nНажмите 'Enter' для завершения...");
        getch(stdin);
        return 0;
    }

    while((file_inp=fopen(argv[1],"rb"))==NULL)
    {
        printf("\nНе могу открыть файл с именем:%s!",argv[1]);
        printf("\nВведите правильное имя входного файла или
                                                    Ctrl+C для выхода:\n");
        scanf("%s",argv[1]);
    }

    strcpy(namef,argv[1]);

    n=0;

```

```

do
{
    if( (argv[1][n]=='.') || (argv[1][n]==0) ) break;
    n++;
}
while(1);

if( ((argv[1][n+1]=='s')||(argv[1][n+1]=='S')) &&
    ((argv[1][n+2]=='h')||(argv[1][n+2]=='H')) &&
    ((argv[1][n+3]=='f')||(argv[1][n+3]=='F')) )
{
    shf=1; // Файл зашифрован
    for(i=1;i<=3;i++)
    {
        if((c=getc(file_inp))==EOF)
        {
            printf("Входной файл не является зашифрованным
                    программой secret!");
            printf("\nНажмите 'Enter' для завершения...");
            getch(stdin);
            exit(0);
        }
        namef[n+i]=c;
    }
}

if(shf==0) // Если файл не зашифрован
{
    namef[n+1]='s';
    namef[n+2]='h';
    namef[n+3]='f';
}

file_out=fopen(namef,"w+b");

if(shf==0) for(i=1;i<=3;i++) putc(argv[1][n+i],file_out);

printf("\nВведите пароль шифрования (от 1 до 1000 любых
                                             символов):\n");

scanf("%s",pw);
l=strlen(pw);

i=0;
do
{
    if((c=getc(file_inp))==EOF) break;
    cl = c ^ pw[i];
    putc(cl,file_out);
    i++;
    if(i==l) i=0;
}
while(1);

if(shf==0)
    printf("\nЗашифрованный файл сохранен с именем %s.",namef);
else
    printf("\nРасшифрованный файл сохранен с именем %s.",namef);
printf("\nНажмите 'Enter' для завершения...");
getch(stdin);
getch(stdin);

return 0;
}

```

При желании можно усложнить процедуру шифрования, например, применив специальный алгоритм перебора символов из пароля, или добавив другие хитрости, тем самым повысив степень секретности шифрования. Но даже в приведенном виде программа имеет достаточно высокую степень защиты от “взлома” и позволяет сразу же приступить к защите ваших данных.

Литература

1. ГОСТ 28147-89.

Режим UDMA в Windows XP

А.ГОРЯЧКИН,

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

Не так давно на моем "компе" во время просмотра видеофильма произошел сбой в работе операционной системы Windows XP, что само по себе является большой редкостью. После перезагрузки начался долгий процесс полной проверки жесткого диска. Поскольку "винт" у меня не маленький (Seagate 60 Гб), и, к тому же, прак-

тессора. От режима DMA режимы UDMA 33/66/100/133 отличаются тем, что в них скорость обмена повышена за счет передачи данных, как по переднему, так и по заднему фронту сигнала чтения или записи. Кроме того, в режиме UDMA используется механизм обнаружения ошибок и верификации при передаче и хранении данных с помощью циклического контрольного кода (CRC).

Итак, после сбоя система решила, что оптимальным режимом для моего "винта" вместо UDMA будет режим PIO (PIO — программируемый ввод/вывод). Меня, конечно, это не устраивало, но все мои попытки включить режим UDMA на "винте" не приводили к положительному результату. Переустановку системы и реформатирование "винта" я отложил на крайний случай, как оставляют последний патрон, отстреливаясь от наседающего врага. Мириться с создавшимся положением дел мне не хотелось,

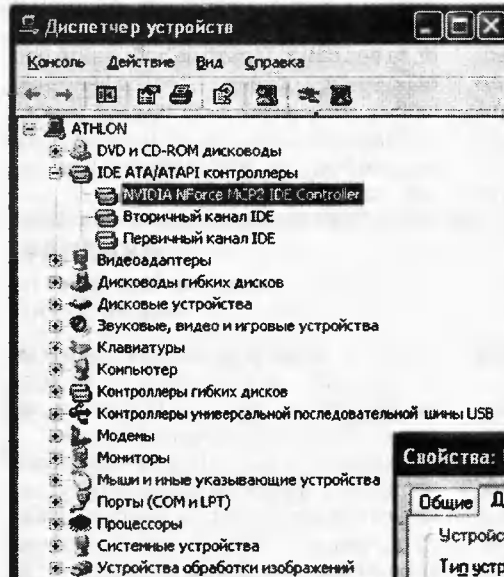
нием "NVIDIA nForce MCP2 IDE Controller" (рис.1). Именно этот контроллер в чипсете nForce2 отвечает за работу жестких дисков, CD-ROM и других устройств с интерфейсом IDE. Пугаться этого не стоит, поскольку удаление происходит, как вы сами понимаете, не на аппаратном, а на программном уровне. Проще говоря, мы совершаем обычное удаление драйверов, с последующей их установкой самой операционной системой в автоматическом режиме без запроса вставить какие-либо компакт-диски.

Позволив мне удалить драйверы, "эпишника" молча проглотила обиду и предложила компромиссный вариант — подобно "Матрице-2", произвести перезагрузку. После чего система выдала сообщение о том, что найдены новые устройства — жесткий диск, CD-ROM и т.д. После окончания загрузки я не замедлил проверить, какой режим установился на "винте" после моей атаки на Windows XP, и был несказанно обрадован тем, что теперь все стало на свои места (рис.2).

Запустив для пущей убедительности утилиту Sandra 2002, я загрузил модуль File System Benchmark и остался вполне доволен результатами тестирования. Проверить включение режима UDMA можно также используя утилиту Nero InfoTool, кликнув по вкладке "Configuration" (рис.3).

Таким образом, мне удалось решить свои проблемы с компьютером "малой кровью", поскольку я обошелся без переустановки операционной системы,

Рис. 1



тически до отказа забит "мягкой рухлядью", именуемой файлами, работа Scandisk'a заняла продолжительное время. Но это еще, как говорится, только цветочки. После окончания проверки "винта", операционная система выдала подробный отчет, в который, помимо всего остального, было включено сообщение о двух найденных плохих секторах (bad sectors), что очень меня огорчило. И вдобавок ко всему, после сбоя работа диска заметно замедлилась, что подтвердилось тестом утилиты Sandra 2002 и отсутствием режима UltraDMA (Мой компьютер — Свойства системы — Оборудование — Диспетчер устройств — IDE ATA/ATAPI контроллеры — Первичный канал IDE). Что характерно, BIOS при загрузке нормально распознавала жесткий диск, его объем и активный режим UDMA100.

Здесь я должен сделать небольшое лирическое отступление и напомнить пользователям о том, что такое режим DMA, и для чего он нужен. DMA (Direct Memory Access) — прямой доступ к памяти. В режиме прямого доступа обмен данными между оперативной памятью и периферийными высокоскоростными устройствами происходит через каналы DMA без использования ресурсов центрального про-

и я стал искать пути решения этой проблемы, хотя мысли о покупке нового "винта" уже начали закрадываться в мою голову. Подключив "винт" к другому "компю", режим UDMA на капризном "винте" все же удалось включить. Но при возврате "винта" на "родное" место, режим UDMA по-прежнему устанавливаться не хотел. Поскольку терять мне было нечего, я постарался нанести ответный удар по Windows XP и удалил из системы в списке "IDE ATA/ATAPI контроллеры" устройство под назва-

Рис. 2

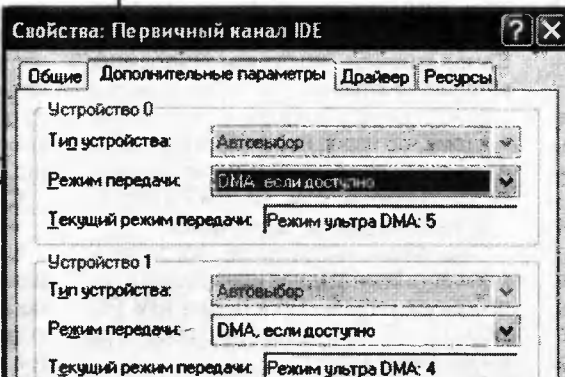
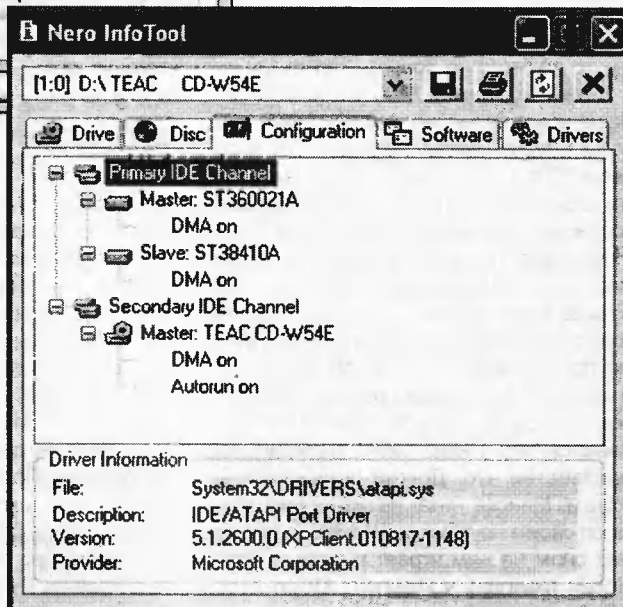


Рис. 3



переформатирования жесткого диска, и тем более, избежал потери практически



всех данных, хранившихся на "винте" (были повреждены всего два видеофильма). Кроме того, мне не пришлось идти на финансовые затраты, связанные с покупкой нового винчестера.

Так что, после различных сбоев, при заметных замедлениях в работе устройств с интерфейсом IDE, весьма полезной будет привычка периодически проверять, включен ли на них режим UDMA. Тем более, замечено, что приводы CD-ROM и CD-R/RW часто самопроизвольно устанавливаются в режим PIO. И разумеется, при отсутствии режима UDMA можно и нужно самим включать его для ускорения работы жестких дисков, CD-ROM и других устройств с интерфейсом IDE.

Анализируя причины происшедшего с моим "винтом", я пришел к выводу о том, что, скорее всего, при появлении сбоев в работе винчестера, Windows XP авто-

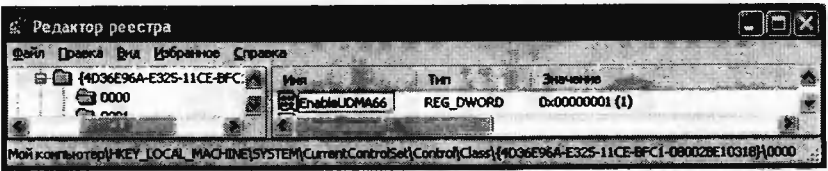


Рис. 4

матически понижает скорость работы жесткого диска, принудительно устанавливая вместо режима UDMA менее скоростной режим PIO.

Дополнительно к вышесказанному могу сообщить следующее. В операционных системах Windows 2000/XP имеется возможность включить режим UDMA66. Для этого надо немного поколдовать в системном реестре — запустить редактор реестра (Regedit.exe) и выбрать раздел [HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Class\{4D36E96A-E325-11CE-BFC1-08002BE10318}\0000]. Создайте в этом

разделе новый параметр типа REG_DWORD и назовите его EnableUDMA66. Установите его значение равным 1, как показано на рис.4. Пользователям, которые не любят сами ковыряться в системном реестре, я советую для установки режима UDMA66 использовать утилиту Customizer XP (<http://www.tweaknow.com>), которая все сделает за вас сама. Только не забывайте про одно важное правило — после внесения любых изменений в системный реестр необходимо произвести перезагрузку компьютера, для того чтобы эти изменения вступили в силу.

Теория "downgrade"

Каждый сходит с ума по-своему! (кто-то очень знающий)

Не подумайте ничего плохого, читая заголовок. Просто сегодня я хочу рассказать о новом веянии в компьютерном мире — downgrade, сторонником которого являюсь и я. Если говорить коротко, суть этого движения в следующем — берем самый "крутой" компьютер, ставим на него старые программы (а лучше, если еще сделать так, чтобы эти самые программы тормозили) и стараемся при их помощи выполнять задачи, как и при обычной работе компьютера. То, что все это называется downgrade, я узнал относительно недавно, хотя подобными экспериментами занимаюсь уже не первый год, просто буржуи на западе дали имя этому явлению, а у нас как-то до этого не дошло. А жаль...

Зачем это нужно? Кому это интересно? Чаще всего возникают именно такие вопросы. Ответу. Никому не нужно и не интересно! Это просто такое хобби, такое же, как собирать марки или коллекционировать всяческие произведения искусства. Так и "downgrade" при всем желании не станет массовым помешательством. Хотя... Все в наше время возможно. Например, мне известен случай, когда один юзер из Минска просто ради интереса собрал 486 IBM и каким-то неведомым природо способном всунул туда Windows 98. И после этого радовался, что у него *.MP3 тормозит, как неизвестно что. Другой пользователь, тоже из Минска, никак не может расстаться со своим ноутбуком IBM 386, даже перед сном на нем играет в старые игры. Некий товарищ, у которого Athlon 1700, десятками скачивает старые программы и доказывает всем на форуме, как это интересно — играть в старые добрые игры.

С периодичностью в несколько месяцев в наших СМИ появляются публикации про DOS, хотя, казалось бы, практического повода для их появления уже нет. Там освещаются вопросы давно минувших дней. Самое странное, что отзывы, приходящие авторам, заставляют по письмам писать еще несколько публикаций. К чему же это? А к тому, что народ никак не забудет XT- и AT-машины, которые неожиданно становятся дополнением к MAC G4 или Pentium 4.

В лаборатории "downgrade"

Свои опыты я ставлю не просто так, у меня есть благородная идея собрать все программы 1981...1994 гг. для IBM PC. Нереально? Ну и пусть! Зато работы до конца жизни хватит.

Самым главным инструментом "downgrade" является компьютер. Лучше, если их два — старенький класса 286-й IBM и новый — класса Pentium 4. Лично я провожу свои "зверские" эксперименты на двух машинах такой конфигурации: IBM 286, 4 Мб RAM, 160 Мб HDD, EGA и AMD K6-2 450, 128 Мб RAM, 20 Гб HDD, XGA. Поверьте, скромненько, но со вкусом.

Так что же можно сотворить, имея два таких компьютера? В принципе, очень многое. Взять, к примеру, локальную сеть. Можно играть в игры, а можно пугать друзей разнообразием коллекции иконок на жестком диске IBM 286. Кстати, провести локальную сеть под MS-DOS — это целое искусство. Ведь это даже не Nowell, и тем более, не Windows. Попотеть приходится изрядно. Но зато в конце работы вы с благодарностью печатаете "net view", и у вас на EGA-мониторе появляется ваш драгоценный Pentium. Ну, чем не развлечение? Тем более, что практическая отдача от вашего опыта очень большая — чем с большим количеством старых приложений вы рабо-

таете, тем меньше проблем потом возникает с более навороченными их потомками для Windows, ведь идея остается та же, только интерфейс полностью другой.

Теперь хочу перейти к практической части, т.е. к вопросам, которые будут касаться каждого вашего "downgrade". Между прочим, для этого у нас используют два компьютера, а на западе только один. Так что мы хоть в чем-то более "продвинутые".

Соединение компьютеров через порты

Компьютеры соединяют через порты в разных ситуациях. Например, вам нужно перенести несколько десятков или сотен мегабайтов информации на другой компьютер, а у вас нет ни привода CD-ROM, ни сетевой карты. Или вы желаете совместно использовать какие-либо файлы или группы файлов.

Да и стоит ли покупать привод CD-ROM, если ваш компьютер приблизительно равен ему по цене? Конечно, нет! Здесь нам приходят на помощь старые добрые DOS-утилиты Interlink и Norton Commander. Кому что по душе. Я поделюсь своим опытом по работе с каждой из них. Начну с несколько более сложной, но функциональной программы Interlink, которая входит в поставку комплекта MS DOS 6.22. Она способна организовать связь между компьютерами, связанными через параллельные и последовательные порты.

Последовательные порты соединяются последовательным семижильным нуль-модемным кабелем или таким же трехжильным. Параллельные порты связываются двунаправленным параллельным кабелем. Последовательный кабель должен иметь 9- или 25-штырьковые разъемы на каждом из его концов, а параллельный должен иметь на каждом

Р.КАРПАЧ,

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



конец разъем типа DB-25. При этом его кабель образует 11 жил, связывающих штырьки двух разъемов.

Если вы определились с кабелем, то можно переходить к работе с самим программным обеспечением. Вам следует знать или, вернее, сделать следующее. При создании сети при помощи программы Interlink, по умолчанию, как только вы установите соединение с другой машиной, ее диски становятся доступными для выполнения большинства команд. Более того, вы можете запускать программы с другого компьютера, одним словом, действуют все принципы локальной сети. Что же нужно, чтобы заставить работать утилиту Interlink? Самое первое и главное условие — это одинаковые версии Interlink на ведущем и ведомом компьютере. Если это условие соблюдено, то проблем не должно возникнуть. Итак, в файл Config.sys компьютера-клиента впишем следующую строку:

```
DEVICE(HIGH)=C:\DOS\interlnk.exe /  
DRIVERS:8 /NOPRINTER /COM:1 /AUTO.
```

А теперь расшифруем.

DEVICE — по умолчанию загрузит драйвер в основную память.

DEVICEHIGH — загрузит драйвер в расширенную память, при условии, что такая имеется, и загружен драйвер расширенной памяти himem.sys.

/DRIVERS — указывает количество дисков, которые могут быть доступны для чтения. Если вы выставите 0, то ни один из накопителей компьютера сервера не будет доступен.

/NOPRINTER — эта запись обеспечивает использование на компьютере-клиенте своих портов вместо портов сервера.

/COM:1 — указывает порт, через который проводится соединение.

/AUTO — включает драйвер тогда, когда другой компьютер готов к сеансу связи. Также этот переключатель реально экономит вашу память. Ведь он включается только тогда, когда другая машина к связи готова. Это очень важно на компьютерах с процессорами 8088 и ОЗУ 1 Мб.

А что же сделать на компьютере-сервере? На головной машине мы запустим программу INTERSVR.EXE, в моем случае с ключом /COM2. Вы увидите красивую таблицу, в которой будет указан порт, через который идет соединение, накопители одного и второго компьютера и их объемы.

Теперь, что касается оболочки Norton Commander. Идеология связи в ней предельно упрощена, но при этом нет возможности запуска программ на удаленном компьютере. А ее функциональность сводится лишь к банальному переносу файлов. Для связи при помощи Norton Commander нужно соблюсти два условия:

- использовать одну и ту же версию оболочки на обеих машинах;
- знать, через какие порты вы собираетесь подключаться. Смешно звучит, однако, встречаются пользователи, которые не понимают, куда и что они втыкают.

Чтобы соединиться, выберите в верхнем меню правой или левой панели опцию "Связь". В ней укажите, через какой именно порт вы желаете подключиться. Также выберите режим связи — "Ведущий" или "Ведомый". На второй машине произведите аналогичные действия, а затем на обоих компьютерах нажмите кнопку "Связь".

Си-ди-ромчик

Для транспортировки информации на АТ-машину можно использовать и CD-ROM. Есть два варианта подключения: через шлейф, параллельно с HDD, или через звуковую карту с контроллером IDE. Мне везет, и я пользуюсь первым методом. Аппаратно у меня никаких проблем не возникает, а вот программно есть трудности с установкой драйвера CD-ROM под DOS, хотя на самом деле это не сложно. В папке C:\DOS у вас находится утилита mscdex.exe. А на диске, что вам дали с CD-ROM, драйвер, в моем случае он называется gscdrom.sys. Я его копирую в папку C:\DOS и пишу в файле autoexec.bat

```
C:\DOS\MSCDEX.EXE /S /d:mscd001
```

и, соответственно, в config.sys

```
DEVICE=C:\DOS\gscdrom.sys /d:mscd001.
```

Все, проблема решена! Перезагрузившись, можно начинать работу с CD-ROM на АТ. Но проблема может возникнуть в другом, я никак не мог ее решить. На 386-й машине был установлен не IDE HDD, а, может, помните, такой "блин" — Segate ST251 на 40 Мб. Так вот, при установке звуковой карты с IDE-контроллером, которая так хорошо работала на 286-й, работа с CD ROM не получилась. Так что хочу заранее всех предупредить, что с такой конфигурацией могут возникнуть проблемы.

Память — не склероз

На 286-й материнской плате имеется проблема с памятью, а вернее, с ее установкой. Установив планки по 1 Мб во все слоты, при включении компьютера я не заметил никакого теста, и в BIOS памяти тоже не было. Облавив множество сайтов и опросив с полсотни знакомых, я так и не нашел ответа на вопрос, куда делись мои 4 Мб. И вот, когда я в очередной раз разобрал корпус, меня "пробило". Вокруг самих банков памяти находилась группа одинаковых чипов. На свой страх и риск я выковырял их отверткой. Сразу предупреждаю, не перепутайте с микросхемами BIOS, а то я опасаюсь, что некоторые индивидуумы могут это сделать. Когда я включил компьютер, чудосвершилось — при загрузке высветилось 3798 Кб RAM. Так старенький IBM заимел еще 4 Мб памяти.

Наш дом — DOS!

Поскольку все же IBM 286 — компьютер с 16-разрядным процессором, то, следовательно, и DOS нужно оптимизировать для работы именно с ним. И главное, не дать ему забыть, что у него есть еще и расширенная память! Запустив NC, мы видим, что диск работает не очень шустро, и что при открытии папки с большим ко-

личеством файлов он что-то долго "хрустит", а уж затем выдает результат. Нет, здесь проблема не в слабом 12-мегагерцовом процессоре, она просто в том, что вы не настроили кэши и кое-что еще. Поэтому я приведу примеры конфигурации файлов своего компьютера. Напомню лишь, что он обладает 12-мегагерцовым процессором и 4 Мб RAM.

Autoexec.bat:

@ECHO OFF — отключает эхо-отображение запуска командных файлов.

C:\keyrus.com — драйвер русской клавиатуры для DOS.

C:\mouse.com — драйвер мыши (кстати, бывают и такие, которые подключаются из файла config.sys).

C:\DOS\smartdrv.exe /x 1024 512 — запуск администратора кэша с начальным размером 1024 Кб и минимальным 512 Кб. Эта программа кардинальным образом увеличивает быстродействие дисковой подсистемы, но при условии, что ваш жесткий диск имеет маленькую буферную память, как у меня, иначе это может вызвать противоположный эффект. К тому же, если у вас меньше 1 Мб RAM, лучше smartdrv не использовать, а пользоваться строкой buffers в файле config.sys.

PATH C:\WIN3.1;C:\DOS — эти параметры вам пропишет Windows 3.1 после своей установки. В данном случае это облегчает запуск "окон" из любого каталога командой win.

C:\SHELLS\NC4\nc.exe — запуск оболочки Norton Commander 4.0.

Config.sys:

DEVICEHIGH=C:\DOS\himem.sys — devicehigh, в отличие от команды device, загружает драйвер в расширенную память. В данном случае загружается драйвер расширенной памяти himem.sys, без которого никак нельзя. Для компьютеров с 386-м процессором имеется отличная утилита Microsoft memaker, которая может оптимизировать память автоматически.

NUMLOCK=OFF — эта команда отключает NumLock при загрузке.

DEVICEHIGH=C:\DOS\SETVER.EXE — без этой утилиты, на мой взгляд, никак нельзя, так как часто встречаются программы, которые пишут Incorrect Dos version. К ним относится и Windows 286 или 1.04. В драйвере сразу записана таблица файлов с версиями.

DOS=HIGH — параметр HIGH/LOW определяет, должна ли MS-DOS пытаться загружаться в HMA (HIGH), или загружать все свои компоненты в обычную память (LOW). По умолчанию устанавливается LOW.

FILES=20 — количество файлов, с которыми одновременно может работать программа.

BUFFERS=30 — этот параметр особенно нужен, когда у вас менее 1 Мб оперативной памяти, и, следовательно, вы не можете организовать эффективное кэширование дисков. Вот вам и прихо-



дится заниматься буферизацией.

FCBS=1 — задает число управляющих блоков файлов (FCB), которые MS-DOS может одновременно открыть. Эту команду можно использовать только в файле CONFIG.SYS. FCB — это структура данных, в которой хранится информация о файле.

SHELL=C:\DOS\COMMAND.COM
C:\DOS\ /p — команда указывает путь к командному интерпретатору command.com.

Очень важно!!! Не форматируйте жесткий диск для вашей 286-й машины с загрузочной дискеты Windows 98!!! Это вызовет лишь одни проблемы и с Norton Utilities, и с Windows 3.1. Не поленитесь сделать загрузочный диск DOS. Сделайте все, что надо, с него. Затем скопируйте установку системы на диск, запустите install.exe и помните, что вам будет нужна еще одна пустая дискета, так как у "старушки" DOS мания создавать себе копии.

Что-то я все про 286-й. А что же можно угодить на Pentium?

Раскрасим консоль DOS

Для этого вам нужно из комплекта Norton Utilities 8.0 использовать маленькую полезную программку be.exe. В файле autoexec.bat напишите: C:\NU\BE SA GREEN — и цвет букв в вашей консоли окрасится в зеленый цвет. Также с помощью be.exe можно писать мелодии по нотам, которые будут проигрываться через динамик.

Дискеты 720 Кб на 800 Кб

Дискеты на 800 Кб? Реально. Утилиты 800.com или ru1700.com можно скачать на www.fdd5-25.net. Только предупреждаю, что некоторые BIOS будут с ними работать некорректно. Впрочем, если не поленились, то сами прочитаете мануал. Еще бы порекомендовал вам утилиту Fformat. Она работает не из командной строки DOS, имеет приятный интерфейс, а также огромный выбор форматов дисков.

Интернет под DOS

Под DOS тоже имеется некоторое количество браузеров, один из них — DOS Internet browser for Dial-Up PPP Connection (www.Nettamer.net). Поэтому даже для моего IBM 286 существует возможность выхода в сеть.

Также я бы рекомендовал вам посетить сайт <http://kiarchive.relocm.ru>, в этом архиве имеется огромная коллекция разных утилит для Windows 3.1. Для IBM 286 рекомендую текстовый редактор E+, а также просмотрщик картинок View It (он без проблем работает с EGA-монитором). Там же имеется прекрасный архиватор, который, судя по всему, пришел на IBM с MAC — это Aladdin archiver for Windows 3.1. Мне он больше по душе, чем пресловутый RAR.

Помните старый "КОРБЕТ"? Тогда загляните на сайт www.Cpm.z80.de. Там представлена огромная коллекция программ для CPM DOS, а также эмуляторы, которые без труда работают на "двойке".

Если подумать, получается парадоксальная ситуация. При помощи самых совре-

менных технологий мы можем дать вторую жизнь даже самым старым компьютерам, на которых, в свою очередь, мы сможем эмулировать еще более старые системы. В этом и парадокс "downgrade" в практике!

Практический DOS для WINDOWS

Мне никогда не нравилось, да и вам, полагаю, тоже, что при перезагрузке Windows в режим эмуляции MS-DOS появляется "несчастный" command.com, лишенный всяких изысканий. Вот что я вам предлагаю сделать. В папке Windows, создать файл dosstart.bat. В нем прописать следующие строки:

```
ECHO OFF
ECHO y >yes
del C:\Windows\win386.swp <yes
del yes
{удаление Swar файла, при каждой
перезагрузке в режим DOS}
C:\windows\smartdrv.exe {каширование диска}
{также здесь вы можете прописать команду на
запуск драйвера звука под MS-DOS}
C:\mouse.com {драйвер мыши, который вы сами
поставите}
C:\Nc\nc.exe {загрузка оболочки Norton
Commander}
```

В последнее время я заметил, что если неизощренный пользователь попадает в режим эмуляции MS-DOS, то он даже не знает, что делать. Поэтому я приведу список базовых команд, которые вам помогут перемещаться по диску.

DIR — показывает содержимое каталога или диска.

MKDIR или **MD** — создание каталога.

CHDIR — отображает имя или изменяет текущий каталог.

CD — переключение по папкам, например, CD games. Наберите CD <диск>, чтобы узнать имя текущего каталога на указанном диске. Наберите CD без параметров, чтобы узнать текущий диск и каталог. Наберите CD.. — и окажетесь в корневом каталоге.

RMDIR или **RD** — удаление каталога.

COPY — команда копирования.

DEL или **ERASE** — удаляет файл или группу файлов.

RENAME или **REN** — переименовыва-ет файл или группу файлов.

Чтобы подробнее узнать о возможностях какой-либо из команд DOS, следует набрать интересующую вас команду с ключом /?.

Выбираем древнюю материнскую плату!

Покупая новые комплектующие, нужно быть внимательным! А покупая старые, нужно быть внимательным вдвойне. Основные беды материнских плат:

-непропаи — распространенный дефект, как правило, часто встречается в изделиях производителей, у которых нарушен техпроцесс при изготовлении плат. Проявляется при малейшей деформации платы, к примеру, при установке в некачественный корпус с отклонениями размещения точек крепления (как говорится, "в корпус вставляю — не работает, на столе собираю — работает"). Это довольно сложный дефект, т.к. при огромном количестве навесных эле-

ментов, их миниатюрности и многослойности платы (также встречаются непропаи внутри чипов) найти точку глюка довольно проблематично. В этом случае иногда помогает метод "глобального потепления" — прогрев платы до температуры плавления припоя в месте предполагаемого непропая. Метод — варварский, и требует опыта и аккуратности, чтобы не привести к более плачевному результату;

-разрушение BIOS — происходит в большинстве случаев из-за активизации вирусных программ, очумелых ручек пользователей, реже — из-за выхода из строя флэш-микросхемы. Для лечения нужно иметь программатор, поддерживающий запись данного типа микросхем, или, как худший вариант — материнскую плату с аналогичным типом флэшки на панельке и программу-прошивальщик (awdflash, amiflash). Для "родной" платы программирование осуществляется "на горячую" — запускаем плату с "родной" флэшкой, затем аккуратно вынимаем флэшку, вставляем программируемую микросхему и запускаем соответствующую программу-прошивальщик. Могут возникнуть проблемы с некоторыми программами-прошивальщиками в виде отказа "заливать" неродную для этой платы версию прошивки BIOS;

-механические повреждения — от чего и почему они появляются, понятно каждому. Здесь главное — быть внимательным при осмотре. Ведь, казалось бы, маленькая и безобидная царапина на самом деле могла отрезать пару дорожек. Чаще всего повреждения встречаются около отверстий точек крепления платы, крепления радиатора процессора, также "страдают" разъемы памяти, слоты CPU, PCI, ISA и т.д.;

-неисправности электрических цепей — происходят, как правило, из-за превышения питающих напряжений (к примеру, неисправен блок питания), подключения неисправных устройств или подсоединения плат расширения и различных девайсов "на ходу", неисправностей кабелей подсоединения внешних устройств (из-за переломанного провода в кабеле мышки, клавиатуры или принтера запросто загибается мульти-чип на материнской плате) и нередко из-за непродуманных инженерных решений производителя и некачественных радиоэлементов. Иногда эти дефекты можно наблюдать визуально, в виде вздувшихся конденсаторов, следов горения элементов либо сильного нагрева чипов.

Эпилог

То, что вы прочитали — лишь малая часть тех сумасшедших экспериментов, которые я проводил со своими компьютерами. Может, кому-то показалось, что в них нет смысла. Ну, а какой должен быть смысл, кроме практической пользы для ума? Ведь программы, выпущенные десять, а то и пятнадцать лет назад, намного сложнее заставить работать, чем новые, гладкие и красивые, Windows-приложения.



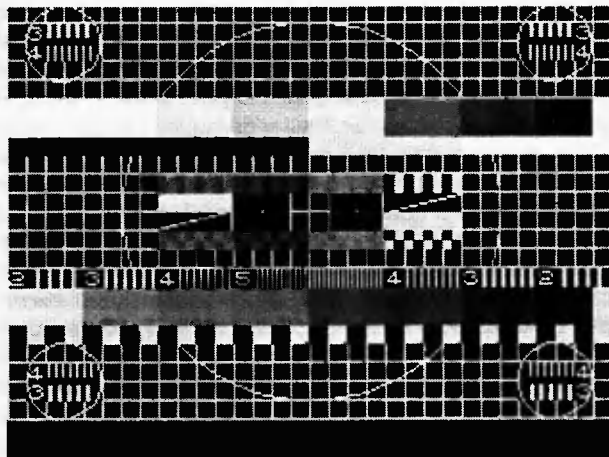
Телетест на Sinclair — 2

Е.ИЛЯСОВ,
г. Балашов,
Саратовской обл.

В ноябрьском номере журнала за 2003 г. был предложен программный способ генерации аналогов телевизионных испытательных сигналов. Приведенная в статье [1] программа "TV-test" реализована на Sinclair-совместимых компьютерах и формирует необходимый и достаточный комплект испытательных сигналов для разносторонней оценки параметров изображения устройств видеовывода. К ее недостаткам можно отнести отсутствие среди генерируемых сигналов комплексного испытательного изображения, позволяющего в совокупности оценить самые разнообразные видеопараметры.

За время, прошедшее после разработки "TV-test", у членов нашего компьютерного кружка была возможность ознакомиться еще с некоторыми программами подобного назначения. Среди них наше внимание привлекла программа "RGB test" (В.В.Нагишев, г.Москва, 1990 г.), где в числе прочих формировалось изображение, напоминающее универсальную электрическую испытательную таблицу УИЭТ. Фрагмент этой программы, генерирующий такое изображение, послужил прототипом для доработки программы "TV-test", которая позволяет вместо малоинформативного испытательного сигнала, выводимого на экран при выборе пункта "Select table" \ "Special" [1], генерировать вполне удовлетворительный аналог универсальной электрической испытательной таблицы (рис.1).

Рис. 1



Вашему вниманию предлагается листинг этой доработки для самостоятельного набора, который допускает двоякое использование. Во-первых, приводимый текст программы вполне самостоятелен и может работать автономно, в качестве отдельной программы-генератора испытательного телесигнала. А во-вторых, сохранив набранный и отлаженный фрагмент на внешнем носителе, можно подгрузить его к уже набранной и отлаженной программе "TV-test" и получить таким образом ее усовершенствованную версию под условным названием "TV-test+". Это возможно благодаря тому, что нумерация строк приведенного фрагмента является "сквозной" по отношению к базовой программе "TV-test", продолжая и дополняя оригинальную нумерацию.

Листинг "+"

```
3 REM last edition: 2.09.03

20 LET b$="0": LET p$="0": LET i$="7"

120 PRINT TAB 10;"TV TESTING +""

6270 GO TO 7100

6370 GO TO 7100

6400 REM - Special -
6410 BORDER 0: PAPER 0: INK 7: CLS
6420 FOR i=8 TO 255 STEP 8
6430 PLOT 0+i,0: DRAW 0,175
6440 NEXT i
6450 FOR i=8 TO 175 STEP 8
6460 PLOT 0,0+i: DRAW 255,0
6470 NEXT i
6480 CIRCLE 128,88,80
6490 FOR i=0 TO 7
6500 PRINT AT 5,i*4:BRIGHT 1;INK 7-i;"???"
6510 PRINT AT 6,i*4:BRIGHT 1;INK 7-i;"???"
6520 PRINT AT 15,i*4;INK 7-i;"???"
6530 PRINT AT 16,i*4;INK 7-i;"???"
6540 NEXT i
6550 PRINT AT 9,8:PAPER 3;INK 4;"???"PAPER 1;INK 2;"???"
      PAPER 5;INK 6;"???"PAPER 7;INK 0;"???"PRINT AT 10,8;
      "???"": PRINT AT 11,8;"???"
6560 PRINT AT 12,8:PAPER 3;INK 4;"???"PAPER 1;INK 2;"???"
      PAPER 6;INK 5;"???"PAPER 0;INK 7;"???"
6570 FOR i=0 TO 31 STEP 2
6580 PRINT AT 17,i;" ?"
6590 PRINT AT 7,0;" ";BRIGHT 1;" ";BRIGHT 0;
      "??????";BRIGHT 1;"??????"
6600 NEXT i
6610 FOR i=120 TO 136 STEP 8
6620 PLOT i,80: DRAW 0,15
6630 NEXT i
6640 PLOT 120,88: DRAW 15,0
6650 PLOT 108,88: PLOT 148,88
6660 PRINT AT 14,0;"2 3 4 5 4 3 2 "
6670 FOR i=1 TO 4
6680 PLOT i*5+8,56: DRAW 0,8
6690 PLOT i*5+7,56: DRAW 0,8
6700 PLOT i*5+231,56: DRAW 0,8
6710 PLOT i*5+230,56: DRAW 0,8
6720 NEXT i
6730 FOR i=1 TO 6
6740 PLOT i*4+38,56: DRAW 0,8
6750 PLOT i*4+37,56: DRAW 0,8
6760 PLOT i*4+198,56: DRAW 0,8
6770 PLOT i*4+197,56: DRAW 0,8
6780 NEXT i
6790 FOR i=3 TO 24 STEP 3
6800 PLOT i+70,56: DRAW 0,8
6810 PLOT i+166,56: DRAW 0,8
6820 NEXT i
6830 FOR i=106 TO 159 STEP 2
6840 PLOT i,56: DRAW 0,8
6850 NEXT i
6860 PRINT AT 1,1;"3 ";AT 2,1;"4 ";AT 19,1;"4 ";AT 20,1;
      "3 ";AT 1,27;" 3";AT 2,27;" 4";AT 19,27;" 4";
      AT 20,27;" 3"
6870 FOR i=1 TO 5
6880 PLOT i*4+14,162: DRAW 0,6
6890 PLOT i*4+14,8: DRAW 0,6
6900 PLOT i*4+218,162: DRAW 0,6
6910 PLOT i*4+218,8: DRAW 0,6
6920 PLOT i*4+13,162: DRAW 0,6
```



```

6930 PLOT i*4+13,8: DRAW 0,6
6940 PLOT i*4+217,162: DRAW 0,6
6950 PLOT i*4+217,8: DRAW 0,6
6960 NEXT i
6970 FOR i=3 TO 21 STEP 3
6980 PLOT i+14,152: DRAW 0,6
6990 PLOT i+14,18: DRAW 0,6
7000 PLOT i+218,152: DRAW 0,6
7010 PLOT i+218,18: DRAW 0,6
7020 NEXT i
7030 CIRCLE 23,16,16: CIRCLE 23,159,16: CIRCLE 231,16,16:
      CIRCLE 231,159,16
7040 PLOT 63,80: DRAW 32,8
7050 PLOT 160,87: DRAW 32,8
7060 PLOT 95,80: DRAW 0,15
7070 PLOT 159,80: DRAW 0,15
7080 PLOT 0,0: DRAW 0,175: DRAW 255,0: DRAW 0,-175: DRAW -255,0
7100 REM - Change border -
7110 LET b=VAL bs
7120 BEEP .1,10
7130 LET o$=INKEY$: IF o$="" THEN GO TO 7130
7140 IF o$<>" " THEN GO TO main
7150 LET b=b+1: IF b>7 THEN LET b=0
7160 BORDER b: GO TO 7120

```

Если предполагается автономная работа фрагмента, без его присоединения к базовой программе, строки с номерами 6270 и 6370 должны быть при наборе пропущены. Если же этот фрагмент предполагается использовать в качестве дополнения к программе "TV-test", то в этом случае можно опустить при наборе строку с номером 20.

Для построения изображения в программе широко используются символы блочной псевдографики из оригинальной кодовой таблицы компьютеров, совместимых с Sinclair ZX-Spectrum. Так, в строках 6500, 6510, 6520, 6530, 6580, 6590 и в двух последних группах символов в строке 6550 должны быть символы с кодом 143 (#8F — "закрашенный квадрат" ■). В первой и в третьей группах символов в строке 6550 — символы с кодом 133 (#85 — "закрашенная правая половина квадрата" ■), а во второй и четвертой группах символов в этой же строке — символы с кодом 138 (#8A — "закрашенная левая половина квадрата" ■). В строке 6560 в первой и четвертой группах использованы символы с кодом 134 (#86 — "закрашенные верхний левый и нижний правый углы" ■), а во второй и четвертой группах — символы с кодом 137 (#89 — "закрашенные верхний правый и нижний левый углы" ■). Для облегчения набора и снижения вероятности внесения ошибок, на рис.2 приводится "скриншот" (экранный фрагмент), иллюстрирующий "псевдографический" участок листинга.

В заключение хотелось бы еще раз отметить, что хотя контроль технических параметров аппаратуры вполне доступен в любительских (домашних, либо школьных и кружковых) условиях, однако настройку и ремонт сложной техники (не только устройств видеовывода!) следует поручать квалифицированному специалисту, не понаслышке знакомому с техникой безопасности и осознающему ответственность за результаты своих действий.

Рис. 2

```

6530 PRINT AT 6,1:4: BRIGHT 1: I
6540 PRINT AT 15,1:4: INK 7-1:
6550 PRINT AT 16,1:4: INK 7-1:
6560 NEXT 1
6570 PRINT AT 9,0: PAPER 3: INK
6580 PRINT AT 10,0: PAPER 1: INK 2:
6590 PRINT AT 11,0: PAPER 5: INK 6:
6600 PRINT AT 12,0: PAPER 7: INK 0:
6610 PRINT AT 13,0: PAPER 3: INK 4:
6620 PRINT AT 14,0: PAPER 1: INK 2:
6630 PRINT AT 15,0: PAPER 5: INK 6:
6640 PRINT AT 16,0: PAPER 7: INK 0:
6650 PRINT AT 17,0: PAPER 3: INK 4:
6660 PRINT AT 18,0: PAPER 1: INK 2:
6670 PRINT AT 19,0: PAPER 5: INK 6:
6680 PRINT AT 20,0: PAPER 7: INK 0:
6690 PRINT AT 21,0: PAPER 3: INK 4:
6700 PRINT AT 22,0: PAPER 1: INK 2:
6710 PRINT AT 23,0: PAPER 5: INK 6:
6720 PRINT AT 24,0: PAPER 7: INK 0:
6730 PRINT AT 25,0: PAPER 3: INK 4:
6740 PRINT AT 26,0: PAPER 1: INK 2:
6750 PRINT AT 27,0: PAPER 5: INK 6:
6760 PRINT AT 28,0: PAPER 7: INK 0:
6770 PRINT AT 29,0: PAPER 3: INK 4:
6780 PRINT AT 30,0: PAPER 1: INK 2:
6790 PRINT AT 31,0: PAPER 5: INK 6:
6800 PRINT AT 32,0: PAPER 7: INK 0:
6810 PRINT AT 33,0: PAPER 3: INK 4:
6820 PRINT AT 34,0: PAPER 1: INK 2:
6830 PRINT AT 35,0: PAPER 5: INK 6:
6840 PRINT AT 36,0: PAPER 7: INK 0:
6850 PRINT AT 37,0: PAPER 3: INK 4:
6860 PRINT AT 38,0: PAPER 1: INK 2:
6870 PRINT AT 39,0: PAPER 5: INK 6:
6880 PRINT AT 40,0: PAPER 7: INK 0:
6890 PRINT AT 41,0: PAPER 3: INK 4:
6900 PRINT AT 42,0: PAPER 1: INK 2:
6910 PRINT AT 43,0: PAPER 5: INK 6:
6920 PRINT AT 44,0: PAPER 7: INK 0:
6930 PRINT AT 45,0: PAPER 3: INK 4:
6940 PRINT AT 46,0: PAPER 1: INK 2:
6950 PRINT AT 47,0: PAPER 5: INK 6:
6960 PRINT AT 48,0: PAPER 7: INK 0:
6970 PRINT AT 49,0: PAPER 3: INK 4:
6980 PRINT AT 50,0: PAPER 1: INK 2:
6990 PRINT AT 51,0: PAPER 5: INK 6:
7000 PRINT AT 52,0: PAPER 7: INK 0:
7010 PRINT AT 53,0: PAPER 3: INK 4:
7020 PRINT AT 54,0: PAPER 1: INK 2:
7030 PRINT AT 55,0: PAPER 5: INK 6:
7040 PRINT AT 56,0: PAPER 7: INK 0:
7050 PRINT AT 57,0: PAPER 3: INK 4:
7060 PRINT AT 58,0: PAPER 1: INK 2:
7070 PRINT AT 59,0: PAPER 5: INK 6:
7080 PRINT AT 60,0: PAPER 7: INK 0:
7090 PRINT AT 61,0: PAPER 3: INK 4:
7100 PRINT AT 62,0: PAPER 1: INK 2:
7110 PRINT AT 63,0: PAPER 5: INK 6:
7120 PRINT AT 64,0: PAPER 7: INK 0:
7130 PRINT AT 65,0: PAPER 3: INK 4:
7140 PRINT AT 66,0: PAPER 1: INK 2:
7150 PRINT AT 67,0: PAPER 5: INK 6:
7160 PRINT AT 68,0: PAPER 7: INK 0:
7170 PRINT AT 69,0: PAPER 3: INK 4:
7180 PRINT AT 70,0: PAPER 1: INK 2:
7190 PRINT AT 71,0: PAPER 5: INK 6:
7200 PRINT AT 72,0: PAPER 7: INK 0:
7210 PRINT AT 73,0: PAPER 3: INK 4:
7220 PRINT AT 74,0: PAPER 1: INK 2:
7230 PRINT AT 75,0: PAPER 5: INK 6:
7240 PRINT AT 76,0: PAPER 7: INK 0:
7250 PRINT AT 77,0: PAPER 3: INK 4:
7260 PRINT AT 78,0: PAPER 1: INK 2:
7270 PRINT AT 79,0: PAPER 5: INK 6:
7280 PRINT AT 80,0: PAPER 7: INK 0:
7290 PRINT AT 81,0: PAPER 3: INK 4:
7300 PRINT AT 82,0: PAPER 1: INK 2:
7310 PRINT AT 83,0: PAPER 5: INK 6:
7320 PRINT AT 84,0: PAPER 7: INK 0:
7330 PRINT AT 85,0: PAPER 3: INK 4:
7340 PRINT AT 86,0: PAPER 1: INK 2:
7350 PRINT AT 87,0: PAPER 5: INK 6:
7360 PRINT AT 88,0: PAPER 7: INK 0:
7370 PRINT AT 89,0: PAPER 3: INK 4:
7380 PRINT AT 90,0: PAPER 1: INK 2:
7390 PRINT AT 91,0: PAPER 5: INK 6:
7400 PRINT AT 92,0: PAPER 7: INK 0:
7410 PRINT AT 93,0: PAPER 3: INK 4:
7420 PRINT AT 94,0: PAPER 1: INK 2:
7430 PRINT AT 95,0: PAPER 5: INK 6:
7440 PRINT AT 96,0: PAPER 7: INK 0:
7450 PRINT AT 97,0: PAPER 3: INK 4:
7460 PRINT AT 98,0: PAPER 1: INK 2:
7470 PRINT AT 99,0: PAPER 5: INK 6:
7480 PRINT AT 100,0: PAPER 7: INK 0:
7490 PRINT AT 101,0: PAPER 3: INK 4:
7500 PRINT AT 102,0: PAPER 1: INK 2:
7510 PRINT AT 103,0: PAPER 5: INK 6:
7520 PRINT AT 104,0: PAPER 7: INK 0:
7530 PRINT AT 105,0: PAPER 3: INK 4:
7540 PRINT AT 106,0: PAPER 1: INK 2:
7550 PRINT AT 107,0: PAPER 5: INK 6:
7560 PRINT AT 108,0: PAPER 7: INK 0:
7570 PRINT AT 109,0: PAPER 3: INK 4:
7580 PRINT AT 110,0: PAPER 1: INK 2:
7590 PRINT AT 111,0: PAPER 5: INK 6:
7600 PRINT AT 112,0: PAPER 7: INK 0:
7610 PRINT AT 113,0: PAPER 3: INK 4:
7620 PRINT AT 114,0: PAPER 1: INK 2:
7630 PRINT AT 115,0: PAPER 5: INK 6:
7640 PRINT AT 116,0: PAPER 7: INK 0:
7650 PRINT AT 117,0: PAPER 3: INK 4:
7660 PRINT AT 118,0: PAPER 1: INK 2:
7670 PRINT AT 119,0: PAPER 5: INK 6:
7680 PRINT AT 120,0: PAPER 7: INK 0:
7690 PRINT AT 121,0: PAPER 3: INK 4:
7700 PRINT AT 122,0: PAPER 1: INK 2:
7710 PRINT AT 123,0: PAPER 5: INK 6:
7720 PRINT AT 124,0: PAPER 7: INK 0:
7730 PRINT AT 125,0: PAPER 3: INK 4:
7740 PRINT AT 126,0: PAPER 1: INK 2:
7750 PRINT AT 127,0: PAPER 5: INK 6:
7760 PRINT AT 128,0: PAPER 7: INK 0:
7770 PRINT AT 129,0: PAPER 3: INK 4:
7780 PRINT AT 130,0: PAPER 1: INK 2:
7790 PRINT AT 131,0: PAPER 5: INK 6:
7800 PRINT AT 132,0: PAPER 7: INK 0:
7810 PRINT AT 133,0: PAPER 3: INK 4:
7820 PRINT AT 134,0: PAPER 1: INK 2:
7830 PRINT AT 135,0: PAPER 5: INK 6:
7840 PRINT AT 136,0: PAPER 7: INK 0:
7850 PRINT AT 137,0: PAPER 3: INK 4:
7860 PRINT AT 138,0: PAPER 1: INK 2:
7870 PRINT AT 139,0: PAPER 5: INK 6:
7880 PRINT AT 140,0: PAPER 7: INK 0:
7890 PRINT AT 141,0: PAPER 3: INK 4:
7900 PRINT AT 142,0: PAPER 1: INK 2:
7910 PRINT AT 143,0: PAPER 5: INK 6:
7920 PRINT AT 144,0: PAPER 7: INK 0:
7930 PRINT AT 145,0: PAPER 3: INK 4:
7940 PRINT AT 146,0: PAPER 1: INK 2:
7950 PRINT AT 147,0: PAPER 5: INK 6:
7960 PRINT AT 148,0: PAPER 7: INK 0:
7970 PRINT AT 149,0: PAPER 3: INK 4:
7980 PRINT AT 150,0: PAPER 1: INK 2:
7990 PRINT AT 151,0: PAPER 5: INK 6:
8000 PRINT AT 152,0: PAPER 7: INK 0:
8010 PRINT AT 153,0: PAPER 3: INK 4:
8020 PRINT AT 154,0: PAPER 1: INK 2:
8030 PRINT AT 155,0: PAPER 5: INK 6:
8040 PRINT AT 156,0: PAPER 7: INK 0:
8050 PRINT AT 157,0: PAPER 3: INK 4:
8060 PRINT AT 158,0: PAPER 1: INK 2:
8070 PRINT AT 159,0: PAPER 5: INK 6:
8080 PRINT AT 160,0: PAPER 7: INK 0:
8090 PRINT AT 161,0: PAPER 3: INK 4:
8100 PRINT AT 162,0: PAPER 1: INK 2:
8110 PRINT AT 163,0: PAPER 5: INK 6:
8120 PRINT AT 164,0: PAPER 7: INK 0:
8130 PRINT AT 165,0: PAPER 3: INK 4:
8140 PRINT AT 166,0: PAPER 1: INK 2:
8150 PRINT AT 167,0: PAPER 5: INK 6:
8160 PRINT AT 168,0: PAPER 7: INK 0:
8170 PRINT AT 169,0: PAPER 3: INK 4:
8180 PRINT AT 170,0: PAPER 1: INK 2:
8190 PRINT AT 171,0: PAPER 5: INK 6:
8200 PRINT AT 172,0: PAPER 7: INK 0:
8210 PRINT AT 173,0: PAPER 3: INK 4:
8220 PRINT AT 174,0: PAPER 1: INK 2:
8230 PRINT AT 175,0: PAPER 5: INK 6:
8240 PRINT AT 176,0: PAPER 7: INK 0:
8250 PRINT AT 177,0: PAPER 3: INK 4:
8260 PRINT AT 178,0: PAPER 1: INK 2:
8270 PRINT AT 179,0: PAPER 5: INK 6:
8280 PRINT AT 180,0: PAPER 7: INK 0:
8290 PRINT AT 181,0: PAPER 3: INK 4:
8300 PRINT AT 182,0: PAPER 1: INK 2:
8310 PRINT AT 183,0: PAPER 5: INK 6:
8320 PRINT AT 184,0: PAPER 7: INK 0:
8330 PRINT AT 185,0: PAPER 3: INK 4:
8340 PRINT AT 186,0: PAPER 1: INK 2:
8350 PRINT AT 187,0: PAPER 5: INK 6:
8360 PRINT AT 188,0: PAPER 7: INK 0:
8370 PRINT AT 189,0: PAPER 3: INK 4:
8380 PRINT AT 190,0: PAPER 1: INK 2:
8390 PRINT AT 191,0: PAPER 5: INK 6:
8400 PRINT AT 192,0: PAPER 7: INK 0:
8410 PRINT AT 193,0: PAPER 3: INK 4:
8420 PRINT AT 194,0: PAPER 1: INK 2:
8430 PRINT AT 195,0: PAPER 5: INK 6:
8440 PRINT AT 196,0: PAPER 7: INK 0:
8450 PRINT AT 197,0: PAPER 3: INK 4:
8460 PRINT AT 198,0: PAPER 1: INK 2:
8470 PRINT AT 199,0: PAPER 5: INK 6:
8480 PRINT AT 200,0: PAPER 7: INK 0:
8490 PRINT AT 201,0: PAPER 3: INK 4:
8500 PRINT AT 202,0: PAPER 1: INK 2:
8510 PRINT AT 203,0: PAPER 5: INK 6:
8520 PRINT AT 204,0: PAPER 7: INK 0:
8530 PRINT AT 205,0: PAPER 3: INK 4:
8540 PRINT AT 206,0: PAPER 1: INK 2:
8550 PRINT AT 207,0: PAPER 5: INK 6:
8560 PRINT AT 208,0: PAPER 7: INK 0:
8570 PRINT AT 209,0: PAPER 3: INK 4:
8580 PRINT AT 210,0: PAPER 1: INK 2:
8590 PRINT AT 211,0: PAPER 5: INK 6:
8600 PRINT AT 212,0: PAPER 7: INK 0:
8610 PRINT AT 213,0: PAPER 3: INK 4:
8620 PRINT AT 214,0: PAPER 1: INK 2:
8630 PRINT AT 215,0: PAPER 5: INK 6:
8640 PRINT AT 216,0: PAPER 7: INK 0:
8650 PRINT AT 217,0: PAPER 3: INK 4:
8660 PRINT AT 218,0: PAPER 1: INK 2:
8670 PRINT AT 219,0: PAPER 5: INK 6:
8680 PRINT AT 220,0: PAPER 7: INK 0:
8690 PRINT AT 221,0: PAPER 3: INK 4:
8700 PRINT AT 222,0: PAPER 1: INK 2:
8710 PRINT AT 223,0: PAPER 5: INK 6:
8720 PRINT AT 224,0: PAPER 7: INK 0:
8730 PRINT AT 225,0: PAPER 3: INK 4:
8740 PRINT AT 226,0: PAPER 1: INK 2:
8750 PRINT AT 227,0: PAPER 5: INK 6:
8760 PRINT AT 228,0: PAPER 7: INK 0:
8770 PRINT AT 229,0: PAPER 3: INK 4:
8780 PRINT AT 230,0: PAPER 1: INK 2:
8790 PRINT AT 231,0: PAPER 5: INK 6:
8800 PRINT AT 232,0: PAPER 7: INK 0:
8810 PRINT AT 233,0: PAPER 3: INK 4:
8820 PRINT AT 234,0: PAPER 1: INK 2:
8830 PRINT AT 235,0: PAPER 5: INK 6:
8840 PRINT AT 236,0: PAPER 7: INK 0:
8850 PRINT AT 237,0: PAPER 3: INK 4:
8860 PRINT AT 238,0: PAPER 1: INK 2:
8870 PRINT AT 239,0: PAPER 5: INK 6:
8880 PRINT AT 240,0: PAPER 7: INK 0:
8890 PRINT AT 241,0: PAPER 3: INK 4:
8900 PRINT AT 242,0: PAPER 1: INK 2:
8910 PRINT AT 243,0: PAPER 5: INK 6:
8920 PRINT AT 244,0: PAPER 7: INK 0:
8930 PRINT AT 245,0: PAPER 3: INK 4:
8940 PRINT AT 246,0: PAPER 1: INK 2:
8950 PRINT AT 247,0: PAPER 5: INK 6:
8960 PRINT AT 248,0: PAPER 7: INK 0:
8970 PRINT AT 249,0: PAPER 3: INK 4:
8980 PRINT AT 250,0: PAPER 1: INK 2:
8990 PRINT AT 251,0: PAPER 5: INK 6:
9000 PRINT AT 252,0: PAPER 7: INK 0:
9010 PRINT AT 253,0: PAPER 3: INK 4:
9020 PRINT AT 254,0: PAPER 1: INK 2:
9030 PRINT AT 255,0: PAPER 5: INK 6:
9040 PRINT AT 256,0: PAPER 7: INK 0:
9050 PRINT AT 257,0: PAPER 3: INK 4:
9060 PRINT AT 258,0: PAPER 1: INK 2:
9070 PRINT AT 259,0: PAPER 5: INK 6:
9080 PRINT AT 260,0: PAPER 7: INK 0:
9090 PRINT AT 261,0: PAPER 3: INK 4:
9100 PRINT AT 262,0: PAPER 1: INK 2:
9110 PRINT AT 263,0: PAPER 5: INK 6:
9120 PRINT AT 264,0: PAPER 7: INK 0:
9130 PRINT AT 265,0: PAPER 3: INK 4:
9140 PRINT AT 266,0: PAPER 1: INK 2:
9150 PRINT AT 267,0: PAPER 5: INK 6:
9160 PRINT AT 268,0: PAPER 7: INK 0:
9170 PRINT AT 269,0: PAPER 3: INK 4:
9180 PRINT AT 270,0: PAPER 1: INK 2:
9190 PRINT AT 271,0: PAPER 5: INK 6:
9200 PRINT AT 272,0: PAPER 7: INK 0:
9210 PRINT AT 273,0: PAPER 3: INK 4:
9220 PRINT AT 274,0: PAPER 1: INK 2:
9230 PRINT AT 275,0: PAPER 5: INK 6:
9240 PRINT AT 276,0: PAPER 7: INK 0:
9250 PRINT AT 277,0: PAPER 3: INK 4:
9260 PRINT AT 278,0: PAPER 1: INK 2:
9270 PRINT AT 279,0: PAPER 5: INK 6:
9280 PRINT AT 280,0: PAPER 7: INK 0:
9290 PRINT AT 281,0: PAPER 3: INK 4:
9300 PRINT AT 282,0: PAPER 1: INK 2:
9310 PRINT AT 283,0: PAPER 5: INK 6:
9320 PRINT AT 284,0: PAPER 7: INK 0:
9330 PRINT AT 285,0: PAPER 3: INK 4:
9340 PRINT AT 286,0: PAPER 1: INK 2:
9350 PRINT AT 287,0: PAPER 5: INK 6:
9360 PRINT AT 288,0: PAPER 7: INK 0:
9370 PRINT AT 289,0: PAPER 3: INK 4:
9380 PRINT AT 290,0: PAPER 1: INK 2:
9390 PRINT AT 291,0: PAPER 5: INK 6:
9400 PRINT AT 292,0: PAPER 7: INK 0:
9410 PRINT AT 293,0: PAPER 3: INK 4:
9420 PRINT AT 294,0: PAPER 1: INK 2:
9430 PRINT AT 295,0: PAPER 5: INK 6:
9440 PRINT AT 296,0: PAPER 7: INK 0:
9450 PRINT AT 297,0: PAPER 3: INK 4:
9460 PRINT AT 298,0: PAPER 1: INK 2:
9470 PRINT AT 299,0: PAPER 5: INK 6:
9480 PRINT AT 300,0: PAPER 7: INK 0:
9490 PRINT AT 301,0: PAPER 3: INK 4:
9500 PRINT AT 302,0: PAPER 1: INK 2:
9510 PRINT AT 303,0: PAPER 5: INK 6:
9520 PRINT AT 304,0: PAPER 7: INK 0:
9530 PRINT AT 305,0: PAPER 3: INK 4:
9540 PRINT AT 306,0: PAPER 1: INK 2:
9550 PRINT AT 307,0: PAPER 5: INK 6:
9560 PRINT AT 308,0: PAPER 7: INK 0:
9570 PRINT AT 309,0: PAPER 3: INK 4:
9580 PRINT AT 310,0: PAPER 1: INK 2:
9590 PRINT AT 311,0: PAPER 5: INK 6:
9600 PRINT AT 312,0: PAPER 7: INK 0:
9610 PRINT AT 313,0: PAPER 3: INK 4:
9620 PRINT AT 314,0: PAPER 1: INK 2:
9630 PRINT AT 315,0: PAPER 5: INK 6:
9640 PRINT AT 316,0: PAPER 7: INK 0:
9650 PRINT AT 317,0: PAPER 3: INK 4:
9660 PRINT AT 318,0: PAPER 1: INK 2:
9670 PRINT AT 319,0: PAPER 5: INK 6:
9680 PRINT AT 320,0: PAPER 7: INK 0:
9690 PRINT AT 321,0: PAPER 3: INK 4:
9700 PRINT AT 322,0: PAPER 1: INK 2:
9710 PRINT AT 323,0: PAPER 5: INK 6:
9720 PRINT AT 324,0: PAPER 7: INK 0:
9730 PRINT AT 325,0: PAPER 3: INK 4:
9740 PRINT AT 326,0: PAPER 1: INK 2:
9750 PRINT AT 327,0: PAPER 5: INK 6:
9760 PRINT AT 328,0: PAPER 7: INK 0:
9770 PRINT AT 329,0: PAPER 3: INK 4:
9780 PRINT AT 330,0: PAPER 1: INK 2:
9790 PRINT AT 331,0: PAPER 5: INK 6:
9800 PRINT AT 332,0: PAPER 7: INK 0:
9810 PRINT AT 333,0: PAPER 3: INK 4:
9820 PRINT AT 334,0: PAPER 1: INK 2:
9830 PRINT AT 335,0: PAPER 5: INK 6:
9840 PRINT AT 336,0: PAPER 7: INK 0:
9850 PRINT AT 337,0: PAPER 3: INK 4:
9860 PRINT AT 338,0: PAPER 1: INK 2:
9870 PRINT AT 339,0: PAPER 5: INK 6:
9880 PRINT AT 340,0: PAPER 7: INK 0:
9890 PRINT AT 341,0: PAPER 3: INK 4:
9900 PRINT AT 342,0: PAPER 1: INK 2:
9910 PRINT AT 343,0: PAPER 5: INK 6:
9920 PRINT AT 344,0: PAPER 7: INK 0:
9930 PRINT AT 345,0: PAPER 3: INK 4:
9940 PRINT AT 346,0: PAPER 1: INK 2:
9950 PRINT AT 347,0: PAPER 5: INK 6:
9960 PRINT AT 348,0: PAPER 7: INK 0:
9970 PRINT AT 349,0: PAPER 3: INK 4:
9980 PRINT AT 350,0: PAPER 1: INK 2:
9990 PRINT AT 351,0: PAPER 5: INK 6:
1000 PRINT AT 352,0: PAPER 7: INK 0:

```

Литература

1. Ильясов Е.В. Телетест на Sinclair. — Радиомир. Ваш компьютер, 2003, №11, С.43.

А.ВЕНДИЛОВСКИЙ,
г.Минск.

Need for Speed Underground

Город окутала ночь, улицы стали пустынные, словно насыщенная дневная жизнь была лишь иллюзией, странной пародией на жизнь. Пустынные остановки автобусов и безлюдные парки вызвали ощущение сюрреалистичности, по крайней мере, у Дедушки Геймера. Он уже битый час пытался понять две вещи. Первое — почему Автор послал именно его, заслуженного специалиста по шуттерам и стратегиям, разбираться с автосимулятором? И второе — где шляется его напарник Малдер? Нет, Дед вовсе не чурался всевозможнейшего автотранспорта, и даже всегда был рад прокатиться на очередном четырехколесном создании, если у того при этом была хорошая броня и несколько качественных скорострельных пушек. Сейчас же он чувствовал себя несколько не в своей тарелке. Там, где стоял Дед, толпились около полусотни зевак и пяток автомобилей, раскрашенных и расфуфыренных "дальше некуда". Чувствовалось, что не только внешняя сторона машин подверглась изменению. По силуэтам и некоторым мелким деталям, подмеченным острым взглядом Деда, под капотом каждой из этих малышек стояли детали, явно не входившие в стандартную заводскую поставку, да и мощность движков была намного выше обычной. Вокруг Деда толпились поклонники уличных гонок и сами гонщики.

... слабый звук, приближавшийся издали, превратился в забивающий уши рев. Из-за поворота на бешеной скорости выскочили два автомобиля. На какое-то мгновение их подбросило на горочке вверх, и они словно летели в свете ночного города. Полет прервался глухим ударом колес об асфальт. Запах горелой резины и бензиновые пары окутали Деда, когда последние участники столь экстравагантным способом прибыли на место гонки. Но еще больше удивился Дед, признав в одном из водителей Малдера, который пытался о чем-то договориться с весьма очаровательной устроительницей этого шоу. Издали долетали лишь обрывки слов о какой-то трансмиссии, шинах и компьютерном контроле. Взмахом руки Малдер приветствовал напарника, указал ему на автомобиль, а сам продолжал размахивать кошельком и явно просить скидку на приобретение новых спойлеров и фар. Жаркая дискуссия закончилась лишь через полчаса, когда обе стороны ударили по рукам, и кошелек Малдера сильно похудел. Но зато к его автомобилю механики стали прикручивать какие-то детали, о которых Дед имел весьма смутное представление.

— Здорово, Дед! Как тебе моя новая тачка? — Малдер был несколько возбужден. — У нашего Автора странное чувство юмора, раз он послал нас на тестирование этой игры. Но надо отдать ему должное — все игры серии NFS всегда были непревзойденными хитами. Будем надеяться, что и седьмая часть не ударит лицом в грязь.

Интерфейс? — Дед был сосредоточен на карте трассы предстоящего заезда, прикидывая, не найдется ли по пути нескольких более коротких и желательных прямых объездов, поэтому сформулировал свои вопросы коротко и четко. — Близок к идеальному, насколько это возможно, никаких проблем с привыканием, все кнопки расположены именно по степени их использования, что снимает проблему "раско-



ряжавшихся пальцев" и не отвлекает от процесса игры.

Машины? — Порядка двадцати моделей, в основном, японские, хотя американских и европейских тоже достаточно.

Трассы? — Сто двенадцать штук. Но тут, правда, есть один нюанс. Гонки проходят внутри одного большого мегаполиса, так что трассы складываются из разных улиц, по которым уже приходилось проезжать. Но я бы не относил это к недостаткам, скорее, это даже достоинство игры. С одной стороны — все по-новому, с другой — уже знаешь, где можно педаль утопить до пола, а где перед поворотом притормозить. Ну и, конечно, можно подобрать оптимальный маршрут.

Режимы игры? — Стандартные гонки из серии "Будь первым", гонки на время, гонки с выбыванием, гонки с подсчетом очков на чекпоинтах, игры с переключением передач и показательные выступления на артистизм вождения. Что касается артистизма, то добытые тобой очки суммируются, и в конце концов изменяют твой уровень репутации, что открывает тебе возможности приобретения новых деталей, некоторого увеличения банковского счета и использования новых трасс.

Напарники огляделись. Движок игры вызывал восхищение. Даже на слабенькой второй GeForce он поражае своей красотой. Конечно, некоторые опции были деактивированы напрочь, но когда Дед, послав к дьяволу предупреждения, выставил все имеющиеся параметры на максимум, игра совсем не превратилась в слайд-шоу, и более того, оказалась достаточно играбельна! Но то, что увидели бойцы на экране мониторов после установки видеокарты последнего поколения, привело их в трепет. Все сияло яркими огнями, город поражал игрой света от рекламных вывесок, витрин и фонарей — Лас-Вегас просто отдыхает перед этим фантастическим безумием красок. Все детали были настолько до мелочей проработаны, что первое слово, которое пришло им в голову — фотореалистичность! Отражение машины на мокром асфальте, отражение в проезжающих автомобилях, свет фар, далеко вперед вырывающий дорогу из темноты. "Графика — десять баллов!" — подумал Дед.

Малдер выжимал газ по максимуму, пытаясь разницей между автоматической и ручной коробками передач рассказать о физической модели автомобилей, когда висевший на хвосте соперник достаточно ощутимо стукнул их сзади. На повороте Малдер на какое-то время потерял управление, машину на всей скорости развернуло, и на мокром асфальте ее повело юзом. Весь этот "балет" закончился лишь тогда, когда их автомобиль со всего маху врезался в дерево.

"Деревья, Малдер, это такие создания, которые так и норовят броситься на твой автомобиль!" — съехидничал Дед. С дерева прямо на ветровое стекло спикировала бумажная табличка, и взору несколько ошарашенных столкновением бойцов открылась надпись "Вечно зеленая секвойя — привезена в наш город из Австралии".

— Лучше бы она там и оставалась, — буркнул Малдер, вновь заводя мотор и бросаясь вдогонку за соперниками. — Последний произошедший с нами эпизод свидетельствует о двух вещах: во-первых, AI у наших противников очень хорош, и про легкую победу можем сразу забыть — они ошибок не прощают; во-вторых, эти гонки — аркадные, и столкновения никак не влияют на автомобиль и гонщика. Ну а самая главная цель игрока — не просто прийти первым, а превратить свой автомобиль в четырехколесное чудо техники, ночного дьявола дорог, чтобы потом попасть на обложки журналов. Каждая победа приносит нам неко-

торую сумму денег, которую мы можем потратить на приобретение не только таких абсолютно необходимых для каждого уличного гонщика вещей как новые колеса, фары или двигатель (про новые трансмиссии или аппараты для подачи закиси азота я говорить не буду), но и всевозможнейших наворотов для тюнинга, начиная от тонированных стекол, и заканчивая эмблемами и неоновой подсветкой, что превратит твою машину в неповторимый образчик постиндустриального искусства XXI века.

На колени Деда рухнуло несколько увесистых каталогов с огромным перечнем всевозможнейших деталей, от одного вида которых любой уличный гонщик пришел бы в благоговейный восторг.

— Кстати, еще об автомобилях. Не смотри, что гонка аркадная, автомобиль ведет себя очень правильно и предсказуемо на любом участке пути, и ты сам почувствуешь, как отличается езда на различных скоростях, либо на мокром и сухом асфальте. Физическая модель очень честная, что не может не радовать.

Поговорить о геймплее у Деда и Малдера времени уже не хватило, они догоняли соперников, и их автомобиль черной стрелой неся по городу, на сумасшедшей скорости лавируя в рядах городского транспорта, в самое последнее мгновение уходя от лобового столкновения на встречной полосе во время очередного опаснейшего обгона. Контролируемые заносы следовали один за другим. Они были во власти невероятного драйва, когда два автомобиля выскакивают на финишную прямую, соревнуясь за каждый миллиметр и сбивая на своем пути дорожные знаки.

Дед и не заметил, как игра захватила его, наполняя кровь адреналином под мощную и ритмичную музыку и объемный звук. Он уже присмотрел несколько очень нужных вещей для их машины. Главное — победить в этом заезде, а уж в следующем... Он довольно ухмыльнулся, представляя, на что будет способен их механический друг, когда на него установят турбо-кит и новый бортовой компьютер. Черный автомобиль бойцов неумолимо рвался к финишу...

Minimum system requirements

Windows 98, 98 Second Edition, ME, 2000 Professional, XP Professional or XP Home operating system;

Intel Pentium III (or compatible) CPU running at 700Mhz or higher;

32MB DirectX 9 compatible video card (Geforce2 or higher) using a supported chipset (see below);

128MB of RAM (256 MB using Windows 2000 or XP);

DirectX compatible sound card;

2 GB of HD space;

8X CD-ROM.

Minimum system requirements for online play

Windows 98, 98 Second Edition, ME, 2000 Professional, XP Professional or XP Home operating system;

Intel Pentium III (or compatible) CPU running at 700Mhz or higher;

32MB DirectX 9 compatible video card (Geforce2 or higher) using a supported chipset (see below);

256MB of RAM;

DirectX compatible sound card;

2 GB of HD space;

8X CD-ROM;

Network card for broadband connectivity.

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



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

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



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

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

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

Куплю контроллер жесткого диска и программное обеспечение для ПК "Сантак-002".
Приму в дар системный блок компьютера.
425568, Респ.Марий Эл,
Куженерский р-н, д.Шудумарь. Нараев М.А.

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

Куплю два картриджа (чистый оптический диск) типа OC301-1 или OC301-2, желательно фирмы "Maxell".
456785, г.Озерск, Челябинская обл., а/я 643.
E-mail: dkfbvbn@telecom.ozersk.ru

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

Продаю компьютер 486 по частям: материнская плата: O3Y — 12 Мб SIMM; процессор — Intel 80486 DX2S, 120 МГц; видеокарта — Cirrus Logic (SVGA/4 Мб); винчестер — 472 Мб; мультикарта — HDD/FDD (5,25", 3,5")/2xLPTxCOM; дисководы 3,5" и 5,25"; мышь — Genius, 3 кнопки, для COM-порта (DB9); клавиатура — Microsoft Natural (разъем 5 DIN).

Для ответа приложить чистый конверт.
692665, Приморский край, Михайловский р-н, с.Ивановка, ул.Восточная, 19 — 2, Гевик А.И.

Куплю запрограммированное ПЗУ для контроллера TR-DOS (ПЛ №2/1993, Королев, Исаев, Бурылов) или готовый контроллер со схемой другой версии (5.03, 5.04).
393774, Тамбовская область, г. Мичуринск, ул. Мичурина, д.30, ВНИИС, Зайцеву Ю.К.

Помогу с утилизацией старого "железа"; 286...5x86, Атари, Коммодор, Практик, Поиск, ЕС 1840...1841, Истра 4816.
E-mail: amstaffoskarowner@torba.com.

Продам на кассетах по низкой цене коллекцию программ для ZX-Spectrum.
692446, Приморский край, г.Дальнегорск-6, а/я 34.
Email: alecs-dms@mail.ru

Куплю: схемы телевизора "Альфа 51ТЦ-4301Д", радиоприемника "Меридиан-210"; книгу "Программирование на Бейсике для персональных ЭВМ радиолюбителя" изд. "Патриот".
169312, г.Ухта, ул.Строителей, 19 — 6, Голивец В.Г.

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. Горячкин.
E-mail: anatoliygor@nm.ru

Куплю компьютерные комплектующие (б/у); аксессуары для компьютеров (мышки, колонки и т.п.) в нерабочем состоянии.
640023, г.Курган-23, а/я 2055, Анатолий.
E-mail: anatoli25@rambler.ru

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 "Роспечать", стр.392 или по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь", стр.171 (раздел "Издания РО, 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			Ресценки на 1 экз. (с учетом пересылки)		
	Радиолобитель. Ваш компьютер	Радиолобитель. КВ и УКВ	Радиолобитель. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
1999	3, 9, 10, 11	1—6, 10, 11, 12	5, 6	20	700	5
2000	3—6, 8—12	2—8, 10—12	6, 7, 9, 11, 12	20	800	5
2001	2, 4—6	2, 5, 6	2—6	25	900	5
	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2001	7—10, 12	7, 9—12	7, 8, 10, 11	30	1000	6
2002	1—12	1—12	1—3, 5—12	35	1500	7
2003	1—12	1—12	1, 5—8, 10—12	40	1800	8
2004	1—12	1—12	1—12	45	2100	9

Наши платежные реквизиты:

- для жителей России и стран СНГ (кроме Беларуси) —
получатель: ООО "Радиомир Пресс", ИНН 772904741, КПП 772901001;
р/с 40702810900010000084 в ООО КБ "Агропромкредит", ф-л Центральный, г.Москва,
корр. счет 30101810500000000109, БИК 044525109
Адрес банка: 125315, г.Москва, Ленинградский пр-т, дом 76, корп. 4;

- для жителей Беларуси —
получатель: УП "РЛД", УНН 190218688
р/с 3012000004882 в ф-ле №524 АСБ "Беларусбанк" в г.Минске код 121.
Адрес банка: 220028, г.Минск, ул.Физкультурная, 31.

При оплате через Сбербанк в графе "назначение платежа" необходимо написать свой почтовый индекс, полный адрес, фамилию, имя и отчество полностью и точно перечислить, какие конкретно номера какого из журналов Вы заказываете.

При оплате почтовым переводом все сведения о заказе необходимо указать в графе "Для письма".

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

Приобретение отдельных номеров журналов

В РОССИИ:

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

В магазинах радиодеталей "ЧИП и ДИП":

- г.Москва, ул.Гиляровского, д.39 (ст. метро "Проспект Мира" — радиальная);
- г.Москва, ул.Ивана Франко, д.40, к.1, стр. 2 (платф.Рабочий поселок,
15 мин. от Белорусского вокзала);
- г.Москва, ул.Беговая, д.2;
- г.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 "ТРАНТЭК": 111024, г.Москва, 4-я Кабельная, 2А (ст. метро "Авиамоторная"),
тел. (095) 258-91-94, 258-91-95. E-mail: abook@mail.ru, abook@inbox.ru

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

На УКРАИНЕ:

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

В КАЗАХСТАНЕ:

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

В БЕЛАРУСИ:

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

Need for Speed Underground





- 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