

радиомир

2 • 2003

Ваш компьютер

Microsoft
WINDOWS XP SP 1
10 Операционных систем
на одном диске!



ТАТУИРОВКИ
ШРАМИРОВАНИЕ
ПИРСИНГ
КУЛЬТУРА
РИТУАЛ



8 АЛЬБОМОВ
128 kHz • 44 kHz • STEREO

MP3

Н.Лутковская.

Процессору не будет жарко и без кулера



Рис. 1. Технология изготовления проводов из сверхпроводящего материала диборида магния.



Рис. 2. Сверхпроводящий провод из диборида магния в разрезе.

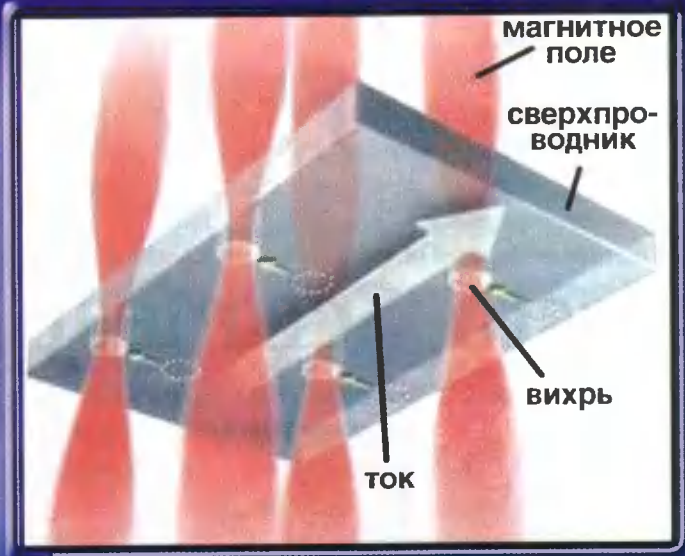


Рис. 3. Возникновение в пластине сверхпроводника вихрей, рассеивающих энергию.

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 график —
Наталья БЕЛЬСКАЯ,
Александр ГОРАЮТИН,
Мария КАЛАБУХОВА

Оформление обложки —
Наталья БЕЛЬСКАЯ,
Надежда БОГОМОЛОВА

Адреса для писем:
119454, РФ, г.Москва, а/я 37,
факс (095) 131-97-17;
220095, РБ, г.Минск-95, а/я 199,
факс (017) 221-01-10
E-mail: rl@radiopage.by
rm@radio-mir.com
WWW: http://radio-mir.com

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

Адрес банка: 113054, г.Москва,
ул.Зацепа, 43Б

Материалы для публикации принимаются
в рукописном, печатном и электронном
вариантах

Требования к графическим материалам
рекламного характера в электронном виде:
CorelDRAW до 9.0, все шрифты в кривых;
bitmaps 300 dpi; TIFF 300 dpi; CMYK.
Приложить печатную копию.

За достоверность рекламной и другой
публикуемой информации несут ответ-
ственность рекламодатели и авторы.
Мнение редакции не всегда совпадает
с мнениями авторов

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

Подписано к печати 30.12.2002 г.
Формат 60 x 84 1/8,
Печать офсетная, 6 печ. л.
Цена свободная.

Отпечатано в типографии
ЗАО "Красногорская типография".
Заказ №3.

радиомир Ваш компьютер

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

№2/февраль/2003

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

КОМПЬЮТЕРНЫЕ ГОРИЗОНТЫ

Н.ЛУТКОВСКАЯ. Процессору не будет жарко и без кулера 2

НЕ ТОЛЬКО НОВИЧКУ

В.КУЦ. Что такое системная плата 4

О.ПЕТРАКОВ. Поведенческое моделирование в PSPICE 7

А.ГРИНЧУК. Работа с редактором деловой графики Visio 2002.
Работа с трафаретами 9

А.ГОРЯЧКИН. На вершине музыкального Олимпа.
Аудиоформат MP3 & Player WinAmp 11

КОММУНИКАЦИИ

С.ГРИНЧУК. Копирование файлов из Интернета 13

ДАЙДЖЕСТ 17

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

М.ДОЛИНСКИЙ. Решение задач методом дихотомии 21

Serrgio /HI-TECH. Низкоуровневое программирование 24

ДИАЛОГ ПРОГРАММИСТОВ

А.ТЕРЁШКИН /HI-TECH. Перехват вызовов API
в системах семейства Windows 28

И.РОЩИН. Тайна ненайденного файла 31

О.ВАЛЬПА. Стрелочные часы 32

В.ФЕДОРОВ. Программа для расчета микрополосковых линий 34

РЕЦЕПТЫ

А.СЕДУНОВ. Буферное устройство для параллельного порта IBM PC 35

А.ГОРЯЧКИН. Сказание о "черном" кулере
(оптимизация системы охлаждения CPU) 37

Б.ШИЛЬНИКОВ. Ремонт мышки 38

МИР 8 БИТ

И.РОЩИН. Японские кроссворды 39

Е.ИЛАСОВ. "Тормоз" для винчестера
(счастливым обладателям "винтов" посвящается) 43

ИГРОТЕКА

М.ВАЛЬПА. Star Wars Jedi Knight 2: Jedi Outcast 45

К.КЛИЩЕНКО. Голливуд — родина "Колец" 46

ДОСКА ОБЪЯВЛЕНИЙ

Куплю, продам, обменяю 48

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



Н.ЛУТКОВСКАЯ,
E-mail: SRLSA@bsu.by

Процессору не будет жарко и без кулера

Современный компьютер невозможно себе представить без проводов, транзисторов и прочих подобных деталей. Но, как мы знаем из школьного курса физики, все провода обладают сопротивлением, на котором выделяется тепло. Поэтому применение обычных материалов неизбежно связано с их нагреванием. Больше всего проблем вызывает нагрев процессора, поэтому для охлаждения ставят специальный вентилятор — кулер. Но чем мощнее процессор, тем больше тепла он выделяет, поэтому его перегрев — одно из главных препятствий на пути повышения производительности процессора. Решением столь глобальной проблемы может стать применение сверхпроводников — материалов, которые практически не оказывают сопротивления электрическому току, или, иначе говоря, обладают нулевым сопротивлением.

В январе 2001 г. был открыт новый сверхпроводник — диборид магния. Дешевый в изготовлении, просто охлаждаемый, легко принимающий форму проводов, он может значительно расширить область применения сверхпроводников. "Этот материал мог бы занять место ниобий-титана, традиционного низкотемпературного сверхпроводника, в будущих МРО-магнитах", — заявил Пол Грант (Paul Grant), сотрудник Исследовательского института электроэнергии в Пало Альто, Калифорния.

До открытия диборида магния, при попытке применить сверхпроводимость для решения реальных задач инженеры не находили полностью подходящих материалов. Низкотемпературные сверхпроводники (металлические) сами по себе не являются очень дорогими и при этом обладают хорошими механическими свойствами. Но для получения температур для их охлаждения (0...4,2 К) требуются дорогостоящие технологии. Высокотемпературные (керамические) сверхпроводники могут быть охлаждены со значительно меньшими затратами до менее "холодной" температуры 77 К. Но для их производства требуется значительное количество дорогого материала — серебра.

На температурной шкале диборид магния находится посередине между этими двумя типами. До температуры сверхпроводящего состояния в 39 К его удобно охлаждать криогенным охладителем или жидким водородом (у обоих точек

кипения — 20,2 К), т.е. способом, давно используемым в промышленности.

Порошок, который есть в любой хорошо оборудованной химической лаборатории, никогда не проверяли на сверхпроводимость. Но только до недавнего времени! На небольшой конференции в японском городе Сендай в январе 2001 г. пять японских же исследователей объявили о своем открытии. Новость разлетелась с быстротой молнии. И ночами в лабораториях по всему миру горели огни, освещающие ученых, которые сверхурочно работали над изучением свойств недорогой металлической смеси, чтобы извлечь весь содержащийся в ней потенциал.

Усилия ученых и энергозатраты принесли свои плоды спустя две недели, когда Пол С.Кэнфилд (Paul C.Canfield) из Айовского государственного университета, Эймс и сотрудники его лаборатории астрофизического отдела создали трехсантиметровые провода из диборида магния с помощью пропускания паров магния через тонкие трубки бора. "Этот материал просто наполнен хорошими новостями!" — заявил довольный изобретатель.

Два месяца спустя после первого объявления, Американское физическое общество провело очередное ежегодное собрание в Сизтле, шт. Вашингтон. Под диборид магния выделили специальное вечернее заседание. Большой банкетный холл отеля, выбранный для собрания, был забит учеными до отказа. Было выслушано несколько десятков выступлений. Ученые мужи разошлись только в час ночи.

Начальные данные показали исследователям, что кое-что им еще предстоит сделать. Чтобы материал стал по-настоящему полезным, его пропускной ток и устойчивость к магнитному полю должны быть повышены, и к тому же, производственным способом. Должен быть разработан экономически выгодный метод производства проводов. Сегодня, два года спустя, новости весьма обнадеживающие.

Цены

Магний и бор, которые входят в состав диборида магния, являются очень дешевыми. Есть мнение, что, возможно, провода из этого материала будут сравнимы по стоимости с проводами из меди. На момент откры-

тия сверхпроводящих свойств диборида магния, килограмм этого соединения стоил около \$750. Но с увеличением спроса цена может резко упасть, и восстановление пентагидрата бора до металлического бора будет стоить около \$10. Чтобы магний и бор вступили в реакцию, потребуется еще около \$10. Т.о., стоимость диборида магния составит примерно \$20/кг.

Следует учитывать, что это нижняя граница стоимости материала. Как верхний предел, стоимость промышленно изготовленного диборида магния уменьшится до \$300/кг.

В электроэнергетике соотношение "цена к производительности" измеряется в стоимости за 1 килоампер-метр. Цена производства провода из диборида магния составит около \$1/кА-м. Для сравнения, высокотемпературные сверхпроводящие керамические провода сегодня можно купить по \$200/кА-м. Предполагается, что цена упадет до \$50/кА-м в ближайшие 2...3 года.

Длина — не проблема!

В настоящее время две компании занимаются производством проводов из диборида магния для изготовления из них кабелей, катушек и т.п. — Hyper Tech Research в Трое, шт. Огайо, США и Diboride Conductors в Кембридже, Великобритания. В первой из них сделали провода длиной 100 м, но электрические свойства протестировали только на более коротких образцах.

Как заявил президент Hyper Tech Майк Томсик (Mike Tomsic), метод протягивания проводов состоит в "упаковывании" сверхпроводника в железную трубку по всей длине. Как видно из рис.1 (все рисунки — на 2-й стр. обложки), U-образная железная трубка заполняется порошком, а затем закрывается, формируя круглый провод диаметром 6 мм, который затем вытягивается до четверти первоначальной толщины. Разрез провода показан на рис.2. Чтобы сделать многожильный провод, несколько нитей собирают в одной трубке, которую затем вытягивают до диаметра 1,4 мм.

Прогресс в производстве проводов и лент из диборида магния был очень быстрым. Уже в мае 2001 г. Суньо Джин (Sungho Jin) со своими коллегами из Agere Systems (Элленгаун, шт. Пенсильвания), приемники Lucent



Technologies, сделал ленту из диборида магния длиной почти 1 м с очень небольшими дефектами. Железные трубки с внешним диаметром 6 мм они наполнили порошком диборида магния, расплющили их и протянули. В результате получились ленты шириной 5 мм и толщиной 0,5 мм. В завершение ленты прокалили при температуре 900°C. При этом порошок расплавился, застыл и стал твердым. Используя такую технологию, можно получить провода любой длины.

Комфортная температура

Отличительной особенностью высокотемпературных керамических сверхпроводников является то, что они могут быть охлаждены дешевым жидким азотом, поддерживать температуру фазового перехода которого (77 К) стоит дешево. Обычные сверхпроводники, такие как ниобий-титан, необходимо охлаждать до очень низкой температуры жидкого гелия. Дибориду магния надо 20...30 К, чтобы он "стал полезным". Хотя эта температура ниже, чем у жидкого азота, она находится в пределах обычных промышленных криогенных охладителей, и поэтому не будет дорогой.

Диборид магния вполне подходит для таких применений как трансформаторы для электропередачи, электронные приборы, компьютеры. Скорее всего, именно эти применения позволят занять материалу устойчивые позиции на рынке.

Президент Huper Tech предполагает, что цена на транзисторы из диборида магния, работающие при температуре 25 К, составит около \$59/кА-м, большей частью за счет цены охлаждения. Для медных трансформаторов, работающих при 300 К (комнатных температурах), эта цифра колеблется около \$65, в основном, из-за потерь на сопротивлении. Для высокотемпературных керамических сверхпроводников, работающих при 77 К, в зависимости от цены провода цена колеблется от \$80 до \$180.

Эти провода могут использоваться и для повышения производительности электромоторов и генераторов, т.к. сверхпроводящие катушки могут проводить больший ток, чем медные. Тонкие сверхпроводящие пленки могут служить как электронные компоненты и удобные датчики магнитного поля. Другими потенциальными применениями могут стать сверхпроводящие магниты и беспроводные передающие радиостанции.

Диборид магния является очень привлекательным материалом для беспроводных сверхчувствительных приемников и передатчиков со сверх-

низким фазовым шумом.

Многие ученые, так или иначе, понимают революционное значение этого нового сверхпроводника. Как заявил Кава (Cava), принстонский эксперт, написавший комментарий по поводу важности открытия: "Если бы диборид магния открыли в 60...70-х гг., целая культура исследований сверхпроводников была бы совершенно другой".

Нет ничего идеального

Несмотря на свои уникальные свойства, у диборида магния есть и недостатки, которые надо преодолеть.

Раньше всего обнаружилась неспособность материала пропускать большой ток. В мае 2001 г. Джин и его коллеги из Agere Systems сообщили, что их образцы проводов могут пропускать 35 000 А/см². Этого, безусловно, мало для широкого применения. В настоящее время 200 000 А/см² при напряженности магнитного поля в 1 Тл является типичной цифрой для трансформаторов и электромоторов.

Большее беспокойство причиняет то, что диборид магния не выдерживает сильных магнитных полей. Данные показали, что при полях с напряженностью больше 2 Тл сверхпроводящие свойства исчезают. При этом в сплаве образуются магнитные вихри (рис.3), которые двигаются под воздействием силы Лоренца, создаваемой током. Это движение сопровождается рассеиванием энергии, что похоже на сопротивление.

Для решения этой проблемы предлагается сделать материал неидеальным по структуре. Группа Дэвида Кэплина (David Carlin) из Лондонского имперского колледжа обнаружила, что наличие индуцированных дефектов в структуре сплава связывает вихри и удерживает их от движения. Ученые бомбардировали образцы диборида магния протонами и показали, что полученные дефекты значительно увеличили способность материала управлять током с помощью магнитного поля. Но т.к. бомбардировка протонами непрактична для широкомасштабного производства проводов, участники группы предложили использовать химические добавки. "Даже с небольшим уровнем неупорядоченности (около 1%) материал становится технологически привлекательными независимо от того, каким способом он обработан — механическим или химическим", — заявил Кэплин.

Внесение химических примесей уже приносит ожидаемые улучшения. Чанг-Биом Иом (Chang-Beom Eom), профес-

сор по изучению и разработке материалов из Висконсинского университета, создал технологию добавки кислорода в тонкие сверхпроводящие пленки.

Кислород проникает в решетку материала и замещает некоторые атомы бора. Это позволяет улучшить как величину пропускаемости тока, так и устойчивость к магнитному полю. При добавлении кислорода также возникают небольшие количества оксида магния, что повышает неупорядоченность.

Но Иом не уверен, что технология, используемая им для повышения сверхпроводящих свойств в тонких пленках, подойдет и для изготовления проводов. Президент Huper Tech Томсик также предупредил о том, что, несмотря на подтверждение возможности создания способов улучшения связывания вихрей, сделать то же самое в проводах будет сложной задачей.

Если вышеперечисленные недостатки будут решены без нанесения ущерба стоимости производства, то провода из диборида магния могут быть использованы в медицинских диагностических приборах, работающих на мощных магнитах, для магнитного резонансного отображения.

Мечты о будущем

Окрыленные успехом последних лет, некоторые ученые мечтают о том дне, когда провода из диборида магния получат широкое применение. Но даже если материал подтвердит свои преимущества перед другими традиционными сверхпроводниками, он может столкнуться с другим так называемым высокотемпературным сверхпроводником второго поколения — соединением иттрия, бария, меди и кислорода (ИБМК), находящимся на ранней стадии разработки.

ИБМК становится сверхпроводником при температурах выше температуры жидкого азота. Но по сравнению с другими высокотемпературными сверхпроводниками, он обладает лучшими характеристиками в магнитном поле и более дешевый. Интересно заметить, что диборид магния и ИБМК находятся на одном временном уровне развития. Томсик предполагает выпустить диборид магния в продажу к 2004...2005 гг. по цене \$1...2/кА-м. Филип Дж. Пеллегрино (Philip J. Pellegrino), президент IGC-SuperPower, Нью-Йорк, планировал начать продавать ИБМК \$10/кА-м уже в конце 2002 г.

Так что, видимо, наступивший год не будет жарким!

По материалам журнала
"IEEE Spectrum".



В.КУЦ,

E-mail: victor_kootz@mail.ru

Что такое системная плата

Не будет преувеличением сказать, что системная плата, являясь основным связующим элементом, "скелетом" компьютера, определяет не только возможность совместной работы всех его подсистем и взаимодействие с дополнительными устройствами и элементами, но и накладывает определенные ограничения, устанавливая пределы потенциальных возможностей системы по модернизации и расширению ее потребительских функций в будущем. Поэтому при покупке нового или модернизации существующего компьютера проблеме правильного выбора системной платы необходимо уделить повышенное внимание, а для этого необходимо знать, какие элементы могут находиться на системной плате, и для чего они предназначены.

Типоразмеры системных плат

Для самых первых компьютеров типа IBM AT были разработаны системные платы стандарта Full Size AT, имевшие впечатляющие размеры 35,6 x 30,5 см. До настоящего времени такие монстры, конечно, не дожили, однако их уменьшенная модификация Baby-AT (24,4 x 21,8 см) оказалась довольно удачной и составила основную массу системных плат для процессоров трех поколений — от 386 до Pentium включительно. Но все хорошее когда-нибудь кончается, и господство плат формата Baby-AT закончилось с появлением в 1995 г. нового стандарта системных плат — ATX, предложенного корпорацией Intel. Платы этого типоразмера, сочетая в себе все лучшее из предыдущих разработок в этой области, сегодня являются основным стандартом (рис. 1).

Спецификацией ATX определен размер платы 30,5 x 24,4 см. Системные платы ATX обеспечивают поддержку широкого набора дополнительных функ-

ций, таких как перевод компьютера в "спящий" режим и выход из него программными методами или с помощью определенной комбинации клавиш, использование унифицированной задней панели с разъемами ввода/вывода, более рациональное размещение разъемов и устройств на плате. Кроме того, на базе стандарта ATX созданы следующие типоразмеры:

- Mini-ATX — 28,4 x 20,8 см, особенно распространения не получил и уже ушел в небытие;

- Micro-ATX — 24,4 x 24,4 см, популярный форм-фактор для недорогих решений начального уровня, офисных компьютеров или сетевых терминалов.

Отдельно необходимо отметить типоразмер Flex-ATX, ориентировочный размер платы которого составляет 23 x 19 см. Хотя в его наименовании присутствует аббревиатура ATX, он далеко не по всем параметрам соответствует этой спецификации. Так как платы типоразмера Flex-ATX предназначены, в первую очередь, для нестандартных малогабаритных корпусов, вошедших в моду после выпуска компанией Apple компьютера iMac, популярность которого связана исключительно с новаторским внешним видом, то конфигурация внешних разъемов на платах Flex-ATX может

не соответствовать общему стандарту и в каждом конкретном случае быть индивидуальной. Но наименьший на сегодняшний день формат системных плат Mini-ITX предложила в начале 2002 г. фирма VIA. Размер плат в этом стандарте — 17 x 17 см, что на 30% меньше Flex-ATX.

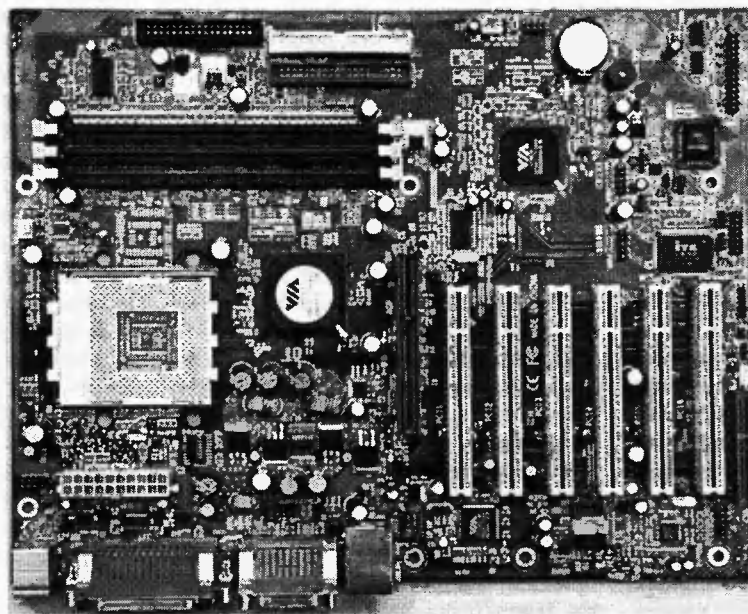
С появлением мощных процессоров Pentium 4 фирма Intel немного усовершенствовала спецификацию ATX. В новой редакции ATX 2.03 предусмотрено использование отдельного разъема для подачи напряжения питания +12 В, а также большое внимание было обращено на повышение надежности крепления массивного радиатора системы охлаждения процессора. Все эти изменения вынуждают, при использовании системных плат стандарта ATX 2.03 для модернизации компьютера, обновить не только блок питания (так как старые не имеют дополнительного разъема +12 В), но и корпус компьютера, так как новая система крепления радиатора задействует и отдельные корпусные элементы.

Сердце системы — процессор и его окружение

Главное место на любой системной плате занимает процессорный разъем. Современные компьютеры имеют до-

вольно большое количество разновидностей форм-факторов гнезд для процессоров, и их число постоянно растет. К сожалению, времена систем на базе единого процессорного гнезда (совсем недавно это был Socket 7) безвозвратно ушли в прошлое. Конкурентная борьба не на жизнь, а на смерть, обострившаяся в последнее время, а также маркетинговая политика компьютерных гигантов, направленная на получение максимальных прибылей, приводит к тому, что уже сейчас практически каждая новая серия процессоров имеет свой, от-

Рис. 1





личный от предшественников и, тем более, от конкурентов, тип разъема. Некоторые типы процессорных разъемов, такие как Socket A для процессоров компании AMD, без изменения используются уже несколько лет, позволяя без особых проблем производить относительно недорогой апгрейд системы. Иллюстрацией прямо противоположной тенденции может служить разъем Socket 423 для процессоров Pentium 4, срок активной жизни которого оказался меньше года.

Все современные процессоры используют разъемы типа ZIF Socket, обеспечивающие нулевое усилие при установке и различающиеся, в основном, только количеством выводов процессора. Их основные типы, которые можно встретить сегодня или в ближайшем будущем, перечислены в таблице:

Разъем	Количество выводов	Поддерживаемые процессоры	Примечание
Socket 370	370	Pentium III, Celeron	Устаревший
Socket A	462	Athlon, Duron	
Socket 423	423	Pentium 4 (Willamette)	Устаревший
Socket 478	478	Pentium 4 (Willamette, Northwood)	
Socket 754	754	Athlon XP 64	Разрабатывается

Работу процессора и его взаимодействие с памятью и внешними устройствами обеспечивает набор микросхем системной логики, или, короче, чипсет. Именно он в настоящее время определяет как производительность, так и саму архитектуру системной платы. Чаще всего чипсет состоит из двух основных частей — микросхем северного и южного мостов. Северный мост обеспечивает взаимодействие процессора с памятью, организует работу шин AGP и, в устаревших моделях, PCI. Южный мост, по большому счету, является контроллером периферийных устройств. Обычно он содержит контроллер шины PCI (поддерживающий до 6 мастер-слотов), двухканальный IDE-контроллер (ATA33/66/100, а чипсеты от VIA и SiS, кроме того, еще и ATA133), два-три канала USB (до 6 портов), звуковой контроллер AC97 (самые современные чипсеты обеспечивают 6-канальный звук и цифровые выходы типа S/PDIF) и сетевой контроллер. По типу связи между северным

и южным мостами все чипсеты подразделяются на 2 большие группы — классические и с хабовой архитектурой. Мосты классических чипсетов соединяются между собой по шине PCI, тогда как у чипсетов с хабовой архитектурой — по специальной высокоскоростной магистрали, обеспечивающей значительно большую пропускную способность шины по сравнению с PCI. Большое значение широкой полосе пропускания этой магистрали стали придавать в последнее время, когда в составе чипсетов появились контроллеры внешних шин новых типов (USB 2.0, IEEE1394), позволившие организовать работу ПК с самыми современными высокоскоростными периферийными устройствами.

Отдельной группой стоят интегрированные чипсеты, в северный мост

которых входит графический контроллер, что, ценой незначительного увеличения стоимости самого чипсета и, соответственно, системной платы на его основе, позволяет за счет исключения достаточно дорогого внешнего видеоадаптера существенно снизить стоимость готовой системы. Таким образом, интегрированные чипсеты в основном используются для построения недорогих маломощных систем, например, офисных или школьных компьютеров.

Системная память

На сегодняшний день сложилась парадоксальная ситуация, когда в

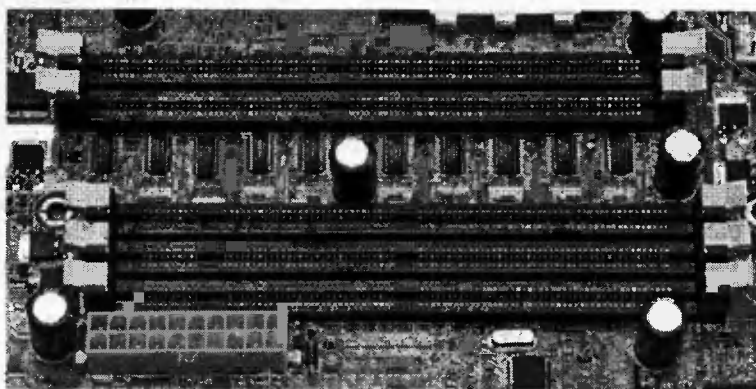
компьютерах используется сразу три различных, не совместимых между собой типа памяти. Это, в первую очередь, старая добрая SDRAM, пока еще достаточно широко распространенная, но стремительно устаревающая и поэтому совершенно бесперспективная. На смену ей пришла значительно более быстрая память DDR SDRAM, ставшая сегодня отраслевым стандартом де-факто. Третьим типом современной системной памяти является RDRAM — еще более быстрая чем DDR, но, из-за ряда как технических, так и, самое главное, маркетинговых просчетов разработчиков, занимающая сегодня мизерную долю рынка памяти и не имеющая абсолютно никаких перспектив ее увеличить.

На системных платах, использующих память типа SDRAM, применяются 168-контактные разъемы для модулей типа DIMM. Изначально модули памяти SDRAM работали на частоте 66 МГц, но сейчас ни их, ни более быстрых модулей стандарта PC100, работающих на частоте 100 МГц, найти в продаже уже невозможно. Практически вся существующая память соответствует спецификации PC133 с рабочей частотой 133 МГц, что обеспечивает ей пропускную способность 1064 Мб/с.

Подавляющее большинство современных системных плат используют память DDR SDRAM, модули типа DIMM которой имеют разъем с 184 контактами. Передавая информацию по обоим фронтам синхроимпульса на частоте 133 МГц, память DDR SDRAM обеспечивает скорость передачи до 2,1 Гб/с, что и отразилось в названии спецификации — PC2100 (или DDR266). Но прогресс не стоит на месте, и ей на смену пришли следующие варианты — PC2700 (DDR333), имеющий, как легко догадаться, скорость передачи, достигающую уже 2,7 Гб/с на рабочей частоте 166 МГц, и PC3200 (DDR400). Существенное значение может иметь, в первую очередь для мобильных систем, еще и то, что память DDR SDRAM имеет напряжение питания 2,5 В, в отличие от обычной SDRAM, питающейся от 3,3 В.

Слоты для модулей памяти разных типов,

Рис. 2



хоть и различаются по числу контактов, но очень похожи друг на друга (рис.2, два верхних слота — для DDR SDRAM, а три нижних — для SDRAM).

И, наконец, третий тип современной памяти — RDRAM — был разработан фирмой Rambus при широкой поддержке Intel, но, по причине слишком высоких лицензионных отчислений, установленных фирмой Rambus, первоначальные цены на этот тип памяти оказались неоправданно высокими, и возможность завоевания рынка была безвозвратно упущена. Модули памяти RDRAM устанавливаются в 168-контактные разъемы типа RIMM, причем, так как ширина шины памяти составляет всего 16 бит, чипсеты имеют двухканальные контроллеры памяти, поэтому модули устанавливаются на плату попарно. Для передачи информации используются оба фронта тактового импульса, что фактически удваивает частоту работы. В настоящий момент самая быстрая RDRAM работает на частоте 400 МГц, обеспечивая пиковую пропускную способность одного канала 1,6 Гб/с. Но в реальных приложениях очень высокая скорость RDRAM часто нивелируется ее не менее высоким показателем латентности (запаздывания) сигнала.

Кроме рассмотренных трех типов системной памяти, если очень хорошо поискать, на старых платах формата Baby-AT для первого Pentium еще можно иногда увидеть 72-контактные разъемы для модулей типа SIMM, используемые для памяти совсем уж устаревших типов — FPM и EDO RAM. Модули типа SIMM разрабатывались еще в эпоху 486-х процессоров, поэтому имели 32-разрядную ширину шины. В системные платы для процессоров типа Pentium, имеющих 64-разрядную шину данных, такие модули необходимо вставлять только попарно.

Дисковые интерфейсы

Для подключения различных дисковых накопителей, таких как жесткие диски (винчестеры) или CD-ROM и их производные (CD-R/RW, DVD и пр.), в подавляющем большинстве случаев используют различные варианты интерфейса IDE (Integrated Device Electronics — устройство с интегрированным контроллером), часто называемого еще и ATA (рис.3).

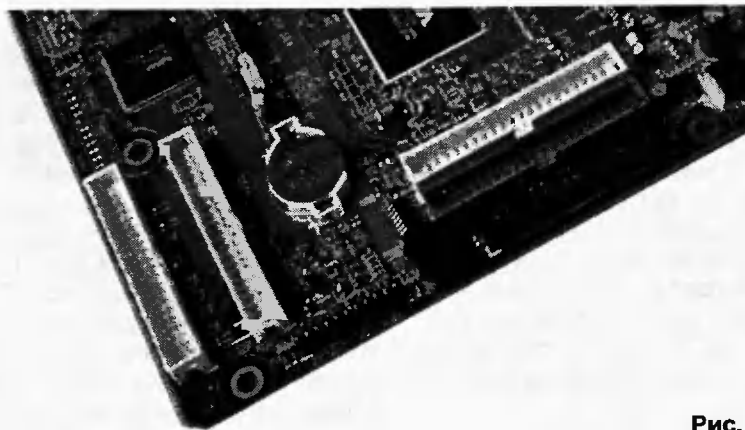


Рис. 3

Практически любая современная системная плата имеет два канала IDE-контроллера, интегрированных в чипсет — первичный (Primary) и вторичный (Secondary), каждый из которых представляет собой, по сути, отдельный контроллер, имеющий собственный разъем на плате и отдельный шлейф с двумя оконечными разъемами, к которому можно подключить один или два накопителя. Таким образом, стандартный IDE-контроллер обеспечивает одновременное подключение до 4-х дисковых накопителей.

Сегодня самым распространенным вариантом интерфейса IDE является Ultra ATA/100, обеспечивающий передачу данных со скоростью 100 Мб/с. Чтобы подключить жесткий диск, поддерживающий спецификацию Ultra ATA/100, необходим специальный 80-жильный кабель. Для обеспечения стабильности передачи данных на высоких частотах каждая из 40 информационных линий шлейфа чередуется с общим проводом, а в остальном его разводка ничем не отличается от старого 40-жильного кабеля, с которым возможна работа дисковых устройств стандарта Ultra ATA/33 и ниже. В 2001 г. известный производитель жестких дисков, фирма Maxtor, представила спецификацию на новый интерфейс Ultra ATA/133, однако его использование не приводит к реальному увеличению скорости работы современных жестких дисков по сравнению со старым Ultra ATA/100. Тем не менее, интерфейс Ultra ATA/133, активно поддерживаемый тайваньскими производителями чипсетов и высокомерно проигнорированный Intel, завоевал определенную популярность, тем более, что благодаря новому режиму BigDrive появилась возможность преодолеть барьер емкости IDE-дисков в 137 Гб.

В скором времени на смену параллельному интерфейсу ATA, практически полностью исчерпавшему свои возможности по наращиванию скорости обмена информацией с дисками, должен прийти новый последовательный интерфейс Serial ATA. Он будет поддерживать все накопители, включая винчестеры, CD, DVD, флоппи-дисководы и другие подобные устройства при подключении их к системным платам. Как следует из названия, в Serial ATA вместо передачи данных в параллельном режиме по многожильному кабелю используется принцип последовательной передачи данных. В самом начале 2002 г. была анонсирована первая спецификация Serial ATA, но его широкое применение начнется только в начале нынешнего года, когда получат распространение чипсеты со встроенными Serial ATA контроллерами. Первое поколение Serial ATA будет иметь пропускную способность 1,5 Гбит/с, затем скорости возрастут до 3 Гбит/с во второй спецификации, и позднее — до 6 Гбит/с, при этом сохранится обратная совместимость с Serial ATA первой версии. Интерфейс предназначен для подключения внутренних устройств, поэтому максимальная длина кабеля относительно невелика (1 метр), тем не менее, это в 2 раза больше, чем у существующего параллельного ATA-интерфейса (всего 45 см). Кроме того, использование тонких кабелей позволяет более рационально расположить их в корпусе компьютера, тем самым уменьшив помехи движению воздушных потоков от вентиляторов системы охлаждения компьютера.

(Окончание следует)



Поведенческое моделирование в PSPICE

О. ПЕТРАКОВ,
E-mail: petrakov@mtu-net.ru

НЕ ТОЛЬКО НОВИЧКУ

Радиолюбители уже давно оценили возможности программ моделирования. Наиболее популярные и доступные из них — ELECTRONIC WORKBENCH, MICROCAP-7, DESIGN LAB и OrCad-9.2. Как правило, после этапа освоения такой программы остро встает вопрос создания моделей новых компонентов. В этой статье дается обзор способов пополнения симуляторов моделями, подробно рассматривается поведенческое моделирование, которое особенно эффективно при создании моделей новых компонентов, особенно сложных.

1. Методы расширения возможностей PSPICE

Общее у программ моделирования ELECTRONIC WORKBENCH, MULTISIM, MICROCAP-7, DESIGN LAB и OrCad-9.2 — то, что все они базируются на PSPICE. Будучи доступными, обладая простыми интерфейсом и входным языком, они получили наибольшее распространение как среди любителей, так и среди профессионалов на территории СНГ. Перечисленные программы имеют схожие возможности, хотя обладают разным, а порой и оригинальным интерфейсом. Например, ELECTRONIC WORKBENCH и MULTISIM имитируют реальные измерительные приборы, которые можно подключать к схеме, нарисованной на экране. При должной глубине познания с помощью любой из этих программ можно эффективно решать очень сложные схемотехнические задачи. При этом во многом успех моделирования определяется наличием хороших библиотек PSPICE-моделей компонентов. Недостаток моделей легко компенсируется умением их разрабатывать. Некоторым опытом в этом направлении мы поделимся с читателями. Данная статья рассчитана на тех, кто уже имеет опыт работы с такими программами и общается с компьютером "на ты". Все примеры в статье разработаны в OrCAD-9.1, но их легко модифицировать под любую другую программу.

Любые симуляторы электронных схем, в том числе и PSPICE, содержат встроенные модели для ограниченного количества устройств. Моде-

лирование схемы, содержащей компонент, не входящий во встроенный набор, требует расширения возможностей симулятора каким-либо способом. Есть три пути расширить возможности симулятора, и, в частности PSPICE:

- модифицировать исходный код симулятора, чтобы добавить новую модель компонента;
- формировать макромоделю на основе внутреннего устройства компонента;
- поведенческое моделирование.

Модификация исходного кода

Модификация исходного кода PSPICE — сильный метод. Однако использование этого метода связано со значительными проблемами. Среда, для которой готовится код нового SPICE-компонента, далеко не простая. Требуется высококлассный программист-эксперт, способный проверить, что дополнительный код обслуживается верно, и что сам PSPICE в целом продолжает после этого действовать правильно. Далеко не всем доступно подобное знание и понимание какой-либо реализации PSPICE. Вероятно, этот способ доступен только элите в университетской среде.

Кроме этого, как только PSPICE модифицирован, он становится нестандартным. Это потенциальная проблема, если моделирование когда-либо должно проводиться в другом месте со стандартным PSPICE. К тому же, всякий раз, когда приобретается новая (разумеется, не модифицированная) версия PSPICE, весь труд по перенесению и проверке работоспособности модификаций придется повторить.

Формирование макромоделей

Формирование макромоделей устройства производится с использованием существующих встроенных в симулятор примитивов. Этот метод хорош для моделирования составных устройств, таких как, например, оптроны. Обычно есть одна или несколько связей между компонентами составного устройства, и они макромоделюются, хотя некоторые

функции могут моделироваться абстрактно с использованием управляемых источников напряжения или тока. Данный метод работает не так хорошо, когда характеристики устройства даны в форме уравнения или как набор измеренных величин. В этих случаях, по возможности, необходимо прибегать к методам, подобным тем, которые применяются, например, при синтезе логарифмической зависимости тока от напряжения, когда передается ток через идеальный диод или считается напряжение с диода. Макромодели, смонтированные этим методом, часто получаются весьма сложными, разработчику их трудно сопровождать, они могут быть медленным и неточным.

Поведенческое моделирование

Поведенческое моделирование (Behavioral Modeling) является процессом разработки модели компонента или системы с точки зрения внешне наблюдаемого поведения, а не из описания их внутреннего устройства.

Отметим два важных приложения поведенческого моделирования: моделирование устройств нового типа и моделирование сложных систем в виде черного ящика. Далее мы рассмотрим инструменты, имеющиеся в PSPICE, поддерживающие возможности поведенческого моделирования и приведем примеры.

2. Моделирование новых компонентов

Задание произвольных напряжений

Источник напряжения можно задать произвольным выражением, которое может содержать константы, параметры, напряжения и токи узлов, время, математические функции, включая log, exp и тригонометрические. На рис. 1 приведен пример задания источника напряжения выражением.

Рис. 1



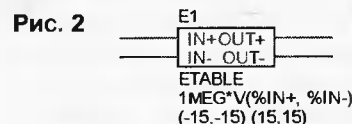
Табл. 1

Имя, назначение	Символ
ABM, число π (источник ЭДС)	
ABM1, делитель 1:10 (управляемый источник ЭДС)	
ABM2, среднее двух напряжений (источник ЭДС)	
ABM3, среднее трех напряжений (источник ЭДС)	
ABM/I, число (источник тока)	
ABM1/I, входное напряжение +100 В, деленное на 1000 (источник тока)	
ABM2/I, среднее двух напряжений (источник тока)	
ABM3/I, среднее трех напряжений (источник тока)	
ABS, абсолютная величина	
ARCTAN, арктангенс (источник ЭДС)	
ATAN, арктангенс (источник ЭДС)	
BANDPASS, чебышевский полосовой фильтр	
BANDREJ, чебышевский режекторный фильтр	
CONST, источник постоянного напряжения	
COS, функция косинус	
DIFF, вычитание напряжений	
DIFFER, дифференцирующее устройство	
EXP, функция экспонента	
FTABLE, табличное задание передаточной функции в частотной области	
GAIN, усилитель	
GUMPT, жесткий усилитель-ограничитель	
HILO, усилитель-ограничитель с характеристикой тангенс гиперболический	
HIPASS, фильтр верхних частот	
INTEG, интегратор (источник тока, пропорциональный интегралу входного напряжения)	
LAPLACE, преобразование Лапласа	
LIMIT, ограничитель	

Имя, назначение	Символ
LOG, натуральный логарифм	
LOG10, десятичный логарифм	
LOPASS, фильтр нижних частот	
MULT, перемножитель	
PWR, степенная функция	
PWRS, степенная функция	
SIN, функция синус	
SOFTLIM, мягкий усилитель-ограничитель (с характеристикой тангенс гиперболический)	
SQRT, корень квадратный	
SUM, суммирование напряжений	
TABLE, табличное задание передаточной функции	
TAN, функция тангенс	
EFREQ, табличное задание комплексного коэффициента передачи	
ELAPLACE, задание коэффициента передачи преобразованием Лапласа	
EMULT, перемножитель напряжений	
ESUM, сумматор напряжений	
ETABLE, табличное задание передаточной функции	
EVALUE, нелинейная передаточная функция	
GFREQ, табличное задание комплексного коэффициента передачи источника тока	
GLAPLACE, задание коэффициента передачи источника тока преобразованием Лапласа	
GMULT, перемножитель напряжений (источник тока)	
GSUM, сумматор напряжений (источник тока)	
GTABLE, табличное задание передаточной функции источника тока	
GVALUE, нелинейная передаточная функция источника тока	
TABLE, передаточная функция источника напряжения, заданная таблично	

Моделирование операционного усилителя

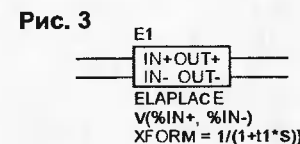
Приведем простой пример использования управляемого источника напряжения, заданного таблицей, как операционного усилителя без использования его полной, гораздо менее быстродействующей модели (рис.2):



Для реализации операционного усилителя использован аналоговый блок ETABLE (табличное задание передаточной функции). При этом выходное напряжение ограничено значениями ± 15 В, т.е. пределами физически реальных границ. В этих границах характеристика аппроксимируется прямой линией, усиление составит 10^6 .

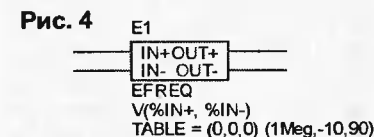
Лапласовы выражения

Лапласовы выражения при задании передаточной функции могут включать константы, параметры и математические функции от S , включая $\log$, $\exp$, и тригонометрию. Пример задания передаточной функции Лапласовым выражением показан на рис.3.



Табличное задание АЧХ

При табличном задании амплитудно-частотной характеристики амплитуда и фаза линейно интерполируются в таблице. На рис.4 приведен пример табличного задания АЧХ.



Полный перечень аналоговых блоков, содержащихся в библиотеке ABM.SLB и предназначенных для поведенческого моделирования, приведен в табл.1. Как видим, инструментарий достаточно богатый. В дальнейшем мы им частично воспользуемся.

(Продолжение следует)



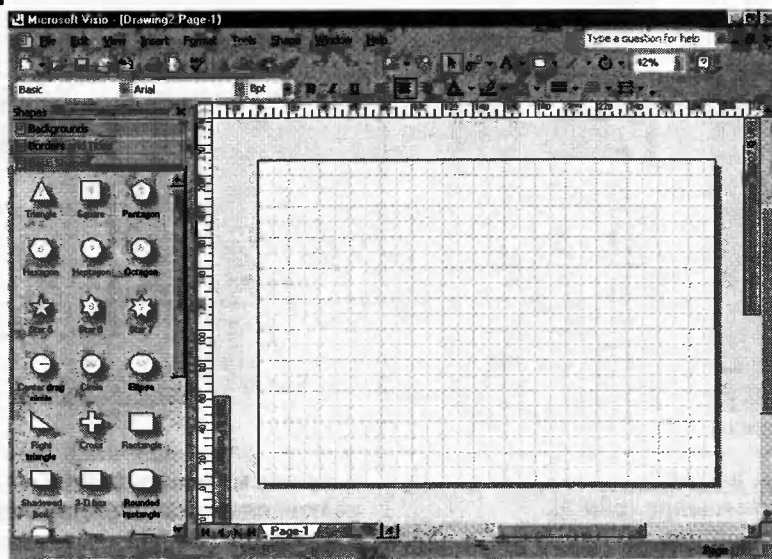
Работа с редактором деловой графики Visio 2002

(Продолжение. Начало в №№12/2002, 1/2003)

Работа с трафаретами

В предыдущей статье мы рассмотрели основные приемы работы с фигурами в Visio. Здесь еще раз можно вспомнить иерархию основных объектов программы: фигуры (**Shapes**) объединяются в трафареты (**Stencils**), а трафареты, в свою очередь — в шаблоны (**Templates**). Мы немного подробнее рассмотрим работу объектами "второго этажа" — с трафаретами.

Рис. 1



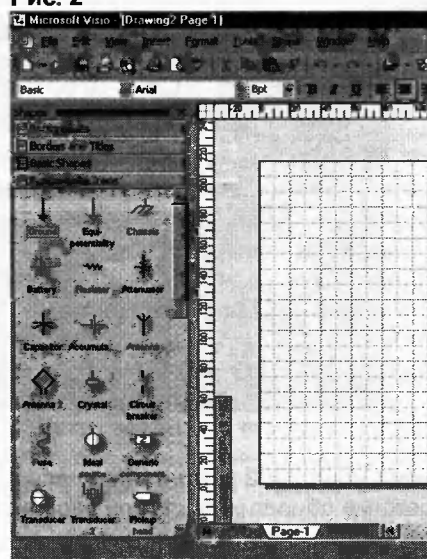
Чаще всего приходится сталкиваться со следующей проблемой. При создании рисунков на основе стандартных шаблонов, возникает необходимость использовать фигуру из совершенно другого набора. Например, как в простейшую блок-схему (**Basic Diagram**) включить, к примеру, электрическую цепь?

Итак, начинаем действовать — после загрузки Visio выбираем категорию **Block Diagram**, а в ней шаблон **Basic Diagram**. Стандартный набор трафаретов (рис.1) содержит только фоновые рисунки (**Backgrounds**), заголовки и границы (**Borders and Titles**) и основные фигуры (**Basic Shapes**). А для рисования электродеталей нам необходимо еще воспользоваться трафаретом **Fundamental**

Items из шаблона **Basic Electrical**. Данная проблема решается просто — в меню **File** выбираем **Stencil/Electrical Engineering/Basic Electrical**. В результате этих манипуляций в левой половине рабочего окна появляется четвертый трафарет (рис.2), и блок-схему можно разукрасить изображением электрической схемы. Можно также поступить и более радикально — начать с пустой страницы и вызвать все

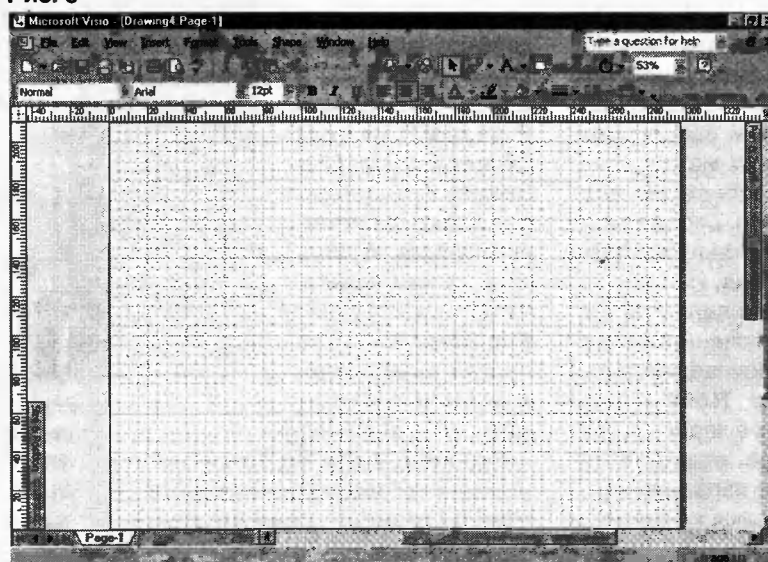
Однако разница между трафаретами не всегда заключается только в начертании фигур — имеется несколько специализированных трафаретов с достаточно специфическими свойствами. Примером может служить набор фоновых рисунков **Backgrounds** из **Visio Extras** (рис.4). Во-первых, размер рисунка при перетаскивании автоматически подстраивается под размер рабочей страни-

Рис. 2



необходимые шаблоны. Для этого используется команда **File/New/New Drawing** (рис.3) и выбор необходимых трафаретов **File/Stencil/...**

Рис. 3



цы. Во-вторых, добавление фона зависит еще и от способа установки программы. В Visio 2000 при перетаскивании фонового рисунка автоматически создавалась дополнительная фоновая страница, и

Рис. 4

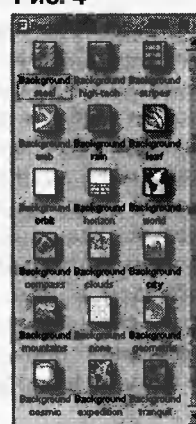


Рис. 5

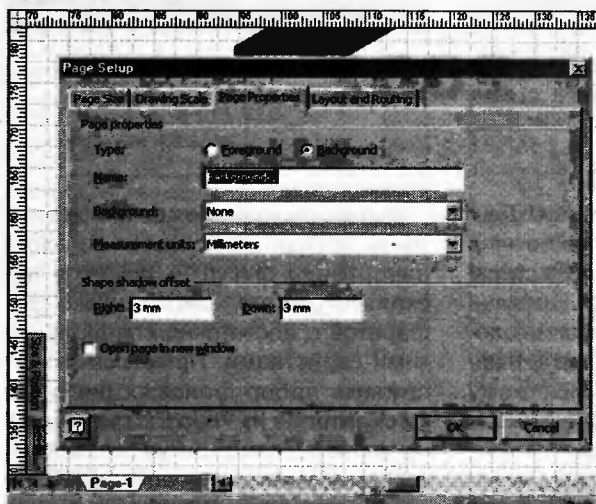
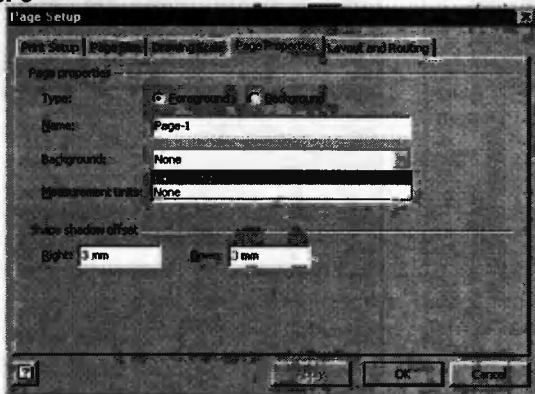


Рис. 6



если Visio 2002 устанавливалась в режиме обновления старой версии, то наследуется старый режим работы. При установке версии 2002 "с нуля" эта процедура выглядит сложнее. Прежде всего, необходимо создать фоновую страницу — **Insert/New Page**, а в диалоговом окне установить переключатель в положение **Background** (рис.5). Фоновый рисунок перетаскивается на новую страницу. Внизу рабочего окна видны ярлычки с названиями страниц, эти ярлычки используются для переключения между страницами. Сейчас необходимо вернуться на исходную страницу и, после выбора команды **File/Page Setup/Page Properties**, указать, какая именно страница является фоновой (рис.6), отключение же фоновой страницы производится выбором **Background: None**. Таким образом, в Visio, как в хорошем графическом редакторе, можно создавать многослойные изображения. Фигуры, подстраивающие размеры

площадь, а **Room** — указывает размеры помещения (рис.8).

Разные версии и подверсии Visio содержат различный набор трафаретов, и перенос этих объектов также не является сложной операцией. Шаблоны хранятся в виде файлов с расширением *.vss в соответствующей папке Visio, например, в случае Visio 2002 в папке **C:/Program Files/Microsoft Office/Visio10/1033/Solutions** вы найдете папки с шаблонами. Список именно этих папок открывается по выбору меню **File/Stencil**. Нужно просто скопировать файлы в соответствующую директорию.

Впрочем, если же вам все-таки не хватает стандартных средств, то можно создать и пользовательские

под размеры рабочего листа, содержатся и на трафарете **Borders and Titles** (границы и заголовки — рис.7). Другой пример активных фигур можно найти на трафарете **Walls, Doors and Windows** из набора **Building Plan**. Например, фигура **Space** автоматически вычисляет свою пло-

Рис. 7



трафареты. В первую очередь, желательно создать папку (к примеру, с именем **My**) в **C:/Program Files/Microsoft Office/Visio10/1033/Solutions**. Далее делаем заготовку трафарета — **File/Stencil/New Stencil**. На рабочем листе создаем и группируем фигуры. Для создания трафарета перетаскиваем при помощи мыши фигуры на заготовку (рис.9). Фигурам на трафарете можно дать произвольное имя (даже на русском языке) и сопроводить комментарием — сделать щелчок правой клавишей мыши по фигуре/**Master Properties**. В диалоговом окне (рис.10) указывается новое имя фигуры и текст всплывающей подсказки. После этого при на-

Рис. 8

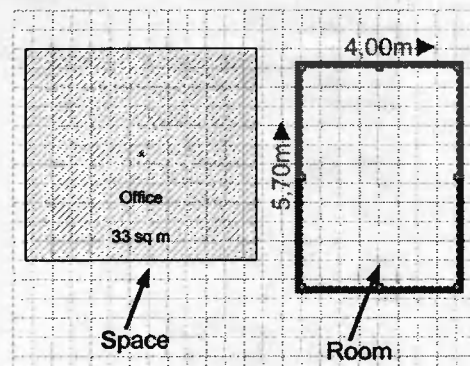
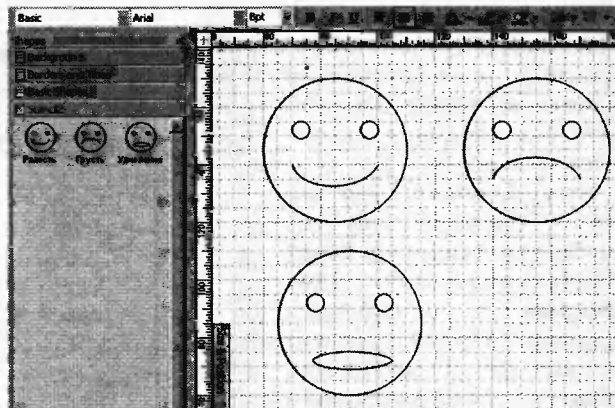
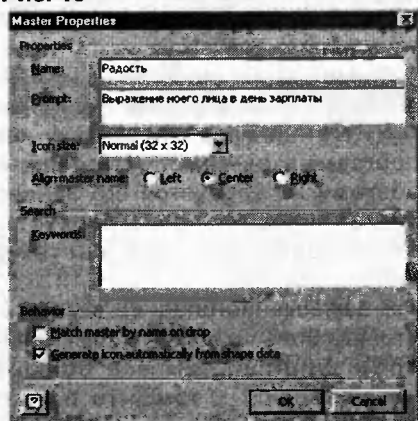


Рис. 9



диалоговом окне сохранения файла указывается папка (**My**). Наш новый трафарет становится доступным из меню **File/Stencil** (рис.12).

Рис. 10



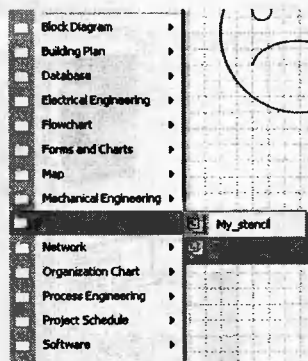
В любой момент результаты нашего труда можно подредактировать — сделать щелчок правой клавишей мыши по заголовку трафарета/Edit, символ редактирования — красная звездочка в заголовке окна.

Рис. 11



Из этих примеров видно, что программа Visio не только обладает большим набором инструментальных средств, но и позволяет в этот набор включать пользовательские разработки. Для эффективного их использования необходимо хотя бы в общих

Рис. 12



чертах ориентироваться в этой массе объектов. Поэтому в следующий раз мы сделаем краткий обзор основных шаблонов программы.

(Продолжение следует)

На вершине музыкального Олимпа

Аудиоформат MP3 & Player WinAmp

А.ГОРЯЧКИН,

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

У пользователей персональных компьютеров давно возникло желание прослушивать звуковые файлы с хорошим качеством. Но размеры звуковых файлов профессионального качества довольно велики и занимают очень много места (20...70 Мб). Стерефонический звуковой сигнал длительно в одну минуту с качеством аудио-компакт-диска потребует на винчестере около 10 Мб. Поэтому хранить такие файлы на жестком диске или передавать их по сети Интернет — дело довольно хлопотное и дорогостоящее.

Для уменьшения объема аудио-файлов стали применять программы-кодеры, которые производят сжатие по специальным алгоритмам. Использование этих кодеров позволяет уменьшить объем звуковых файлов в 10...13 раз почти без потери качества. В алгоритм сжатия закладывается удаление избыточной информации, отсутствие которой практически неощутимо на слух. В результате получается близкое к CD качество звука при значительно меньшем размере файла.

Сегодня самый популярный среди форматов сжатых аудиофайлов — MP3 (MPEG 1.0 layer 3). После того как формат MP3 начал стремительно набирать популярность, появилось множество программных плееров, позволяющих прослушивать файлы MP3. Одна из самых известных программ такого рода — WinAmp (<http://www.winamp.com>). Для кодирования/декодирования файлов формата MP3 широко применяется аудиокодек Fraunhofer IIS. Информацию об этом кодеке и других продуктах технологии MPEG Layer-3 audio можно получить, посетив сайт в Интернете <http://www.iis.fhg.de/audio>. При кодировании файлов задают параметр bitrate — чем он выше, тем качественнее становится звучание, и, соответственно, увеличивается размер

файла. В Интернете наиболее распространены файлы формата MP3, кодированные с битрейтом 128 или 160 Кбит/сек. Есть аудиокодеры, которые позволяют кодировать, используя метод переменной пропускной способности VBR (Variable Bitrate). Кодирование с переменным битрейтом позволяет получать файлы MP3 более компактными, так как при этом экономится дисковое пространство. Более того, стабилизируется качество звучания файлов MP3. Файл, кодированный с переменным битрейтом, при проигрывании в WinAmp'e можно "опознать" по постоянно изменяющемуся числу в окошке "kbps". Программ с такими возможностями не очень много. Один из таких кодировщиков, "умеющих" кодировать файлы MP3 с переменным битрейтом — AudioCatalyst версии 2.1 и выше [1].

Есть сведения о том, что уже разработана обновленная версия формата MP3, которая называется MP3 Pro. Как утверждают создатели нового формата MP3 Pro, качество звучания, равнозначное прежнему формату, достигается уже при вдвое меньшем битрейте. Файлы нового формата можно прослушивать обычными MP3-плеерами, однако качество звучания при воспроизведении с использованием старого алгоритма декодирования будет значительно хуже.

WinAmp на сегодняшний день является бесспорным лидером среди программных проигрывателей звуковых файлов. Разработчик этого популярного плеера — фирма NullSoft (<http://www.nullsoft.com>). Плеер поддерживает огромное количество аудиоформатов с расширениями MP1, MP2, MP3, WMA, WAV, CDA, MID и др. Кроме того, WinAmp способен проигрывать аудиотреки с CD-дисков. Неповоротливому и тяжелому на подъем Windows Media Player'у от "мелкомягких" трудно соперничать с быст-



рым и мощным WinAmp. Несомненное преимущество плеера заключается в его способности использовать дополнительные плагины (plugins). Не будет преувеличением сказать, что для WinAmp создано громадное количество разнообразных плагинов и скинов. Плагины — это модули, расширяющие функциональные возможности программы. В физическом смысле плагины представляют собой файлы, как правило, имеющие расширение *.dll. Все плагины, созданные для WinAmp, по своему назначению разделяются на следующие категории:

- плагины ввода файловых форматов, поддерживаемых программой WinAmp;

- плагины вывода файлов с последующим их сохранением на жестком диске;

- плагины визуализации, создающие видеоэффекты на экране монитора во время прослушивания звуковых файлов. Обычно видеоэффекты выводятся в окне, но есть и такие, которые работают в полноэкранном режиме;

- плагины audio (DSP/эффекты), значительно улучшающие качество звука и придающие полноценное звучание даже записям с низким битрейтом. Существуют DSP-плагины с разнообразными звуковыми эффектами, например, с объемным звучанием, реверберацией и т.д.;

- плагины общего назначения.

Имена файлов плагинов ввода обычно начинаются с in..., плагинов вывода — out..., плагинов визуализации — vis..., например: in_mp3.dll, out_wave.dll, vis_avs.dll.

Есть плагины, с помощью которых плеер WinAmp может преобразовывать файлы MP3 в файлы формата WAV, или позволяющие просматривать видеофайлы форматов MPEG, AVI, MOV (Quick Time). Удобно то, что пользователю не надо самому выяснять, к какой категории относится тот или иной имеющийся в наличии плагин. Достаточно поместить все плагины в одну папку, например, в C:\Players\Winamp\Plugins\, и указать в настройках программы путь к местоположению подключаемых модулей, а WinAmp в процессе загрузки автоматически их распознает и отсортирует по назначению. Для того чтобы какой-либо

из плагинов был задействован программой, надо предварительно произвести его настройку. Плагины для WinAmp можно скачать с <http://www.winamp.com/plugins>. Однако перед тем как скачивать, надо учесть то обстоятельство, что не все плагины для WinAmp бесплатные. Помимо плагинов и скинов, на сайте разработчика можно найти русификаторы к программе WinAmp, а также обновленную версию самого проигрывателя WinAmp. Много полезной информации можно отыскать на русском сайте <http://www.winamp.ru>.

Кроме основного окна (см. рису-



нок), в программе имеется возможность открыть вспомогательные окна:

- список воспроизведения (playlist);
- 10-полосный эквалайзер;
- минибраузер Интернета для подключения к сайтам производителя.

С помощью playlist'a можно прочитать, а при необходимости и отредактировать информацию об имени исполнителя, названии композиции и альбома, жанре, годе выпуска альбома, а также внести свои комментарии (ID3 Tag Editor) и сохранить все это в файле.

Скины — сменные панели, позволяющие изменить внешний вид главного окна плеера на любой вкус. Есть скины, позволяющие менять внешний вид не только основного окна, но и всех остальных вспомогательных окон. Причем создание скинов стало делом не только дизайнеров из фирмы NullSoft, а скорее, интернациональным. Это только лишь раз свидетельствует о популярности WinAmp. Плеер способен работать с архивами ZIP, поэтому для экономии дискового пространства скины на жестком диске

можно хранить в заархивированном виде. Скины можно скачать с <http://www.winamp.com/skins>.

При установке WinAmp не обязательно устанавливать его в папку "Program Files", поскольку как программа он самодостаточен, и поэтому не требуется его заново устанавливать после переустановки операционной системы Windows 9x/Me.

В заключение следует заметить, что чем больше пользуешься плеером WinAmp, тем все больше убеждаешься в том, что разработчики этой программы определенно постарались сделать ее много-

функциональной и максимально удобной для пользователя.

Литература

1. А.Горячкин AudioCatalyst 2.1. — Радиомир. Ваш компьютер, 2002, №2, С.2.



Рисунок О.Попова



Копирование файлов из Интернета

С.ГРИНЧУК,

E-mail: grinchuk@nihe.niks.by

(Продолжение. Начало в №1/2003)

Наше прошлое занятие было посвящено общему знакомству с программой FlashGet. Кроме того, мы обсудили вопрос добавления и выполнения новых заданий на загрузку файлов из Интернета. Однако наверняка вы заметили, что у рассмотренного способа загрузки файлов есть два существенных недостатка. Во-первых, до начала загрузки необходимо настроить параметры каждого выполняемого задания — указать категорию FlashGet, в которую автоматически переместится файл после выполнения задания; определить папку, в которую будет помещен загруженный файл; указать, на сколько частей FlashGet должен разбить загружаемый файл; а также определить, как именно будет выполняться новое задание — немедленно, по расписанию, или же FlashGet должен просто добавить новое задание в список категории **Задания**. Конечно же, подобная настройка каждого задания весьма трудоемка. Во-вторых, требуется указать полный и точный адрес копируемого из Интернета файла, что уж совсем неудобно, т.к. возможны случайные опечатки при вводе. Поэтому сегодня мы поговорим о том, как можно облегчить процедуру добавления и настройки новых заданий на загрузку. А для этого необходимо осуществить уже знакомый нам

1. Запуск FlashGet с помощью команды Пуск/Программы/FlashGet/FlashGet.

2. Настройка стандартных параметров выполнения заданий загрузки.

Итак, давайте рассмотрим, какие средства предлагает FlashGet для оптимизации вашей работы. В том случае, если для большинства заданий используются похожие настройки, можно указать FlashGet, какие параметры следует использовать для каждого нового задания. При этом уже не требуется выполнять рутинную работу по настройке каждого задания в отдельности,

хотя, конечно же, за вами остается право для каждого добавляемого задания изменить какие-то параметры его выполнения. Для определения стандартных параметров выполнения заданий загрузки выполните команду **Сервис/Параметры закладки по умолчанию...** или щелкните по кнопке  на панели инструментов. На экране появится диалоговое окно **Свойства закладки по умолчанию** (рис. 4), которое

заклучение настройте запуск новых заданий вручную, для чего в разделе **Старт** установите переключатель **Вручную**. Напомним, что при этом FlashGet добавляет новые задания в список категории **Задания**, но выполнение заданий не начинается. Подтвердите выбор установленных параметров щелчком по кнопке **ОК**.

Давайте проверим, будет ли использовать FlashGet указанные нами параметры для всех новых заданий. Для этого нам нужно просто добавить новое задание. Как вы помните, выполнить эту операцию можно либо с помощью команды **Задания/Новое задание**, либо с помощью кнопки  на панели инструментов. Убедитесь в том, что диалоговое окно **Добавить новую закладку** действительно содержит установленные нами настройки.

Теперь для добавления нового задания на загрузку файла остается просто в поле **URL** указать адрес файла. Таким образом, первую сложность с настройкой заданий мы решили. Не забудьте закрыть диалоговое окно **Добавить новую закладку** щелчком по кнопке **Отмена**.

3. Способы добавления новых заданий загрузки.

А сейчас поговорим о том, как можно обойти вторую сложность — заполнение поля **URL** вручную. Для упрощения ввода адреса загружаемого файла, FlashGet предлагает несколько решений. Во-первых, можно настроить FlashGet на перехват содержимого буфера обмена, тем самым менеджер по загрузке файлов будет отслеживать всю информацию (в том числе и адреса),

содержит те же параметры, что и уже знакомое нам диалоговое окно добавления нового задания. Для начала определим стандартные параметры сохранения принятых файлов. Для этого в поле **Категория** установите **Принятые**, а для выбора папки, в которую будет помещен загруженный файл, щелкните по кнопке , расположенной рядом с полем **Сохранить**, и выберите свою рабочую папку. Не забудьте подтвердить свой выбор щелчком по кнопке **ОК**. Для настройки количества одновременно закупаемых частей, на которые будет разбито каждое новое задание, в поле **Разбить на** установите **5**. И в

Рис. 4

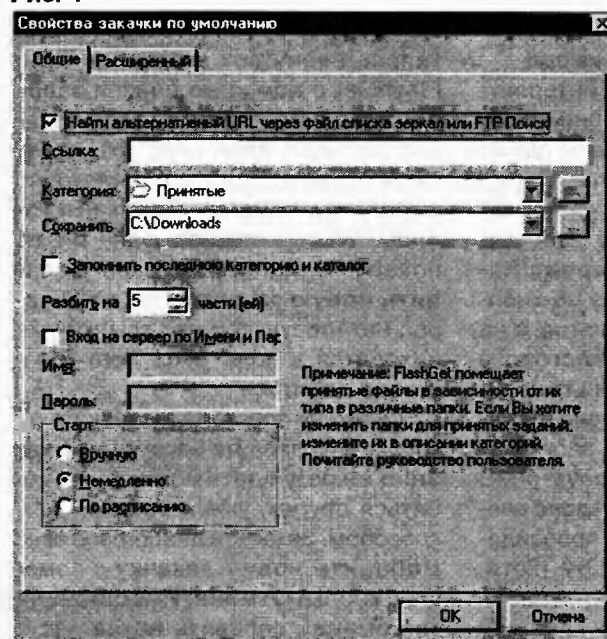
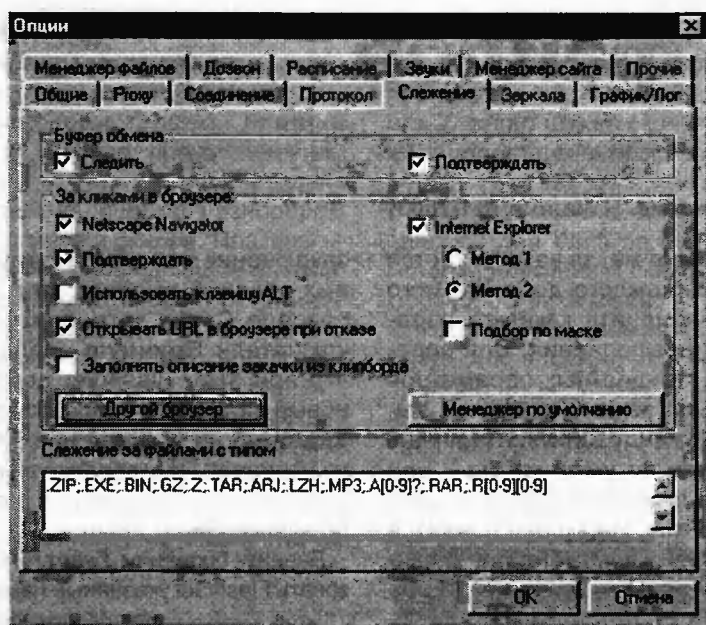


Рис. 5



помещаемую в буфер обмена. Продемонстрируем это на примере.

Предположим, что нас интересует текст договора на абонентское обслуживание, предлагаемое одним из провайдеров в Беларуси. Перед началом работы требуется проверить настройки FlashGet. Для этого выполните команду **Сервис/Настройки...** или щелкните по кнопке  на панели инструментов, и на вкладке **Слежение** диалогового окна **Опции** (рис.5) проверьте наличие в разделе **Буфер обмена** флажка **Следить**. Вот теперь загрузите хорошо знакомый вам браузер MS Internet Explorer, и в поле **Адрес** введите адрес сайта нужного провайдера — <http://www.unibel.by> (хотя, это может быть любой другой сайт). Не забудьте нажать на клавишу **Enter** на клавиатуре. Перейдите по ссылке **Услуги**, найдите на этой странице ссылку **Договор**. Установите курсор мыши на эту ссылку и просмотрите информацию в строке состояния браузера — там отобразится адрес интересующего нас документа. Каким же образом можно передать задание по загрузке этого документа FlashGet? Ведь не будем же мы в самом деле запоминать этот длинный и витиеватый путь к требуемому файлу. Достаточно щелкнуть правой кнопкой мыши по ссылке, ведущей к интересующему нас файлу, и в контекстном меню выбрать команду **Копировать ярлык**. При этом URL ссылки будет поме-

щен в буфер обмена. Теперь нам осталось вернуться в рабочее окно FlashGet с помощью панели задач Windows и определить новое задание загрузки, используя URL из буфера обмена. Для этого достаточно выполнить команду **Правка/Вставить URL**, и у вас на экране тут же появится диалоговое окно **Добавить новую загрузку**. При этом адрес интересующей нас ссылки уже находится в поле **URL**, чего мы и хотели добиться.

Однако не торопитесь щелкать по кнопке **OK**, давайте рассмотрим, как этого же результата можно было добиться другим, уже известным нам способом. Закройте диалоговое окно **Добавить новую загрузку** с помощью кнопки **Отмена**. И просто дайте команду **добавить новое задание: Задания/Новое задание** или кнопка  на панели инструментов. Посмотрите, результат будет точно такой же — появится диалоговое окно **Добавить новую загрузку** с указанием всех требуемых параметров. Вот теперь подтвердите добавление задания щелчком по кнопке **OK**.

Чтобы начать выполнение задания, выделите имя файла в категории **Задания** и выполните команду **Задания/Старт**, или же воспользуйтесь кнопкой  на панели инструментов. Дождитесь окончания копирования файла, после чего перейдите в категорию **Принятые**. На прошлом занятии для просмотра загруженного из Интернета файла мы

использовали Проводник. Чтобы несколько ускорить этот процесс, выполните двойной щелчок мышью по имени загруженного файла **contract.htm**.

Как видите, использование буфера обмена позволяет не только отказать от ввода адреса вручную, но и закачивать файлы любого типа, «стоящие» за гиперссылками. Однако, вызывая контекстное меню для гиперссылки по правой кнопке мыши, вы наверняка обратили внимание, что после установки FlashGet на ваш компьютер, программа автоматически расширила контекстное меню Internet Explorer, добавив в него две дополнительные команды — **Закачать при помощи FlashGet** и **Закачать все при помощи FlashGet**. Эти команды предоставляют возможность либо загрузить одну выбранную ссылку, либо загрузить все ссылки в пределах страницы.

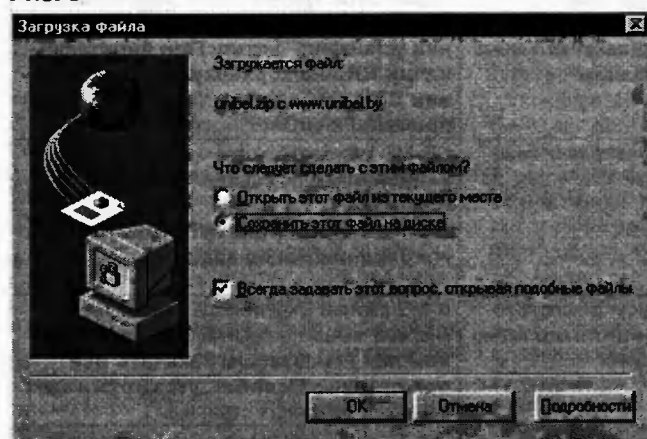
В качестве демонстрации в MS Internet Explorer с помощью кнопки **Назад** на панели инструментов вернитесь на предыдущую страницу (<http://www.unibel.by/rus/win/services/index.htm>). На загруженной странице выполните щелчок правой кнопкой мыши, например, по ссылке **Условия предоставления услуг**. Выберите в контекстном меню команду **Закачать при помощи FlashGet**. На экране вновь появится диалоговое окно **Добавить новую загрузку**, в котором в поле **URL** будет отображен адрес выбранного файла. Подтвердите добавление нового задания щелчком по кнопке **OK**. Выполните новое задание на загрузку, для чего выделите имя файла в категории **Задания** и воспользуйтесь кнопкой  на панели инструментов. Дождитесь окончания копирования файла, перейдите в категорию **Принятые** и двойным щелчком по имени файла просмотрите скопированный документ.

Просто, не правда ли? Однако для использования подобных способов добавления новых заданий следует помнить о существовании правой кнопки мыши. Хорошо было бы, чтобы FlashGet не только следил за нашим путешествием по просторам Всемирной паутины, но еще и анализировал, какие файлы мы пытаемся загрузить с помощью браузе-



ра, и только в случае необходимости (например, если загружаемый файл слишком велик) брал на себя задачу копирования файла. Оказывается, FlashGet в какой-то мере умеет и это. Только опять-таки предварительно следует настроить параметры связи FlashGet и Интернет-броузеров. Для этого еще раз выполните команду **Сервис/Настройки...** или щелкните по кнопке  на панели инструментов. Снова переместитесь на вкладку **Слежение** (рис.5). Обратите внимание, что здесь можно указать, что FlashGet должен следить не только за буфером обмена, но и за вашим Интернет-броузером — перехватывать запросы на получение файлов определенного типа. А вот за какими именно файлами будет следить FlashGet, указано в поле **Слежение за файлами с типом** (причем этот список при необходимости можно дополнять). По умолчанию в этом списке присутствуют в основном архивные и музыкальные файлы, т.е. именно те типы файлов, которые потенциально могут иметь большой размер. Таким образом, как только в процессе путешествия по Всемирной паутине вы наткнетесь на файл указанного в списке типа, автоматически загрузится FlashGet (даже если до этого он у вас и не был запущен), и задание на загрузку этого файла будет передано именно ему, что, конечно же, ускорит процесс получения требуемого файла из Интернета.

Рис. 6



В этом же диалоговом окне вы можете указать, какой браузер вы предпочитаете, и соответственно, щелчки в каком именно браузере FlashGet должен отслеживать. Рекомендуем вам оставить установлен-

ными флажки как для **Netscape Navigator**, так и для **Internet Explorer**. Проверьте также установку флажков **Подтверждать и Открывать URL в браузере при отказе**, благодаря которым для каждого отдельного случая вы сможете принять решение, следует ли задание по загрузке выбранного файла передавать FlashGet, или же вы хотите скопировать этот файл средствами браузера. Таким образом, при щелчке по гиперссылке, указывающей на любой файл типа, перечисленного в поле **Слежение за файлами с типом**, автоматически активизируется FlashGet, который сначала выведет на экран диалоговое окно добавления нового задания, и только в случае вашего согласия это задание будет передано на выполнение FlashGet. Если же вы выполните щелчок по кнопке **Отмена**, на экране появится диалоговое окно **Загрузка файла** (рис.6), и вы получаете возможность скачать этот файл стандартным способом.

С помощью флажка **Использовать клавишу ALT** можно указать, что FlashGet должен отслеживать только те щелчки по гиперссылкам, которые были выполнены при нажатой клавише **Alt**. В этом случае у вас появляется еще одна возможность разграничить полномочия браузера и FlashGet и без установки флажков **Подтверждать и Открывать URL в браузере при отказе**. Обычный щелчок мышью по гиперссылке приведет к загрузке

файла средствами браузера, а щелчок при нажатой клавише **Alt** — к передаче загрузки этого файла FlashGet. Т.к. мы уже установили флажки **Подтверждать и Открывать URL в браузере при отказе**, проследите за тем, чтобы флажок **Использовать клавишу ALT**

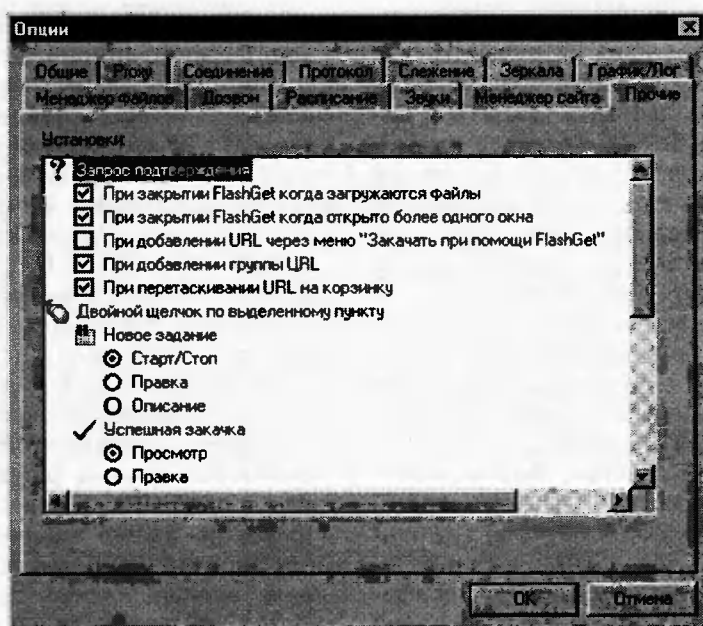
был снят. Вот теперь щелчком по кнопке **ОК** можно закрыть диалоговое окно **Опции** с сохранением выбранных параметров.

В качестве тестового примера давайте воспользуемся той же самой

Web-страницей, загруженной в браузере. Переключитесь в MS Internet Explorer, снова с помощью кнопки **Назад** вернитесь к предыдущей странице и на загруженной странице в самом конце документа найдите ссылку **Договор и приложения**. Установите курсор мыши на ссылку и просмотрите URL ссылки в строке состояния браузера. Ссылка указывает на архивный файл **unibel.zip**. А сейчас выполните по ссылке щелчок мышью. На экране появится диалоговое окно **Добавить новую загрузку**. Убедитесь, что имя выбранного файла находится в поле **URL**, и подтвердите добавление нового задания щелчком по кнопке **ОК**. Начните выполнение задания, воспользовавшись кнопкой  на панели инструментов. Дождитесь окончания копирования файла, перейдите в категорию **Принятые** и убедитесь в получении файла.

Вот вам еще один способ избежать ошибок при вводе адреса загружаемого файла вручную. Но оказывается, что и это еще не все. FlashGet предлагает еще один метод добавления новых заданий на загрузку, основанный на использовании так называемой Корзинки. Чтобы воспользоваться услугами Корзинки, необходимо вначале убедиться в ее наличии. Для этого откройте меню **Вид** и проверьте наличие флажка **Корзинка**. После этого поищите на экране вашего компьютера пиктограмму Корзинки . Теперь для добавления нового задания на загрузку вам достаточно просто перетащить с помощью мыши любую ссылку из окна браузера в Корзинку. Следует отметить, что FlashGet поддерживает также перетаскивание сразу нескольких ссылок. Чтобы застраховаться от случайностей, можно настроить FlashGet на выдачу запроса о подтверждении добавления нового задания при подобном перетаскивании ссылок в Корзинку. Для этого снова выполните команду **Сервис/Настройки...** или щелкните по кнопке  на панели инструментов, выберите вкладку **Прочие**, и в разделе **Запрос подтверждения установки флажков При перетаскивании URL на корзину** (рис.7). Кстати, здесь же с помощью флажка **При добавлении URL через меню "За-**

Рис. 7



качать при помощи FlashGet" определяется, будет ли появляться такое же диалоговое окно добавления нового задания при использовании в браузере указанной команды контекстного меню. Закройте диалоговое окно с помощью кнопки ОК. Для удобства сверните окно FlashGet щелчком по кнопке  строки заголовка (в прошлый раз мы уже говорили о том, что вместо закрытия программы это действие приводит к ее сворачиванию до пиктограммы на панели задач Windows). Обратите внимание, что Корзинка при этом на экране остается. А вот теперь переключитесь в MS Internet Explorer, найдите на загруженной странице ссылку **Порядок регистрации** и перетащите мышью эту ссылку на пиктограмму Корзинки. Снова на экране появится диалоговое окно добавления нового задания. Подтвердите добавление задания, выполните задание и убедитесь в получении требуемого файла. Не забудьте после этого завершить работу с Internet Explorer (Файл/Заккрыть).

Таким образом, мы познакомились со всеми возможными способами добавления новых заданий на загрузку файлов:

- ввод адреса загружаемого файла вручную;
- слежение за буфером обмена;
- использование расширенного контекстного меню браузера Internet Explorer;
- слежение за щелчками в браузере;

- перетаскивание ссылок из окна браузера в Корзинку.

Каждый из способов по-своему хорош, и надеемся, что все они вам пригодятся при работе в Интернете. На этом наше сегодняшнее занятие мы заканчиваем, и нам осталось выполнить

4. Завершение работы с FlashGet.

Напомним, что для этого необходимо выполнить команду **Файл/Выход** или воспользоваться кнопкой  на панели инструментов.

Основные операции, рассмотренные нами на этом занятии, приведены в таблице. В следующую нашу встречу мы поговорим о том, каким образом можно настроить FlashGet для копирования информации из сети по расписанию.

Действие	Реализация
Включение/выключение отображения на экране Корзинки	Вид/установить/снять флажок Корзинка
Настройка стандартных параметров выполнения новых заданий	Сервис/Параметры закладки по умолчанию... или 
Настройка перехвата содержимого буфера обмена	Сервис/Настройки... или  /вкладка Слежение/в разделе Буфер обмена установить флажок Следить
Настройка параметров связи FlashGet и Интернет-браузеров	Сервис/Настройки... или  /вкладка Слежение/в разделе За кликами в браузере установить флажки Netscape Navigator, Internet Explorer/установить флажки Подтверждать, Открывать URL в браузере при отказе/снять флажок Использовать клавишу ALT/в поле Слежение за файлами с типом проверить список отслеживаемых типов файлов
Настройка появления диалогового окна добавления нового задания при перетаскивании ссылок в Корзинку или при выборе в браузере команды контекстного меню "Закачать при помощи FlashGet"	Сервис/Настройки... или  /вкладка Прочие/в разделе Запрос подтверждения установить флажки При перетаскивании URL на корзинку, При добавлении URL через меню "Закачать при помощи FlashGet"
Добавление нового задания с использованием буфера обмена	1. В MS Internet Explorer — щелчок правой кнопкой мыши по ссылке/Копировать ярлык; 2. В FlashGet — Правка/Вставить URL или Задания/Новое задание ()
Добавление нового задания с использованием расширенного контекстного меню браузера MS Internet Explorer	В MS Internet Explorer: щелчок правой кнопкой мыши по ссылке/Закачать при помощи FlashGet или Закачать все при помощи FlashGet для загрузки всех ссылок в пределах страницы
Добавление нового задания с использованием Корзинки	Перетащить ссылку из окна браузера в Корзинку 
Добавление нескольких ссылок из фрагмента Web-страницы в список новых заданий	Выделить фрагмент Web-страницы в окне браузера/перетащить фрагмент из окна браузера в Корзинку 

(Продолжение следует)



ЛИСТАЯ СТРАНИЦЫ

Скрестить ежа и ужа (Linux и Windows) пытается компания Lindows.com. Что из этого получится, посмотрим. А в заметке "Lindows — даа в одном" (CHIP, №8/2002, С.7) рассказывается о планах компании.

Компания Lindows.com анонсировала выпуск своей ОС на ядре Linux, которая должна быть полностью совместима с ПО для Windows. Время выпуска полнофунк-

циональной Lindows пока не определено, но планируется на текущий год. Анонсирован выпуск Lindows OS SPX, которая базируется на ядре Linux и пытается достичь полной совместимости с Windows-программами при помощи технологии Wine. При этом Lindows не будет бесплатной, ее стоимость составит \$99. По одной лицензии она может устанавливаться на несколько ПК. Пока что разработчики не гарантируют полной совместимости со всеми Windows-приложениями. Зарегистрированные пользователи одновременно

заплатят непосредственно за Lindows OS SPX, а также за доступ к библиотеке программного обеспечения. Активизировать приложения можно посредством "онлайн-склада" Click-N-Run (www.lindows.com/clickn-run). Установка отдельных модулей программного обеспечения существенно экономит время на установку всей системы. Второе назначение "склада" — создание рынка сторонних производителей ПО, которым позволено публиковать и продавать свои разработки по технологии Click-N-Run.

Статья А. Кузнецова "Насколько хорош новый Celeron?" (ПЛ: компьютеры, №9/2002, С.76...81) посвящена результатам сравнительного тестирования нового процессора от Intel.

Вместе с новым Pentium 4 Celeron 1,7 ГГц тестировались процессоры Pentium 4 1,7 ГГц, Pentium 4 1,8 ГГц, Pentium 4 2,0 ГГц от Intel и Athlon XP 1700+, Athlon XP 1800+, Athlon XP 2100+ от AMD. Технические характеристики, заявленные производителем, приведены в табл. 1.

Табл. 1

Процессор	Модель	Реальная частота, МГц	Ядро	Технология, мкм	Объем кеша первого уровня, Кб	Объем кеша второго уровня, Кб	Частота системной шины, МГц	Набор протоколов расширений	Разъем	Ориентировочная цена, USD
Pentium 4 Celeron	1700	1700	Willamette	0,18	8+12000 инструкций	128	400	SSE2	Socket 478	83
Pentium 4	1700	1700	Willamette	0,18	8+12000 инструкций	256	400	SSE2	Socket 478	135
Pentium 4	1800	1800	Willamette	0,18	8+12000 инструкций	256	400	SSE2	Socket 478	155
Pentium 4	2000	2000	Northwood	0,13	8+12000 инструкций	512	400	SSE2	Socket 478	188
Athlon XP	1700+	1466	Palomino	0,18	64+64	256	266	3DNow! Professional	Socket A	80
Athlon XP	1800+	1533	Palomino	0,18	64+64	256	266	3DNow! Professional	Socket A	89
Athlon XP	2100+	1733	Palomino	0,18	64+64	256	266	3DNow! Professional	Socket A	180

Все процессоры Intel тестировались на материнской плате Albatron PX845, а продукты AMD — на платформе MSI KT3 Ultra. Остальные параметры:

- оперативная память — 512 Мб DDR SDRAM PC2100;
- графический адаптер — ASUS V7100 Pro 32 Мб;
- жесткий диск — IBM IC35L040 40 Гб ATA/100.

Испытания проходили под операционной системой Windows 98 SE с установленным DirectX 8.1. Для видеокарты был установлен референсный драйвер nVIDIA версии 29.42. Системная логика VIA работала с драйвером версии 4.39. Для чипсета Intel использовалась информационная утилита версии 4.00.1013 и программный акселератор версии 2.2.

Для тестирования использовались несколько программ. SiSoftware Sandra Professional 2002 служила для оценки нескольких синтетических показателей. Производились тесты CPU Arithmetic Benchmark, CPU Multi-Media Benchmark и Memory Bandwidth Benchmark. Первый из них определяет быстрдействие процессора в арифметических операциях — целочисленных и с плавающей точкой. Второй тест показывает производительность процессора при вы-

полнении мультимедийных инструкций — при целочисленных операциях и с плавающей точкой. Наконец, третье испытание проводилось для определения пропускной способности подсистемы памяти (ЦПУ — чипсет — память) отдельно для целочисленных операций и для операций с плавающей точкой. Еще одним синтетическим тестом стала программа CPU Benchmark 2002 beta3. Результат рейтингового показателя вычисляется на основе четырех отдельных испытаний. К ним

компрессии файлов JPEG, Zlib-компрессии и декомпрессии файлов, поиске текста, компрессии и декомпрессии файла формата Ogg Vorbis, близкого по структуре к MP3, расчете преобразований трехмерной векторной графики. С помощью этой же программы производилось также тестирование подсистемы памяти и определялась скорость преобразования файла MPEG1 в файлы ASF низкого и высокого разрешения (входит в видеотест). Следует добавить, что

относится оценка производительности арифметико-логического устройства, блока вычислений с плавающей точкой, подсистемы памяти и общей производительности системы. В меню программы можно сразу указать набор потоковых расширений, с которым предполагается тестировать процессоры.

Утилита CPU RightMark версии 1.01 (www.ixbt.com) служит для определения объективного быстродействия процессора в различных вычислительных задачах, как то: численное моделирование физических процессов и работа с трехмерной графикой. Тест состоит из двух блоков. Один из них производит расчет физической модели, второй — отображение (рендеринг) сцены. Утилита допускает расчет физической модели с использованием FPU или набора SSE2. Визуализация сцены происходит с помощью либо FPU, либо набора SSE. Таким образом, можно оптимально использовать сильные стороны того или иного процессора. Тестирование выполнялось при установках по умолчанию: экранное разрешение — 1024x768, глубина цвета — 32 бита.

Программа компании MadOnion.com PCMark2002 использовалась для определения индекса производительности процессоров. Итоговый показатель выводится в ней на основе результатов, показанных при де-

PCMark2002 поддерживает набор потоковых расширений SSE2.

Производительность процессоров при работе с офисными приложениями определялась с помощью теста Ziff Davis Media Business Winstone 2001. Необходимо оговориться, что итоговый результат основан на последовательном выполнении офисных программ и не учитывает режим многозадачности, который очень часто используется в реальной жизни.

Для оценки процессоров в работе с трехмерными графическими приложениями использовалась еще одна программа компании MadOnion.com — 3Dmark2001 SE build 330. С ее помощью по трем первым игровым сценам определялся индекс производительности в DirectX3D при следующих системных установках: цветовое разрешение — 16 бит; экранное разрешение — 640x480; антиалиасинг — выключен; кадровый буфер — двойной; формат текстур — сжатый; рендеринг — программный; Z-буфер — 16 бит.

Следующей программой, позволяющей определить практическую эффективность процессоров при работе с графикой, стала демоверсия Quake 3 Arena (demo001). В этой игре в режиме OpenGL определялось максимальное количество кадров в секунду при воспроизведении тестового ролика. Систем-



ные установки были такими: экранное разрешение — 640х480; цветовое разрешение — 16 бит; полноэкранный режим — включен; освещение — Lightmap; геометрические детали — высокий уровень; текстурные детали — минимальный уровень; текстурное качество — 16 бит; текстурная фильтрация — билинейная.

Третьей программой, тестирующей работу процессора при работе с графикой в OpenGL, стала игра "DroneZmark", позволяющая определить скорость в режиме теста, выраженную в кадрах в секунду. Ее выбрали потому, что она поддерживает набор потоковых расширений SSE2. Системные установки при испытаниях были следующие: экранное разрешение — 640х480; цветовое разрешение — 32 бита; Z-буфер — 24 бита; цветовое разрешение текстур — 32 бита; качество текстур — высокое.

Быстродействие процессоров в такой практической задаче как архивация файлов проверялось с помощью архиватора WinRAR версии 3.0. Был выбран уровень сжатия Best. Архивировался каталог общим объемом 149 Мб. Результат определялся по времени, затраченному на эту процедуру.

К результатам тестирования еще двумя программами надо подходить с известной долей скепсиса. Во-первых, это программа N-Bench версии 2.0. Разработана она для определения быстродействия процессоров AMD, что, в общем-то, объясняет ее направленность и внушает естественные опасения за объективность результатов при тестировании процессоров Intel. Но при этом с ее помощью определяется и 3D-быстродействие в графике. Данная программа поддерживает самые последние потоковые расширения 3DNow! Professional и SSE2.

Во-вторых, программа MultimediaMark 99 от Futuremark Corporation, последняя из выбранных для тестирования, уже достаточно стара. Она поддерживает только существовавшие на момент ее создания наборы инструкций MMX, KNI (SSE). В этой

программе определяются мультимедийные возможности системы на основе четырех тестов: обработки и воспроизведения файла MPEG1, обработки изображений и работы с аудио-эффектами.

Результаты тестирования приведены в табл. 2 и 3.

Подводя итоги, автор отмечает, что блин а лице Pentium 4 Celeron комом не вышл. Совершенно естественно, что этот процессор не превосходит по производительности полноценный Pentium 4. Этого по определению быть не может. Но отстает он в большинстве тестов ненамного. В тех задачах, где определяющим фактором становится объем кеша второго уровня, отставание немного больше, а в ситуациях, когда более существенно влияние набора потоковых расширений SSE2, разница в результатах становится меньше. Учитывая, что цена моделей Celeron примерно в полтора раза меньше стоимости процессоров Pentium 4, работающих на одинаковой с ними частоте, можно сделать интересный для пользователя вывод. По эффективности вложения денег процессоры Celeron выглядят явно более привлекательно, чем настоящие Pentium 4. Это не может не радовать ограниченных в средствах покупателей, не предпочитающих абсолютную производительность любой ценой. Единственное, что пока огорчает — наличие в продаже только двух моделей Pentium 4 Celeron. Но это дело времени.

Как видно из результатов тестирования, самым крупным и неоспоримым достоинством нового Celeron является поддержка набора потоковых расширений SSE2. Поэтому при использовании приложений, работающих с этим набором, Celeron обходит в ряде случаев по скорости даже процессор Athlon 2100+. Поскольку все больше разработчиков софта начинают поддерживать SSE2, для пользователя это может стать решающим фактором при выборе нового компьютера. Редакция журнала "ПЛ: компьютеры" без колебаний присуждает процес-

сору Intel Pentium 4 Celeron титул "Лучшая покупка". Очень вероятно, что в будущем ему на смену придет процессор Pentium 4 на ядре Willamette. Это произойдет в том случае, если корпорация Intel сможет перевести основную часть производства на 0,13-мкм технологический процесс, по которому производятся процессоры Pentium 4 на ядре Northwood с кешем второго уровня 512 Кб, и начать производство его очередной версии, выполненной по 0,09-мкм технологии (ядро Prescott). Объем кеша второго уровня у последнего составит уже 1 Мб. Привычная расстановка сил в этом случае сохранится.

И хотя статья была посвящена рассмотрению возможностей нового Pentium 4 Celeron, автор сказал несколько слов и о процессорах AMD. Они по-прежнему выглядят очень привлекательно по критерию цена/производительность. Процессоры Athlon XP, маркированные по рейтинговой системе и фактически имеющие более низкую частоту, весьма успешно конкурируют по производительности с более быстрыми по реальной частоте Pentium 4. В ряде задач они даже опережают конкурентов. Правда, в основном это происходит тогда, когда определяющим аспектом становится большой объем кеша первого уровня. В то же время, по результатам тестов видно отставание от Pentium 4, когда решающим моментом становится пропускная способность подсистемы памяти. Еще нагляднее отставание при использовании программ, которые могут работать с набором потоковых расширений SSE2. Ведь процессоры Athlon XP поддерживают только SSE. Это дает фору Pentium 4. В дальнейшем разрыв будет увеличиваться, пока не появятся новые модели процессоров AMD, в которых реализован набор SSE2. Правда, и компания Intel не собирается стоять на месте. А уже увидевшие свет новые Athlon на ядре Thoroughbred (0,13-мкм технология) могут помочь только в плане увеличения рабочих частот.

Табл. 2

Процессор	Модель	Sandra 2002						CPUBench2002	CPU RightMark		
		Arithmetic ALU	Arithmetic FPU	Multi-Media Integer	Multi-Media Float-Point	Memory Bandwidth Integer	Memory Bandwidth Float-Point		Math Solving	Rendering	CPU
Pentium 4 Celeron	1700	3205	2077	6175	6190	1674	1635	4244	195,7	28,4	3840
Pentium 4	1700	3174	2079	6721	8186	1997	1987	4321	196	29,5	4039
Pentium 4	1800	3369	2209	7121	8669	2043	1935	4602	207,4	31	4279
Pentium 4	2000	3739	2445	7906	9637	2002	1931	4924	230,6	32,7	4819
Athlon XP	1700+	4029	2025	8016	9298	1911	1742	4839	170,7	32,4	4327
Athlon XP	1800+	4234	2117	8381	9722	1924	1774	5064	177,9	33,8	4502
Athlon XP	2100+	4787	2394	9478	10991	1894	1779	5702	201,7	38,1	5026

Табл. 3

Процессор	Модель	PCMark2002			Business Winstone 2001	3DMark 2001 SE	Quake 3 Arena demo001	Drone Zmark	WinRAR (время в сек)	N-Bench 2.0		Multimedia-Mark 1999
		Memory	ASF Low	ASF High						CPU	3D	
Pentium 4 Celeron	1700	3997	82,6	18,1	42,2	2548	172,8	99,89	264	836	1116	2683
Pentium 4	1700	4105	83,4	18,1	48,2	2905	193,6	102,89	237	908	1143	2761
Pentium 4	1800	4022	92,7	19,5	49,4	3018	198	104,08	235	962	1169	2879
Pentium 4	2000	4540	104,8	21,8	56,9	3771	212,2	109,74	212	1176	1214	3462
Athlon XP	1700+	2898	67,5	14,5	56,7	3419	185,9	90,46	302	1533	1499	4055
Athlon XP	1800+	2953	68,7	15	57,4	3553	191,4	92,15	297	1609	1504	4143
Athlon XP	2100+	2986	76,8	16,6	60,1	3749	198,6	92,97	289	1846	1586	4626



Недорогой, стоимостью примерно 77 USD, сканер может стать хорошим дополнением практически к любому компьютеру. Об одном из таких устройств рассказывается в статье А.Алексеева "Plustek OpticPro S6 — шаг за шагом" (ПЛ: компьютеры, №9/2002, С.42).

Что нам, в принципе, надо от сканера, уточняет автор? Не так много. Чтобы текст распознавался с минимумом ошибок, а фотографии можно было организовать в альбом и отправить по электронной почте друзьям. Если такой сканер еще и стоит недорого, то это то, что нужно.

Процесс инсталляции сканера русифицирован, так что даже у неподготовленного пользователя не должно возникнуть вопросов. Причем, это первый сканер в тестовой лаборатории журнала "ПЛ: компьютеры", при установке которого проверяется его серийный номер. Если указать неверный, то воспользоваться сканером не удастся. Трудно назвать это достоинством, считает автор.

О новом чипсете фирмы Silicon Integrated System сообщается в заметке "Монополия Intel пошатнулась" (CHIP, №9/2002, С.12).

Чипсет SiSR658 предназначен для рабочих станций и высокопроизводительных настольных компьютеров, так как работает с быстрой (и дорогой) памятью RDRAM. Это первый в мире чипсет, поддерживающий двухканальную память RDRAM 800/1066 МГц, максимальный объем которой может достигать 4 Гб. Благодаря техно-

Сверхминиатюрный плеер под названием MuVo выпустила известная фирма Creative. Об этом сообщается в заметках "Многофункциональный MP3-проигрыватель" (CHIP, №9/2002, С.8) и "Creative MuVo: сверхкомпактный MP3-плеер" (Компьютер Пресс, №9/2002, С.129).

Это устройство представляет собой самый маленький в мире MP3-проигрыватель с размерами 73х35х16 мм и весом всего 28 г. Он позволяет сохранять и воспроизводить музыку двух форматов — MP3 и WMA. При этом

Еще одна компьютерная новинка — читающая ручка. О ней сообщается в заметке "Волшебное перо" (ПЛ: компьютеры, №9/2002, С.16).

На рынке периферийных устройств в скором времени, возможно, появится интересный продукт, обещающий в значительной мере облегчить жизнь владельцам карманных компьютеров, а также порадовать утомленных постоянной работой с клавиатурой пользователей настольных ПК и ноутбуков. Речь идет о небольшой, абсолютно обычной на первый взгляд чернильной ручке Vpen израильской компании OTM Technologies, способной распозна-

О новой беспроводной компьютерной мыши от Microsoft сообщается в заметке с красноречивым названием "Microsoft Wireless IntelliMouse Explorer 1.0 — если бы они так делали Windows" (ПЛ: компьютеры, №9/2002, С.46).

Технические характеристики сканера следующие:

- фотоэлемент — цветная CCD-матрица;
- площадь сканирования — 216х297 мм;
- оптическое разрешение — 600х1200 dpi;
- интерфейс — USB 1.1;
- глубина цвета — 48 бит;
- серая шкала — 16 бит;
- память буфера обмена — 512 Кб;
- габариты — 457х270х64 мм;
- масса — 2,2 кг.

Они вполне традиционны для данного класса устройств. Однако в нем используется технология Pixel by Pixel, характерная для куда более дорогих изделий. При этом цвет обрабатывается сразу для конкретного пиксела, что исключает неточность при назначении цветов.

Для начала отсканировали в Adobe Photoshop 7.0 цветной тестовый образец. При размещении оригинала на стекле отсчет координат идет от верхнего правого угла, что

гии MuTIOL, чипсет может обеспечить скорость передачи данных между северным и южным мостами 1 Гб/с. Он поддерживает звук в формате 5.1, отвечающий спецификации AC'97, 10/100 Мбит Ethernet, до шести слотов PCI и столько же портов USB 2.0/1.1. Кроме того, имеется двухканальный интерфейс Ultra ATA 133/100/66. Введена поддержка нового интерфейса для видеокарт AGP 8х.

SiSR658 предназначен для процессоров Intel Pentium 4 с частотой до 3 ГГц,

объем флэш-памяти MuVo составляет 128 Мб, что позволяет воспроизводить до четырех часов высококачественной музыки в формате WMA или до двух часов в формате MP3 с битрейтом 128 Кбит/с. MuVo питается от одной батарейки типа AAA, которая обеспечивает непрерывную работу в течение 12 часов. Его можно использовать как обычное устройство для переноса и хранения данных, что позволяет сохранить информации в 88 раз больше, чем на одной дискете.

Плеер состоит из двух частей — модуля

вать рукописный текст. Главное ее преимущество перед другими подобными устройствами — она не требует для письма специальной бумаги.

Это устройство объединило в себе усилия сразу нескольких компаний. Так, процессом перевода замысловатых движений руки в цифровые символы будет заниматься зашито в ручку программное обеспечение SimpliWrite, разработанное фирмой ART, а дальнейшая передача информации на ближайший компьютер будет осуществляться с помощью беспроводной технологии передачи данных Bluetooth. Кстати, сама Vpen полностью совместима со все-

оптические мыши от программного гиганта уже давно пользуются популярностью, несмотря на высокую цену. Новая модель — первая беспроводная мышь из линейки Microsoft IntelliMouse. По сравнению с другими устройствами этой серии, в нее вне-

не совсем удобно, т. к. приходится тянуться через весь стол. Сканирование начинается с предварительного просмотра. Через 10 с на экране получено изображение. При разрешении 600 dpi и режиме "Цветное фото" (24 бит), был получен файл размером 98,4 Мб. Всего через 125 с он уже был доступен для редактирования. Цвет передается достаточно равномерно, картинка — резкая и без видимых дефектов. Повторное сканирование с цветностью 48 бит не дало существенной прибавки в качестве. Зато подавление раstra средствами драйвера срабатывает просто отлично.

Второе испытание — это распознавание мелкого текста. При этом программные аппроксимации ABBYY Finereader 4.0 работали минимально. Через 30 с был получен практически идеально распознанный текст. С помощью кнопок Plustek OpticPro S6 можно отправить картинку по E-mail, скопировать прямо на принтер, перейти в спящий режим. Еще одну кнопку можно назначить по своему усмотрению.

работающих при частоте системной шины 533/400 МГц. Пока материнских плат на новом чипсете еще никто не представил, но уже в скором времени ожидается их появление. Судя по предварительным тестам, чипсет от SiS будет, как обычно, немножко уступать в производительности аналогичным решениям Intel. При этом платы на нем, как и в случае с другими чипсетами SiS, будут дешевле. Так что перед потребителями опять встанет выбор между ценой и производительностью.

памяти и батарейного отсека. Модуль памяти вставляется непосредственно в USB-порт. Компьютер опознает этот модуль как внешний диск, после чего пользователь может перемещать туда любые файлы обычным методом, например, drag-and-drop. Чтобы прослушать музыку, необходимо соединить модуль памяти и батарейный отсек и подключить наушники.

Ориентировочная цена плеера составит 170 USD за 128 Мб-версию и 130 USD — за 64 Мб-вариант.

ми существующими Bluetooth-устройствами и не потребует для работы особой настройки компьютера. Кроме того, Vpen можно использовать и как обычную компьютерную мышь.

Интересно, что сама компания OTM Technologies не собирается заниматься производством Vpen. Эта непосильная для небольшой израильской компании ноша ляжет на плечи многочисленных OEM-производителей, от которых, во многом зависит, станет ли новое устройство в будущем таким же популярным, как мышь или джойстик сегодня. Разработчики надеются, что конечная рыночная цена Vpen не превысит 80 USD.

сен ряд полезных изменений.

По традиции мышь идеально лежит в руке, а прибавление в весе (2 батарейки AA) сделало ее еще более устойчивой. Изменения коснулись боковых кнопок — сначала необходимо привыкнуть к их расположению,



зато случайные нажатия исключены полностью. Единственным недостатком механической части является чересчур жесткое вращение колесика. При прокрутке вверх то и дело срабатывает средняя кнопка.

В эксплуатации мышь показала себя с лучшей стороны. 6000 считываний датчика в секунду (вдвое больше, чем у обычной оптической мыши) позволяют играть

Очередная новинка от Philips — беспроводной монитор. О нем сообщается в заметке "Концептуальный монитор" (CHIP, №9/2002, С.17).

Основной особенностью этого устройства является использование беспроводной технологии Wi-Fi 802.11b, благодаря которой этот монитор может работать на

Новая клавиатура компании BTC, названная Smart Office ("BTC 8190 — "умная" клавиатура для офиса", ПЛ: компьютеры, №9/2002, С.45), позволит значительно повысить производительность труда многих сотрудников фирм.

Клавиатура имеет нестандартный внешний вид — двухцветный серо-белый корпус со скругленными краями и увеличенное пространство под нижним рядом клавиш, позволяющее не напрягать запястья рук в течение всего рабочего дня. В верхней части корпуса находятся клавиши, обеспечивающие прямой вызов "народных" программ компании Microsoft: Word, Excel, Power Point и

Очередному шагу в миниатюризации карт флэш-памяти посвящена статья С.Асмакова "xD-Picture: новый формат флэш-памяти" (КомпьютерПресс, №9/2002, С.188).

xD-Picture за последние два года стал уже третьим форматом, завоевавшим звание "самой миниатюрной карты флэш-памяти". Размер носителей составляет 20x25x1,7 мм, а масса — 2 г. Предыдущими обладателями этого почетного звания были Memory Stick Duo (20x31x1,6 мм), представленные компанией Sony осенью прошлого года, а до них — Multimedia Card (24x32x1,4 мм), появившиеся на свет в ноябре 1997 г. благодаря совместным усилиям фирм SanDisk и Siemens.

Как следует из совместного заявления Olympus и FujiFilm, 30 июля этого года объявивший о выпуске карт нового формата, новый носитель должен прийти на смену морально устаревшим картам SmartMedia. По официальной версии, появление xD-Picture вызвано сразу несколькими объективными причинами.

Во-первых, это тенденция к уменьшению размеров цифровых фотокамер. Во-вторых, потребность в увеличении емкости сменных носителей — в силу технологических причин максимальная емкость выпускаемых сегодня карт SmartMedia составляет "всего" 128 Мб. Первая линейка xD-Picture, которая появится в продаже уже осенью текущего года, будет включать носители емкостью 16, 32, 64 и 128 Мб (их производством будет заниматься компания Toshiba). К концу нынешнего года должна появиться 256-мегабайтная версия xD-Picture, а в течение следующего — 512-мегабайтная и, возможно, гигабайтная.

даже в современные 3D-шутеры, не испытывая каких-либо заметных затруднений, а диапазон уверенной работы превысил заявленный. Оригинальные технологии экономии заряда батарей должны продлить им жизнь до 8 месяцев.

Программа IntelliPoint 4.0, поставляемая в комплекте, не только предупреждает о снижении заряда, но и назначает кнопкам

удалении до 100 м от базового ПК. Detachable Monitor имеет сенсорный экран с диагональю 15 дюймов и весит около 2,2 кг. При этом его толщина составляет всего 25 мм. Установленные аккумуляторы обеспечивают около 3 часов непрерывной работы. Кроме того, монитор может комплектоваться устройством

"Календарь" из Outlook. Специальная кнопка рядом позволяет использовать 12 функциональных клавиш в расширенном режиме. В этом случае с их помощью можно напрямую работать с наиболее популярными функциями из экранного меню (такими как Help, New, Open, Save, Replace и т. д.).

Еще одна группа из восьми кнопок отвечает за навигацию и работу в Интернете, а также за вызов электронной почты. В правой верхней части клавиатуры находятся еще три группы функциональных клавиш. Первая группа открывает опцию Мой компьютер, запускает калькулятор или диалоговое окно Log Off. Следующая группа из семи

Если же говорить о более далеких планах, то в перспективе предусмотрено увеличение емкости этого носителя до 8 Гб. Столь значительный рост емкости миниатюрного носителя стал возможен благодаря использованию многослойной технологии.

Обнародованы также основные технические характеристики карт xD-Picture. Так, максимальная скорость чтения данных составляет 5 Мб/с, а скорость записи — 3 Мб/с (для карт емкостью 16 и 32 Мб — 1,3 Мб/с). Напряжение питания — 3,3 В; потребляемая при работе мощность — 25 мВт. Как и SmartMedia, карты xD-Picture содержат только флэш-память и не оснащаются встроенным контроллером (в отличие, например, от CompactFlash).

В условиях довольно жесткой конкуренции, существующей сегодня на рынке сменных карт флэш-памяти, весьма остро встает проблема обеспечения максимальной совместимости нового носителя с уже имеющимися у пользователей оборудованием, рассчитанным на другие форматы. Поэтому одновременно с самими картами xD-Picture будет начат выпуск адаптеров-переходников для форм-факторов PC Card и CompactFlash, а также внешних карт ридеров с интерфейсом USB.

В перспективе использование карт xD-Picture не ограничится только цифровыми фото- и видеокамерами. Планируется внедрять новые носители в карманные компьютеры, портативные цифровые плееры, мобильные телефоны и другие подобные устройства.

Однако, несмотря на радужные прогнозы FujiFilm и Olympus, появление xD-Picture

действия в зависимости от приложений. Вообще, по богатству функций и удобству их использования — это один из лучших мышиных драйверов.

Со времени выпуска цена мыши упала с 75 до 45 USD, поэтому сейчас называть ее высокой для манипулятора с подобными характеристиками уже не совсем корректно.

для зарядки аккумуляторов и клавиатурой для традиционной работы с компьютером.

Монитор способен работать автономно, как самостоятельное устройство, которое можно использовать для написания заметок или проверки электронной почты.

клавиш позволяет осуществлять контроль за мультимедийными функциями. С ее помощью производится регулировка звука, управление прослушиванием записей. А двумя кнопками а крайнем верхнем углу клавиатуры можно выключать компьютер или переводить его в "спящий режим".

Заметные отличия в функциональных возможностях данной модели по сравнению со стандартными образцами наблюдаются и в нижней части корпуса. Там находится координатное устройство в виде маленького джойстика в кольце, а под ним — четыре клавиши. С помощью этого функционального блока можно работать с Clipboard.

влечет за собой и негативные последствия. Например, довольно спорным представляется тезис об "обеспечении максимальной совместимости цифровых камер различных производителей". Скорее, совсем наоборот — ведь появление еще одного стандарта карт флэш-памяти только добавит головной боли пользователям и сотрудникам цифровых фотолабораторий (потребуется срочно запастись соответствующими адаптерами), а также производителям карт-ридеров (им придется в спешном порядке обновлять линейки своих изделий).

Не вселяют оптимизма и предварительные цены на карты xD-Picture: 32-мегабайтный носитель будет стоить 30 USD, 64-мегабайтный — 50 USD, а 128-мегабайтный — 90 USD. Причем это цены для западноевропейского рынка, а следовательно, в отечественных магазинах они будут на 20...30% выше.

Также пока непонятно, удастся ли влиять FujiFilm и Olympus заинтересовать своей новинкой других производителей портативной цифровой техники. А ведь именно от этого в значительной степени будет зависеть популярность как самого формата xD-Picture, так и изделий FujiFilm и Olympus, в которых он используется. Так, например, если продвигающей носители Secure Digital организации SD Association удалось пару лет назад, еще на стадии разработки стандарта, заручиться поддержкой более двухсот производителей, то экзотические карты Memory Stick (кстати, появившиеся еще в 1998 г.) так и остались эксклюзивной принадлежностью устройств, выпускаемых компанией Sony, и это не лучшим образом отразилось на их популярности.



Решение задач методом дихотомии

Введение

Дихотомия, дихотомический поиск, двоичный поиск — это способ нахождения значения величины, которая является ответом задачи, или, зная которую, можно вычислить ответ задачи. При этом сам поиск ведется в некотором смысле "подбором".

Вначале из условий задачи мы определяем диапазон, внутри которого заведомо содержится искомое значение [Left, Right]. Т.е. мы точно знаем, что искомое значение больше либо равно Left и меньше либо равно Right. На текущем шаге мы в качестве "проверяемого" значения выбираем Middle, равное среднему из Left и Right:

$Middle := (Left + Right) / 2.$

Далее условиями задачи определяется целевая функция Target. Мы вычисляем ее значение на аргументе Middle и сравниваем его со специальным значением Required ("должно быть").

В зависимости от результатов сравнения мы принимаем решение, какую из границ (Left или Right) заменить значением Middle, и продолжаем процесс, пока не найдем значение Middle с нужной точностью Diff. Например, так:

```
procedure Dichotomy;  
begin  
  while (abs(Right-Left)>Diff) do  
  begin  
    Middle:=(Left+Right)/2;  
    if Target(Middle)<Required  
    then Left := Middle  
    else Right := Middle;  
  end;  
end;
```

Далее в статье приводятся решения методом дихотомии нескольких задач, предлагавшихся на олимпиадах для школьников (Белорусская республиканская, Гомельская областная по информатике) и студентов (Северный и Центральный четвертьфиналы и Центрально-Европейский полуфинал командного ACM первенства мира по программированию среди вузов).

По традиции, предлагается в качестве наиболее эффективного следующий способ изучения материала.

После прочтения условия задачи нужно попытаться решить ее самостоятельно. Написав решение, отослать его на проверку на сайт <http://dl.gsu.unibel.by>. Проверка осуществляется автоматически, и обычно вы получаете ответ о результатах проверки в течение нескольких минут. В случае не прохождения вашим решением какого-либо из тестов, вы имеете возможность получить сам тест (входные данные и правильные результаты), затем найти и исправить ошибку в решении и снова отослать его на тестирование.

Если вам удалось решить задачу самостоятельно, можно пропускать (или бегло просмотреть) соответствующий раздел статьи, переходя сразу к следующей задаче.

Если же самостоятельно решить задачу не удастся, нужно прочитать идею решения, предложенную автором, и вновь попытаться проделать остальную работу самостоятельно описанным выше способом. Если же идеи тоже недостаточно, тогда нужно внимательно ознакомиться с текстом предложенного решения и подробными пояснениями.

И так для каждой задачи!

В случае проблем с доступом в Интернет, вы можете использовать электронную почту для отсылки решений и получения результатов — просто напишите пустое письмо на адрес dl-service@gsu.unibel.by, указав RUS в поле "ТЕМА", и вы получите ответ на русском языке от почтового робота о том, как им пользоваться.

1. Задача "Арбузы"

(Белорусская республиканская олимпиада по информатике, 1994 г.)

В ряд лежат N арбузов, пронумерованных от 1 до N. Нам известно, что:

- массы первого и N-го арбузов — $m(1)$ и $m(N)$ соответственно;

- масса i-го арбуза $m(i)$ есть среднее арифметическое масс двух соседних арбузов, увеличенное на d:

$$m(i) = d + (m(i-1) + m(i+1)) / 2.$$

По введенным $m(1)$, $m(N)$, d и j найти $m(j)$. Ограничение — $N < 200$.

Если найти $m(j)$ по введенным данным невозможно, вывести фразу: "данные некорректны".

Формат ввода: **Формат вывода:**

```
M(1)                      M(j)  
M(N)  
N  
D  
J
```

Пример 1

Ввод: **Вывод:**

```
1.0                      85.0  
1.0  
21  
1.0  
15
```

Пример 2

Ввод: **Вывод:**

```
30.0                      данные некорректны  
210.0  
10  
-5.0  
8
```

Эта задача является характерным примером применения дихотомического поиска — мы будем подбирать вес второго арбуза так, чтобы расчеты веса N-го арбуза по заданной в условиях формуле давали значение веса N-го арбуза, заданного в исходных данных.

По ходу расчетов мы последовательно будем вычислять веса всех арбузов от 3-го до N-го, и поэтому после завершения подбора методом дихотомии мы можем вывести вес любого из арбузов.



Прежде всего рассмотрим тело программы:

```
begin
  InputData;
  Dichotomy;
  OutputData;
end.
```

Оно выглядит предельно лаконично, в нем вызываются три процедуры:

- InputData — для ввода и инициализации исходных данных;
- Dichotomy — для нахождения решения методом дихотомии;
- OutputData — для вывода результатов.

Начнем с классической процедуры Dichotomy, осуществляющей дихотомию на множестве вещественных чисел:

```
procedure Dichotomy;
begin
  while (abs(Right-Left)>Diff) do
  begin
    Middle:=(Left+Right)/2;
    if Target(Middle)<Required
    then Left := Middle
    else Right := Middle;
  end;
end;
```

Если левая и правая границы поиска отличаются более чем на Diff (где Diff — задаваемая нами точность вычислений, например, 0.000001), то вычисляется Middle — середина отрезка [Left, Right].

Далее переопределяется значение левой или правой границы поиска в зависимости от результата сравнения значения целевой функции Target с аргументом Middle и с Required (значение, которому функция Target должна быть по условию задачи равна при аргументе со значением Middle).

Сущность задачи отражает именно алгоритм вычисления функции Target:

```
function Target(Middle:double): double;
begin
  m[2]:=Middle;
  for i:=3 to N do
    m[i]:=2*m[i-1]-m[i-2]-2*D;
  Target:=M[N];
end;
```

По условию задачи сказано, что веса арбузов связаны соотношением

$$m(i)=d+(m(i-1)+m(i+1))/2,$$

которое описывает, как вычислять вес арбуза, если известны веса его левого и правого соседей.

Для вычислений весов арбузов по двум заданным весам рядом лежащих арбузов, гораздо более удобна формула, легко получающаяся из предыдущей:

$$m(i+1):=2*m(i)-m(i-1)-2*d;$$

или, если вычислять вес i -го арбуза, а не $(i+1)$ -го, то

$$m(i):=2*m(i-1)-m(i-2)-2*d;$$

Вес первого арбуза нам известен, а вес второго арбуза мы подбираем так, чтобы получить правильное значение для N -го арбуза, вес которого нам тоже задан.

Теперь перейдем к рассмотрению процедуры ввода:

```
procedure InputData;
begin
  assign(input,'input.txt'); reset(input);
  readln(m[1]);
  readln(mn);
  readln(N);
  readln(D);
  Readln(j);
  close(input);
  Left := 0;
  Right := 1000;
  Required := mn;
end;
```

Она вводит исходные данные, а также устанавливает начальные значения левой и правой границ дихотомического поиска (Left и Right) и величину, которую мы будем использовать как эталонную для целевой функции (Required).

И, наконец, процедура вывода:

```
procedure OutputData;
begin
  assign(output,'output.txt'); rewrite(output);
  i:=1;
  while (i<=N) and (M[i]>0.001) do inc(i);
  if i>N
  then writeln(M[j]:0:3)
  else writeln('данные некорректны');
  close(output);
end;
```

Вообще, от нас требуется вывод с точностью до 0,001 веса арбуза с номером j , заданным в исходных данных.

Однако в результате расчетов вес арбуза может оказаться нереальным (т.е. меньше нуля или пренебрежимо малым — почти равным нулю), в этом случае, по задумке авторов задачи, требуется выводить ответ "данные некорректны".

Полный текст решения:

```
{N+,E+}
program by94dlt2;
const
  MaxN = 200;
  Diff = 0.000001;
var
  m      : array [1..MaxN] of double;
  N,j,i  : integer;
  D,mn   : double;

  Right, Left, Middle, Required : double;

function Target(Middle:double): double;
begin
  m[2]:=Middle;
  for i:=3 to N do
    m[i]:=2*m[i-1]-m[i-2]-2*D;
  Target:=M[N];
end;

procedure Dichotomy;
begin
  while (abs(Right-Left)>Diff) do
  begin
    Middle:=(Left+Right)/2;
    if Target(Middle)<Required
    then Left := Middle
    else Right := Middle;
  end;
end;

procedure OutputData;
```



```

begin
  assign(output, 'output.txt'); rewrite(output);
  i:=1;
  while (i<=N) and (M[i]>0.001) do inc(i);
  if i>N
    then writeln(M[j]:0:3)
    else writeln('данные некорректны');
  close(output);
end;

procedure InputData;
begin
  assign(input, 'input.txt'); reset(input);
  readln(m[1]);
  readln(mn);
  readln(N);
  readln(D);
  Readln(j);
  close(input);
  Left := 0;
  Right := 1000;
  Required := mn;
end;

begin
  InputData;
  Dichotomy;
  OutputData;
end.

```

2. Задача "Голодание"

(Гомельская областная олимпиада по информатике, 1997 г.)

Курс лечебного голодания длился 21 день. В результате ежедневного взвешивания пациента были получены результаты — $m(1)$, ..., $m(21)$. Оказалось, что изменение веса между i -м и $i+1$ -м взвешиваниями ($i=1, \dots, 20$), прямо пропорционально весу в i -й день — $m(i)$.

Вводятся веса $m(1)$ и $m(21)$. Необходимо найти $m(2)$ — вес человека во 2-й день. Ответ вывести с точностью 4 цифры после запятой.

Пример ввода **Пример вывода**

```

70                70.0987
72

```

Обозначим K — коэффициент увеличения веса в день. Тогда получаем:

$$\begin{aligned}
 m(2) &= K * m(1) \\
 m(3) &= K * m(2) = K * K * m(1) = (K^2) * m(1) \\
 m(4) &= K * m(3) = \dots = (K^3) * m(1) \\
 &\dots \\
 m(21) &= K * m(20) = \dots = (K^{20}) * m(1) \\
 \text{Отсюда } K^{20} &= m(21) / m(1) \text{ и} \\
 K &= \sqrt[20]{m(21) / m(1)}
 \end{aligned}$$

Таким образом, чтобы ответить на вопрос задачи, нам нужно вычислить корень 20-й степени из значения $m(21)/m(1)$.

Поскольку в Паскале нет такой функции (например, для извлечения корня второй степени есть — SQRT), мы должны самостоятельно реализовать соответствующие вычисления. Хороший способ сделать это — применить дихотомию для подбора K .

Рассмотрим подробнее реализацию решения. Начнем с главной программы:

```

begin
  InputData;
  Dichotomy;

```

```

  OutputData;
end.

```

Это стандартный образец. Такой же стандартной является и процедура Dichotomy:

```

procedure Dichotomy;
begin
  while (abs(Right-Left)>Diff) do
    begin
      Middle:=(Left+Right)/2;
      if Target(Middle)<Required
        then Left := Middle
        else Right := Middle;
    end;
  end;
end;

```

Как уже говорилось ранее, специфика решения задачи заключается в функции Target, вызываемой из процедуры Dichotomy:

```

function Target(Middle:double): double;
var Res : Double;
begin
  Res:=m1;
  for i:=1 to 20 do Res:=Res*Middle;
  Target:=Res;
end;

```

В данном случае функция Target в соответствии с условиями задачи вычисляет 20-ю степень числа Middle — в результате выполнения операции

$\text{Target} = \text{Middle}^{20}$;

Рассмотрим теперь процедуру InputData:

```

procedure InputData;
begin
  assign(input, 'input.txt'); reset(input);
  readln(m1);
  readln(m21);
  close(input);
  if m21/m1 < 1
    then
      begin
        Left := 0.0;
        Right := 1.0;
      end
    else
      begin
        Left := 1.0;
        Right := m21/m1;
        Middle := 1.0;
      end;
  Required := m21;
end;

```

В ней, кроме ввода исходных данных $m1$ и $m21$, необходимо присвоить начальные значения величинам Left, Right и Required, используемым в процедуре Dichotomy.

Известно, что корень любой степени из числа меньше 1 — число меньшее 1, и потому для случая $(m21/m1)<1$ устанавливаются границы Left и Right, равные 0 и 1 соответственно.

Аналогично, в случае $(m21/m1)>1$ устанавливаются границы Left и Right равными 1 и $m21/m1$ соответственно.

Кроме того, для случая $(m21/m1)=1$ устанавливаем заведомо известный ответ $\text{Middle}=1.0$, поскольку в процедуре Dichotomy значение Middle не будет вычисляться



вследствие равенства Left и Right.

И наконец, вывод результата:

```
procedure OutputData;
begin
  assign(output, 'output.txt'); rewrite(output);
  writeln(m1*Middle:0:4);
  close(output);
end;
```

По условию задачи требуется вывести m2 с точностью до 4-х знаков после запятой. Понятно, что в результате дихотомии мы нашли $K=Middle$, поэтому и выводится $m2=m1*Middle$.

Полный текст решения:

```
{S,N+,E+}
program go97dlt4;
const
  Diff = 0.000000001;
var
  m1,m21 : double;
  i       : integer;

  Right, Left, Middle, Required : double;

function Target(Middle:double): double;
var Res : Double;
begin
  Res:=m1;
  for i:=1 to 20 do Res:=Res*Middle;
  Target:=Res;
end;

procedure Dichotomy;
begin
  while (abs(Right-Left)>Diff) do
  begin
    Middle:=(Left+Right)/2;
```

```
    if Target(Middle)<Required
    then Left := Middle
    else Right := Middle;
  end;
end;

procedure OutputData;
begin
  assign(output, 'output.txt'); rewrite(output);
  writeln(m1*Middle:0:4);
  close(output);
end;

procedure InputData;
begin
  assign(input, 'input.txt'); reset(input);
  readln(m1);
  readln(m21);
  close(input);
  if m21/m1 < 1
  then
    begin
      Left := 0.0;
      Right := 1.0;
    end
  else
    begin
      Left := 1.0;
      Right := m21/m1;
      Middle := 1.0;
    end;
  Required := m21;
end;

begin
  InputData;
  Dichotomy;
  OutputData;
end.
```

(Продолжение следует)

НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ

(Продолжение. Начало в №№9-12/2001, №№1-10, 12/2002, №1/2003)

Программирование под WIN32

10. Библиотека

#1. Программы имеют свойство расти. Через некоторое время количество строчек в листинге становится настолько велико, что хранить их в одном файле становится нецелесообразно.

Здесь мы сталкиваемся с такой диалектической категорией как **рост — развитие**. Желаящие поупражняться в нахождении наиболее общих закономерностей бытия, вот вам тема для медитирования: что есть разница в размерах операционных систем Windows 2000 и Windows XP — **рост** либо **развитие**?

Недавно в конференции RTFM_Helpers прозвучал такой вопрос: "Подскажите, можно ли как-то вынести про-

цедуру в один файл? А то главный *asm*-файл здоровый получается, тяжело лазить по нему". Как совершенно верно ответил *vk!m*, существует два основных способа сделать это — один "кривой", а другой правильный.

"Кривой" способ заключается в создании отдельного *asm*-файла, записывании туда процедуры и включения его в главный файл директивой *include*.

Правильный способ выглядит следующим образом — создаем отдельный *asm*-файл с процедурой, отдельный *inc*-файл с прототипом, создаем объектник и линкуем с главным объектным файлом. Если процедура универсальная, то можно создать свою библиотеку (*library*) и добавить ее туда. Потом можно подключать библиотеку к проекту опциями линкера или директивой *includelib*.

Вот о последнем мы сегодня и поговорим, то есть о создании собственной библиотеки.

Здесь, однако, стоит заметить, что *ml.exe* обладает одним досадным глюком. Проявляется он при ассемблировании сразу нескольких файлов и заключается в том, что имена каталогов, содержащие пробелы, воспринимаются некорректно. Существующий патч от *promethee* эту ошибку исправляет только для каталогов с именем до 8 символов. Имейте это в виду, и да не наступите вы на грабли!

Для начала "разобьем" все наши функции таким образом, чтобы каждая из них находилась в отдельном *asm*-файле.

Serrgio /HI-TECH,

E-mail:serrgio@gorki.unibel.by
http://wasrm.ru/



Функция вычисления простейшего хеша — суммы
ASCII-кодов символов:

; Файл SimpleHash.asm

.386

.model flat, stdcall
option casemap :none

.code

```
;-|SimpleHash|-----
;Функция : вычисление суммы ASCII-кодов символов
;Параметры : указатель на строку и ее длина
;Результат : в eax - сумма
;Использует : -
```

```
SimpleHash proc uses EBX ECX EDX pFrom:DWORD,
nIterations:DWORD
    xor EAX,EAX                ;результат
    xor EDX,EDX                ;временная переменная
    mov ECX,[nIterations]
    mov EBX,[pFrom]
@@Lab1:mov DL,[EBX]
    add EAX, EDX
    inc EBX
    loop @@Lab1
    ret
SimpleHash endp
```

end

Функция конвертации строки двойного слова в строку
из восьми 16-ричных символов:

;файл dword2hexstr.asm

.386

.model flat, stdcall
option casemap :none

.code

```
-----
;Функция : конвертация 32-битного числа в строку из 8-ми
;          16-ричных ASCII знаков
;Параметры : 32-битное значение и указатель на буфер
результата
;Результат : строка в буфере, в eax указатель на буфер
;Использует : -
;-----
```

```
dword2hexstr proc public uses ECX EBX nValue:DWORD,
pBuffer:DWORD
```

```
    mov ebx,[pBuffer]
    mov cl,32-4
@@Loop:mov eax,[nValue]
    shr eax,CL
    and al,0Fh ;тетраду в
    cmp al,10 ;hex ASCII знак
    sbb al,69h
    das
    mov [ebx],AL
    inc ebx
    sub cl,4
    jae @@Loop
    mov eax,[pBuffer]
    ret
```

dword2hexstr endp

end

Функция, обратная dword2hexstr — конвертация ASCII
hex-строки в число:

; Файл hexstr2dword.asm

.386

.model flat, stdcall

option casemap :none

include dzofasm.inc
includelib dzofasm.lib

.code

```
;-| hexstr2dword |-----
; Функция : конвертация ASCII hex-строки в число
; Параметры : ASCII hex-строка
;            : длина строки
;            : адрес переменной для указателя на необработанную
;            : часть строки
;            : адрес переменной для счетчика оставшихся знаков
; Результат : число
;            : [указатель на необработанную часть строки]
;            : [счетчик оставшихся знаков]
; Использует : ch2digit
; Примечание : сканирование останавливается на первой "нецифре"
;            : или цифре, вызывающей переполнение результата.
```

```
hexstr2dword proc uses esi ecx ebx
hexstr:dword, strlen:dword,
strafter:dword, remained:dword
```

```
;-|----- подготовка параметров и результатов
```

```
    mov esi,[hexstr] ; указатель на ASCII-строку
    mov ecx,[strlen] ; длина строки
    xor eax,eax ; результат
```

```
    jcxz @@xs2dret ; нулевая длина -> ничего не делать
    mov bh,0Fh ; последняя hex-цифра (radix-1)
```

```
;-|----- основной цикл
```

```
@@xs2dloop:
    xchg ebx,eax
    mov al,[esi] ; преобразовать знак
    call ch2digit ; ...в значение
    xchg eax,ebx
    cmp bl,bh ; "цифра"?
    ja @@xs2dret ; нет -> выход
    cmp eax,100000000h ; если старший "разряд" не ноль
    jae @@xs2dret ; ... (будет переполнение) -> выход
    shl eax,4 ; "сдвинуть" "цифру"
    add al,bl ; ...в младший разряд
    inc esi ; "принять" знак
    loop @@xs2dloop
```

```
;-|----- возврат результата
```

```
@@xs2dret:
    mov ebx,[strafter] ; адрес переменной для...
    mov [ebx],esi ; указателя на необработанную
; часть строки
    mov ebx,[remained] ; адрес переменной для...
    mov [ebx],ecx ; счетчика оставшихся знаков
    ret
hexstr2dword endp
```

end

Функция конвертации ASCII-цифры в число (для "внутреннего использования" функцией hexstr2dword):

;файл ch2digit.asm

.386

.model flat, stdcall
option casemap :none

.code

```
;-|ch2digit|-----
;Функция : конвертация ASCII-цифры любого radix'a в число
;Параметры : AL - hex-символ
;Результат : AL - число
;Использует : -
```



;Примечание : если на входе AL содержит слишком большую цифру,
; то значение на выходе будет больше, чем radix ;)
;-----

```
ch2digit proc public
    cmp     al,'9'
    jbe     done
    dec     ax                ; OPTIMIZE: instead DEC AL
    and     al,not 20h        ; upper case
    sub     al,'A'-'0'-11
done:      sub     al,'0'
    ret
ch2digit endp
```

end

Обратите внимание на то, что в конце этих листингов стоит директива *end* без указания точки входа. Помните, что главная процедура в программе только одна!

И, кстати говоря, "end startname" в модуле со стартовой меткой можно смело заменить на "просто end", но тогда придется указывать точку входа с помощью опции линкера /ENTRY:startname.

Теперь вызываем *ml.exe* следующей командной строкой (естественно, исходя из предположений, что операционной системе известен путь к этой программе, а все четыре вышеописанных исходника находятся в одном каталоге):

```
ml /c /coff *.asm
```

В результате должно получиться четыре объектных файла: *ch2digit.obj*, *dword2hexstr.obj*, *hexstr2dword.obj* и *SimpleHash.obj*.

Теперь запускаем "мясорубку" *link.exe* следующим образом:

```
link -lib *.obj /out:dzofasm.lib
```

Все, библиотека готова ;). Однако прежде чем мы сможем ее использовать, необходимо позаботиться об одноименном файле с прототипами. Открываем текстовый редактор F4 и пишем там следующее:

```
ch2digit      PROTO
dword2hexstr  PROTO :DWORD, :DWORD
hexstr2dword  PROTO :DWORD, :DWORD, :DWORD, :DWORD
SimpleHash    PROTO :DWORD, :DWORD
```

#2. Как известно, в *win32* принято передавать параметры через стек, а в процедуре *ch2digit* передача проводится через регистр — разве это правильно? С другой стороны, если сделать передачу параметра через стек, то не слишком ли "тяжела" подобная "артиллерия" для вызова подобной "мелочи"? А с третьей стороны, правильно ли вообще было выносить этот код в отдельную процедуру?

Отвечаю на эти три вопроса: правильно, правильно, правильно и ("не может быть, чтобы все это было правильно!") еще раз ;), правильно. И, надев на себя маску Йоды, в очередной раз делаю ожесточенную попытку объяснить читателю сущность программистского дзена...

Процесс программирования вообще, и разработка библиотечных функций в частности, сводится к следующим "целям":

1. Универсальность (желательна переносимость кода с как можно меньшими затратами по "вырезанию" и "интеграции" в новое место).
2. Размер (желательна краткость кода и его вызовов, минимализм места под данные).
3. Скорость (желателен код, который не выполняет лишних действий помимо своей основной функции или выполняет их как можно быстрее).

Первая цель важна для "управления" сорсами (*source*), сопровождения и повторного использования кода, а также обучения программированию (аналог — бюрократия, управление производством). Вторая и третья цели важны для исполнения программ (аналог — собственно производство). И хотя иногда движение ко второй цели приближает и к третьей (и наоборот), в большинстве случаев эти цели противоречат друг другу ("конфликт пространства—времени"), а также первой цели.

Например, если нам важнее размер, и не так важна скорость, то выделение кода в отдельную процедуру — полезно. Если нам важна скорость, и не так важна компактность кода, то лучше использовать макросы и вообще прямую вставку кода. Если нам опять важен размер кода, то нужно смотреть на размер самого кода, размер кода вызова и обрамления процедуры и на количество вызовов (читай — нужно выбрать минимум из $p \cdot f$ и $p \cdot c + p \cdot f$, где: p — количество вызовов; f — размер кода; а c и p — размеры кода вызова и обрамления процедуры).

Таким образом, нет единых, универсальных рекомендаций/требований "делай всегда так-то и так-то" — все зависит от целей ;). Желающих более пристально помедитировать над данной темой отсылаю к книге ДЗ Бентли "Жемчужины программирования".

Говоря о перлах, не могу не привести пример одной из моих любимых "жемчужин" в программировании. Это функция, аналогичная вышеприведенной *dword2hexstr*, и взята она из библиотеки *m32lib/MASM32*:

```
dw2a proc dwValue:DWORD, lpBuffer:DWORD
    invoke sprintf,lpBuffer,ADDR fmtStrInG,dwValue
    ret
dw2a endp
```

#3. Программа, которую мы с вами сегодня напишем, будет реверсировать хеш приблизительно таким же образом, как это делают известные и всенародно любимые программы от Elcomsoft'a. Сразу оговорюсь, что наша "простая хеш-сумма" отнюдь не является "необратимым" хешем, и реверсировать его можно просто "решая уравнения". Но, как говорится, "примера ради" мы поступим с ним грубо. А заодно и поюзаем нашу свежееиспеченную библиотеку.

Программа будет просить ввести хеш и методом перебора искать все возможные комбинации паролей, печатая при этом подходящие (ну да, их несколько, не удивляйтесь, функция неинъективна). Вот ее исходник:

```
.386

.model flat,stdcall
option casemap:none

include \masm32\include\kernel32.inc
include \masm32\include\windows.inc
include \include\dzofasm.inc

includelib kernel32.lib
includelib dzofasm.lib

.data ;-|глобальные переменные и константы|-----

S_consoleTitle db 'SimpleHashReverser Program',0
; заголовок окна

S_prompt      db 'Enter hash value > '
; приглашение
szPrompt      = $-S_prompt
; размер приглашения
```



```
S_alphabet      db '0123456789'
                db 'AaBbCcDdEeFfGgHhIiJjKkLlMm'
                db 'NnOoPpQqRrSsTtUuVvWwXxYyZz'
                db '~!@#$%^&*()_+?/\|' ; алфавит
szAlphabet      = $-S_alphabet ; число символов алфавита

szBuffer        = 120 ; размер буферов ключа и пароля
S_password      db szBuffer dup(?) ; буфер пароля
S_key           db szBuffer dup(?) ; буфер ключа

szHash          = 8 ; размер буферов хеша-результата
                ; и ввода
S_hash          db szHash dup(0) ; буфер хеша-результата
S_input         db szHash dup(0) ; буфер ввода

.code
=====
main proc uses ebx esi ebp edi
;----- локальные переменные

    local hinput :dword ; хэнгл ввода
    local houtput :dword ; хэнгл вывода
    local nwritten :dword ; "напечатано символов"
    local nread :dword ; "введено символов"

    local strafter :dword ; указатель на необра-
                        ; ботанную часть строки
    local remained :dword ; счетчик оставшихся знаков

    local targethash :dword ; реверсируемый хеш

;----- подготовительные операции

    invoke SetConsoleTitle,addr S_consoleTitle ; задать
                        ; заголовок окна
    invoke GetStdHandle,STD_INPUT_HANDLE ; получить
                        ; хэнгл ввода
    mov [hinput],eax ; ...и сохранить его
    invoke GetStdHandle,STD_OUTPUT_HANDLE ; получить
                        ; хэнгл вывода
    mov [houtput],eax ; ...и сохранить его

;----- ввод значения хеша, для которого подбирается пароль

    invoke WriteConsole,houtput,addr
S_prompt,szPrompt,addr nwritten,NULL
    invoke ReadConsole,hinput,addr S_input,szHash,addr
nread,NULL
    invoke hexstr2dword,addr S_input,nread,addr
strafter,addr remained
    mov [targethash],eax

;----- рабочая часть

    mov esi,offset S_key ; указатель на ключ
    mov edi,offset S_password ; указатель на пароль
    mov ecx,esi ; старший разряд ключа

@@keylenloop:
    cmp ecx,offset S_key+szBuffer ; вылезли за пределы
                        ; буфера?
    jae @@break ; да - закончить генерацию
    mov byte ptr [esi],0 ; нет - инициализировать
                        ; новый
                        ; старший разряд ключа

@@passwordloop:
    movzx ebx,byte ptr [esi] ; значение разряда ключа
    mov al,S_alphabet[ebx] ; ...в символ "алфавита"
    mov [edi],al ; ...установить разряд
                        ; пароля

;-[A]-----
; вычисление хеш-функции пароля и сравнение его с целевым хешем

    push ecx
```

```
    mov edi,offset S_password ; указатель на пароль
    sub ecx,offset S_key-1 ; длина пароля
    invoke SimpleHash,edi,ecx
    pop ecx

.if [targethash] == eax
    push ecx
    sub ecx,offset S_key-3 ; длина пароля+два байта CRLF
    mov word ptr [edi+ecx-2],0A0Dh ; CRLF после пароля
    invoke WriteConsole,houtput,edi,ecx,addr nwritten,NULL
    pop ecx
.endif

;-[B]-----

    mov esi,offset S_key ; указатель на ключ
    mov edi,offset S_password ; указатель на пароль
@@carryloop:
    inc byte ptr [esi] ; увеличить разряд ключа
    cmp byte ptr [esi],szAlphabet ; максимальное
                        ; значение разряда?
    jb @@passwordloop ; НЕТ - следующий ключ готов

    mov byte ptr [esi],0 ; ДА - обнулить разряд ключа
    mov al,S_alphabet[0]
    mov [edi],al ; ...и соответствующий
                        ; разряд пароля
    inc esi ; следующий разряд ключа
    inc edi ; следующий разряд пароля
    cmp esi,ecx ; вылезли за пределы
                        ; текущей длины?
                        ; (исчерпаны все значения
                        ; ключа?)
    jbe @@carryloop ; нет - увеличить
                        ; следующий разряд

    inc ecx ; да - добавить новый
                        ; разряд в ключ

    jmp @@keylenloop
@@break:

;----- на прощание

    invoke ExitProcess,0 ; ВЫХОД ;)

main endp

end main
```

По сравнению с нашей предыдущей программой, здесь только два кардинальных изменения. Во-первых, мы сделали ввод хеша, и, следовательно, возникла потребность в подпрограмме конвертации строки шестнадцатеричных символов в соответствующее число (см. процедуру `hexstr2dword`. Под нее же, кстати говоря, была заведена и парочка локальных переменных — `strafter` и `remainder`).

Во-вторых, добавлена функция вычисления хеша текущего пароля (см. процедуру `SimpleHash`) и сравнение его с реверсируемым. В том случае, если хеш-сумма очередного пароля равна реверсируемому, пароль будет печататься. По этому поводу, домашнее задание — посмотрите, в какие инструкции ассемблируется высокоуровневая конструкция

```
.if выражение
;какой-то код
.endif
```

перед тем как я расскажу вам об этом сам ;)

(Продолжение следует)



Перехват вызовов API в системах семейства Windows

А.ТЕРЁШКИН /HI-TECH,
г. Майкоп,
WWW: <http://wasm.ru/>

(Продолжение. Начало в №1/2003)

2.3. Изменение таблиц импорта

Когда компилятор встречается в исходном тексте вызов функции, которая присутствует не в компилируемом исполняемом файле, а в некотором другом — чаще всего в dll, в простейшем случае он генерирует "call" на этот символ. Впоследствии линкер исправляет этот псевдовызов на вызов переходника ("stub"), используя библиотеку импорта, содержащую переходники для всех экспортируемых символов в указанных библиотеках. Такие переходники состоят из одной инструкции — "jmp [x]", где x — адрес двойного слова в таблице импорта PE-файла. Эти адреса загрузчик PE-файла заполняет корректными значениями при инициализации модуля, опираясь на данные, указанные в таблице импорта.

В более сложных случаях (при непосредственном указании импортируемой функции) компилятор генерирует "call [x]", минуя переходник. Таблица (директория) импорта должна располагаться в секции, имеющей атрибуты "инициализированные данные" и "читаемая" (IMAGE_SCN_CNT_INITIALIZED_DATA и IMAGE_SCN_MEM_READ). Таблица импорта состоит из массива структур — дескрипторов импорта (IMAGE_IMPORT_DESCRIPTOR), завершающим элементом которого является нулевая структура. Дескриптор импорта выглядит следующим образом:

```
IMAGE_IMPORT_BY_NAME    STRUC
IBN_Hint                 DW      ?
IBN_Name                 DB      1 DUP (?) ;длина не
фиксирована
IMAGE_IMPORT_BY_NAME    ENDS

IMAGE_THUNK_DATA         STRUC
UNION
TD_AddressOfData         DD      IMAGE_IMPORT_BY_NAME PTR ?
TD_Ordinal               DD      ?
TD_Function              DD      BYTE PTR ?
TD_ForwarderString       DD      BYTE PTR ?
ENDS
IMAGE_THUNK_DATA        ENDS

IMAGE_IMPORT_DESCRIPTOR  STRUC
UNION
ID_Characteristics       DD      ?
ID_OriginalFirstThunk    DD      IMAGE_THUNK_DATA PTR ?
ENDS
ID_TimeDateStamp         DD      ?
ID_ForwarderChain        DD      ?
ID_Name                  DD      BYTE PTR ?
ID_FirstThunk            DD      IMAGE_THUNK_DATA PTR ?
IMAGE_IMPORT_DESCRIPTOR ENDS
```

ID_OriginalFirstThunk и ID_FirstThunk содержат относительные виртуальные адреса (RVA) структур IMAGE_THUNK_DATA, описывающие импортируемые функции.

ID_TimeDateStamp содержит предполагаемый "штамп времени" модуля-экспортера и используется при технике линковки "bound imports". Если импорты не связаны, то значение этого поля равно 0.

ID_ForwarderChain содержит RVA первого форварда в списке импортируемых функций. Если форварды отсутствуют, то значение этого поля равно -1. ID_Name содержит RVA имени импортируемого модуля.

Массивы ID_OriginalFirstThunk и ID_FirstThunk идут параллельно. Два массива необходимы для сохранения ин-

формации о импортируемых функциях — массив ID_OriginalFirstThunk загрузчиком изменен не будет, а массив ID_FirstThunk будет заполнен адресами требуемых функций (RVA имен функций уничтожатся).

Модуль может экспортировать функции, реализация которых в нем не присутствует, а лишь импортируется из другого модуля. Этот прием называется "форвардинг функций". Его можно проиллюстрировать на примере wsock32.dll. Эта библиотека экспортирует множество функций, часть из которых — переходники к функциям ws2_32.dll, а часть просто отсутствует в модуле — при импорте отсутствующей, но экспортируемой функции из wsock32, загрузчик, анализируя форварды wsock32, обнаруживает, что он должен экспортировать эту функцию из другого модуля, в частности, из ws2_32.dll.

Смысл перехвата функций методом исправления таблицы импорта состоит в следующем:

- определяется адрес перехватываемой функции;
- исходя из данных PE-заголовка модуля-жертвы вычисляется адрес его таблицы импорта;
- среди дескрипторов импорта ищется тот, который описывает импорты из модуля, содержащего реализацию перехватываемой функции;
- перебираются все структуры IMAGE_THUNK_DATA, начиная с RVA=ID_FirstThunk — найденного дескриптора в поисках полей TD_Function, которые содержат адрес перехватываемой функции;
- найденные адреса заменяются адресом обработчика перехваченной функции.

Код функции, осуществляющей вышеперечисленные шаги, может выглядеть так (форвардинг функций не учитывается):

```
hook_api proc modbase:dword,modname:dword,\
             procname:dword,hook_proc:dword
             local oldproc:dword
             local dummy:dword

             pushad
             @SEH_SetupFrame "jmp bad_exit"
             call IsBadCodePtr,hook_proc ; проверка
                                           ; корректности вызова

             test eax,eax
             jnz bad_exit
             push procname
                                           ; шаг 1 - узнаем
                                           ; адрес
             call GetModuleHandleA,modname ; перехватываемой
                                           ; функции

             call [realGetProcAddress],eax
             test eax,eax
             mov oldproc,eax
             jz bad_exit
             mov edi,modbase
             call IsBadReadPtr,edi,40h
             test eax,eax
             jnz bad_exit
             cmp word ptr [edi],'ZM'
             jnz bad_exit
             mov ebx,[edi.MZ_lfanew]
             push 0F8h
             add ebx,edi
             call IsBadReadPtr,ebx
             test eax,eax
             jnz bad_exit
             cmp dword ptr [ebx],'EP'
             jnz bad_exit
```



```

mov     esi,[ebx.NT_OptionalHeader\ ; шаг 2:
        .OH_DirectoryEntries\; получение
        .DE_Import\ ; адреса таблицы
        .DD_VirtualAddress] ; импорта

or      esi,esi
jz      bad_exit
add     esi,edi
cmp     esi,ebx
jz      bad_exit
stc
mov     eax,[esi.ID_Name]
test    eax,eax
jz      noimps

next_imp_desc: ; шаг 4: перебираем
              ; дескрипторы импорта
push    esi
push    edi
nxtchar_: call patchthisidesk

pop     edi
pop     esi
mov     eax,[(esi\
+IMAGE_SIZEOF_IMPORT_DESCRIPTOR).ID_Name]
add     esi, IMAGE_SIZEOF_IMPORT_DESCRIPTOR
test    eax,eax
jnz     next_imp_desc

noimps: popad
xor     eax,eax
jc      simpleret
mov     eax,oldproc

simpleret: @SEH_RemoveFrame
ret

bad_exit: @SEH_RemoveFrame
popad
xor     eax,eax
stc
ret

bad_exit4proc:
pop     eax
jmp     mismatch

patchthisidesk: ; шаг 3: ищем вход
                ; перехватываемой
                ; функции в данном
                ; дескрипторе импорта
sub     esi,0ch
test    eax,eax
jnz     bad_exit4proc
mov     eax,[esi.ID_Name]
test    eax,eax
jz      bad_exit4proc
mov     esi,[esi.ID_FirstThunk]
add     esi,edi
call    IsBadReadPtr,esi,4
test    eax,eax
jnz     bad_exit4proc
mov     eax,[esi]
test    eax,eax
jz      bad_exit4proc
mov     edi,oldproc

next_thunk:
cmp     eax,edi
jnz     next_thunk_
call    VirtualProtect,esi,1000h,4,offset dummy,esi
pop     esi

mov     eax,hook_proc
mov     [esi],eax ; шаг 5: заменим адрес
                  ; на свой
clc

next_thunk_:
mov     eax,[esi+4] ; проверим TD_Function
                  ; следующего блока на 0

add     esi,4
test    eax,eax
jnz     next_thunk
db      0c3h;ret

hook_api endp

```

2.4. Сплайсинг функций ядра Windows 9x

В Win9x, в отличие от WinNT/Win2k, память, начиная с 2 Гб, спроецирована на все контексты, присутствующие в системе, т.е. ее содержимое одинаково для всех процессов. В этом регионе памяти находятся модули ядра, объекты ядра и пользовательские объекты, доступные всем контекстам — проекции файлов. Документированный Windows API не предоставляет средств для внесения изменений в память выше 2 Гб. VirtualProtect + WriteProcessMemory завершается неудачно. Дж.Рихтер в своей книге "Programming Applications for Microsoft Windows" так поясняет эту ситуацию: "On Windows 98, the main Windows DLLs (Kernel32, AdvAPI32, User32, and GDI32) are protected in such a way that an application cannot overwrite their code pages. You can get around this by writing a virtual device driver (VxD)". Однако эту проблему можно решить и без написания собственного VxD. Достаточно вызвать VxDCall0_PageModifyPermissions. VxDCall0 — это всегда первая экспортируемая Kernel32.dll функция. Нижеследующий код разрешит на чтение/запись первую страницу kernel32 (идея впервые описана Vecna).

```

call    GetModuleHandleA,offset kernel32
mov     ebx,eax
mov     eax,[ebx.MZ_lfanew]
lea     edi,[eax.ebx]
mov     esi,[edi.NT_OptionalHeader.\
OH_DirectoryEntries.DE_Export.\
DD_VirtualAddress]
mov     esi,[esi.ebx.ED_AddressOfFunctions]
mov     ecx,[esi.ebx]
add     ecx,ebx ; ecx==VxDCall10
shr     ebx,12
push    020060000h
push    00h
push    01h
push    ebx
push    001000dh ;_PageModifyPermissions
call    ecx

```

Надо учитывать спроецированность памяти выше 2 Гб, ведь если некоторый регион будет разрешен для записи, а потом его память будет изменена, то эти изменения произойдут одновременно во всех контекстах. Таким образом, если удастся изменить ядро для передачи управления некоторым функциям другим обработчикам, расположенным в памяти ниже 2 Гб (память пользовательского процесса), то при вызове перехваченной функции процессом другого контекста произойдет сбой — по указанному адресу не окажется соответствующего кода обработчика. Следовательно, код глобальных обработчиков функций ядра необходимо располагать также в памяти ниже 2 Гб. Код в памяти сверх 2 Гб можно расположить несколькими способами. Если размер кода небольшой, то его можно уместить в межсекционное пространство какого-либо модуля ядра (размер кода не больше разницы между физической и виртуальной длиной секции PE-модуля). Следующий код пытается найти удовлетворяющее этому требованию место в регионе, занятом kernel32, и, в случае успеха, копирует в найденное пространство код обработчика:

```

call    GetModuleHandle,offset kernel32
mov     ebx,eax
mov     eax,[ebx.MZ_lfanew]
movzx   ecx,word ptr [eax.ebx.NT_FileHeader.\
FH_NumberOfSections]
lea     esi,[eax.ebx+SIZE_IMAGE_NT_HEADERS]

try_next_section:
mov     eax,[esi.SH_Characteristics]
and     eax,IMAGE_SCN_MEM_WRITE\
+IMAGE_SCN_MEM_READ\
+IMAGE_SCN_CNT_INITIALIZED_DATA
cmp     eax,IMAGE_SCN_MEM_WRITE\

```



```

        +IMAGE_SCN_MEM_READ\
        +IMAGE_SCN_CNT_INITIALIZED_DATA
jne     next_section
mov     eax,[esi.SH_SizeOfRawData]
mov     edi,[esi.SH_VirtualSize]
sub     eax,edi
cmp     eax,CODE_SIZE
jnb     next_section
add     edi,[esi.SH_VirtualAddress]
add     edi,ebx
jmp     copy_code
next_section:
add     esi,IMAGE_SIZEOF_SECTION_HEADER
loop   try_next_section
jmp     section_not_found
copy_code:
mov     esi,offset IMPLANT_CODE
mov     ecx,CODE_SIZE
cld
rep     movsb ;скопировать код в найденное пр-во

```

Плюс этого метода в том, что процесс, осуществивший запись в межсекционное пространство, никак не связан с этой памятью. Он может быть закрыт, и память не будет освобождена.

Однако чаще всего свободной памяти, найденной таким путем, не хватает. В таком случае можно "разбросать" части своего обработчика по всем межсекционным пространствам модулей, загруженным выше 2 Гб, а потом "склеивать" части обработчика на лету. Этот метод используется, к примеру, в вирусе Win9X.CIH (он распределяет свой код в межсекционном пространстве зараженного модуля — при этом физический размер модуля на диске не изменяется). Таким способом можно размещать обработчики размером не больше 10 Кб. Понятно, что любой серьезный проект будет превышать этот предел.

Существует более простой и надежный метод выделения памяти >2 Гб. Можно выделить память под объект — проекцию файла, например, из своп-файла. Содержимое подобного объекта будет расположено системой в памяти >2 Гб, все выделенные страницы могут быть помечены атрибутом "исполняемые".

```

call    CreateFileMappingA,0xffffffff,0,NULL,\
PAGE_READWRITE,0,IMPLANT_SIZE,0
call    MapViewOfFile,eax,FILE_MAP_WRITE,\
SECTION_MAP_EXECUTE,0,0,IMPLANT_SIZE
mov     edi,eax
mov     esi,offset IMPLANT
mov     ecx,IMPLANT_SIZE
cld
rep     movsb

```

С помощью этого метода можно выделить память сколь угодно большого размера, но она будет иметь хозяина — процесс, ее выделивший. Поэтому если процесс, установивший глобальный обработчик в память >2 Гб, завершится, то выделенная им память освободится; любой вызов в ядре, приводящий к передаче управления на эту память, в лучшем случае приведет к аварийному завершению процесса, поток которого совершил "незаконное" действие, а в худшем — к краху системы. Следовательно, глобальный перехватчик функции ядра не должен завершиться.

Общим недостатком этих методов является неопределенность базового адреса копируемого кода. В любом случае очевидно, что базовый адрес, указанный линковщиком при компиляции модуля-перехватчика, не совпадет с его истинной базой в памяти. Также возникает вопрос о вызовах API-функций таким обработчиком — таблица импорта отсутствует.

Самое простое решение первой проблемы состоит в применении техники базонезависимого кода. Идея в том, что в самом коде хранятся лишь относительные адреса данных. При инициализации кода он определяет свою базу и, прибавляя

ее к относительным адресам, вычисляет абсолютные адреса данных. Код на ЯВУ не может быть базонезависимым в силу негибкости компиляторов (уточнение — может, но только в случае неиспользования им глобальных меток).

Пример базонезависимого кода:

```

call    delta
delta:   pop     ebp
sub     ebp,offset delta-code_start
lea     esi,[ebp+offset _data-code_start]
...
_data   db      'Text',0

```

Рассмотрим решение второй проблемы применительно к функциям ядра. Так как модули ядра присутствуют во всех контекстах по одному и тому же адресу, то для вызовов их функций из обработчиков достаточно построить в них переходники, а необходимые адреса функций в них будут записываться еще до копирования кода обработчика в регион >2 Гб. Типичный код переходника и кода, его заполняющего, выглядит так:

```

; код в секции кода процесса-перехватчика
call    GetModuleHandleA,offset user32
call    GetProcAddress,eax,offset msgboxa
mov     msgboxa_,eax
...
; код в секции данных процесса-перехватчика:
; он будет скопирован в память >2 Гб
_MessageBoxA: mov     eax,12345678h
org      $-4
msgboxa_ dd      0
jmp      eax

```

Используя вышеприведенную технику, вызов функций ядра становится тривиальным:

```

mov     eax,[ebp+offset _uType-code_start]
push    eax
lea     eax,[ebp+offset _caption-code_start]
push    eax
lea     eax,[ebp+offset _text-code_start]
push    eax
mov     eax,[ebp+offset _hWnd-code_start]
push    eax
call    _MessageBoxA

```

Но как быть в случае, когда необходимо вызвать функцию из библиотеки, расположенной ниже 2 Гб? В разных контекстах она может быть загружена (если загружена) в результате каких-то коллизий по разным адресам — следовательно, ее адрес нельзя заранее записать в переходник. В этом случае сначала нужно определить адрес библиотеки в текущем контексте (если библиотека туда не загружена, загрузить ее), а затем найти адрес требуемой функции.

Для надежности не следует сразу пользоваться функциями LoadLibrary*, т.к. если библиотека уже находилась в контексте, то эта операция инкрементирует ее счетчик использования. Если окажется так, что эта dll была загружена в контекст единственный раз, то процесс-хозяин, выполнив FreeLibrary, должен был бы выгрузить ее, но в результате непредусмотренного вызова LoadLibrary* FreeLibrary лишь декрементирует ее счетчик, и может нарушиться логика выполнения программы-хозяина. Таким образом, для более "невидимого" вмешательства в чужой процесс необходимо сначала проверить присутствие требуемого модуля функцией kernel32!GetModuleHandle, и только в случае его отсутствия воспользоваться LoadLibrary* (после выполнения функций, dll следует выгрузить с помощью FreeLibrary по тем же причинам).

```

lea     ebx,[ebp+offset ws2_32-offset code_start]
call    _GetModuleHandleA,ebx
push    ebp
xor     ebp,ebp
or      eax,eax
jnz     dlloaded

```



```

call _LoadLibrary,ebx
mov ebp,eax
dllloaded: lea ecx,[ebp+(offset clsock-offset code_start)]
call _GetProcAddress,ebx,ecx
mov ecx,[ebp+(offset hSocket-offset code_start)]
call eax,ecx
call _FreeLibrary,ebp ; пройдет успешно, если
pop ebp ; dll загрузили мы

```

Итак, любая память доступна нам для чтения/записи, наш код может выполняться корректно вне зависимости от его базы и вызывать любые API функции. Теперь можно приступить к исправлению кода функций ядра для передачи управления нашему обработчику.

Ebp указывает на начало найденной памяти. Для начала сохраним первые 5 байтов перехватываемой функции:

```

call GetProcAddress,ebx,offset lla
mov edi,eax
lea esi,[ebp+(offset lla_code-offset code_start)]
push edi
xchg esi,edi
movsb
movsd

```

Затем изменим флаг разрешения записи атрибута страницы, содержащей начало перехватываемой функции, на истинный:

```

mov eax,[esp]
pushad
shr eax,12
push 020060000h
push 00h
push 01h
push eax
push 001000dh ;_PageModifyPermissions
mov eax,[vxdcall0]
call eax
popad

```

pop edi
Построим пятибайтовый JMP в начале функции:

```

mov al,0e9h
stosb ;прибавим 4 к edi
stosd

```

```

lea eax,[ebp+(offset lla_entry-offset code_start)]
sub eax,edi
mov [edi-4],eax

```

При таком методе организации сплайсинга обработчик перехваченной функции может выглядеть так (синтаксис функции — одно двойное слово):

```

lla_entry: call swap_lla
push dword ptr [esp+4]
call _lla ;вызов настоящей функции
call swap_lla
... ;какие-то действия
ret 4

```

```

_lla: mov eax,12345678h
org $-4
lla: dd 0
jmp eax

```

```

swap_lla: push esi edi ebp
call delta
delta: pop ebp
sub ebp,offset delta-code_start

```

```

lea esi,[ebp+(offset lla_code-code_start)]
mov edi,[ebp+(offset lla_code_start)]
push eax
mov eax,[edi]
xchg [esi],eax
mov [edi],eax
mov al,[edi+4]
xchg [esi+4],al
mov [edi+4],al
pop ebp edi esi
ret

```

(Продолжение следует)

Тайна ненайденного файла

Хочу рассказать об одной истории — возможно, мой опыт вам поможет.

Запускаю демо "Dive" с компакт-диска. Все работает. После этого копирую на винчестер и запускаю оттуда. Не работает — при запуске выдает сообщение "Can't find data file" ("Не могу найти файл данных"). Хотя все файлы я скопировал!

В чем же дело? Может быть, в директории D:\SCENE&~1\DIVЕ, откуда я копировал файлы, был какой-то файл с атрибутом "hidden" ("скрытый"), не отображающийся в панели Dos Navigator'a и поэтому не скопированный? Чтобы проверить это, включил опцию "показывать скрытые файлы" — никаких скрытых файлов не обнаружилось.

Версию о том, что причина неработоспособности — сработавшая защита от копирования, я отбросил практически сразу: защищена может быть какая-нибудь коммерческая программа, но никак не демо.

Может быть, программа жестко привязана к директории D:\SCENE&~1\DIVЕ и только там ищет свои файлы? Чтобы проверить это, я отключил драйвер CD-ROM, создал директорию C:\BAK\SCENE&~1\DIVЕ, скопировал в нее

файлы программы и командой subst сопоставил директорию C:\BAK и логический диск D. Но и это не помогло...

Тогда я попробовал зайти с другой стороны. Сообщение "Не могу найти файл...", очевидно, появляется после того как программа пытается открыть какой-то файл, и эта операция завершается неудачно. Для открытия файла служит функция 3Dh прерывания 21h. При ее вызове в DS:DX помещается адрес ASCII-строки с именем открываемого файла, а в AL — код доступа: 0 — открыть файл только для чтения, 1 — только для записи, 2 — для чтения и записи. После вызова функции, если все нормально, флаг C сброшен, а в регистре AX — номер открытого файла. Если же открыть файл не удалось, флаг C установлен, а в AX — код ошибки.

Я написал резидентную программу, которая перехватывает прерывание 21h и, если вызвана функция открытия файла, печатает значения регистров и строку, на которую указывает DS:DX, до и после вызова функции. При запуске демо эта резидентная программа напечатала следующее:

```

FLAGS=7046 AX=3D02 BX=A372 CX=0000 DX=A3B0 name=DIVE.PRJ
FLAGS=7047 AX=0005 BX=A372 CX=0000 DX=A3B0 name=DIVE.PRJ

```

И.РОЩИН,

г.Москва,

Fido: 2:5020/689.53

ZXNet: 500:95/462.53

E-mail: asder_ffc@softhome.net

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

То есть программа открывает файл DIVE.PRJ (который, как я сразу же проверил, присутствует на диске) для чтения и записи (AL=2). Функция завершается с ошибкой (нечетное значение регистра флагов — 7047h — соответствует установленному флагу C, так как этот флаг размещается в самом младшем бите регистра флагов). Код ошибки — 5, что значит “нет доступа” (а вовсе не “файл не найден”, чему соответствует код 2). Очевидно, программа не анализирует код ошибки и при любой ошибке выдает “Can not find data file”.

Почему же нельзя получить доступ к файлу DIVE.PRJ для чтения и записи? Может быть, у этого файла установлен атрибут “read only” (“только чтение”)?

Проверил — так оно и оказалось. И после того как этот атрибут был снят, все заработало!

Но позвольте, а как же тогда все работало при запуске demo с компакт-диска? Ведь и там у файла DIVE.PRJ был установлен атрибут “read only”!

При запуске demo с компакт-диска резидентная программа напечатала следующее:

```
FLAGS=7046 AX=3D02 BX=A372 CX=0000 DX=A3B0 name=DIVE.PRJ
FLAGS=7046 AX=0005 BX=A372 CX=0000 DX=A3B0 name=DIVE.PRJ
```

Как видите, тот же самый вызов функции открытия файла почему-то завершается успешно (значение регистра флагов четно — значит, флаг C сброшен). То есть файл, находящийся на компакт-диске, почему-то можно открыть для чтения и записи, несмотря на установленный атрибут “read only”...

Кстати говоря, в рассматриваемом случае, очевидно, достаточно было открыть файл DIVE.PRJ только для чтения — если бы программе и впрямь было бы необходимо что-то записывать в этот файл, она не запустилась бы нормально с компакт-диска...

Итак, тайна ненайденного файла была раскрыта. После этого я вспомнил, что ранее уже сталкивался с похожей ситуацией. При запуске с винчестера (или с дискеты) еще одного demo на экране можно было увидеть следующее:

```
C:\WORK\UNSORTED>verses.exe
Fatal error: VERSES.EXE not found!
```

Если верить этому сообщению, программа не может отыскать... собственный exe-файл! :-)

Проверив атрибуты этого файла, я обнаружил, что атрибут “read only” также установлен. После его снятия и это demo нормально запустилось. Причина неработоспособности, как и в первом случае, была не в том, что программа не могла найти нужный файл, а в том, что она не могла открыть этот файл для чтения и записи.

Так что, если вы столкнетесь с аналогичной ситуацией, советую проверить атрибуты файла (файлов) программы. В Dos Navigator'е это можно сделать, нажав комбинацию клавиш Alt+F. Также в Dos Navigator'е удобно включить автоматическое снятие атрибута “read only” при копировании с CD-ROM (Настройки\Конфигурация\Системные установки). Когда я вначале копировал файлы, эта опция была выключена.

Ну а при написании своих программ желательно предусмотреть, чтобы выводимые сообщения об ошибках не вводили пользователя в заблуждение...

Литература

1. П.Абель. Ассемблер и программирование для IBM PC. — М.: Мир, 1985.

О.ВАЛЬПА,

E-mail: sandh@narod.ru

Стрелочные часы

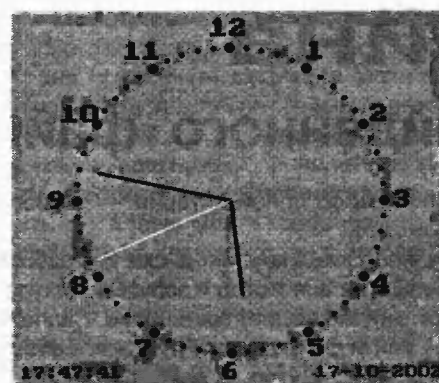
Иногда при создании интерфейса пользователя или экранной заставки появляется необходимость отображать часы. Для придания большей привлекательности изображению, часы можно отображать в виде циферблата с движущимися стрелками. Я хочу рассказать о том, как это сделать.

Встроенная библиотека языка СИ позволяет решить данную задачу весьма красиво и просто. Для создания такой программы необходимо познакомиться с некоторыми функциями языка СИ, которые позволяют извлечь внутреннее время компьютера. Кроме того, в программе будут использованы графические функции, позволяющие выводить на экран цифры разного формата, а также рисовать окружности и линии в цвете.

Алгоритм решения данной задачи довольно прост. Из компьютера извлекается текущее время. Далее вычисляются координаты концов часовой, минутной и секундной стрелок. Это делается с помощью тригонометрических функций. Естественно, выбирается определенная длина каждой стрелки.

От центра к этим координатам рисуются прямые линии различной толщины или цвета.

Компьютерное время считывается циклически, и каждый раз происходит сравнение старых данных о времени с новыми данными. При изменении времени стрелки стираются путем прорисовки их цветом фона и рисуются новые стрелки. Так создается эффект движения стрелок на циферблате. Копия экрана часов представлена на рисунке.



Для привлекательности изображение часов дополнено отображением на циферблате текущего времени и даты в цифровом виде.

Ниже приведен текст программы. Он снабжен подробными комментариями, которые позволяют легко разобраться в программе и при желании внести в нее новые элементы. Например, можно поместить на циферблате красочный рисунок или календарь на текущий год. От этого программа станет более полезной. Не забывайте, что в каталоге, из которого запускается программа, должен присутствовать файл графической библиотеки egavga.bgi. Желаю всем читателям творческих успехов в программировании.



Текст программы

```
// =====
// Программа «clock» отображает на экране в графическом режиме
// циферблат стрелочных часов и показывает текущее время и дату
//
// Файл:clock.cpp
// Дата:15.10.2002
// Автор: Вальпа Олег
// =====

// Библиотечные функции
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <bios.h>
#include <dos.h>
#include <math.h>

// Координаты сообщений
#define XKM 400
#define YKM 30
#define XCL 480
#define YCL 30
#define XKR 550
#define YKR 30
#define PI 3.1415

// Скан-код клавиши ESC
#define ESC 0x1b

// Цвета
#define COL_FON LIGHTGRAY // Цвет фона
#define COL_MET BLACK // Цвет меток
#define COL_NUM BLACK // Цвет цифр
#define COL_TIME BLUE // Цвет времени
#define COL_DATE GREEN // Цвет даты
#define COL_MSTR BLACK // Цвет минутной стрелки
#define COL_CSTR BLACK // Цвет часовой стрелки
#define COL_SSTR YELLOW // Цвет секундной стрелки

// Начало программы
int main(void)
{
    // Инициализация переменных
    int gdriver = DETECT, gmode, errorcode, i;
    int x, y, mx, my, sx, sy, xmax, ymax, ret, bt;
    int god, den, mes, oden, omes, ogod;
    int xtime, ytime, xdate, ydate;
    int xsec, ysec, xchs, ychs, xmin, ymin, sec, ssec, min, chs;
    int oxsec, oysec, oxchs, oychs, oxmin, oymin, osec, ossec, omin, ochs;
    char msg[80], klav, mklav;
    union REGS regs;
    struct dostime_t t;
    struct dosdate_t d;
    struct SREGS segs;
    struct REGPACK regpack;

    /* Инициализация графики */
    initgraph(&gdriver, &gmode, "");

    /* Чтение и обработка результата инициализации */
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Ошибка инициализации графики: %s\n",
            grapherrormsg(errorcode));
        printf("Нажмите любую клавишу");
        getch();
        exit(1);
    }

    /* Очистка экрана */
    cleardevice();
    ret=0; x=0; y=0;
    xmax = getmaxx(); // Получить предельные
    ymax = getmaxy(); // координаты экрана

    // Закрасить экран цветом фона
    setfillstyle(SOLID_FILL, COL_FON);
    bar(0, 0, xmax, ymax);

    // Вычислить координаты вывода времени и даты
    xtime=xmax*0.2; ytime=ymax*0.95;
    xdate=xmax*0.8; ydate=ymax*0.95;

    // Вычисляем координаты и рисуем часовые метки
    setcolor(COL_MET);
    for (i=1; i<=12; i++)
    {
        x=cos((90-30*i)*PI/180)*ymax/2*0.8 + xmax/2;
        y=sin((270-30*i)*PI/180)*ymax/2*0.8 + ymax/2;
        circle(x, y, 5); setfillstyle(SOLID_FILL, COL_MET);
        floodfill(x, y, COL_MET);
    }

    // Проставить цифры
    settextstyle(0, 0, 3);
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    setcolor(COL_NUM);
    for (i=1; i<=12; i++)
    {
        x=cos((90-30*i)*PI/180)*ymax/2*0.9 + xmax/2;
        y=sin((270-30*i)*PI/180)*ymax/2*0.9 + ymax/2;
        sprintf(msg, "%d", i);
        outtextxy(x, y, msg);
    }

    // Рисуем минутные метки
    for (i=0; i<=59; i++)
    {
        x=cos((90-6*i)*PI/180)*ymax/2*0.8 + xmax/2;
        y=sin((270-6*i)*PI/180)*ymax/2*0.8 + ymax/2;
        circle(x, y, 2);
    }

    // Главный цикл
    do {
        _dos_getdate(&d);
        god=d.year;
        den=d.day;
        mes=d.month;

        _dos_gettime(&t);
        chs=t.hour;
        min=t.minute;
        sec=t.second;
        ssec=t.hsecond;

        xsec=cos((90-6*sec)*PI/180)*ymax/2*0.75 + xmax/2;
        ysec=sin((270-6*sec)*PI/180)*ymax/2*0.75 + ymax/2;

        xmin=cos((90-6*min)*PI/180)*ymax/2*0.7 + xmax/2;
        ymin=sin((270-6*min)*PI/180)*ymax/2*0.7 + ymax/2;

        xchs=cos((90-(30*chs+6*min/12))*PI/180)*ymax/2*0.5 + xmax/2;
        ychs=sin((270-(30*chs+6*min/12))*PI/180)*ymax/2*0.5 + ymax/2;

        if (sec != osec)
        {
            setcolor(COL_FON);
            sprintf(msg, "%0.2d:%0.2d:%0.2d ", ochs, omin, osec);
            outtextxy(xtime, ytime, msg);
            sprintf(msg, "%0.2d/%0.2d/%0.4d ", oden, omes, ogod);
            outtextxy(xdate, ydate, msg);
            setlinestyle(0, 1, 3); moveto(xmax/2, ymax/2);
            lineto(oxmin, oymin);
            setlinestyle(0, 1, 3); moveto(xmax/2, ymax/2);
            lineto(oxchs, oychs);
            setlinestyle(0, 1, 1); moveto(xmax/2, ymax/2);
            lineto(oxsec, oysec);

            ossec=ssec; osec=sec; omin=min; ochs=chs;
            oden=den; omes=mes; ogod=god;
            oxsec=xsec; oysec=ysec;
            oxmin=xmin; oymin=ymin;
            oxchs=xchs; oychs=ychs;

            settextstyle(0, 0, 2);
            settextjustify(CENTER_TEXT, CENTER_TEXT);

            setcolor(COL_TIME);
            sprintf(msg, "%0.2d:%0.2d:%0.2d ", chs, min, sec);
        }
    } while (1);
}
```



```
outtextxy(xtime,ytime,msg);
```

```
setcolor(COL_DATE);
sprintf(msg,"%0.2d-%0.2d-%0.4d ",den,mes,god);
outtextxy(xdate,ydate,msg);
setcolor(COL_MSTR);
setlinestyle(0,1,3); moveto(xmax/2, ymax/2);
lineto(xmin,ymin);

setcolor(COL_CSTR);
setlinestyle(0,1,3); moveto(xmax/2, ymax/2);
lineto(xchs,ychs);

setcolor(COL_SSTR);
setlinestyle(0,1,1); moveto(xmax/2, ymax/2);
lineto(xsec,ysec);
```

```
moveto(xmax/2, ymax/2);
}
```

```
moveto(x,y);
```

```
if( bioskey(1) != 0 ) // Если нажата клавиша,
    klav=bioskey(0); // читаем код клавиши.
else klav=0; // Иначе - пустой код клавиши
```

```
) while (klav != ESC);
```

```
closegraph();
return 0;
}
```

В. ФЕДОРОВ,

398046, г. Липецк-46, а/я 1341.

Программа для расчета микрорасположенных линий

Предлагаемая программа написана на языке Quick Basic и предназначена для проведения расчетов микрорасположенных линий (МПЛ) СВЧ-диапазона.

После запуска программа требует ввести материал и толщину подложки, а также толщину фольги (покрывающей подложку). Далее вводится требуемое сопротивление. После этого программа выводит график зависимости ширины МПЛ от ее сопротивления.

Далее программа приглашает произвести расчет длины волны в МПЛ с заданным сопротивлением. Для этого нужно ввести рабочую частоту в гигагерцах (10^9 Гц). В результате программа выведет сводку характеристик (геометрическую и эффективную ширину МПЛ, диэлектрическую и эффективную проницаемость подложки, длину волны в воздухе, а также длину волны (полуволны и четвертьволны) в МПЛ).

Программу можно откомпилировать в исполняемый (.EXE) модуль.

```
SCREEN 12
DIM A$(12): DIM E(12): DIM TGA(12): PI = 3.14
DATA ВИНИЛПЛАСТ,3.5,0.06,ГЕТИНАКС,7,0.05, ПОЛИСТИРОЛ,2.5,0.0002
DATA ПОЛИЭТИЛЕН,2.2,0.0003,СЛЮДА,7,0.0004,СТЕКЛОЛИТ,3.5,0.05
DATA ТЕКСТОЛИТ,8,0.07,ТИКОНД,100,0.0006,ФТОРОПЛАСТ-3,3,0.008
DATA ФТОРОПЛАСТ-4,2,0.0002,ЭСКаПОН,2.7,0.0005,ДРУГОЙ,0,0
1 RESTORE: CLS: PRINT: PRINT
FOR I = 1 TO 12
READ A$(I), E(I), TGA(I)
PRINT TAB(11); I; ".": A$(I), E(I), TGA(I)
NEXT I: PRINT
INPUT " ВВЕРИТЕ МАТЕРИАЛ ПОДЛОЖКИ"; I
IF I > 12 OR I < 1 THEN 1
E = E(I): TGA = TGA(I)
IF I <> 12 THEN 2
8 PRINT: INPUT " ВВЕДИТЕ ВЕЛИЧИНУ ДИЭЛЕКТРИЧЕСКОЙ
ПРОНИЦАЕМОСТИ"; E
IF E <= 0 THEN CLS: GOTO 8
9 PRINT: INPUT " ВВЕДИТЕ ВЕЛИЧИНУ УГЛА ПОТЕРЬ"; TGA
IF TGA < 0 THEN CLS: GOTO 9
2 PRINT: INPUT " ВВЕДИТЕ ТОЛЩИНУ ПОДЛОЖКИ (мм)"; H
IF H <= 0 THEN CLS: GOTO 2
11 PRINT: INPUT " ВВЕДИТЕ ТОЛЩИНУ ФОЛЬГИ (мм)"; T
IF T <= 0 THEN CLS: GOTO 11
3 PRINT: INPUT " ВВЕДИТЕ ТРЕБУЕМОЕ СОПРОТИВЛЕНИЕ МПЛ (Ом)"; Z
IF Z <= 0 THEN CLS: GOTO 3
CLS: LINE (10,10)-(10,460), 15
I = 2: B = 1: GOSUB 4
I = 20: B = 2: GOSUB 4
```

```
I = 200: B = 3: GOSUB 4
LINE (10, 460)-(620, 460), 15
I = 4: B = 1: GOSUB 5
I = 40: B = 2: GOSUB 5
I = 400: B = 3: GOSUB 5
LINE (10, 460 - 2 * INT(Z))-(610, 460 - 2 * INT(Z)), 9
W = .025: N = 0: WN = 0
6 Z1 = 377 * H / (SQR(E) * W * (1 + 1.735 * E ^ -.0724 *
(W/H) ^ -.836))
PSET (10 + W * 40, 460 - 2 * INT(Z1)), 15
IF INT(Z1) = INT(Z) THEN N = N + 1: WN = WN + W
W = W + .025
IF W < 15 THEN 6
LOCATE 2, 5: PRINT "1 Ом/дел.": LOCATE 2, 60: PRINT "1 мм/дел."
IF N = 0 THEN LOCATE 3, 25: COLOR 12:
PRINT "ВЫПОЛНЕНИЕ НЕВОЗМОЖНО":
COLOR 15: GOTO 10
W = WN/N
GOSUB 7
LOCATE 30, 2: INPUT "РАСЧЕТ ДЛИНЫ ВОЛНЫ
В МПЛ (Y-ДА/N-НЕТ)"; IS
IF IS = "N" OR IS = "n" THEN 10
12 CLS: PRINT: INPUT " ВВЕДИТЕ РАБОЧУЮ ЧАСТОТУ (ГГц)"; 50
IF F0 <= 0 THEN 12
PRINT: PRINT " ГЕОМЕТРИЧЕСКАЯ ШИРИНА МПЛ "; W; "мм"
W = W + T * (LOG(2 * H/T + 1)) / PI
PRINT " ЭФФЕКТИВНАЯ ШИРИНА МПЛ "; W; "мм"
PRINT " ДИЭЛЕКТРИЧЕСКАЯ ПРОНИЦАЕМОСТЬ ПОДЛОЖКИ"; E
E = (E + 1) / 2 + (E - 1) / (2 * SQR(1 + 10 * H/W))
PRINT " ЭФФЕКТИВНАЯ ПРОНИЦАЕМОСТЬ ПОДЛОЖКИ "; E
L = 30 / F0
PRINT " ДЛИНА ВОЛНЫ В ВОЗДУХЕ "; L; "см"
L = L / SQR(E)
PRINT " ДЛИНА ВОЛНЫ В МПЛ "; L; "см"
PRINT " ПОЛУВОЛНА В МПЛ "; L/2; "см"
PRINT " ЧЕТВЕРТЬВОЛНА В МПЛ "; L/4; "см"
LOCATE 30, 2: INPUT "НОВЫЙ РАСЧЕТ ДЛИНЫ ВОЛНЫ В МПЛ (Y-
ДА/N-НЕТ)"; IS
IF IS = "Y" OR IS = "y" THEN 12
CLS
10 LOCATE 30, 2: INPUT "НОВЫЙ РАСЧЕТ МПЛ (Y-ДА/N-НЕТ)"; IS
IF IS = "Y" OR IS = "y" THEN 1
CLS: END
4 FOR A = 60 TO 460 STEP 1
LINE (10 - B, A)-(10 + B, A), 15
NEXT A
RETURN
5 FOR A = 10 TO 610 STEP 1
LINE (A, 460 - B)-(A, 460 + B), 15
NEXT A
RETURN
7 LINE (10 + W * 40, 460)-(10 + W * 40, 60), 10
RETURN
```

А.СЕДУНОВ,
г.Улан-Удэ.

Буферное устройство для параллельного порта IBM PC

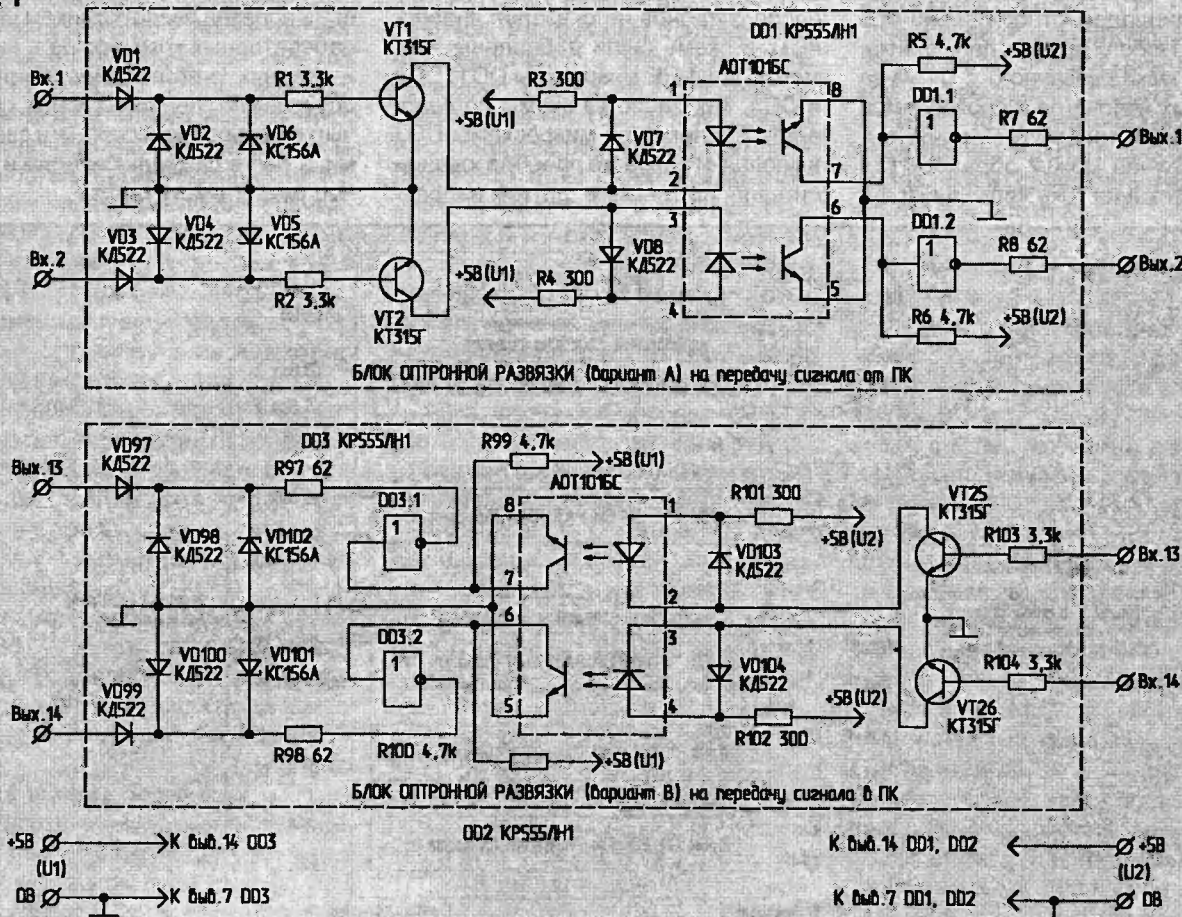
С проблемой защиты своего компьютера наверняка сталкивался каждый. Особенно это актуально, когда вы разрабатываете какое-нибудь устройство, работающее под управлением компьютера либо с его помощью. При проектировании в первую очередь приходится думать о безопасной и безотказной работе самого компьютера. Как правило, для организации управляющих и информационных связей между разработанным устройством и компьютером нужен согласующий модуль, или, другими словами — порт. В персональных компьютерах на сегодняшний момент существует три вида портов — последовательные порты COM1, COM2; параллельный порт принтера LPT, и не так давно появившиеся порты USB. Все их в той или иной мере можно использовать для управления различными устройствами. На мой взгляд, из всех портов наиболее удо-

бен LPT. Простота использования и управления — его достоинства. Наверняка каждый хотел бы максимально обезопасить свой компьютер от различных внештатных ситуаций, которые встречаются во время испытания новых самодельных устройств. Вашему вниманию предлагается принципиальная схема буферного устройства с оптронной развязкой. Данное устройство разрабатывалось для обеспечения максимальной защиты LPT-порта, поэтому на первый взгляд схема может показаться не оптимизированной.

Схема буферного устройства состоит из шести однотипных блоков, разработанных по варианту А, и трех однотипных блоков — по варианту В (рис.1). Схема по варианту А буферизирует выходные шины LPT-порта, а по варианту В — входные (см. таблицу). Для работы буферного устройства необходимо два источника ста-

билизированного напряжения +5 В, гальванически не связанных друг с другом. На схеме они обозначены как U1 и U2. Оба варианта схемы буфера построены с использованием оптрона АОТ101БС. На рис.1 в принципиальной схеме по варианту А диоды VD1, VD2, VD3 и VD4 и стабилитроны VD5 и VD6 обеспечивают защиту от отрицательного напряжения (ниже уровня 0 В, то есть -5 В или -12 В относительно корпуса источника питания ПК), которое может появиться на входе блока по различным причинам (например, на плату упал провод от источника питания). Прямое назначение диодов VD1 и VD4 — пропускать положительный потенциал из LPT-порта, одновременно не пропуская возможный положительный потенциал на вход порта. Например, рассмотрим ситуацию, когда на выводе 1 LPT ПК присутствует напряжение логического "0" (около 0 В со сто-

Рис. 1





Наименование сигнала	Контакт LPT-разъема ПК	Контакт буферного устройства
STROBE	1	Вх.1 (вариант А)
DATA 0	2	Вх.2 (вариант А)
DATA 1	3	Вх.3 (вариант А)
DATA 2	4	Вх.4 (вариант А)
DATA 3	5	Вх.5 (вариант А)
DATA 4	6	Вх.6 (вариант А)
DATA 5	7	Вх.7 (вариант А)
DATA 6	8	Вх.8 (вариант А)
DATA 7	9	Вх.9 (вариант А)
ACK	10	Вых.13 (вариант В)
BUSY	11	Вых.14 (вариант В)
PE	12	Вых.15 (вариант В)
SLCT	13	Вых.16 (вариант В)
AUTO FD	14	Вх.10 (вариант А)
ERROR	15	Вых.17 (вариант В)
INIT	16	Вх.11 (вариант А)
SLCT IN	17	Вх.12 (вариант А)
GROUND	18	к "минусу" U1
GROUND	19	к "минусу" U1
GROUND	20	к "минусу" U1
GROUND	21	к "минусу" U1
GROUND	22	к "минусу" U1
GROUND	23	к "минусу" U1
GROUND	24	к "минусу" U1
GROUND	25	к "минусу" U1

роны ПК). В этот момент происходит замыкание проводника с положительным потенциалом относительно 0 В (U1) или любого другого проводника, гальванически связанного с U1 из-за возможных недочетов в проверяемой схеме, либо по другим причинам. В этом случае может образоваться цепь протекания тока через внутрен-

нюю микросхему LPT-порта. Чтобы исключить такую ситуацию, в схеме буферного устройства установлены диоды VD1 и VD4 на выводах LPT-разъема. Стабилитроны VD5 и VD6 защищают входы порта компьютера от возможного попадания напряжения выше +5 В со стороны буферного устройства. Диоды VD3, VD4 обеспечивают защиту ПК от попадания отрицательного напряжения.

Для надежного срабатывания оптрона в схему были введены транзисторы VT1 и VT2, т.к. изначально оптрон нестабильно срабатывал от импульсов с LPT. Второе назначение транзисторов — максимально разгрузить LPT-порт по току, т.к. при выставлении на выводах LPT уровня логической "1", мощности микросхемы параллельного порта может не хватить для открывания всех оптронов в соответствующих электрических цепочках.

Чтобы сигнал с выходов LPT-порта после оптронов не оказался инверсным, в схему были добавлены элементы ИЛИ-НЕ микросхем DD1, DD2 и DD3. Как видно из схемы по *варианту В*, элементы микросхемы DD3 выполняют такое же преобразование сигнала, но только в другом направ-

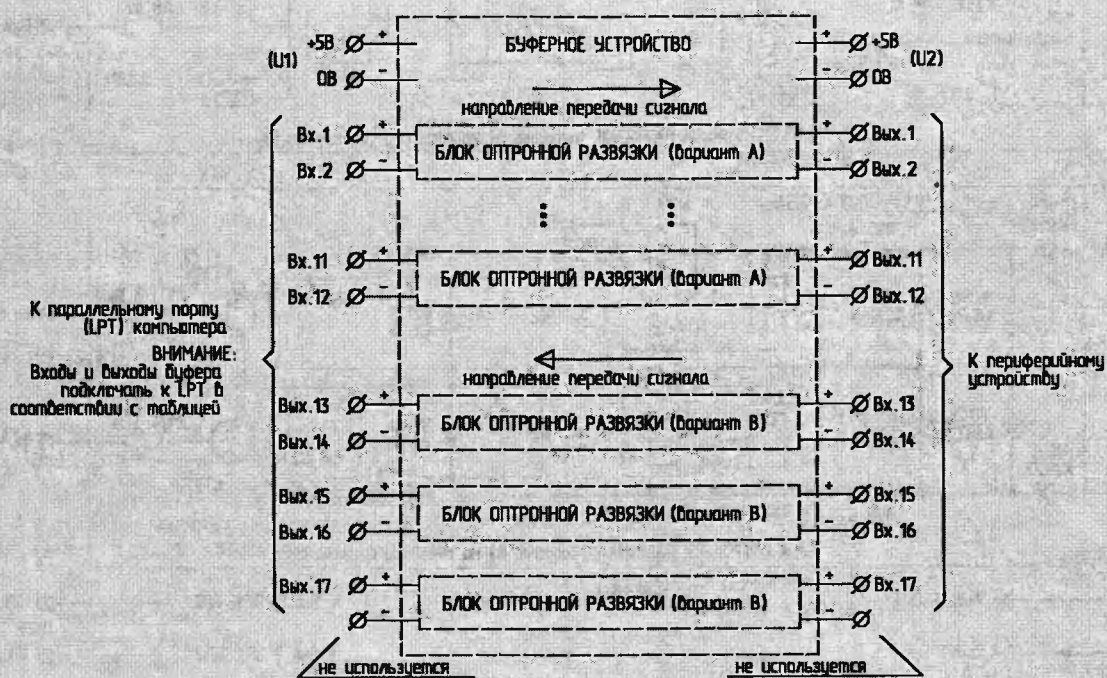
лении. Схема по *варианту В* отличается от схемы по *варианту А* только тем, что оптрон развернут, и поменялись местами транзисторы и логические элементы.

Необходимо отметить тот факт, что буферное устройство не работает в реверсивном режиме. Современные материнские платы позволяют и передавать информацию по шине данных LPT, и принимать, переключив при этом порт LPT на прием. Конструктив данного устройства не позволяет осуществлять прием информации и дальнейший ее перенос на шину данных LPT. Прием информации реализован только для входных управляющих шин LPT-порта.

На **рис.2** представлена блок-схема включения буферного устройства, соответствие сигналов приведено в таблице.

В рассматриваемой схеме применена трехуровневая защита входов LPT-порта на диодах и стабилитронах. Это связано с тем, что данная схема является практически учебным пособием для студентов, и для удобства проведения измерений эксплуатируется в бескорпусном варианте. Материал данной статьи может быть полезен разработчикам учебно-лабораторных комплексов в высших и средних учебных заведениях для обучения по специальностям, связанным с микропроцессорными системами, а также радиолюбителям и инженерам-системотехникам.

Рис. 2





Сказание о “черном” кулере (оптимизация системы охлаждения CPU)

А.ГОРЯЧКИН,

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

С ростом тактовых частот CPU возникает необходимость в эффективном отводе тепла от процессоров. Особенно это актуально для процессоров AMD Athlon, которые, как известно, требуют интенсивного охлаждения. Если CPU нагревается свыше 70°C, это грозит его перегревом. Перегрев процессора может привести к неприятным последствиям — сбоям, зависаниям, и даже к выходу CPU из строя. Поэтому необходимо позаботиться об эффективном охлаждении процессора.

Для охлаждения CPU, как известно, применяют специальное устройство — кулер (cooler — от английского слова cool, охлаждать). Это устройство представляет собой радиатор, принудительно обдуваемый вентилятором для повышения эффективности теплоотвода. Для того чтобы кулер эффективно отводил тепло от процессора, нужно минимизировать тепловое контактное сопротивление между радиатором и процессором. Тепловое контактное сопротивление образуется за счет воздушной прослойки между соприкасающимися поверхностями, возникающей из-за неровностей контактирующих поверхностей и неравномерного их прижима. А воздух, как известно, имеет очень низкую теплопроводность (0,0259 Вт/(м·град) при 20°C).

Снизить величину теплового контактного сопротивления возможно следующими способами:

- уменьшить шероховатость контактирующих поверхностей путем более тщательного шлифования поверхности радиатора, которая прилегает к кристаллу процессора;
- окрасить поверхность радиатора в черный цвет;
- заполнить зазор между соприкасающимися поверхностями процессора и радиатора теплопроводным материалом, с помощью которого произойдет вытеснение воздуха из зазора.

Шлифование радиатора

Шлифованием называют механический процесс обработки для получения ровной и гладкой поверхности. При помощи абразивных материалов с поверхности снимают тонкую стружку.

Для шлифования понадобится наждачная бумага с различной зернистостью, от грубой до самой мелкой. Техника шлифования довольно проста, но потребует немало времени и усердия. Радиатор шлифуют круговыми движениями по наждачной бумаге, которая удерживается на ровной и твердой поверхности, например, на стекле. Разумеется, шлифовать надо только одну плоскость радиатора — ту, которой он соприкасается с процессором.

Перед началом шлифования с кулера снимают вентилятор, чтобы было удобнее шлифовать радиатор. Шлифование поверхности радиатора производят до тех пор, пока она не станет равномерно матово-зеркальной. Шлифовать кристалл процессора не рекомендуется из-за возможного его повреждения.

Анодное оксидирование

Для того чтобы радиатор более эффективно поглощал тепловую энергию, излучаемую процессором, его окрашивают в черный цвет. Это позволяет уменьшить тепловое сопротивление на 15...20% по сравнению с неокрашенным радиатором. Для того чтобы слой краски имел прочное сцепление с поверхностью радиатора, изготовленного из алюминия и его сплавов, его перед покраской подвергают анодированию. Методом электрохимического оксидирования (анодирования) алюминия на поверхности радиатора получают оксидную пленку. Ее используют в качестве грунта для окрашивания поверхности органическими красителями, например, анилиновыми.

Анодирование алюминия проводят в специально оборудованном помещении. Для этого потребуется сернокислый электролит, в качестве которого применяют 15...20-процентный раствор серной кислоты. Концентрация серной кислоты в растворе составляет 180...200 г/л. Анодирование ведут постоянным током. Температура электролита должна поддерживаться в пределах 15...23°C. Процесс ведется при анодной плотности тока 1,5...2 А/дм² для алюминия (напряжение — 2...13 В) и 0,8...1,2 А/дм² для дюралюминия (напряжение —

13...28 В). Продолжительность оксидирования составляет 40...50 минут. После анодирования необходимо тщательно промыть радиатор в проточной воде для полного удаления следов электролита.

Окрашивание радиатора

Для окрашивания радиатора приготавливают красильный раствор. В горячей воде растворяют черный анилиновый краситель до концентрации 5...10 г/л. Полученный раствор кипятят в течение 10...15 минут и затем фильтруют. После фильтрации этот раствор доводят до температуры 60...80°C и на 15...30 минут погружают в него радиатор для окрашивания. После окрашивания радиатор промывают в горячей воде и сушат при температуре 100...150°C. В результате окрашивания алюминия вся поверхность радиатора станет абсолютно черной, что увеличивает эффективность теплоотвода.

Чернение радиатора можно проделывать и перед шлифованием, но в таком случае радиатор будет частично отражать тепловую энергию, излучаемую процессором. Подробно об анодировании и окрашивании алюминия можно прочитать в [1].

Применение теплопроводной пасты КПТ-8

В качестве материала для заполнения зазоров между кристаллом процессора и радиатором вместо той пасты, которая поставляется в комплекте с кулером, можно рекомендовать использовать теплопроводную пасту КПТ-8. Это один из первых теплопроводящих материалов отечественного производства, получивших широкое распространение.

Прежде чем использовать пасту КПТ-8, необходимо знать о ее свойствах, характеристиках и области применения. КПТ-8 — это теплопроводная кремнийорганическая паста. Она применяется в качестве состава, обеспечивающего эффективный тепловой контакт между двумя соприкасающимися или сближенными поверхностями в аппаратуре и оборудовании различного назначения. Применение теплопроводной пасты



КПТ-8 значительно (в 2...3 раза) уменьшает контактное тепловое сопротивление. Эта паста, в отличие от тех материалов, которыми комплектуются кулеры, более плотная, хорошо прилипает и мажется, обладает достаточной вязкостью и сохраняет свою форму. Кроме того, паста КПТ-8 не вытекает из контактного зазора в течение всего времени ее использования. Рекомендуемый для применения интервал рабочих температур — от -60°C до $+180^{\circ}\text{C}$.

Внешне паста КПТ-8 представляет собой однородную теплостойкую массу белого цвета без механических примесей. Она химически инертна, взрывобезопасна, негорюча, не оказывает раздражающего и общетоксического действия на организм. Для приготовления пасты применяют следующие наполнители: аэросил марки А-380 и цинковые белила марок БЦ0 и БЦ1. Плотность КПТ-8 при 20°C — $2,6...3,0 \text{ г/см}^3$. Коэффициент теплопроводности пасты при различных температурах указан в таблице.

Температура	Коэффициент теплопроводности
20°C	$0,7 \text{ Вт/(м}\cdot\text{град)}$
-50°C	не менее $1,0 \text{ Вт/(м}\cdot\text{град)}$
100°C	не менее $0,65 \text{ Вт/(м}\cdot\text{град)}$

Предвосхищая возможные вопросы о том, как достать данный состав, привожу доступные мне сведения.

Паста теплопроводная КПТ-8 ГОСТ 19783-74. Приобрести ее можно в фирменных магазинах "Промэлектроника" (www.promelec.ru). Адрес магазина: г. Москва, ул. Земляной вал, д. 34, метро "Курская", тел. (095) 916-23-21, E-mail: promtech@dol.ru. Стоимость упаковки 19 г — 19 р. 50 к. О производителе пасты информации мне, к сожалению, получить не удалось.

Кроме пасты КПТ-8, можно приме-

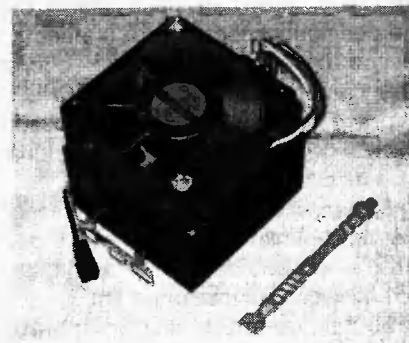
нять и другие теплопроводные пасты. При выборе этих материалов необходимо учитывать, что чем выше его коэффициент теплопроводности, тем лучше будет происходить отвод тепла от процессора. Не рекомендуется вместо пасты использовать всякого рода самоклеющиеся прокладки. Известны случаи, когда при их применении радиатор накрепко "прикипал" к процессору, и снять радиатор с процессора становилось затруднительно.

Установка кулера

После того как будут выполнены шлифование и чернение радиатора, процессор устанавливают в разъем на материнской плате и наносят на кристалл процессора теплопроводную пасту. Пасту нужно наносить тонким равномерным слоем при помощи шпателя. Шпатель можно изготовить самостоятельно из пластмассовой вилки, откусив зубья и сточив напильником край в виде лопатки. Затем закрепляют вентилятор на радиаторе и устанавливают кулер на процессор. При установке кулер слегка поворачивают в разные стороны для лучшего распределения пасты в зазоре, после чего его окончательно фиксируют. Не стоит при установке кулера применять большое усилие, т.к. кристалл процессора очень хрупок, и его можно легко повредить при большом усилии. Это имеет отношение, прежде всего, к процессорам AMD Duron и Athlon.

Далее производят включение компьютера, и после загрузки операционной системы периодически, через определенные промежутки времени измеряют температуру процессора. Отслеживают температурный режим CPU с помощью специальных программ (разумеется, при наличии на материнской плате средств аппа-

ратного мониторинга), например, Motherboard Monitor, утилитами SIV (System Information Viewer) или Easy Tune 4. SIV и Easy Tune 4 — фирменные утилиты производителя материнских плат Gigabyte. На рисунке показан внешний вид кулера с зачерненным радиатором. Габаритные размеры радиатора — $60 \times 60 \times 35 \text{ мм}$, вентилятора — $60 \times 60 \times 20 \text{ мм}$. Рядом с кулером на рисунке расположен шпатель для нанесения теплопроводной пасты.



Заключение

Использование вышеописанной методики снижения теплового сопротивления может оказаться вполне целесообразным. Если в результате применения "черного" кулера и не произойдет существенного снижения температуры процессора, то, во всяком случае, это позволит избежать лишнего его перегрева, улучшит эффективность охлаждения CPU, увеличит "потолок" при его "разгоне". Кроме того, чернение радиатора защищает металл от коррозии, повышает его механическую прочность и придает ему декоративный внешний вид.

Литература

1. Вайнер Я.В., Дасоян М.А. Технология электрохимических покрытий. — Ленинград, Машиностроение, 1972.

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

Ремонт мышки

При длительном использовании мышь чаще всего перестает работать из-за обрыва проводников в кабеле, соединяющем мышь с компьютером. Место вероятного обрыва находится сразу у выхода кабеля из корпуса мышки. Для восстановления работы мышки можно обрезать ка-

бель до места обрыва. Разбираем корпус мышки, выпаиваем ответную часть разъема. Разделяем кабель, залуживаем кончики проводов и впаиваем их в печатную плату мышки вместо разъема. Чтобы не перепутать провода, перед этим запишем цвета проводов и их расположение в разъеме.

Иногда курсор мышки перестает передвигаться по одной из координат. Эту неисправность обычно легко устранить небольшим подгибанием корпусов оптопары по этой координате навстречу

друг другу, но так, чтобы они не задевали за координатный диск. Если курсор по одной из координат не двигается, но заметно небольшое подергивание в нужном направлении, значит, вышла из строя оптопара или один из ее элементов по этой координате. Их можно взять из другой неработающей мышки, где эти элементы исправны.

Такими способами мне удалось отремонтировать несколько мышек в компьютерном классе школы.



ЯПОНСКИЕ КРОССВОРДЫ

Введение

Японские кроссворды (их еще называют "японские сканворды", "японские картинки") — сравнительно недавно появившийся вид головоломок, стремительно набирающий популярность. В этой статье я хочу предложить вашему вниманию программу, которая поможет как решать такие кроссворды, так и создавать их.

Пример кроссворда и его решение

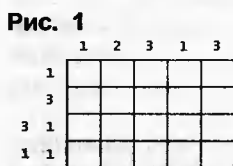
Если вы до сих пор не знали, что собой представляют японские кроссворды, то сейчас узнаете :-). А если знаете, то все равно прочтите этот раздел — здесь я введу термины, которые будут использованы в дальнейших рассуждениях.

В японских кроссвордах нужно отгадывать не слова, а прямоугольное изображение, состоящее из отдельных квадратиков-клеток. Каждая клетка может быть либо белой, либо черной (бывают и многоцветные кроссворды, но здесь я буду рассматривать только черно-белые).

Будем в дальнейшем называть строку или столбец кроссворда общим словом "линия".

Группу идущих подряд черных клеток в линии, ограниченную с каждой стороны либо пустой клеткой, либо границей кроссворда, будем называть "закрашенным участком". Участок может состоять и из единственной клетки.

Так вот, для каждой линии кроссворда приводятся сведения о длине закрашенных участков (в порядке их следования: для строк — слева направо, для столбцов — сверху вниз). По этой информации и нужно разгадать изображение, то есть узнать, какие клетки — белые, а какие — черные.



Это значит, что в этой строке находится одна черная клетка, а остальные — белые. Какое именно положение занимает черная клетка — пока не ясно.

Во второй строке находится участок из трех черных клеток. Понятно, что как его ни располагай, центральная клетка обязательно будет черной.

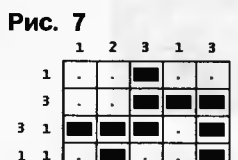
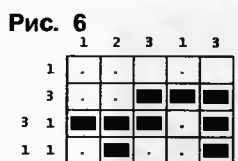
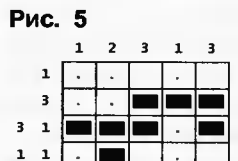
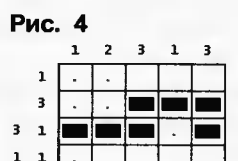
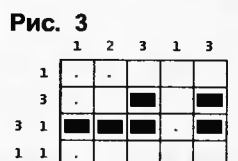
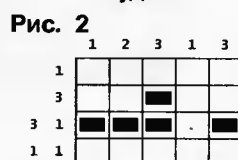
Пример японского кроссворда вы можете увидеть на **рис.1**.

А по какому принципу их решают? Попробуем решить, пока вручную, кроссворд, приведенный на **рис.1**.

Данные о первой строке состоят лишь из одной цифры — "1".

В третьей строке два закрашенных участка, из трех и одной клетки, могут быть расположены единственно возможным образом — так, чтобы между ними находилась одна белая клетка. Итак, об этой строке мы уже знаем все.

Из четвертой строки, как и из первой, информации извлечь не удается.



Отобразим выясненную на данный момент информацию (**рис.2**). Клетки, о которых точно известно, что они белые, будем отмечать точкой.

Теперь перейдем к столбцам. Положение единственной черной клетки в первом столбце нам уже известно. Значит, остальные клетки этого столбца — белые.

Во втором столбце находится черный участок из двух клеток. Положение одной из них нам известно. Вторая клетка обязательно должна примыкать к ней сверху или снизу. Следовательно, самая верхняя клетка этого столбца — точно белая.

Из следующих двух столбцов мы ничего сейчас извлечь не можем, а из последнего столбца ясно, что, как ни располагай в нем участок из трех черных клеток, две средние клетки обязательно будут черными.

Отобразим то, что нам удалось выяснить (**рис.3**).

Снова просмотрим строки кроссворда. Видим, что во второй строке положение закрашенного участка (а значит, и второй белой клетки) полностью определилось. Отмечаем это (**рис.4**).

Тогда во втором и четвертом столбцах положение закрашенных участков тоже однозначно определяется (**рис.5**). Остались лишь четыре невыясненных клетки!

При анализе нижней строки определяется положение второго закрашенного участка (**рис.6**).

Ну а третий и пятый столбцы теперь тоже можно однозначно определить (**рис.7**).

Все, кроссворд решен! Ну а то, что в получившейся картинке не видно какого-либо смысла, — так ведь это был просто пример :-).

Как видим, при решении японского кроссворда мы, получив некото-

И ты не знал...

"Абзац" - новое полиграфическое издание для поклонников компьютера **ZX-Spectrum!**

Наши координаты:

Россия, 160035, г.Вологда, а/я 136

Телефон: (8172) 25-28-71

...что есть газета?



рые сведения о строках, используем их для выяснения информации о столбцах, и наоборот. В этом японские кроссворды похожи на обычные, в которых отгаданные слова по горизонтали дают информацию для отгадывания пересекающихся с ними слов по вертикали, и наоборот.

А теперь рассмотрим, как решать кроссворды (и создавать собственные) уже не вручную, а с использованием моей программы.

Руководство пользователя

После запуска вы увидите на экране меню из пяти пунктов:

1. Resolve new crossword.
2. Resolve old crossword.
3. Test picture.
4. View data.
5. Exit.

Для решения нового кроссворда выбирайте первый пункт. Сначала программа предложит вам ввести размеры кроссворда: ширину (от 1 до 62 клеток) и высоту (от 1 до 42). При необходимости кроссворд можно повернуть на 90 градусов, поменяв при этом информацию о строках и столбцах.

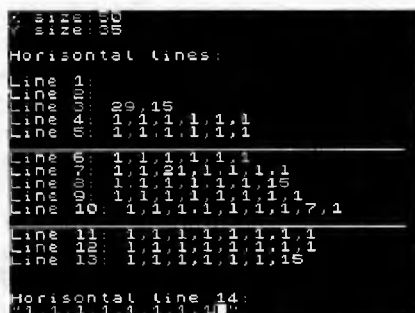
Затем надо ввести информацию о закрашенных участках: сначала о строках (они нумеруются сверху вниз), а потом о столбцах (нумеруются слева направо). Номер текущей вводимой строки (столбца) отображается на экране.

Длины закрашенных участков перечисляются через запятую; если какая-либо линия не содержит закрашенных участков, просто нажимайте Enter.

Уже введенная информация печатается в верхней части экрана, при этом через каждые пять строк выводится разделительная черта — это помогает не перепутать, информацию о какой линии кроссворда вы вводите (в сетке “бумажного” кроссворда через каждые пять клеток разделяющая их линия изображается более жирной).

На **рис.8** показано, как выглядит экран при вводе данных.

Рис. 8



Программа проверяет, являются ли корректными введенные пользователем данные о линии кроссворда. Должно выполняться следующее условие: сумма длин всех закрашенных участков и минимальных промежутков между ними (напомню, что два соседних закрашенных участка должны быть разделены как минимум одной белой клеткой) должна быть не больше длины линии.

Минимальное количество белых клеток, очевидно, на единицу меньше числа закрашенных участков. Тогда данное ограничение можно записать так:

$$\sum_{i=1}^n a_i + n - 1 \leq L$$

где:

- n — количество закрашенных участков;
- a_i — длина i -го закрашенного участка;
- L — длина линии.

То есть, например, если длина линии — 8 клеток, и вы вводите информацию о закрашенных участках: "1,4", то все в порядке — $(1+4)+1 < 8$. А если при вводе клавиша "1" случайно была нажата дважды, и оказалась введена строка "11,4", то программа укажет на некорректность введенных данных — ведь $(11+4)+1 > 8$.

При обнаружении ошибки программа предложит повторно ввести информацию о линии кроссворда.

После ввода всех данных производится еще одна проверка. Смотрите, если сложить длины всех закрашенных участков в строках, то мы, очевидно, получим общее количество черных клеток в кроссворде. Но точно такое же значение должно получиться и при суммировании длин закрашенных участков в столбцах! А если они не совпадают — значит, при вводе допущена ошибка. В этом случае придется вводить данные заново.

При такой проверке всегда обнаруживается одиночная ошибка вида “пропущено число”, “введено лишнее число”, “введено неправильное число”. Да и если таких ошибок несколько, вероятность того, что они взаимно компенсируют друг друга, невелика. А вот если вы при вводе случайно поменяете какие-либо числа местами, это останется незамеченным — ведь от изменения порядка слагаемых сумма не меняется.

Если данные успешно прошли первичную проверку на корректность, программа выдаст приглашение: "Press any key...". После нажатия любой клавиши начнется процесс решения кроссворда.

Информация о ходе решения отображается на экране в таком же виде, как если бы вы решали кроссворд на бумаге. Если клетка кроссворда изображена белой — значит, информация о ней неизвестна; если в клетке стоит точка — значит, эта клетка точно белая; если клетка изображена черной — значит, она черная :-).

Точки, расположенные на левой и верхней сторонах крессворда, служат признаком того, что соответствующие линии крессворда помечены для последующего просмотра. Сначала помечены все линии.

При решении кроссворда программа последовательно просматривает все помеченные строки и столбцы. Точка напротив текущей просматриваемой линии изображается более жирной.

При просмотре линии, возможно, становится известной информация о некоторых ее клетках. Если стало известно, что какая-то клетка черная, она закрашивается черным. Если известно, что какая-то клетка белая, она помечается точкой в центре.

Когда при просмотре строки становится известна информация о некоторых ее клетках, то столбцы, где расположены эти клетки, помечаются. Аналогично, когда при просмотре столбца становится известна информация о некоторых его клетках, то строки, где расположены эти клетки, помечаются.

После просмотра линии точка напротив нее исчезает, т.е. пометка с этой линии снимается.

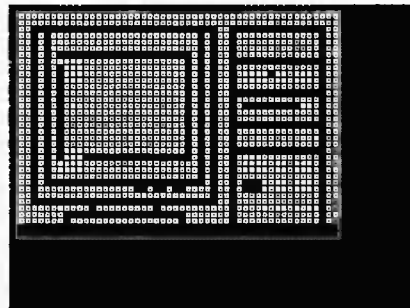
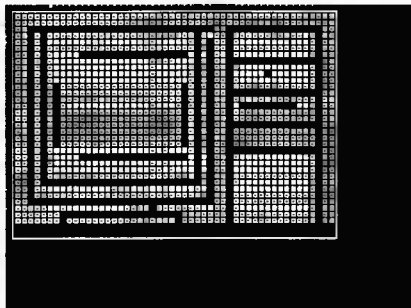
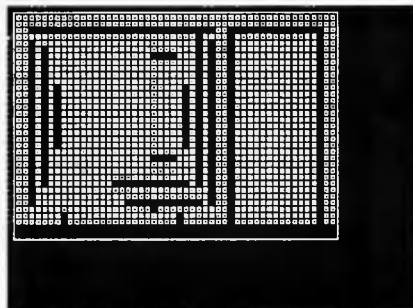
Процесс последовательного просмотра помеченных линий продолжается до тех пор, пока не станет извест-



на информация о всех клетках кроссворда, т.е. пока кроссворд не будет решен.

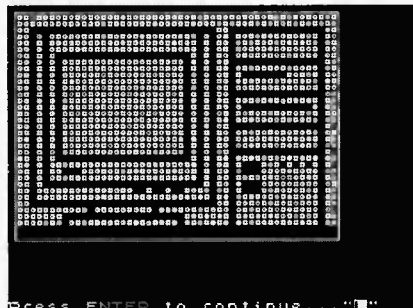
На рис.9 вы можете увидеть несколько скриншотов, сделанных в процессе решения кроссворда.

Рис. 9



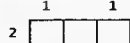
При нормальном завершении вы увидите на экране изображение решенного кроссворда (рис.10), а затем (после нажатия любой клавиши) — время, потраченное на решение. После еще одного нажатия любой клавиши программа выйдет в главное меню.

Рис. 10



Теперь обратим внимание вот на что — даже если данные кроссворда успешно прошли первичную проверку на корректность, этот кроссворд может и не иметь ре-

Рис. 11



шения. Простейший пример такого кроссворда вы можете увидеть на рис.11.

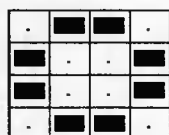
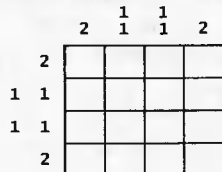
В таком случае при решении кроссворда рано или поздно обнаружится противоречие. Для примера, на рис.11 при анализе первой (и единственной) строки получаем, что средняя клетка будет черной, а при анализе второго столбца — что она должна быть белой.

Программа, обнаружив такое противоречие, выводит сообщение "Incorrect data!", и после нажатия любой клавиши выходит в главное меню.

Также может быть, что кроссворд имеет несколько решений. На рис.12 приведен пример кроссворда с тремя различными решениями.

При решении такого кроссворда рано или поздно наступит момент, когда помеченных линий не останется, а кроссворд еще не будет полностью решен. В этом случае

Рис. 12



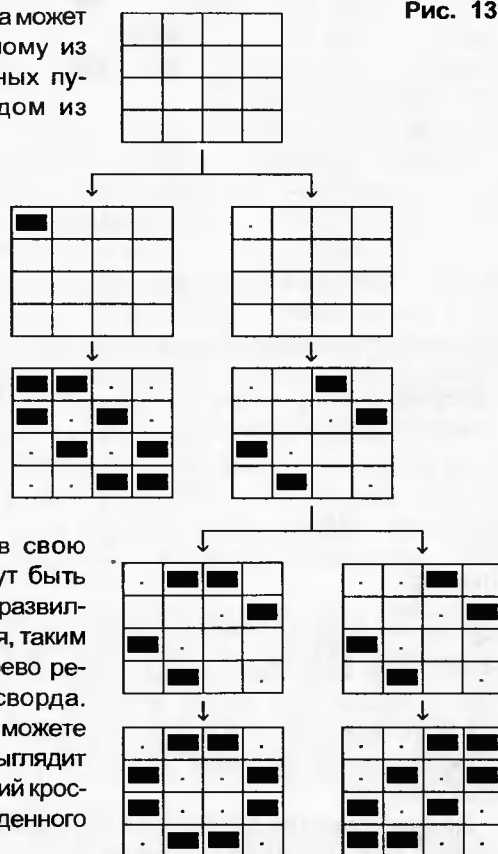
программа выдаст сообщение "Many variants! Continue?" ("Много вариантов. Продолжить?"). Если вы ответите "N" ("Нет") — программа выйдет в главное меню; если "Y" ("Да") — решение кроссворда будет продолжено.

Чтобы программа могла продолжить решение кроссворда, ей необходима дополнительная информация, которую она запросит у вас. Эта информация — цвет первой неизвестной клетки (т.е. такой клетки, цвет которой к настоящему моменту неизвестен; первая такая клетка — та, которая ближе к верхней границе кроссворда, а если таких клеток несколько — та из них, которая ближе к левой границе). Вы можете выбрать, будет ли эта клетка черной или белой, ответив на запрос программы ("(B)lack or (W)hite?") соответственно "B" или "W".

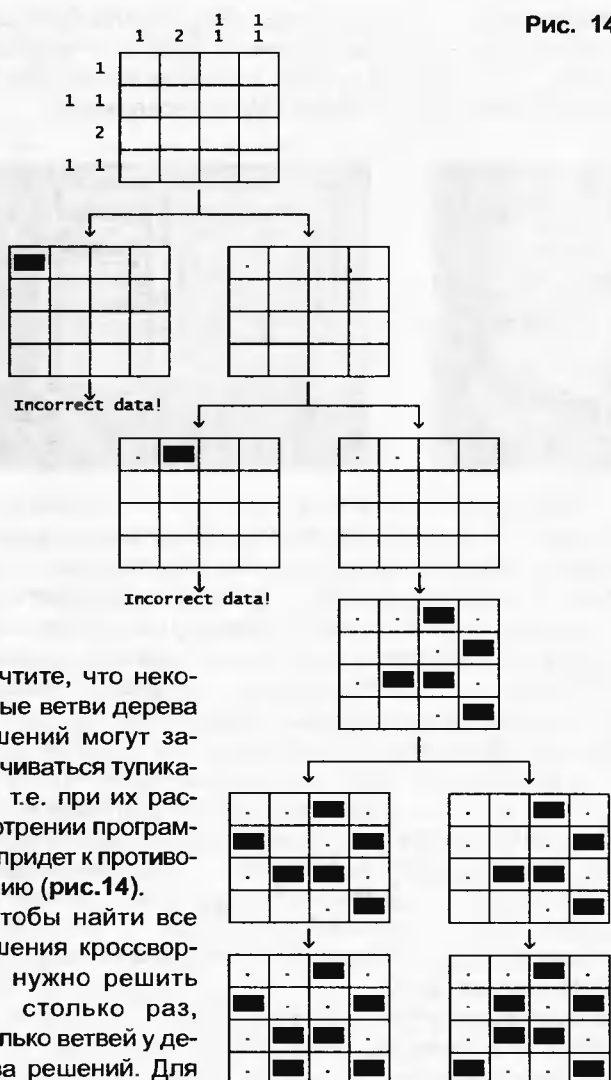
После этого вы увидите, что соответствующая клетка кроссворда примет выбранный вами цвет. Строка и столбец, где находится эта клетка, будут помечены для дальнейшего анализа, и решение кроссворда продолжится.

Момент, когда мы выбираем, черной или белой будет неизвестная клетка, является своеобразной точкой развилки. После него решение кроссворда может пойти по одному из двух возможных путей. На каждом из

Рис. 13



этих путей, в свою очередь, могут быть другие точки развилки. Получается, таким образом, "дерево решений" кроссворда. На рис.13 вы можете увидеть, как выглядит дерево решений кроссворда, приведенного на рис.12.



Учтите, что некоторые ветви дерева решений могут заканчиваться тупиками, т.е. при их рассмотрении программа придет к противоречию (рис.14).

Чтобы найти все решения кроссворда, нужно решить его столько раз, сколько ветвей у дерева решений. Для этого при каждом решении выбирайте цвета неизвестных клеток так, чтобы пройти по новому пути. После получения очередного решения (или если выяснилось, что выбранный путь привел к противоречию), когда программа выйдет в главное меню, выбирайте пункт "Resolve old crossword" ("Решить старый кроссворд"), тогда вам не придется заново вводить информацию о кроссворде.

Теперь о том, как программа поможет вам в составлении собственных кроссвордов. Чтобы составить кроссворд, надо выполнить три шага: нарисовать картинку, получить данные о группах закрашенных участков и проверить, будет ли полученный кроссворд решаться однозначно (иначе тот, кто будет его решать, не узнает, какую именно картинку вы имели в виду). В выполнении всех этих трех шагов вам поможет компьютер.

Рисовать картинку удобно в графическом редакторе. На "ZX-Spectrum" это может быть, например, Art Studio или BGE. Вам нужно получить файл в стандартном экранном формате, причем картинка должна быть расположена в левом верхнем углу экрана. Каждый пиксел картинки соответствует одной клетке кроссворда.

Для выполнения следующих двух шагов как раз поможет моя программа. Выберите в главном меню

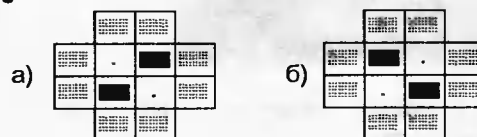
Рис. 14

пункт "Test picture" и укажите имя файла с картинкой. Программа загрузит картинку, и вы увидите ее на экране. Укажите размеры кроссворда. Программа начнет считывать информацию о пикселах картинки (т.е. о клетках кроссворда), при этом обработанные пиксели будут инвертироваться, чтобы вы могли следить за ходом процесса. Затем программа начнет формировать информацию о закрашенных участках, при этом показывая, сколько процентов информации уже сформировано.

Далее программа начнет решать кроссворд по этой информации. Если решение однозначно, то после выхода в главное меню вы, выбрав пункт "View data", сможете переписать данные о закрашенных участках на бумажку :-). Ну а если в процессе решения возникла неоднозначность, значит, картинку надо переделать. В таком случае загружайте графический редактор, и все сначала...

Небольшой совет — если ваша картинка содержит участки как на рис.15а, 15б (где клетки, помеченные "■" — это либо белые клетки, либо граница кроссворда), то решение заведомо будет неоднозначным. Действительно, если в картинке содержится фрагмент как на рис.15а, то, если заменить его на фрагмент, приведенный на рис.15б (или наоборот), то данные о закрашенных участках при этом не изменятся! Т.е. у такого кроссворда будет как минимум два различных решения. Соответственно, следите за тем, чтобы таких участков в вашей картинке не было.

Рис. 15



Тем не менее, даже если таких участков нет, это еще ни о чем не говорит (на рис.12 их нет, а решений несколько). Так что без проверки на однозначность решения все равно не обойтись.

Текст программы

Программа написана на Бейсике для "ZX-Spectrum", но ее должно быть достаточно легко адаптировать и для другого компьютера.

Исходный листинг программы не приводится из-за его значительного объема. Он доступен для скачивания на сайте журнала (<http://radio-mir.com>) либо может быть получен из редакции по предварительному запросу. На сайте также можно скачать уже откомпилированную программу: в архиве jchobeta.zip находятся файлы программы для реального "ZX-Spectrum" (в формате hobeta); в архиве jc_trd.zip находится trd-файл с образом диска для эмуляторов (на этом образе программа записана под именем "boot", чтобы она запускалась автоматически).

Для значительного ускорения работы программы (что особенно важно при решении больших кроссвордов) имеет смысл переписать на ассемблере ключевой участок — обработку одной линии кроссворда (строки 2200...3610). Для "ZX-Spectrum" я так и сделал. Ниже



приведен дампы соответствующей ассемблерной подпрограммы.

```
#F600: 1190 F6CD 9DFB 2323 7E32 B5F6 CD9D FB23
#F610: 237E F532 CCF6 CD9D FB23 237E F532 C0F6
#F620: CD9D FB01 0500 0911 01FD F14F EDB0 11AA
#F630: F6CD 9DFB 0105 0009 F147 1101 FF23 237E
#F640: 1223 1323 2310 F6CD B4F6 F511 A8F6 CD9D
#F650: FB01 0500 09EB 2101 FD3A C0F6 4FED B011
#F660: ACF6 CD9D FB20 242A 595C 2BE5 010C 00CD
#F670: 5516 D121 ACF6 0107 00ED B0EB 3600 2336
#F680: 0023 F177 2336 0023 3600 C9F1 2323 77C9
#F690: A669 7273 F400 A461 7461 6E6C 696E E500
#F6A0: AC65 6E6C 696E E500 C700 8800 A572 726C
#F6B0: 696E E500 C700 A728 673A CCF6 A720 C0C6
#F6C0: 0021 01FD 3630 2C10 FBAF C906 0021 01FF
#F6D0: AF86 2C10 FC47 3AC0 F690 21CC F696 3C5F
#F6E0: 1601 3ACC F647 2101 FF7B BE30 2A7A 8632
#F6F0: FDF6 E526 FD7A 836F 3631 2C7D 3600 20F8
#F700: E11C 1D20 12E5 7D21 CCF6 BEE1 3009 7E82
#F710: E526 FD6F 3630 E17A 863C 572C 10CB AFC9
#F720: 3ACC F6A7 2014 2101 FD3A C0F6 477E FE31
#F730: 3E01 C836 302C 10F5 AFC9 AF32 9CF9 3CDD
#F740: 6721 01FC 3AC0 F647 3620 2C10 FBDD 7C47
#F750: FE02 3E01 3809 0521 01FF 863C 2C10 FB32
#F760: D7F7 DD7C 21CC F6BE 3AC0 F630 127E DD94
#F770: 4726 FFDD 7C3C 6F3A C0F6 963D 2C10 FB32
#F780: A8F7 DD7C 3D28 273A D7F7 D602 DD6F CDBC
#F790: F9A7 2009 DD7D C602 32D7 F718 11DD 2C26
#F7A0: FFDD 7C6F 7EDD 85FE 003E 01D0 18E0 3ACC
#F7B0: F6DD BC28 283A A8F7 C602 DD6F CD67 FAA7
#F7C0: 2009 DD7D D602 32A8 F718 12DD 2D26 FFDD
#F7D0: 7C6F DD7D 3D96 FE00 3E01 D818 FD3A D7F7
#F7E0: 3222 F83E 0132 60F8 327C F826 FFDD 7C6F
#F7F0: 3A22 F886 3D21 A8F7 BE28 163C 6F26 FD3A
#F800: A8F7 953C 477E FE31 2004 AF32 7CF8 2C10
#F810: F43A CCF6 DDBC 2809 7EFE 3120 04AF 327C
#F820: F83E 00FE 0238 0A3D 6F26 FD7E FE31 CA63
#F830: F926 FFDD 7C6F 3A22 F886 473A C0F6 B838
#F840: 0926 FD68 7EFE 31CA 63F9 26FF DD7C 6F46
#F850: 26FD 3A22 F86F 7EFE 30CA 63F9 2C10 F73E
#F860: 00A7 2017 3A22 F8D6 02DD 6F38 0928 07CD
#F870: BCF9 A7C2 63F9 3E01 3260 F83E 00A7 2021
#F880: 26FF DD7C 6F7E 3C21 22F8 86DD 6F21 C0F6
#F890: BE28 0230 07CD 67FA A7C2 63F9 3E01 327C
#F8A0: F826 FFDD 7C6F 7E3D 473A C0F6 2122 F896
#F8B0: 904F 3ACC F647 DD7C 9030 12ED 4447 26FF
#F8C0: DD7C 3C6F 7996 2C25 9624 10F9 4F26 FEED
#F8D0: 7C6F 2C71 0C0D 200F 3ACC F647 DD7C 9030
#F8E0: 06AF 327C F818 993A 22F8 3D4F DD7C 3D28
#F8F0: 0D47 2101 FF79 9625 962C 2410 F94F 26FE
#F900: DD7C 6F71 219C F934 26FE DD7C 6FE5 7EA7
#F910: 281A 473A 22F8 906F 26FC 7EFE 2020 0436
#F920: 3018 06FE 3028 0236 3F2C 10EE 16FF DD5C
```

```
#F930: 1A47 26FC 3A22 F86F 7EFE 2020 0436 3118
#F940: 06FE 3128 0236 3F2C 10EE D113 1AA7 2813
#F950: 477E FE20 2004 3630 1806 FE30 2802 363F
#F960: 2C10 EE21 22F8 3446 26FF DD7C 6F7E 3D80
#F970: 473A A8F7 B838 2426 FD3A 22F8 3D6F 7ED6
#F980: 3120 0332 60F8 E526 FFDD 7C6F 7EE1 856F
#F990: 7ED6 3120 0332 7CF8 C321 F83E 00A7 3E01
#F9A0: C83A CCF6 DDBC 2805 DD24 C34D F7ED 4BC0
#F9B0: F6AF 4721 01FC 1101 FEED B0C9 DD7C FE02
#F9C0: 3010 DD45 2101 FD7E FE31 3E01 C82C 10F7
#F9D0: AFC9 2101 FE36 00DD 7CFE 0338 072C 3601
#F9E0: 2CBD 20FA 26FE DD7C 6FE5 3D47 2E01 DD7D
#F9F0: 9624 9625 2C10 F9E1 7721 01FD 3A01 FEAF
#FA00: 280A 477E FE31 3E01 C82C 10F7 1601 E526
#FA10: FF6A 46E1 7EFE 3028 1E2C 10F8 0601 7EFE
#FA20: 3128 142C 04E5 26FE 6A2C 7EB8 E130 EF14
#FA30: DD7C BA20 D9AF C95A DD7C 9347 26FE 6B2C
#FA40: AF86 2C10 FC47 DD7C 933D B830 146B 34DD
#FA50: 7C3D 3D93 3809 3C47 6B2C 3601 2C10 FB18
#FA60: 831D 20D4 3E01 C9DD 7C21 CCF6 BE38 173A
#FA70: C0F6 DD95 3C47 26FD DD7D 6F7E FE31 3E01
#FA80: C82C 10F7 AFC9 26FE 3ACC F66F E52C 3600
#FA90: E1D6 02DD 9438 083C 473E 0177 2D10 FC3A
#FAA0: CCF6 6F2C 5F16 FFDD 9447 3AC0 F6DD 953C
#FAB0: 96EB 96EB 2D1D 10F8 47DD 7C3C 6F70 3AC0
#FAC0: F632 D2FA 26FE 3ACC F63C 6F7E A728 1F26
#FAD0: FD2E 007E FE31 3E01 C821 D2FA 353A C0F6
#FAE0: 963C 4726 FE3A CCF6 3C6F 7EB8 30E1 3ACC
#FAF0: F65F 1601 26FD 3AD2 FA6F 7EFE 3028 3421
#FB00: D2FA 3514 26FF 6B7E BA30 E916 0126 FE6B
#FB10: 7EA7 2817 26FD 3AD2 FA6F 7EFE 3128 1421
#FB20: D2FA 3514 26FE 6B7E BA30 E91D DD7C BB38
#FB30: C1AF C97B 3238 FB3E 006F DD94 4726 FEAF
#FB40: 862D 10FC 473A 38FB DD94 3DB8 3042 26FE
#FB50: 3A38 FB3C 6F34 DD7C 3C21 38FB BE30 0F3A
#FB60: 38FB 6FDD 943D 4726 FE36 012D 10FB 3ACC
#FB70: F66F DD94 473A C0F6 DD95 3C26 FF96 2C25
#FB80: 962D 2D24 10F7 4725 DD7C 3C6F 70C3 BEFA
#FB90: 2138 FB34 3ACC F6BE 309D 3E01 C92A 4B5C
#FBA0: C5D5 E51A 13A7 2833 BE23 28F7 E1D1 7E23
#FBB0: FE80 282A FE60 381C FE80 0105 0038 19FE
#FBC0: A038 11FE C030 077E 2317 30FB 18EC FEE0
#FBD0: 0E12 3004 4E23 4623 0918 C6C1 C13C C1C9
```

После набора дампа проверьте его правильность — сумма всех байтов, взятая по модулю 65536, должна равняться 45603 (#B223). Запишите соответствующий участок памяти в файл obj <C> (start #F600, length #05E0).

В программе строки 2200...3610 теперь не нужны, а следует добавить такие строки:

```
5 CLEAR 62975: RANDOMIZE USR 15619: REM: LOAD "obj" CODE
2195 RANDOMIZE USR 62976: RETURN
```

“Тормоз” для винчестера (счастливым обладателям “винтов” посвящается)

“Если говорить о каком-то наращивании, то в первую очередь — винт, т.к. он дает самый серьезный выигрыш по любым характеристикам, сравнительно с имеющимися.”

“Очень советую поставить винчестер. Сам уже полтора года с винтом, и уже не представляю себе машины без него. Хотя, говорят, первые PC тоже бывали без винтов.”

“... и полностью разделяю Ваше мнение по поводу работы с “винтом”. Надо было столько лет “юзать” дисковод, чтобы это оценить.”

Все это — “цитаты из классики”, т.е. “надерганные” из информационных сборников “Open Letters” мнения по поводу использования винчестеров с интерфейсом IDE-DRIVE в системе iS-DOS.

Довольно долго они вызывали у меня лишь улыбку. До

Е.ИЛЯСОВ,
Саратовская обл., г.Балашов.



тех пор, пока... Пока архив на дискетах не разбух настолько, что стал серьезно тормозить дела. И вот настало время, когда мне не остается ничего другого, как только присоединить свой голос к вышеприведенным восторгам.

Но у каждой медали две стороны. Есть у Sinclair-совместимого компьютера с винчестером некоторая особенность, не то чтобы очень уж неприятная, но, скажем так — заставляющая время от времени хмурить брови.

Речь в данном случае идет о том, что магнитные диски винчестера, в отличие от своих флорпи-«родственников», вращаются безостановочно, пока включен компьютер. Во время работы в операционной системе iS-DOS это существенно ускоряет операции обмена данными с накопителем и в несколько раз повышает общую производительность системы. А вот когда приходится использовать программы под TR-DOS (любимый ассемблер, любимый графический редактор, любимый архиватор или же любимый гейм-блокбастер), тогда этот же самый фактор способствует лишь бесполезной выработке хотя и значительного, но все же конечного ресурса работы винчестера.

Есть, конечно, способы «заткнуть» движок накопителя. Можно, например, использовать драйвер винчестера под iS-DOS, содержащий функцию «standby» («ide+3sby.blk» или «ide+5sby.blk»), которая «перекрывает кислород» жесткому диску по истечении заданного времени (от одной секунды до, примерно, шестнадцати с половиной минут). Или, если TR-DOS'ная программа была загружена в RAM-диск (это, естественно, только для KAY'ев) утилитой «trd_ram.com», включить в стартовом меню утилиты флажок «Standby Y». По разным причинам нельзя назвать идеальным выходом из положения ни один из этих способов, и тот, кто имеет некоторый опыт работы с винчестером в iS-DOS, хорошо об этом знает.

Здесь предлагается альтернативный вариант «тормоза» — короткая программка, которая также «глушит» винчестер до очередного обращения к нему, но вызывается для этого из-под TR-DOS. Исходным материалом для этой программы послужила iS-DOS'ная (все-таки) утилита «standby.com» (IskraSoft, 1998 г.), практически без изменений перенесенная в TR-DOS и снабженная для загрузки машиннокодированного блока и его устойчивой отработки простейшим интерфейсом на Бейсике.

Варианты использования этой программы могут быть самыми разными — «на вкус и цвет подруги нет». Например, «автопилотный» — помещаем ее на диск под именем «boot», а в Бейсик-загрузчик дописываем строку, которая после свершения программой своего «тормозного» дела запускает «настоящий» boot — естественно, уже имеющий к этому времени другое имя. Если такой «автостоп» будет находиться в начале дискеты, то в этом случае вы даже не заметите его присутствия. Или, скажем, просто добавляем эту скромную утилиту на свой диск с TR-DOS'ными программами, и в самом начале общения с диском (или просто при необходимости) выбираем ее для исполнения.

Именно такой вариант, не требующий дополнительных усилий для реорганизации файловой структуры, и реализован в приведенном листинге Бейсик-загрузчика:

«Standby»

```
10 CLEAR 29999
20 RANDOMIZE USR 15619: REM :
   LOAD "standby" CODE 3e4
30 RANDOMIZE USR 3e4
40 REM CLEAR 65367
50 RANDOMIZE USR 15619: REM :
   RUN "boot"
```

Восстановление стандартного адреса RAMTOP в строке с номером 40 может быть полезным, если заведомо предполагается работа компьютера в режиме 48 Кб (например, «старые» игрушки, адаптированные для загрузки с диска), поскольку сама TR-DOS не очень-то корректно себя ведет с областью памяти UDG, переустанавливая ее адрес по своему собственному усмотрению (точнее, ликвидируя совсем).

Шестнадцатеричная распечатка кодов программы, или так называемый, «дамп»:

«standby» <C>

```
#5DC0 F3 DB F0 07 38 FB 3E A0
#5DC8 D3 D0 3E E0 D3 F0 FB AF
#5DD0 C9
```

address: 24000, size: 17

Загрузка «не в свои» адреса введена потому, что вполне обычный для iS-DOS адрес загрузки 24000 слишком низок для использования в рамках системы TR-DOS. Впрочем, ничто не мешает перетранслировать процедуру «затыкания» портов винчестера, благодаря ее релоцируемости, на какой-нибудь другой, более высокий адрес. Ассемблерный листинг в формате iS-DOS Assembler для этой цели также приводится:

```
ORG    #5DC0 ;start address: 24000
DI
L5DC1 IN  A, (#F0)
RLCA
JR      C, L5DC1
LD      A, #A0
OUT     (#D0), A      ;кидаются значения...
LD      A, #E0
OUT     (#F0), A      ;... в порты винта.
EI
XOR                                ;очистить аккумулятор
RET                                ;и с вещами на выход.
```

Удачного торможения!

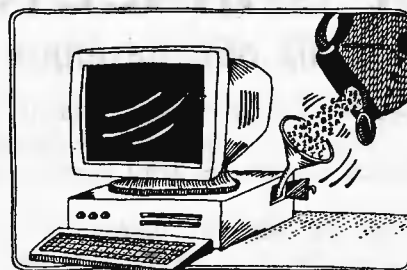


Рисунок О.Попова



М.ВАЛЬПА,
г.Миасс.

Star Wars Jedi Knight 2: Jedi Outcast

Наконец-то Lucasarts произвела на свет долгожданное продолжение некогда хитового экшена "Star Wars Jedi Knight" и окрестила его "Jedi Outcast", т.е. отшельник. Название вполне подходящее, потому что вам действительно предстоит играть за своеобразного отшельника, а если быть более точным — за Джедаю, который отказался от своей Силы (потому что чуть не стал темным джедаем) и стал обычным наемником.

Но не тут-то было. Несколько первых (вполне "безобидных") миссий выводят вас на некоего темного джедая по имени Дессан, который, как вы слышали, убил вашу напарницу Джен и улетел на космическом корабле. Вы в бессильной ярости делаете опрометчивый поступок — летите в долину джедаев, чтобы принять некогда отверженную вами Силу, и тем самым показываете Дессану туда дорогу. Когда вы улетаете в тренировочный лагерь джедаев за своим мечом, Дессан спускается в долину, захватывает ее и с помощью специальных кристаллов начинает штамповать из своих подданных новых темных джедаев. Вернувшись с базы свежим, полным Силы и вооруженным лазерным мечом, вы начинаете последовательно выполнять миссии, основная часть которых заключается в том, чтобы вы нашли космический корабль

Дессана и, соответственно, его самого. Да, кстати, после того как вы заново приняли Силу, в начале каждой новой миссии ее мощь воз-

какой — во вторую. Всего вам будет доступно одиннадцать типов.

Толчок — используется для отталкивания людей, предметов (например, летящих в вас ракет) и для нажатия кнопок.

Перемещение — притягивает предметы и людей, а также притягивает или выбивает оружие из рук противника.

Прыжок — позволяет Джедаю выше прыгать. Чем больше мощь этой Силы, тем выше прыгает Джедай.

Хитрость — действует на врагов, заставляя их не замечать вас, пока вы не начнете атаковать их (не действует на джедаев и роботов).

Скорость — позволяет ускоряться вам и замедляет врагов.

Захват — поднимает и душит вашего врага (не действует на джедаев и роботов).

Молния — позволяет выпустить электрический заряд, способный поразить группу врагов.

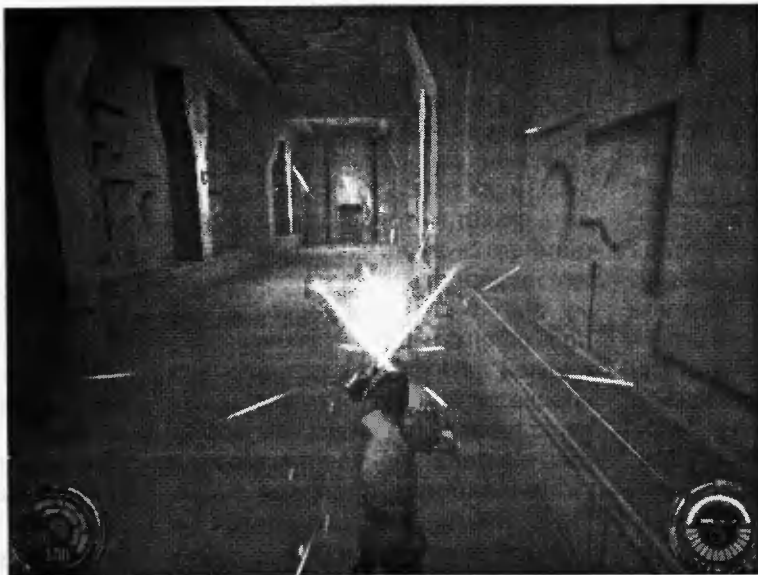
Исцеление — восстанавливает ваше здоровье.

Атака мечом — позволяет более профессионально работать мечом и быстрее наносить удары (как ни странно, тоже относится к Силе).

Защита мечом — влияет на скорость и точность отражения лучей лазеров и ударов лазерных мечей.

Бросок мечом — помимо того, что мечом можно защищаться и атаковать, его еще можно и бросать, при этом он, как бумеранг, возвращается обратно в руку. Уровень этой силы влия-

растает. Всего имеется три ступени мощи Силы: первая — самая слабая, третья — соответственно, самая сильная; правда, жалко, что





ет на дальность полета меча.

Кроме того, существует три стиля обращения с мечом.

Первый — Джедай быстро наносит удары, но они слабые, а также он плохо блокирует удары и плохо отражает лазеры. Этот стиль лучше всего использовать против большого количества врагов.

Второй — сила и скорость ударов, а также защита мечом на среднем уровне. Это самый оптимальный стиль, по ходу игры лучше всего использовать именно его.

Третий — медленные удары, но отличная защита. Этот стиль лучше всего использовать против одного-двух врагов.

Помимо лазерного меча, в ваше распоряжение поступят шокер, ла-

зерный пистолет, бластерная винтовка, снайперская винтовка, энергетический арбалет, пулемет, энергоружье, дробовик, ракетница, гранаты, растяжки и взрывчатка. Дополнительно у каждого вида оружия имеется режим альтернативного огня. Использовать все это богатство по ходу игры можно в двух режимах — от третьего и от первого лица.

Музыка, выполненная в лучших традициях “Звездных Войн”, поддерживает атмосферу игры, делая ее еще более интересной. А отличная графика полностью затянет вас в этот мир космических баталий.

Кстати, несколько слов о сетевой игре. Для тех, у кого нет возможнос-

ти выйти в Интернет или локальную сеть, разработчики создали боты, с которыми вы можете поиграть в разнообразные режимы игры (“захват флага”, “командная игра”, “каждый сам за себя”, “дуэль”).

Так что, на мой взгляд, это отличная игра с прекрасной графикой, занимательным сюжетом и хорошо подобранной музыкой. Превосходная находка для фанатов “Звездных Войн”.

И напоследок, несколько кодов для этой игры:

- devmapall — активирует доступ к другим паролем;
- god — режим бессмертия;
- give all — дает все;
- give ammo — дает оружие и патроны.

К.КЛИЩЕНКО,
г.Минск.

Голливуд — родина “Колес”

*Девять — Перумову для бреда страшного,
И одно Голливуду — ... а впрочем, лучше бы его отдали Перумову.*

Год (а для читающих эти строки, неизвестно сколько) назад вышел первый фильм трилогии “Властелин Колец: Братство Кольца”. Совсем скоро выйдут и “Две Башни”, а монстры игрового мира только теперь выпускают игру по первой части. Впрочем, простите. Я говорю о PC, на приставках уже существуют три игры по Толкиену. Включая великолепную “Two Towers”, о которой я имею лишь восторженные отзывы из чужих уст (жаль, у меня нет Playstation 2). На PC же игры по великой книге скромного профессора филологии идут либо из древности, либо из неизвестно чего... Из древности до нас дошли две части странноватого квеста-РПГ “Lord of the Rings” и стратегия реального времени “War in the Midearth”. Из ... пришла та самая современная игра. Игра, претендующая на затмение “Tomb Raider” и “Blade of Darkness” (т.е. action-adventure). Игра о наивных назгулах, “умной” камере, непонятных боях и специалистах по Толкиену.

Давайте не забудем о хорошем, о музыке. Очень удачный ее выбор иногда отвлекает от бреда, проис-

ходящего на экране. Бред этот выглядит, в общем-то, неплохо, хотя персонажи и напоминают героев фильма. Только вот не очень удачные текстуры делают гнома похожим на истукана, а эльфов — на голодающих детей Судана. И все, на этом приятное закончилось.

Теперь о святом, о геймплее. Приключенческая часть еще скуднее, чем в Diablo, если это вообще возможно. Ладно, диалоги скриптовые, но их так мало. Попытка передать сюжет в столь скудной форме (намного скуднее, чем даже в фильме) не удалась. Мне (прочитавшему книгу 8 раз) чаще всего удавалось понимать, что же заключалось в одной-двух фразах, оставленных специалистами (эту игру якобы консультировали специалисты творчества Мастера). Причем начало игры — ее наиболее приятная часть. Поют птички, солнышко светит. Вы начинаете игру за Фродо — сначала бегаєте, знакомитесь с местностью, и после пары ни на что не влияющих заданий идете дальше по сюжету. Продажа Торбы-на-Круче Любелии и извлечение Кольца Всевластия на свет Эру являются

последними спокойными и приятными действиями в игре. А далее...

В Хоббитоне назгулы. Вам предстоит пробраться незамеченным мимо троих из них. Ах, наивненькие-наивненькие, вы просто не представляете, как они подвержены погоне за камнями. Прямо как маленькие. И при этом они остаются в районе предполагаемого дислоцирования камня. Ожидают, что тот зашевелится, и они поразят его своим клинком. Но не все так просто, авторы постарались, чтобы сгнивший от времени мозг прислужников врага все-таки доставил вам хлопоты. В стратегически важных пунктах стоят всадники, но, сами понимаете, они готовы обменять стратегически важный пункт на простой серый булыжник. И тут “специалисты-толкиенисты” (отдать бы их Углуку на развлечение) подкидывают откровение. Те самые хоббиты, которые ходят бесшумнее и незаметнее мыши (если, конечно, Толкиен для вас является авторитетом), в крадущемся режиме громко трещат чем-то на каждом третьем шаге. “Мыши” уж больно здоровые. А может, это постарались недоброжелательно



настроенные назгулы. Насыпали по всему Средиземью хлопшек и взрывпакетов. По счастливому стечению обстоятельств, они же и не обращают внимания на треск, хотя тот громче падения камня. Вот именно из-за таких "правых" рук и потерпел поражение Вrag.

Следующее испытание — Вековечный Лес. И тут по-настоящему стало заметно, что на Востоке неладно — началась массовая миграция пауков из Лихолесья. Вы пробиваетесь деревянной тросточкой через ряды членистоногих в поисках своих друзей. Но как только находится один из них, игра приобретает разгильдяйский оттенок. Ведь ваши партнеры бессмертны — признание игрой слабости собственного искусственного интеллекта (видимо, чтобы те сами себя не угробили). А вы стоите в стороне и наслаждаетесь. Хотя, как герой, можете рискнуть собственным здоровьем (оное, наверное, восполняется поеданием грибов и пиетов). После сбора команды воедино вам достается одно из самых идиотских занятий, виденных мною за мою долгую игровую практику — поединок с Вязом. Сам по себе, этот факт отдает приставочно-детским проявлением синдрома "Рассвета". Но реализация и вовсе... Вы стоите на одном месте, вполне может быть, что первоначальная задумка авторов предполагала перемещения по площадке (но и так тошно). Рядом с вами по очереди грохочут о землю деревянные кулаки Старого Вяза. И вы должны в режиме от первого лица (а как лучше играть от третьего!) попадать в них. Но не все так просто — попадать надо во внутреннюю сторону ладони, иначе после трех неудачных попыток игра будет проиграна. Вяз крепок, но примитивен, так что в целом получается психodelика уровня демонстрации фильмов Тарковского группе неандертальцев.

Следующий этап игры учит, что "охотники за привидениями" были совершенно неправы. Не нужны для уничтожения призраков сложные излучатели и ловушки — достаточно тросточки, которую подержал в руках Фродо. С помощью этой тросточки я переложил более десятка неупокоенных духов. Кстати, о мерах по укладыванию. Ввиду

того что в честном бою хоббиту не победить без существенных потерь и одного врага за раз, пришлось разработать оригинальную систему боя — бег по кругу и удар на бегу. О воздействии подобных упражнений на психику уже говорилось.

И вот вы играете за Арагорна. Великому воину положено обладать множеством ударов и приемов. Множество ограничено сверху числом 6. При этом четыре из них можно увидеть только в связке, а еще один — это убийство лежащего врага. Блоком можно не пользоваться, просто давите на кнопку.

Кстати, часть игры, ответственная за Пригорье — одна из наиболее удачных. Преодолев преграды города, вы перебираетесь в пустошь. Путь на Заверть вам преграждают стайки волко-крыс. Этот факт, по большому счету, не вызывает нареканий, но вот если учесть обстоятельства, предшествовавшие вашему появлению на горе, то... Хотя, я догадался — тролль и орки (встречающиеся вам в конце пути) принадлежат к личной охране Гэндальфа. После "протыкания" Фродо назгулами вам надо отогнать кольценосцев от поверженного героя. Поджигание соперников — единственный способ, но вот беда, если рядом с телом хоббита их больше трех, то подойти к нему становится нереальным. Но самое забавное скрыто даже не в поединке. Чтобы игра считалась проигранной, вас должны убить, однако это не единственный способ вас победить. Назгулы, добравшиеся до раненого, начинают тыкать (а как это еще обозвать) в него мечами. Чтобы он погиб, ему необходимо пропустить около 20 ударов (а умирает он от передозировки моргульских клинков).

Мория. Вот тут-то и вступит в дело Гэндальф — воитель и маг. Воительность его состоит из четырех ударов, один из них — на добивание. Исполняет Митрандир его посохом, не зря, видно, Гнилоуст требовал отобрать посох. Магия скорее полезна постольку-поскольку

(кроме битвы с Балрогом). Мория в меру красива, но загадки в ней примитивны и не интересны. А жаль. Единственным более или менее забавным моментом мне видится игра за Арагорна. То небольшое время, которое он проводит под вашим управлением у врат Казад-Дума, Бродяжник тратит на отстреливание щупалец у Стража Озера. Само по себе это интересно, но вот скорость выпуска стрел из лука нашего героя заставляет думать, что за спиной у него отряд Леголасов. В самой Мории главную роль играет даже не Гэндальф, а камера. Ее "интеллект" ниже всяческих пределов. Если вы приближаетесь к колонне или стене, эта "умная" камера демонстрирует внешнюю сторону стены и обратную сторону колонны несоответственно. А то вообще потолок покажет.

Что было дальше, я не знаю, потому что игра безнадежно глючит в Лориене и не пускает дальше. Но впечатление сложилось.

Да, еще момент. Авторы в пресс-релизе утверждали, что игра — по книге, а не по фильму. Но те персонажи и пейзажи, которые встречаются и там, и там, до боли похожи друг на друга. В одном обзоре мне встретилась фраза: "Играть в эту игру сможет лишь истинный фанат Толкиена". Это одновременно и правда, и ложь. Правда то, что только светлое имя может заставить играть в этот бред. Ну а играть истинным фанатам, пожалуй, не стоит, потому что желчи и злобы не хватит на создателей этого безумия.

P.S. Термины и названия, приведенные в статье, в основном взяты из перевода Муравьева.

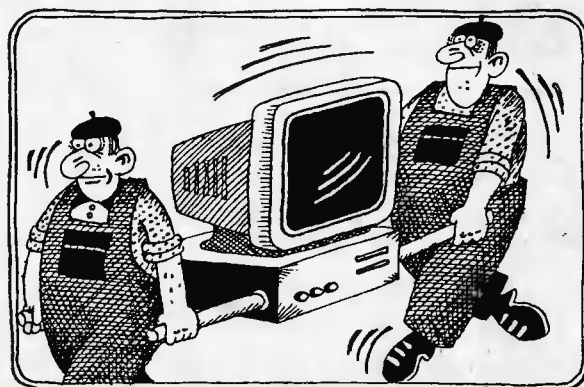


Рисунок О.Попова



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) 221-01-10,
либо через E-mail:
rl@radiopage.by
rm@radio-mir.com
WWW: http://radio-mir.com

Приму в дар компьютер.
462419, г. Орск, ул. Лен. Комсомола, 35 — 21.

Приму в дар или дешево куплю компьютер или комплектующие.
624238, Свердловская обл., г. Нижняя Турва, п. Ис.
ул. Артема, 65 — 12, Кислицын Н.В.

Куплю контроллер жесткого диска для компьютера "Истра 4816-03" (1991 г.), программное обеспечение для компьютера "Искра-1030М".
453500, Башкортостан, г. Белорецк,
а/я 21, Сафонов С.Н.
Тел. (34792) 4-45-67.

Куплю или приму в дар ноутбук и комплектующие к ним.
676244, Амурская обл., г. Зея,
ул. Мухина, 401 — 13, Дмитриенко С.В.

Ищу схему монитора CM322EGA и описание к ней. Олег.
E-mail: novpost@mv.ru

Ищу информацию по модему IDC2442 BXL фирмы INPRO.
E-mail: tdw@mail.ru

Ищу схему раскладки соединительного кабеля принтера MC6313 (аналог MC6307) к Пентийму-3. Нужны положения переключателей.
E-mail: bosernik@vltz.ru

Ищу схему включения и программу прошивки управляющего контроллера для микросхемы ISD33000.
440901, Пензенская обл., г. Заречный, пр-кт Мира, 94 — 11, Котыков В.М.

Куплю м/с KP1820BE3-021 для маршрутного компьютера МК-21093.
Тел. (34792) 5-11-25, или для Башкортостана (292) 5-11-25, Алекс.
E-mail: mailto:room21@bashnet.ru

Ищу описание устройства, принципов работы, схему и информацию о создании программного обеспечения игровых приставок SEGA, Dandy, Sony PS, Nintendo 64.
692320, Приморский край, Алучинский р-н, с. Чернышевка, пер. Таежный, 14, Ромащенко Р.С.

Ищу схему ПК "Ратон-9003" пр-ва Гомельского завода, любое программное обеспечение для ZX Spectrum-совместимых ПК на каскадах.
211030, Витебская обл., г. Орша,
ул. Заднепровская, 4/2 — 135, Кучинский Д.М.

Продаю недорого системный блок IBM PC/AT 386 SX-25 RAM 2 Mb, HDD 170 Mb, Modem, Sound, Keyboard, Mouse.
456318, Челябинская обл., г. Миасс,
ул. Вернадского, 42 — 25, Олег Вальпа.
E-mail: sandh@narod.ru

Ищу схему монитора Videal (Proview Technology) Model 570 и описание ИС EMC 8100, используемой в нем.
E-mail: Oleg@zavodsvet.ru

Ученик 11-го класса с благодарностью примет в дар компьютер и комплектующие.
397600, Воронежская обл., г. Калач,
ул. Гагарина, 13, Лебединский Алексей.
Тел. 27-3-36.

Вышлю схему изготовления кабеля для подключения любых сотовых телефонов к компьютеру (микросхема Max232 заменена на 2 транзистора KT315).
E-mail: n_mar@tut.by

Продаю видеокарту S3 Savage3D (8Mb, AGP 2x, TV-out — композит и S-video), поддержка оверлея,

поддержка 3D, диск с последними драйверами.
Тел. в Минске 258-36-33.

Куплю б/у CD-ROM.
E-mail: taras000@attasua.net

Бесплатно вышлем каталоги радиосхем, радиодеталей, книг, радиоконструкторов, видео- и аудиокассет, дискет с подборкой полезных программ и каталог на CD по радиолубительству. К заявке на каталог необходимо прикладывать оплаченный конверт с обратным адресом.
422981, Татарстан, г. Чистополь, а/я 248,
Романов А.А. (B.R.S.)

Уважаемые читатели!

Подписаться на наши журналы (индексы: 48996 — "Радиомир", 48924 — "Радиомир. КВ и УКВ", 48925 — "Радиомир. Ваш компьютер") в I полугодии 2003 г. можно по каталогу Агентства "Роспечать", стр. 283 или по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь", стр. 135 (раздел "Издания РФ, распространяемые по прямым договорам").

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

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

Наши платежные реквизиты:

- для жителей России и стран СНГ (кроме Беларуси) —
получатель: ООО "Радиомир Пресс", ИНН 7729404741,
р/с 40702810900010000084 в ООО КБ "Агропромкредит", ф-л Центральный, г. Москва,
корр. счет 30101810500000000109, БИК 044525109
Адрес банка: 113054, г. Москва, ул. Зацепы, 43Б;

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

На бланке почтового перевода необходимо очень четко написать свой почтовый индекс, полный адрес, фамилию, имя и отчество полностью. В графе "Для письма" необходимо точно перечислить, какие конкретно номера какого из журналов Вы заказываете.

При оплате платежным поручением нужно предварительно выписать счет. Вся информация по E-mail: rm-sales@radio-mir.com или по тел./факсу в г. Минске (017) 221-01-10, или по тел. (029) 649-41-47.

Приобретение отдельных номеров журналов

В РОССИИ:

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

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 "ТРЕНТЭК": 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 "Тулской" (ст. метро "Тулская", место 412-38).

На УКРАИНЕ:

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

В БЕЛАРУСИ:

В Минске в магазине "Книга XXI век", пр. Ф. Скорины, д. 92, тел. (017) 264-27-97 (ст. метро "Московская").

Голливуд – родина “Колец”



