

радиомир

1 • 2003

Ваш компьютер

Windows XP MUI
Professional
Corporate Edition

ГИГАНТЫ
ГРАФИКИ
v.2.0

Adobe Photoshop v7.0 ACDSee v4.01
AutofX Studio Bundle Pro v.2.0
Ansis Portfolio v6.0 Paint Shop Pro V7.04
Virtual Painter v3.2 MGI PhotoSuite Platinum v.4
Picture Publisher Professional v10.0
PhotoImpact v7.0



ПОЗНАВАТЕЛЬНАЯ



PETROSOFT

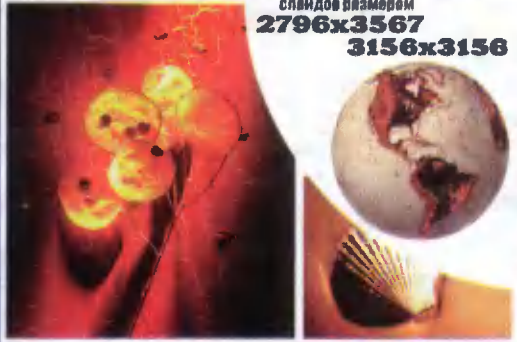
В КОМПЛЕКТЕ ПЕРСОНАЛЬНЫЙ

ФОТОКЛИПАРТ 18

Details of Nature
Global Perspectives

PETROSOFT

Совершенно новая концепция высококачественных
слайдов размером
2796x3567
3156x3156



ДИАНЕТИКА

ЗАЩИТИ СВОЮ СИСТЕМУ!
Лучшая коллекция антивирусов



Doctor Web for Windows 4.28a
популярный русский антивирусный пакет
F-Prot AntiVirus 3.12a Single User
антивирусный пакет для рабочей станции
Kaspersky Anti-Virus Personal Pro 4.0.6.6
RETAIL
Kaspersky Anti-Virus Administration Kit 4.0.5.0
установка AVP в сеть
Kaspersky Anti-virus AVP for MS NT/2000
Server 4.0.5.0
AVP для сервера 2000/NT
McAfee VirusScan Multiplatform v4.5.1 SP1
антивирусный пакет для всех версий Windows
Norton AntiVirus 2002 для
Win98/Me/2K/XP1
русская версия
Trend Micro PC-cillin 2002 9.0.1255
мощный антивирусный пакет

БИБЛИОТЕКА СЭМЛОВО
ПОЛНАЯ КОПИЯ ДИСКА

БИОЛОГИЧЕСКИЙ
ЭНЦИКЛОПЕДИЧЕСКИЙ
СЛОВАРЬ



Более 7.500 статей

PLATINUM 24
ELECTRONIC DRUMS

ФОРМАТ: GIGA+WAV
Drums: Percussion

(:wizoo:)

А.Колотенко.

"Строя миры..."

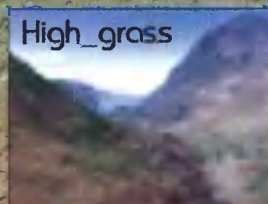
Основы работы с пакетом создания трехмерных ландшафтов Planetside Terragen



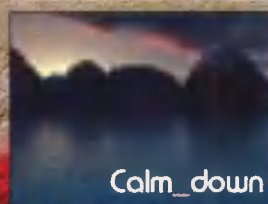
Техника рисования
в Planetside Terragen
основывается
на принципе
"слоеного пирога".

- 1 - корпус
- 2 - текстура/вода
- 3 - атмосфера
- 4 - слой облаков
- 5 - солнце

High_grass



Calm_down



Mountain_waterfall



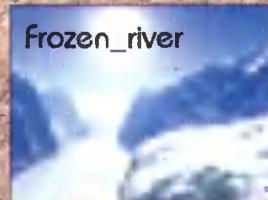
Mars_dawn



Yellow_sunset



Frozen_river



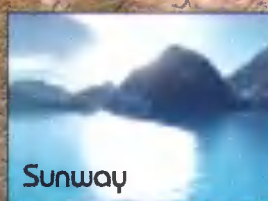
Frozen_peaks



Afternoon



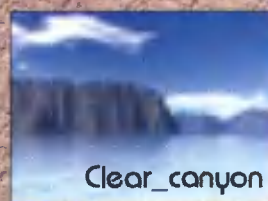
Sunway



Coldwater_Canyon



Clear_canyon



Примеры ландшафтов, созданных при помощи Planetside Terragen.

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 КОРЕБИТ,
Владимир КУЦЕНКО,
Елена ЛЕВИТМАН,
Николай МЕДВЕДЬ,
Геннадий ПЕЧЕНЬ,
Геннадий ШУЛЬГИН

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

Компьютерная верстка —
Янина БЕЛЬСКАЯ

Техническая графика —
Наталья БЕЛЬСКАЯ,
Александр ГОРАЮТИН,
Мария КАЛАБУХОВА

Оформление обложки —
Наталья БЕЛЬСКАЯ,
Надежда БОГОМОЛОВА

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

Зв 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.Коштыганца, 6 — 233, тел. (095) 105-99-89

Подписано к печати 26.11.2002 г.
Формат 60 х 84 1/8.
Печать офсетная. 6 печ. л.
Цена свободная.

Отпечатано в типографии
ЗАО "Красногорская типография".
Заказ 3097.

радиомир Ваш компьютер

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

№1/январь/2003

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

МУЛЬТИМЕДИА

А.КОЛОТЕНКО. "Строя миры..." Основы работы с пакетом
создания трехмерных ландшафтов Planetside Terragen 2

НЕ ТОЛЬКО НОВИЧКУ

А.ГРИНЧУК. Работа с редактором деловой графики Visio 2002 5

В.КУЦ. Знакомимся с офисным пакетом OpenOffice.org 7

А.ГОРЯЧКИН. Создание web-страниц 13

КОММУНИКАЦИИ

С.ГРИНЧУК. Копирование файлов из Интернета 15

Ю.ЛЕВИН. Выбираем персональный файрволл 18

ДАЙДЖЕСТ 20

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

М.ДОЛИНСКИЙ. Использование динамической памяти
в прикладных программах 24

Sergio /HI-TECH. Низкоуровневое программирование 26

ДИАЛОГ ПРОГРАММИСТОВ

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

И.РОЩИН. Исправление набранного в неправильном режиме текста
с помощью резидентной программы 30

И.ШИРКО. Сделай сам:
Управление Winamp'ом 34

РЕЦЕПТЫ

О.ВАЛЬПА. Адаптер ввода-вывода 36

Б.ШИЛЬНИКОВ. Еще раз о "витой паре" 38

МИР 8 БИТ

С.РЮМИК. Музыкальный сопроцессор "Dendy" vs. "ZX-Spectrum" 39

Возвращаясь к напечатанному

№5/01, С.35. И.РОЩИН. Еще об эмуляторах 42

КОМПЬЮТЕРНЫЕ ХРОНИКИ 42

ИГРОТЕКА

А.ГОРЯЧКИН. Его Величество Пасьянс 43

А.ВЕНДИЛОВСКИЙ. Newer Winter Night 45

ДОСКА ОБЪЯВЛЕНИЙ

Куплю, продам, обменяю 48

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

“Строя миры...”

Основы работы с пакетом создания трехмерных ландшафтов Planetside Terragen

А.КОЛОТЕНКО,
г.Харьков,
E-mail: logict@front.ru

Многие из вас, уважаемые читатели, наверное, видели произведение компьютерного искусства “Final Fantasy”. Данный фильм (?) поражает своим качеством, близким к фотографическому, спецэффектами, качеством моделей персонажей (чего стоит только прорисовка волос главной героини) и, конечно же, пейзажа-

ципе “слоеного пирога”, на рис.1 приведена его схема.

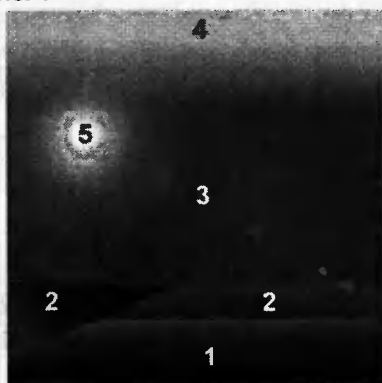
Здесь:

- 1 — каркас. Planetside Terragen обладает достаточно широким спектром алгоритмов создания поверхностей ландшафтов. Каркас является той основой, на которой будут располагаться все остальные детали нашего мира;

Our fantasy can create so much magic worlds like no other “device” can do...

- 2 — текстура/вода. На каркас накладывается текстура. В комплекте с программой поставляется несколько стандартных наборов текстур, которые, тем не менее, позволяют создать довольно интересные картины. Что касается воды — это отдельный слой нашего “пирога”, но если вода прорисовывается, то текстура, естественно, не видна;

Рис. 1



ми. Данное произведение компьютерного искусства было создано при помощи пакетов Maya и Pixar. В этой статье я предлагаю вам попробовать себя в качестве режиссеров и художников ландшафтов. Конечно, фильмы мы снимать не будем, но создать несколько картин для рабочего стола или дизайна веб-страницы особого труда не составит.

Для этого нам потребуются ПК, пользователь и пакет Planetside Terragen версии 0.8.11 или выше. Получить программу можно на сайте производителя <http://www.planetside.co.uk>. Там же вы найдете ее описание на английском языке и много полезной информации. Данное ПО производит рендеринг (от англ. Render), используя мощности **только** процессора. Потому при прорисовке ландшафтов новые GeForce'ы особо не помогут.

Теперь, что касается техники рисования. Тем из читателей, кто уже сталкивался с подобными программами (в частности, с Corel Bruce и Modjo Worlds), будет довольно легко освоиться в новой оболочке. Техника рисования основывается на прин-

Рис. 2

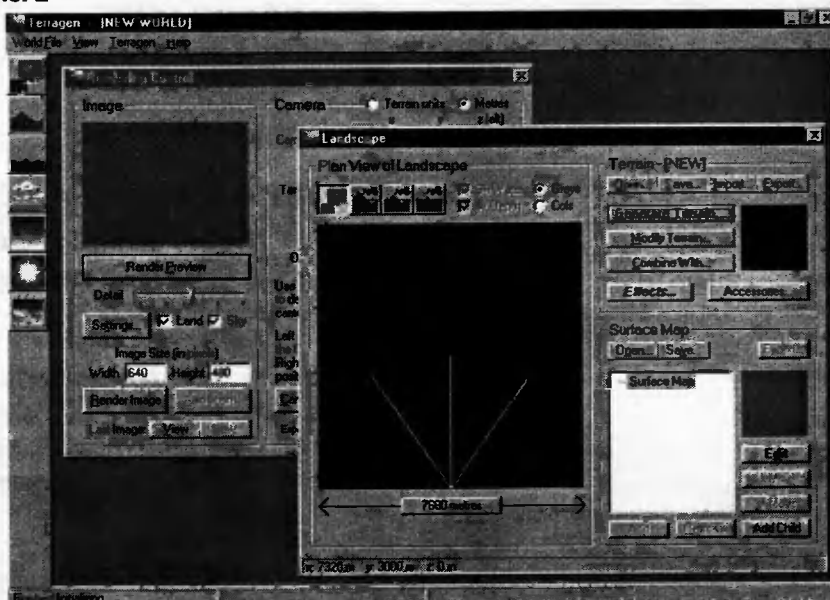


Рис. 3

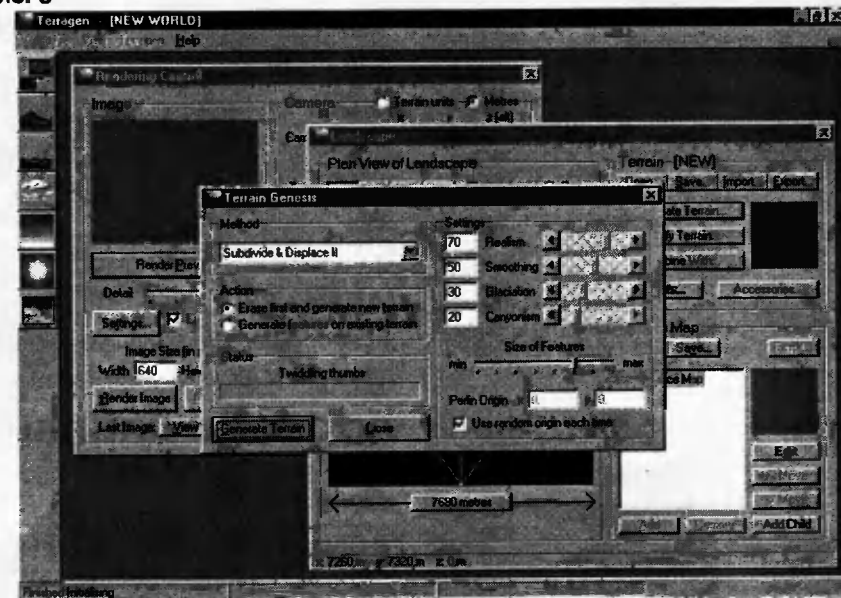
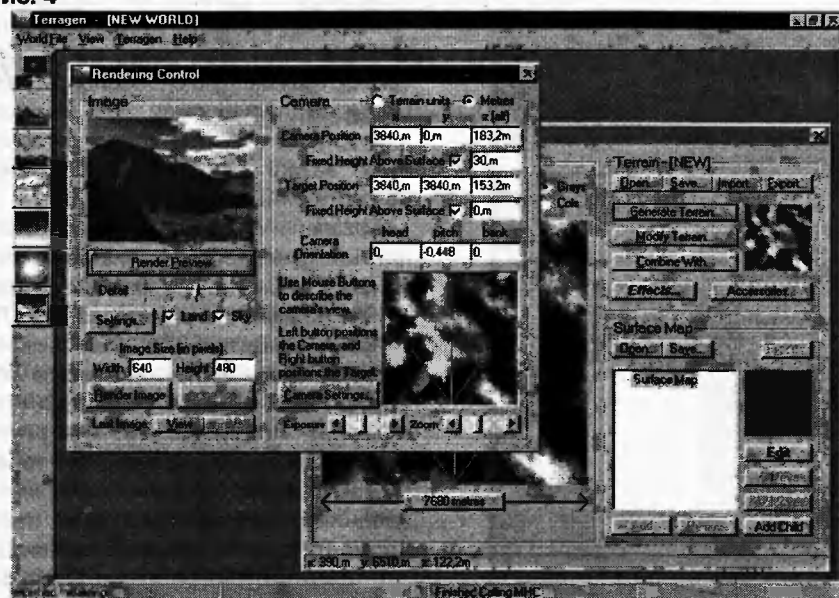


Рис. 4



- 3 — атмосфера. Это слой "воздуха". Сюда относятся все настройки, касающиеся плотности, видимости, цвета атмосферы. Также существуют стандартные шаблоны атмосфер (нормальная, марсианская, дождливая и т.д.);

- 4 — самый верхний слой — слой облаков. Программа позволяет создавать как просто "крышу" из облаков, так и сложные многоуровневые "облачные" 3D-конструкции (с настройкой теней, плотности, размера облаков и т.п.);

- 5 — солнце (или другое "светило"). Здесь вы можете настроить все, что касается освещения: положение звезды над горизонтом и по сторонам света, настройку теней и размера видимой звезды и т.п.

В Terragene используется две системы координат — terrain units и метры. В процессе создания вы видите свой мир в виде карты. Чтобы получить картину мира, необходимо установить камеру (Camera) и цель (Target). Соответственно, получится срез "слоеного пирога" по прямой от камеры к цели, который можно наблюдать в окне предварительного рендеринга.

Теперь обратимся собственно к процессу создания.

Шаг 1. Создаем каркас. Для этого в форме ландшафта необходимо выбрать опцию Generate Terrain (рис.2).

Появится форма создания каркаса. Тут мы можем выбрать один из алгоритмов, а также настроить парамет-

Рис. 5

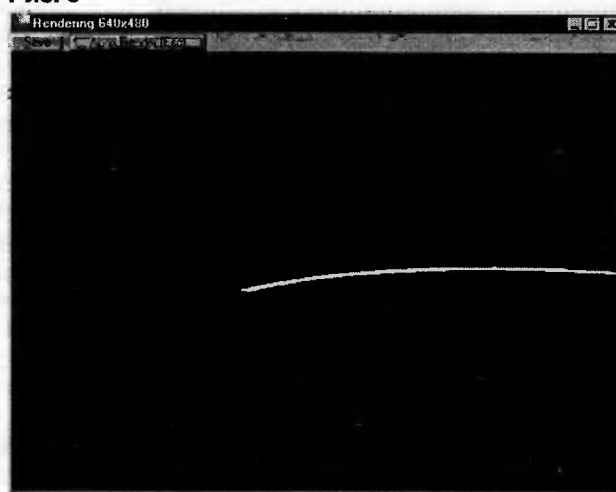
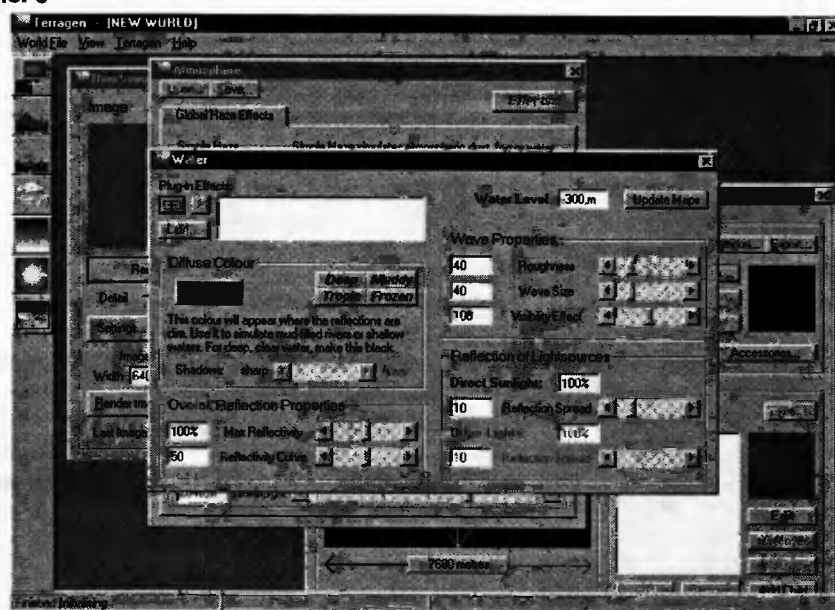


Рис. 6



ры ландшафта "гладкость", "каньонистость", "реализм", "размер отдельных частей каркаса". Наиболее оптимальным (на мой взгляд) и реалистичным алгоритмом является Subdivide and Displace (рис.3).

Шаг 2. После создания каркаса мы можем отредактировать его, используя инструменты ("бульдозеры"), расположенные над картой ландшафта. Щелчок правой кнопкой мыши позволяет сделать углубление, а левой — возвышенность. Всего имеется три "бульдозера" — большой, средний и малый. Расположить камеру и цель можно с помощью четвертого инструмента. По умолчанию в Terragene устанавливается небо и атмосфера, потому можете смело просматривать в Render Preview, что у вас получилось. Бегунок под окном просмотра позволяет отрегулировать детализацию (рис.4).

Шаг 3. Если результат вас устраивает — можете смело жать Render Image, предварительно выставив размеры картинки и настроив параметры камеры в Camera settings.

Скорость процесса рендеринга напрямую зависит от таких факторов как быстродействие ЦПУ, количество полигонов в картине, т.е. от сложности текстур (например, львиную долю времени занимает создание неба), размера картины, детализации, настроек камеры (рис.5).

После окончания рендеринга программа сообщит о времени, затраченном на данный процесс, и количестве полигонов в пейзаже.

Рис. 7

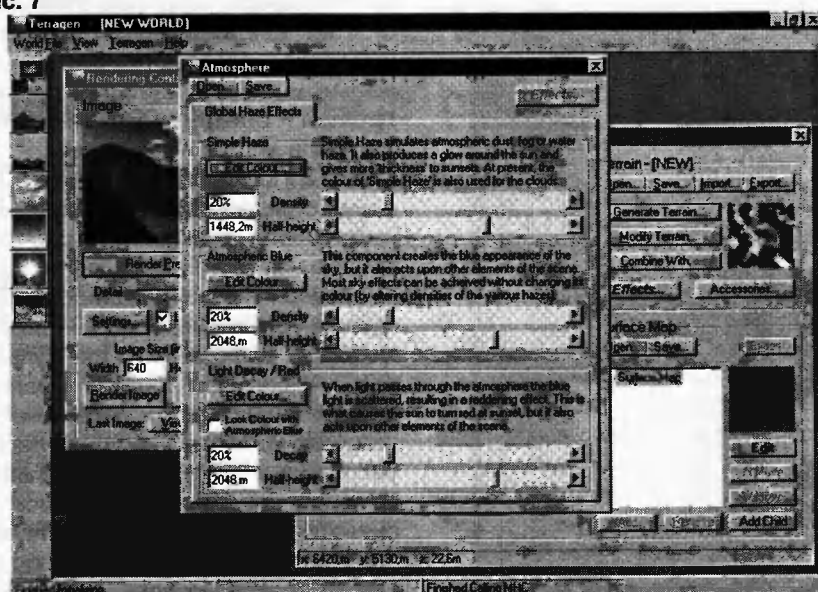


Рис. 8

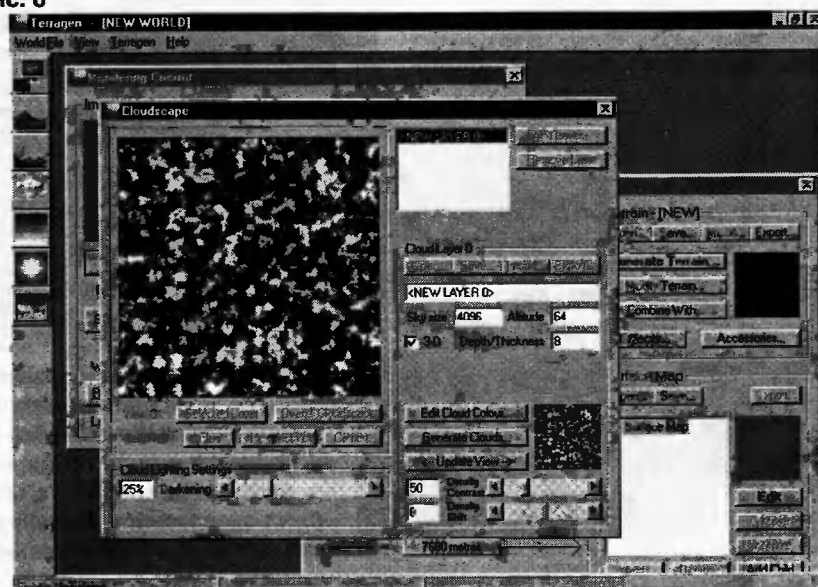


Рис. 9



Выбрав из панели инструментов "Water", вы сможете отрегулировать уровень воды, настроить ее цвет, а также отражающую способность, размер и амплитуду волн. При изменении уровня воды, участки, которые будут "затоплены", будут отображены на карте синим цветом (рис.6).

В разделе "Atmosphere" можно изменить цвет атмосферы, ее плотность (т.е. можно изобразить прозрачный воздух или же устроить пыльную бурю) (рис.7).

При помощи панели "Cloudscape" настраивается все, что касается облаков. Слой облаков создается при помощи инструмента Generate Clouds. Здесь, так же как и при генерации ландшафта, вы можете указать средний размер облаков и их цвет. Кроме того, настраиваются плотность, затенение, глубина/толщина (рис.8).

Настройка освещенности является одним из наиболее важных этапов в создании ландшафта. Terragen предоставляет возможность настроить не только вид и положение "светила", но и поведение остальных объектов, т.е. тени от ландшафта, затенение облаков, второстепенные тени. В стандартном наборе есть несколько красивых калибровок освещения, в частности, рассвет и закат (рис.9).

Следует помнить, что Terragen сохраняет параметры каждого элемента ландшафта в отдельных файлах. Т.е. если вы создали что-либо, но сохранили только каркас, то придется создавать атмосферу и освещение заново. Но в этом есть и свои преимущества, т.к. можно использовать уже готовую атмосферу или освещение для разных ландшафтов.

На рис.10 приведен пример использования готовой картинкой из Terragen.

Рис. 10



Разработчики Terragen обещают добавить новые функции, которые позволят рисовать водопады, добавлять отдельно стоящие валуны, камни, скалы, а также деревья и прочую растительность.



Работа с редактором деловой графики Visio 2002

(Продолжение. Начало в №12/2002)

Формы

После небольшого эксперимента с созданием простейшей диаграммы, в этой статье мы перейдем к более подробному знакомству с основными "действующими лицами" программы Visio. В первую очередь рассмотрим основные свойства рисунков-заготовок, которые на языке Visio имеют название **Фигура (Shape)**. Из-за большого разнообразия этих объектов, им даже сложно дать какое-то общее определение. Как правило, это единый объект (пусть даже состоящий из нескольких элементов), фигуры переносятся с трафалета одним движением, а на рабочем листе выделяются одним щелчком. Остальные свойства уже зависят от конкретной фигуры.

Фигуры подразделяют на три больших класса — одно-, двух- и трехмерные. Одномерные фигуры чаще используются в качестве соединителей. Их отличительная особенность — наличие маркеров начала и конца фигуры (**рис.1**). У стрелок дополнительно имеются маркеры выделения, позволяющие изменить ее толщину. Маркеры начала и конца применяются для склейки с другими фигурами.

Самый важный класс — двухмерные фигуры, фактически все остальные получены их модификацией. Например, одномерная линия и трехмерный куб — это частные случаи прямоугольника. В отличие от других типов, двухмерные фигуры могут быть как замкнутые, так и разомкнутые. При выделении они обрамляются набором из восьми маркеров, угловые используются для пропорционального изменения размеров, верхние и нижние — для растяжения и сжатия фигуры по вертикали, боковые — по горизонтали. Но, несмотря на отличия, одномерные и двухмерные объекты могут быть легко преобразованы друг в друга (**рис.2**) — выделите фигуру/**Format/Behavior**/установите переключатель в положение **Line (1-dimensional)** или **Box (2-dimensional)**.

К сожалению, Visio не содержит полноценных трехмерных фигур.

Рис. 1

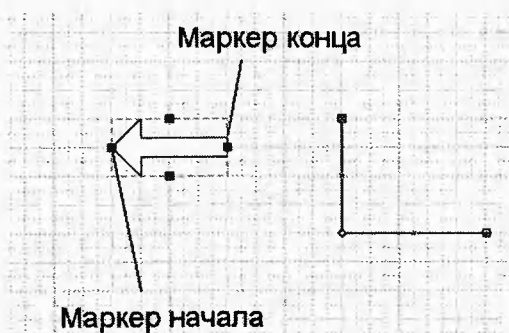


Рис. 2

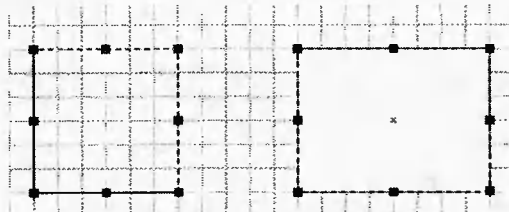


Рис. 3

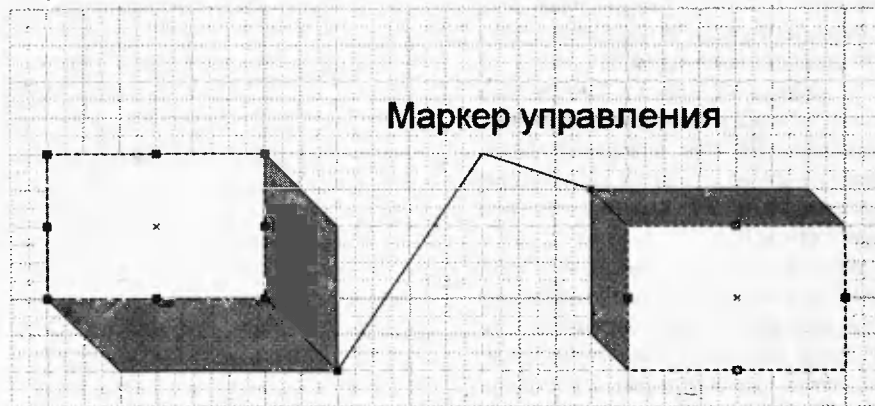


Рис. 4

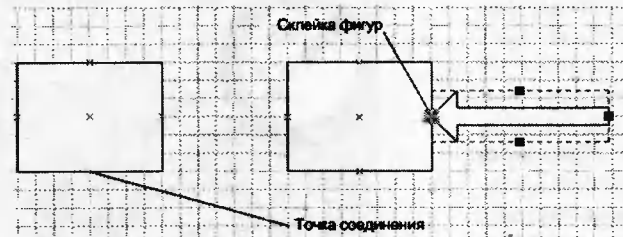
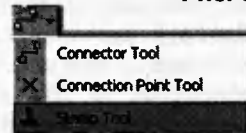


Рис. 5



(рис.5). Затем наведите указатель мыши на точку соединения,

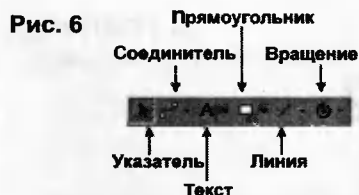
нажмите левую клавишу мыши и перетяните точку соединения.

Маркеры начала и конца одномерной фигуры позволяют изменять не только длину, но и направление фигуры. Вращение двухмерных объектов осуществляется по-другому. Необходимо выделить фигуру и выбрать на панели инстру-

Фактически под 3D-фигурами понимаются двухмерные фигуры с тенью. Соответственно, для управления таким объектом добавляется еще один маркер управления (**рис.3**), перемещая который, можно изменить положение и размеры тени. Во всем остальном — это типичные двухмерные фигуры.

Как видно из этих примеров, все фигуры сопровождаются маркерами. Пока мы имели дело только с маркерами начала и конца, выделения и управления, они появляются в той ситуации, когда фигура выделяется щелчком мыши. Если же фигура (двух- или трехмерная) не выделена, то на ней видны точки соединения — особый тип маркеров, предназначенный для склейки фигур (**рис.4**).

Положением точек соединения можно управлять. Для этого разверните список инструмента соединителя (**Connector Tool**) на панели инструментов Visio и выберите **Connection Point Tool**



ментов **Rotation Tool** (инструмент вращение — рис.6). Это приведет к изменению вида угловых маркеров (кружочек вместо квадратика) и появлению еще одного маркера в центре фигуры. Угловые маркеры — маркеры вращения — позволяют вращать фигуры, центральный — маркер центра вращения — можно и переместить, он задает центр вращения фигуры (рис.7).

Однако средства для работы с одиночной фигурой этими примерами не исчерпываются. После выбора инструмента **Pencil Tool** из списка инструментов линии, на фигуре отображаются управляющие маркеры, перемещение которых позволяет изменить начальный рисунок до неузнаваемости. Если же и это покажется недостаточным, можно, предварительно нажав клавишу **Ctrl**, щелкнуть по границе фигуры и нанести еще один управляющий элемент (рис.8).

Ну а оформление одиночного рисунка можно завершить с помощью меню **Format**. На рис.9 показаны команды, которые позволяют сделать заливку, создать тень и скруглить углы.

Как известно, «один в поле не воин», и Visio предлагает набор команд для действия с группами фигур. Как и во многих других программах, имеются операции группировки и разгруппировки (в меню **Shape/Grouping**). Несколько фигур можно сгруппировать (**Shape/Grouping/Group**) в один объект, который будет перемещаться и поворачиваться как единое целое. Эта операция обратима — группа может быть расформирована выбором **Shape/Grouping/Ungroup**. В группы можно объединять как отдельные фигуры, так группы. Более того, большинство стандартных фигур представляют собой именно группу из более простых элементов. Этим можно воспользоваться для модификации стандартной фигуры (рис.10). В этом случае, правда, нужно учесть, что стандартные фигуры защищены от ряда модификаций, поэтому после разгруппировки необходимо снять защиту

Рис. 7

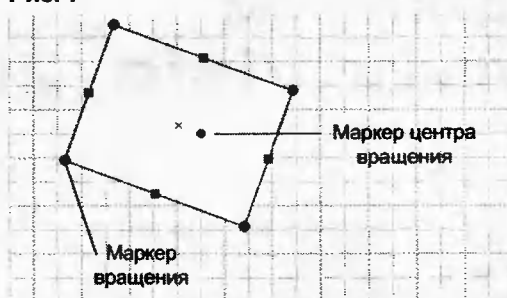


Рис. 8

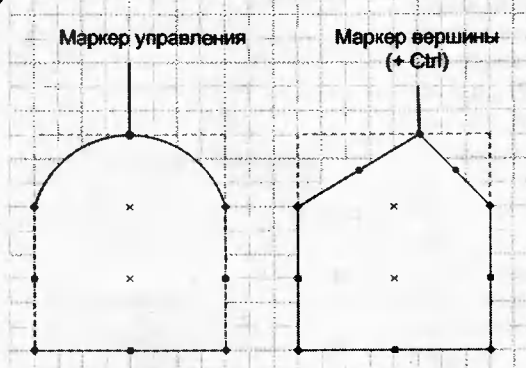


Рис. 9

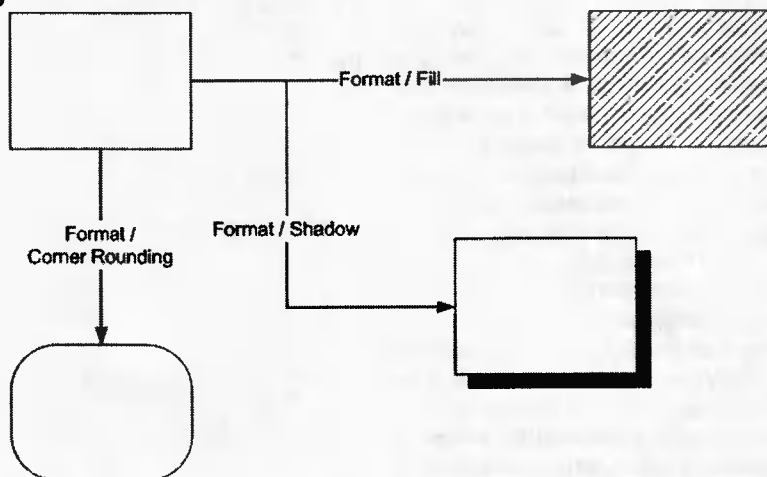
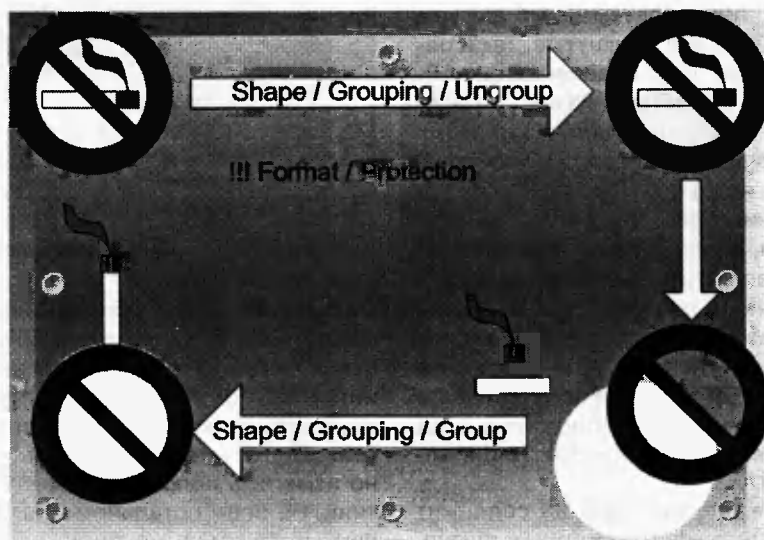


Рис. 10



— **Format/Protection**/снять флажки во всех позициях.

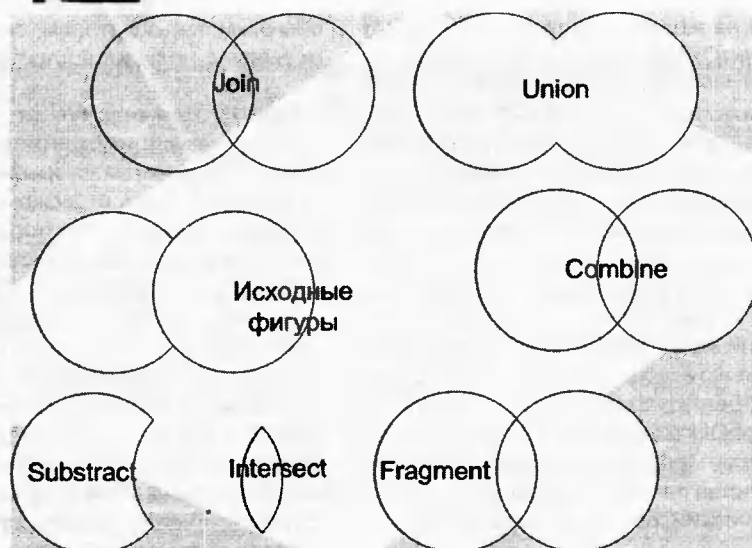
Есть в меню **Shape/Operations** набор в чем-то похожих по действию команд. Однако они не имеют обратной команды и приводят к полной модификации группы объектов. На рис.11 показан результат действия этих команд:

- **Union** — объединение, внутренние линии удаляются;

- **Combine** — комбинирование, объединение с сохранением всех внутренних линий;

- **Join** — соединение, контуры фигур соединяются в единый контур, заливка теряется;

Рис. 11



- **Fragment** — фрагментация, пересекающиеся фигуры разделяются на фрагменты;

- **Intersect** — пересечение, остаются только пересекающиеся части фигур;

- **Subtract** — вычитание.

Итак, мы достаточно "вооружены" для дальнейших действий. В следующем номере мы применим изученные операции на практике, а также, чтобы наши труды по созданию и модификации фигур не пропали даром, рассмотрим процедуру создания пользовательских трафаретов.

(Продолжение следует)

Знакомимся с офисным пакетом OpenOffice.org

В.КУЦ,

E-mail: victor_kootz@mail.ru

Новый пакет офисных приложений OpenOffice.org 1.0 (<http://www.openoffice.org>), предназначенный для установки на компьютеры с операционными системами Windows или Linux, является достойной альтернативой широко распространенному пакету Microsoft Office. Основное достоинство OpenOffice.org, отраженное в его названии, заключается в полной открытости проекта, причем он бесплатен не только для конечных пользователей, но и для разработчиков. Это означает, что любой желающий может получить исходный код программ пакета OpenOffice.org на сайте разработчика, внести в него любые изменения или дополнения, а также использовать его в своих разработках, правда, без права коммерческого распространения программ, использующих этот код. Бесплатность же для пользователей нового пакета приобретает особую актуальность именно сейчас, когда корпорация Microsoft, являющаяся монополистом на рынке программ для офиса, решила немного "закрутить гайки" и резко усилила борьбу с незаконным (по ее мнению) использованием своих программных продуктов, в числе которых офисные пакеты традиционно занимают далеко не последнее место.

Проект OpenOffice.org был основан компанией Sun Microsystems, являющейся извечным конкурентом Microsoft, осенью 2000 г. с целью привлечения широкого круга независимых программистов для совершенствова-

ния и дальнейшего развития интересного, но вызвавшего в свое время немало противоречивых мнений пакета StarOffice 5.2. OpenOffice отличается от своего предшественника, в первую очередь, улучшенной поддержкой Unicode, что существенно расширяет возможности по локализации пакета, значительно переработанными функциями импорта/экспорта в другие форматы, включая Microsoft Office, несколько большей скоростью работы. Код программ, полученный в результате работы сотен, а то и тысяч независимых программистов в рамках проекта OpenOffice.org, ляжет в основу новой версии офисного пакета StarOffice 6.0, но вот его распространение будет осуществляться уже на коммерческой основе. При этом в StarOffice 6.0 будут дополнительно включены оригинальные шрифты, библиотека изображений, лингвистические модули и СУБД Adabas D, лицензированные у сторонних компаний. Тем не менее, оба пакета ни в коей мере не являются конкурентами друг другу, так как StarOffice ориентирован, в первую очередь, на корпоративных заказчиков, а OpenOffice.org — на индивидуальных пользователей.

Но предметом нашего сегодняшнего обзора будет не оригинальный продукт, а дистрибутив OpenOffice.org, собранный российской компанией ALTlinux (<http://www.altlinux.ru>) и отличающийся целым рядом улучшений в плане поддержки не только русского, но и укра-

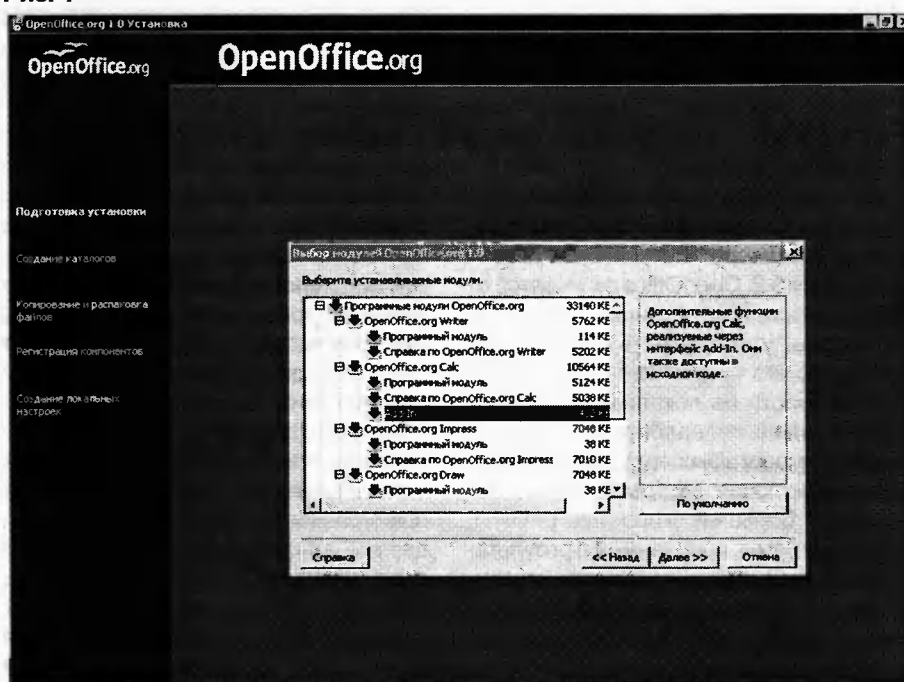
инского языков, отсутствующих в оригинале. Ведь сколько бы ни твердили некоторые "крутые спецы" о необходимости поголовного изучения "главного языка интернационального общения", русскому человеку гораздо сподручнее общаться, даже с компьютерной программой, на своем родном языке. В числе нововведений ALTlinux наиболее интересны полная русификация всех меню и диалогов, поддержка проверки орфографии для русского языка, поддержка автоматической расстановки переносов для русского, украинского и английского языков. На FTP-сайте компании выложены дистрибутивы пакета не только для Linux (что было бы вполне естественно, поскольку компания создает и распространяет собственные варианты дистрибутивов этой популярной ОС), но также и для Windows. Эти дистрибутивы отличаются достаточно скромными размерами, составляя 74 Мб для Linux и менее 53 Мб — для Windows, тогда как хорошо известно, что Microsoft Office XP со всеми его "примочками" с большим трудом втискивается в целый компакт. А вот любителям всевозможной экзотики стоит заглянуть на сайт OpenOffice.org, где можно найти варианты пакета на 24 различных языках не только для Линукса или Окшечек, но и для куда менее распространенных ОС — Solaris, всячески лелеемой фирмой Sun, или для утонченных эстетов — Mac OS X.

Короче говоря, пакет OpenOffice.org доступен всем и каждому без всяких

ограничений, и в этом отношении творениям Microsoft до него очень и очень далеко. А вот как в других отношениях? Для этого, перефразируя Цезаря — “скачал, поставил, посмотрел” — и предлагаются вашему вниманию первые впечатления о новом продукте. Предупреждаю сразу — это только первые, достаточно поверхностные впечатления, так что некоторые моменты могут пройти мимо моего внимания, до некоторых свойств или возможностей пакета я просто не добрался — уж не обесудьте за субъективность оценок.

Итак, знакомство с OpenOffice'ом начинаем, как обычно, с процесса инсталляции пакета (рис.1).

Рис. 1



Весь этот увлекательный процесс, как и положено приличным программным продуктам, происходит быстро, четко и без всяких проблем. Русифицированный мастер позволяет выбрать три варианта установки: стандартный (устанавливаются все компоненты пакета, требуемый объем дискового пространства — около 140 Мб), минимальный (примерно 113 Мб) и выборочный. Последний режим позволяет в известной степени адаптировать содержимое пакета к нуждам конкретного пользователя. Приложения OpenOffice настолько глубоко интегрированы друг с другом, что, как видно из приведенных выше значений, при любом варианте установки необходимо иметь определенный запас свободного ме-

ста на жестком диске, так что пакет OpenOffice если и выигрывает у Microsoft Office по занимаемому дисковому пространству, то только у последних его версий и уступает более ранним вариантам последнего.

OpenOffice.org — по-настоящему интегрированный пакет. Все приложения, входящие в пакет, открываются в виде окон в границах единого рабочего пространства, имеют сходный внешний вид интерфейса и почему-то постоянно ассоциируются с MS Office'ом, особенно текстовый редактор OpenOffice.org Writer и табличный процессор OpenOffice.org Calc. Кроме них, в состав пакета входят векторный графический редактор OpenOffice.org

обычным пользователем, скорее всего, никогда не будет востребована.

OpenOffice.org Writer

Основным приложением любого офисного пакета, на который приходится львиная доля конторских задач, безусловно, является текстовый редактор, в нашем случае — Writer (рис.2).

Выше уже отмечалась определенная схожесть Writer'a с майкрософтовским Word'ом, и это может сыграть далеко не последнюю роль в процессе перехода с одного редактора на другой. Как и его более именитый собрат, Writer располагает двумя настраиваемыми панелями (третья, панель функций, общая для всех приложений пакета) — панели объектов и панели инструментов. Причем по количеству и разнообразию настроек панелей он существенно опережает любые другие текстовые редакторы, не исключая и MS Word. Кроме того, облегчить работу пользователя, особенно начинающего, призваны плавающие панели Навигатора и Стилиста, назначение которых вполне понятно из их названия. Широкий список поддерживаемых типов стилей — кроме стилей абзаца и знака, вид рамок, параметры страницы, списки также настраиваются с помощью набора фиксированных стилей, каждый из которых может быть отредактирован пользователем. Разумеется, можно создавать и свои собственные стили, как на основе уже существующих, так и полностью новых. Но панели Навигатора и Стилиста занимают много места на экране, поэтому, если в данный момент какая-то из них не нужна, ее можно свернуть и задвинуть подальше (или убрать вообще).

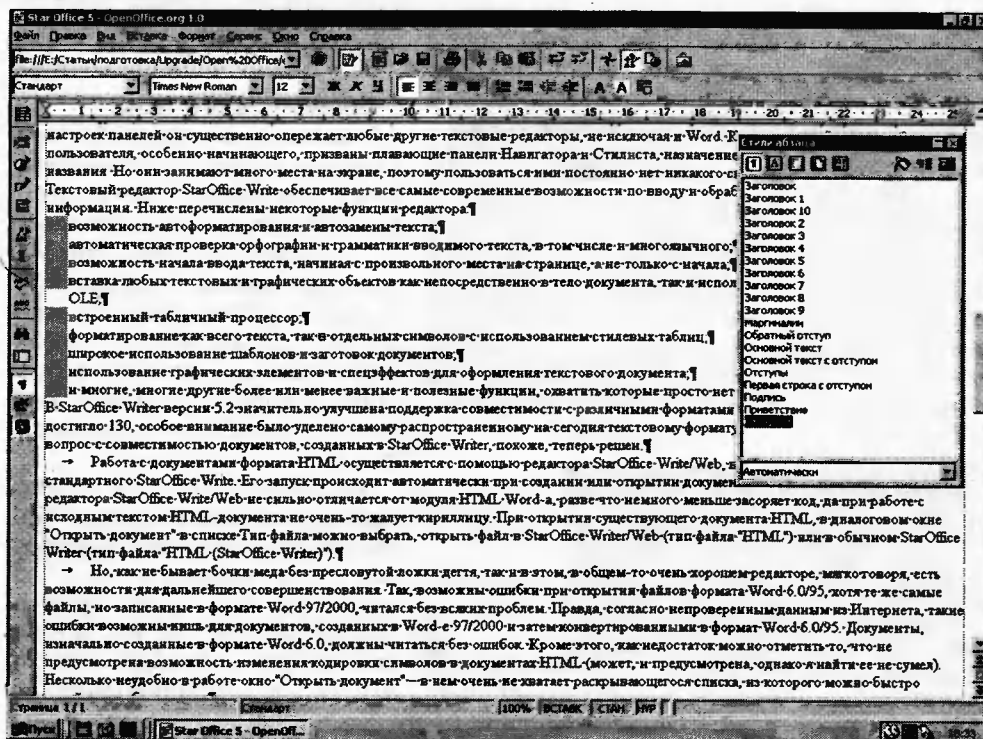
Для любителей оживить создаваемый документ самыми различными “финтифлюшками” удобно будет воспользоваться Галереей — банком графических объектов для вставки. Она реализована как выдвигающаяся, но не плавающая панель, которую при помощи кнопки-заковки (совсем как в Linux'e) можно намертво зафиксировать в главном рабочем окне программы.

Текстовый редактор Writer обеспечивает все самые современные возможности по вводу и обработке текстовой и графической информации. Перечислим некоторые функции редактора:

- возможность автоформатирования и автозамены текста, а также уникальная функция автодополнения при вводе текста;

Draw, web-редактор OpenOffice.org HTML Editor, редактор формул OpenOffice.org Math, приложение для создания презентаций OpenOffice.org Impress. А вот интегрированный Рабочий стол, графический растровый редактор Image и база данных Base, входившие в состав StarOffice 5.2, исчезли (и, честно говоря, потеря от этого невелика). Необходимо отметить, что такая комплектация куда больше соответствует общепризнанному понятию “офисный пакет”, и все составляющие этого пакета в той или иной мере могут быть использованы в процессе практической работы, что опять же выгодно отличает OpenOffice от MS Office, особенно последних версий, предельно перенасыщенных дополнительными функциями, большая часть которых

Рис. 2



- автоматическая проверка орфографии и грамматики вводимого текста, в том числе и многоязычного;
- возможность начала ввода текста, начиная с произвольного места на странице, а не только с начала строки;
- вставка любых текстовых и графических объектов как непосредственно в тело документа, так и используя связывание по технологии OLE;
- встроенные мастер построения диаграмм и табличный процессор;
- форматирование как всего текста, так и отдельных символов с использованием стилевых таблиц;
- широкое использование шаблонов и заготовок документов;
- использование графических элементов и спецэффектов для оформления текстового документа;
- и многие, многие другие, более или менее важные и полезные функции, охватить которые просто нет физической возможности.

В полном соответствии с тенденциями последнего времени, текстовый процессор Writer предоставляет возможность коллективной дистанционной работы, в том числе и через Интернет, что проявилось в появлении функции отправки документа по электронной почте непосредственно из редактора. Очень приятно работать с диалогом вставки гиперссылок, в том числе и перекрестных, а при создании документов текстовый про-

цессор позволяет использовать максимум возможностей буфера обмена, включая разметку встроенных таблиц и сохранение всех гиперссылок. Фактически, Writer берет из буфера не визуализируемую часть кода, а собственно HTML-код вместе с тегами, и встраивает его в свой код, вне зависимости от того, является ли обрабатываемый документ гипер- или же просто текстовым документом. Однако генерируемый Writer'ом HTML-код вполне сопоставим с Word'овским по своей неудобочитаемости и избыточности, вплоть до тегов посреди слова.

В Writer'e "родной" формат сохранения файлов построен на базе технологии гипертекста XML, что позволило заметно уменьшить размер готового документа по сравнению с форматами MS Word, а также значительно улучшить поддержку совместимости с различными форматами документов, так что вопрос с совместимостью документов, созданных в Writer'e, похож, близок к окончательному разрешению.

Но, как не бывает бочки меда без пресловутой ложки дегтя, так и в этом, в общем-то очень хорошем редакторе, мягко говоря, есть возможности для дальнейшего совершенствования. Так, возможны ошибки при открытии русскоязычных файлов формата Word 6.0/95, хотя те же самые файлы, но записанные в формате Word 97/2000, читались без всяких проблем. Правда, такие

ошибки возможны лишь для документов, созданных в Word 97/2000 и затем конвертированных его собственными средствами в формат Word 6.0/95. Документы, изначально созданные в формате Word 6.0, читаются без ошибок. Эта проблема существует давно, еще со времен StarOffice 5.2, однако удовлетворительного ее решения так и не было найдено.

А вообще возможности Writer'a по обмену данными с другими приложениями довольно широки. Кроме документов MS Word, в нем непосредственно открываются тексты в форматах RTF, DOS, электронные таблицы Excel и Lotus 1-2-3 (как DOS-, так и Windows-версий). Возможна вставка изображений практически во всех распространенных растровых форматах. Созданные в Writer'e документы могут быть сохранены в фор-

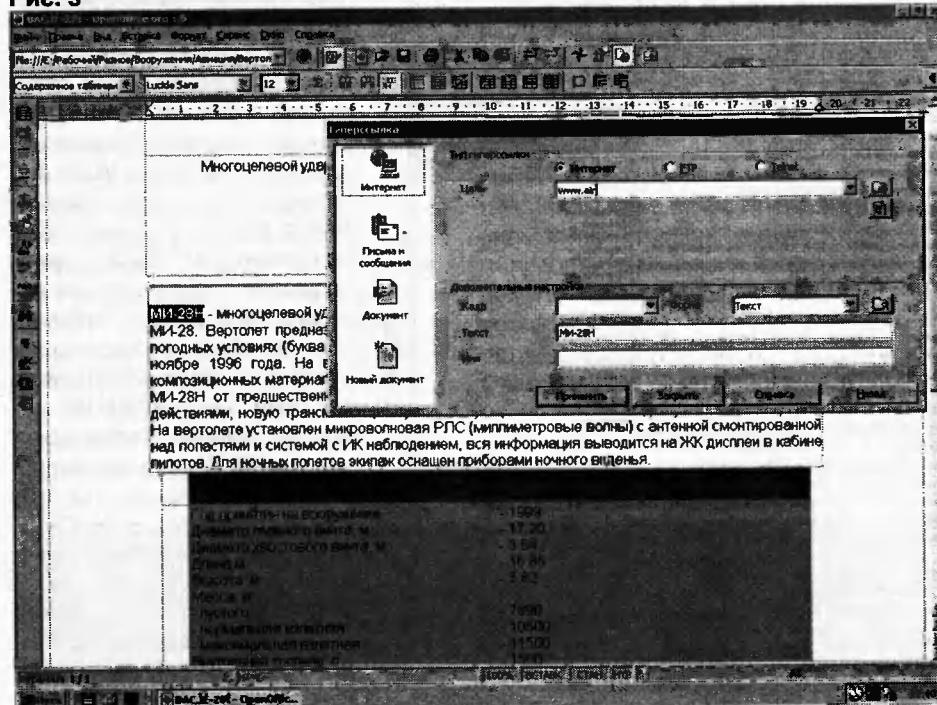
мате HTML и всех текстовых форматах, причем в настройках можно задать тот тип формата, в котором документ будет сохранен по умолчанию. Вроде бы, мелочь, но из таких вот мелочей и складывается общее впечатление об удобстве использования той или иной программы.

OpenOffice.org HTML Editor

HTML Editor — текстовый процессор для создания и редактирования гипертекстовых документов, по своим возможностям мало отличающийся от модуля HTML Word'a, но более удобный в работе. Особенно мне понравился в нем инструмент для создания и редактирования таблиц. Хотя HTML Editor и имеет свой собственный ярлык в меню "Пуск/Программы/...", тем не менее, это, по большому счету, все тот же Writer (рис.3).

Его запуск происходит автоматически при создании или открытии документа HTML. Судя по всему, HTML Editor наконец-то научился более или менее нормально работать с кириллическими шрифтами, в том числе и при работе с исходным текстом HTML, однако и с этим не все так гладко, как хотелось бы (рис.4). Так, например, до сих пор не решены проблемы с отображением русских имен файлов. Беда, конечно, небольшая, но осадок остается не самый приятный.

Рис. 3



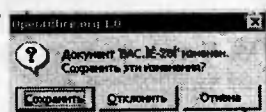
Экспорт документов возможен во все вышеперечисленные форматы, правда, кроме Lotus 1-2-3.

OpenOffice.org Math

Редактор формул OpenOffice Math, в отличие от других приложений пакета, особого впечатления не произвел. Работает он по принципу визуального редактора, который "прост, как правда" — нужно всего лишь ввести строчку формулы в специальном окошке (используя специальное форматирование), а сама формула в уже готовом виде появляется в главном окне программы (рис.6).

По сравнению с Microsoft Equation, редактором формул из состава Office, Math кажется несколько тяжеловесным и менее удобным в работе продуктом, хотя имеет и свои характерные особенности. Так, для вставки формулы в процессе создания текстового документа достаточно ввести ее,

Рис. 4



OpenOffice.org Calc

Вместо описания возможностей и свойств табличного процессора Calc предлагаю читателям посмотреть на скриншот рабочего окна программы и сыграть в детскую игру "найди десять отличий" от Excel'a (рис.5).

А если серьезно, то практически полная идентичность и внешнего вида, и основных возможностей обоих программных продуктов наводит на мысль, что процесс совершенствования интерфейсов табличных процессоров для конторских продуктов на данном этапе подошел к какому-то определенному пределу, и дальнейшие улучшения возможны только после появления принципиально новых, революционных решений, которых пока не наблюдается. Тем не менее, реализация некоторых возможностей процессора Calc имеет свои особенности. Так, разработчики пакета большое внимание уделили его насыщению всяческими функциями, по числу которых он, похоже, даже превосходит Excel. Само собой разумеется, что в рабочий лист можно вставлять самые разнообразные графики и диаграммы, в том числе и трехмерные, причем

Рис. 5

Сводная таблица параметров чипсетов

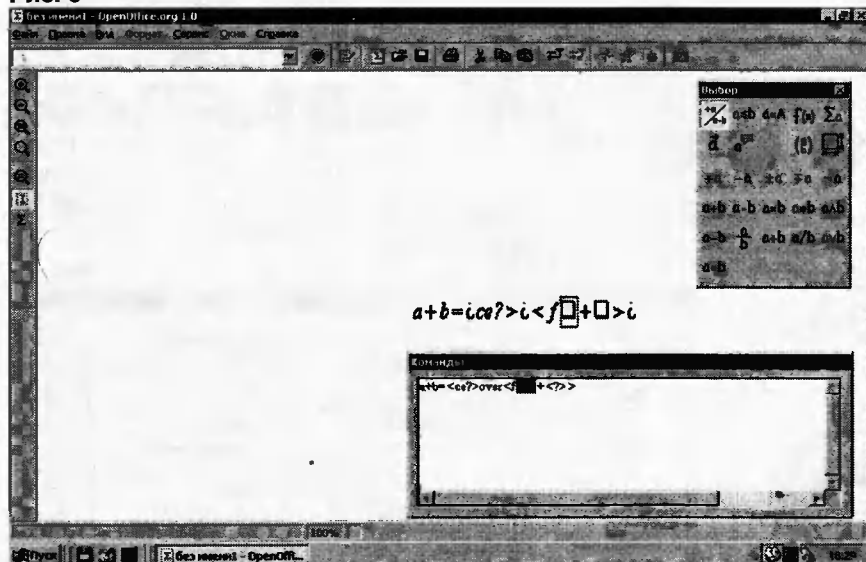
Чипсет	Частота FSB, МГц	Тип памяти	Максимальная объем памяти, Гб	Пиковая пропускная способность памяти, Гб/с	Встроенная графическая Адап	Внешняя Адап	Шина для SATA устройств	Пиковая пропускная способность шины, Мб/с	Число портов USB	Режимы ATA
Sis 735	100/133 (200/266)	PC2100/PC1600	1,5	2,1	нет	4x	MultiIOL	1200	6	100
Sis745	100/133 (200/266)	PC2700/PC2100/PC1600	1,5	2,7	нет	4x	MultiIOL	1200	6	100
Sis746	100/133 (200/266)	PC2700/PC2100/PC1600	1,5	2,7	нет	8x	MultiIOL	533	6 (2.0)	133
nForce 220	100/133 (200/266)	PC2100/PC1600	1,5	2,1	GeForce2 MX	4x	Hyper Transport	800	6	100
nForce 420	100/133 (200/266)	PC2100/PC1600	1,5	4,2	GeForce2 MX	4x	Hyper Transport	800	6	100
nForce 415	100/133 (200/266)	PC2100/PC1600	1,5	4,2	нет	4x	Hyper Transport	800	6	100
nForce 620	100/133 (200/266)	PC2700/PC2100/PC1600	1,5	4,2	GeForce2 MX	4x	Hyper Transport	800	6	100
nForce 615	100/133 (200/266)	PC2700/PC2100/PC1600	1,5	4,2	нет	4x	Hyper Transport	800	6	100
Radeon IGP 320	100/133 (200/266)	PC2100/PC1600		2,1	Radeon VE	4x	A-Link	266	6 (2.0)	100

3D-эффекты которых можно подбирать по своему вкусу. Не желая быть хуже MS Excel, не представляющего для себя жизни без собственного языка программирования Visual Basic for Applications, Calc обзавелся довольно похожим по своим основным возможностям, но несовместимым с ним собственным языком программирования StarBASIC.

Кроме всех текстовых форматов и HTML, Calc "понимает" электронные таблицы Excel и Lotus 1-2-3, а также dBase.

используя буквенные эквиваленты математических символов, выделить полученный текст и выполнить команду "Вставка объекта OpenOffice Math" (кнопка на панели инструментов или соответствующий пункт меню). Такой метод удобен (особенно для тех, кому имеет смысл держать условные обозначения всех математических символов в голове) для создания простейших формул, но по мере их усложнения такой способ ввода превращается в каторжный труд.

Рис. 6



Программа Math умеет конвертировать формулы из мейкрософтовского формата Equation Editor и даже поддерживает MathML.

OpenOffice.org Impress

Программа Impress предназначена для создания презентаций, и по своим функциональным возможностям практически один в один дублирует MS PowerPoint (рис.7).

Она имеет точно такой же инструментальный для рисования и всякого рода эффектов. Но кроме того — набор шаблонов как презентаций в целом (сгруппированных по назначению — научные, финансовые отчеты, бизнес-проекты), так и отдельных страниц. В качестве мастера, сопровождающего начинающих слайдотворителей с самых первых шагов и вплоть до завершающего кадра, выступает Автопилот (рис.8).

В состав слайдов презентации могут быть вставлены: текст, в том числе и многоколоночный, векторные и растровые рисунки, диаграммы, электронные таблицы и просто объекты (по OLE-технологии). Ну и, конечно, доступны всяческие эффекты переходов между слайдами и различные режимы воспроизведения слайд-шоу.

Возможности программы Impress по работе с файлами форматов, отличных от родного, вполне стандартны и ориентированы, прежде всего, на продукты от Microsoft.

OpenOffice.org Draw

С программой Impress тесно интегрирован векторный графический редактор Draw. От своих более или менее навороченных аналогов, в том числе и существующих самостоятельно, он отличается набором шаблонов для создания 3D-фигур. Причем, эти фигуры, соответствующие спецификации OpenGL, являются по-настоящему трехмерными, что выражается в возможности их вращения в любой плоскости. Редактирование созданных трехмерных объектов наиболее простым способом может осуществляться путем применения к ним набора готовых 3D-эффектов — изменения цвета, градиентных заливок, положения, интенсивности и цветовой гаммы источника освещенности, а также наложения текстур (рис.9).

Нет ничего удивительного в том, что, являясь программами одного класса, Draw по своим принципам работы несколько напоминает широко известный CorelDraw. Инструментальная панель слева позволяет создавать различные геометрические фигуры, в том числе и сложной формы, кривые Безье и ограниченные ими полигоны, самые различные линии и стрелки. Все замкнутые объекты могут быть залитыми, заштрихованными или пустыми. Здесь же, с панели инструментов, можно осуществлять различные трансформации объектов (вращение, деформацию, отражение и т.п.), их ранжирование или перемещение. Для любого графического редактора, особенно входящего в состав офисного пакета, весьма важна работа с текстом.

Рис. 7

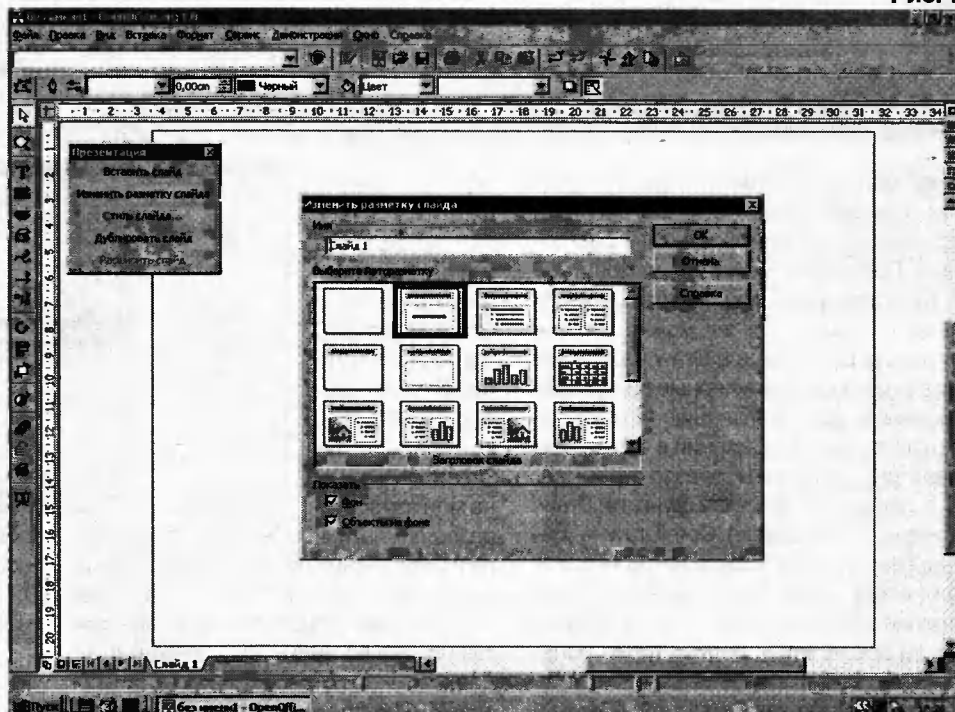
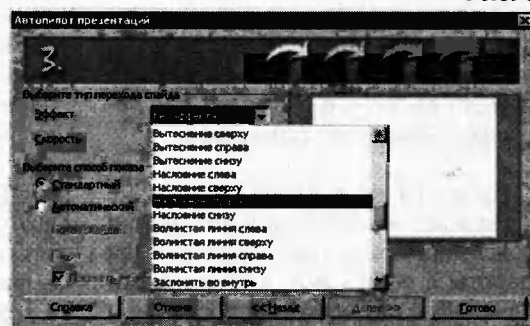


Рис. 8



И у Draw с этим все в полном порядке, тем более, что он, в его нынешней incarnation, вполне сносно научился работать и с кириллическими шрифтами.

Не буду долго и нудно перечислять все те файловые форматы, которые может

Рис. 9



открывать Draw, отмечу только, что в их число входят практически все распространенные, как векторные, так и растровые. Последние могут использоваться в качестве фона для векторных изображений. Точно так же можно экспортировать созданные в Draw изображения в наиболее популярные растровые форматы, но, к сожалению, возможности экспорта изображений в серьезные векторные форматы гораздо скромнее.

В целом, по богатству функций Draw, конечно, проигрывает CorelDraw — безусловному лидеру среди векторных графических редакторов. Однако в Draw вполне успешно можно сделать практически все те вещи, ради которых большинство любителей устанавливает себе такого монстра как CorelDraw. А учитывая его статус бесплатного приложения к офисному пакету, можно, не кривя душой, признать, что редактор Draw вполне достоин занять место в качестве основного векторного редактора для непрофессионального применения.

Ну и что же в итоге?

Подводя окончательный итог, можно констатировать — пакет программ для работы с офисными приложениями OpenOffice.org вполне удался. Свои основные функции — работу с текстовыми документами и электронными таблицами — он способен обеспечить достаточно качественно, ведь основные проблемы его предшественников — слабая поддержка кириллицы и не со-

всем полная совместимость с микрософтовскими офисными форматами, в OpenOffice.org 1.0 решена практически полностью. HTML Editor, конечно, не Dreamweaver, но, на мой взгляд, гораздо симпатичнее, чем MS FrontPage, да и Draw весьма привлекателен. Что касается остальных приложений, так ведь каждый может определить для себя сам, так ли сильно нужны ему презентации, да и как часто потребуются насыщать создаваемые документы сложными формулами или математическими выражениями. Тем более, что возможность выборочной установки реализована достаточно просто и наглядно.

На общем положительном фоне впечатления от пакета в целом, достаточно резко бросаются в глаза мелкие, но от этого не менее неприятные недостатки. Один из самых заметных — то, что работа в целом достаточно шустро, программы пакета буквально засыпают при старте, причем их неторопливость особо не зависит от мощности компьютера — везде тормоза. Кроме того, несмотря на поистине титанический труд программистов компании ALTlinux по русификации интерфейса программ пакета, иногда проскальзывают откровенные ляпы. Даже не

обращая внимания на такие хронические болячки русифицированных программ как неполный перевод отдельных терминов или целых строк, а также "обрезание" строки посреди фразы (рис.10), хочется посоветовать разработчикам в следующий раз соблюдать разумную меру в пе-

Рис. 10

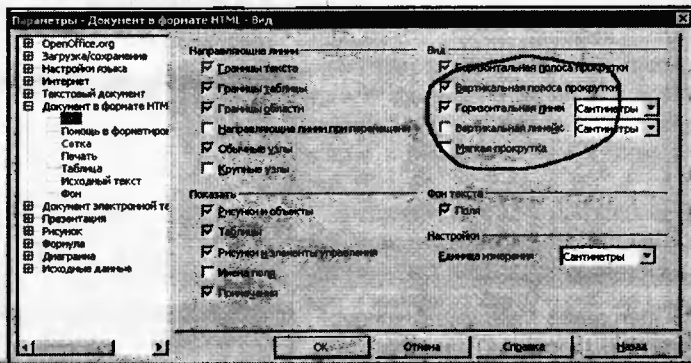
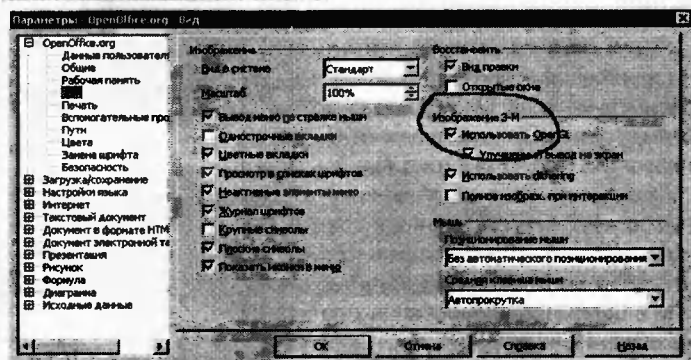


Рис. 11



реводе общеупотребительных терминов. Так, для обозначения трехмерности "3-D" выглядит куда предпочтительнее, чем "3-M" (рис.11).

Заканчивая с проблемами русификации, хочется заметить, что справка только на английском языке также вряд ли будет способствовать росту популярности пакета в нашей стране.

Какие бы ни высказывались мнения (порой, вполне обоснованные) о продуктах корпорации Билла Гейтса, тем не менее, Microsoft Office по праву считается образцом для офисных пакетов, уровня которого по всей сумме параметров не удалось достичь пока никому другому офисному пакету. Тем более отрадно отметить определенный успех OpenOffice, который сумел не только практически вплотную приблизиться к общепризнанному эталону, но и имеет достаточные перспективы для завоевания в своих последующих реинкарнациях "желтой майки" лидера в категории офисных пакетов.



Создание web-страниц

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

Знаете ли вы, что такое язык HTML и для чего он нужен? Приходилось ли вам когда-нибудь создавать web-страницы (файлы формата HTML)? Какие программы нужны для этого? Эта статья, прежде всего, предназначена тем, кто желает прикоснуться к увлекательно-му миру web-дизайна, кто только начинает осваивать создание web-страниц начального уровня сложности и желает постичь премудрости языка HTML.

Свой первоначальный опыт по созданию web-страниц я приобрел, используя текстовый редактор Word из пакета MS Office 97. А толчком к созданию web-страниц послужила непомерно выросшая коллекция ярлыков Интернета (ссылок, закладок), после того как у меня появился доступ к сети Интернет. Ярлыки Интернета — это файлы с расширением URL (Uniform resource locator — унифицированный указатель ресурса). Оказывается, все эти файлы можно разместить на одной web-странице, которую несложно создать самому. Какой в этом смысл, спросите вы? Дело в том, что URL'ы — файлы по размеру небольшие (сотни, а то и десятки байтов), и они очень неэффективно расходуют место на жестком диске, кластеры остаются незаполненными чуть ли не на 98%. Да и доступ к ресурсам Интернета становится более удобным, если "стартовать" с такой страницы — не нужно тратить время на поиски нужной закладки, все находится под рукой. Когда на моей web-странице URL'ам стало тесно (на ней поместилось не более 40 закладок), пришлось создавать страницы с сортировкой по темам. Освоение процесса создания web-страниц шло своим чередом, и меня, в общем-то, все устраивало. Но потом я установил у себя новый Word 2000 и вкусил все его прелести в плане создания файлов с расширением DOC.

Однако, было и досадное разочарование. Оказалось, что в Word 2000 файлы формата HTML получаются

уж слишком громоздкими, но возвращаться из-за этого к старому 97-му Word'у мне уже не хотелось. Вся "фишка" в том, что чем компактнее будет HTML-документ, тем быстрее он будет грузиться, да и места на жестком диске будет меньше занимать. В общем, передо мной встал вопрос — какой HTML-редактор использовать для создания несложных web-страниц и где его взять? Мне был нужен простой, удобный, компактный, русифицированный HTML-редактор. Microsoft FrontPage 98 меня явно не устраивал из-за своего большого размера (его инсталлятор "весил" 26,4 Мб), а вот его усовершенствованная версия — Microsoft FrontPage Express v.2.0 — мне вполне подошла. Тем более, что этот HTML-редактор создавал web-страницы даже еще более компактные, чем Word 97. Microsoft FrontPage Express входит в дистрибутив операционной системы Windows 9x/Me, и при желании его можно в любой момент установить. Однако вполне может возникнуть такая ситуация, когда в вашей версии Windows отсутствует Microsoft FrontPage Express, но он есть в другом дистрибутиве. Не устанавливая же из-за этого другую операционную систему!

Выход есть. Так как установка производится по протоколам, хранящимся в INF-файлах, нужно найти в дистрибутиве Windows, в архивных файлах с расширением CAB, файл **fxprsr16.inf** (для Windows 98 — это архив **precopry2.cab**). Извлекать нужные файлы из архивных CAB-файлов можно с помощью современных архиваторов, например WinRar v.2.71. Далее нужно скопировать извлеченный из архива файл **fxprsr16.inf** в отдельное место, например, на диск A: и можно начинать установку, следуя по маршруту "Пуск\Настройка\Панель управления\Установка и удаление программ\Установка Windows\Установить с диска". Можно просто скопировать Microsoft FrontPage Express, установленный на другом компьютере, и переписать его на

свой. После того как будет произведена установка Microsoft FrontPage Express, можно приступить к освоению программы и к созданию страниц HTML. Описывать в этой статье приемы и правила работы с программой Microsoft FrontPage Express я не буду по той причине, что это тема для отдельного разговора.

Создавать web-страницы можно не только с помощью программ-редакторов, которые умеют сохранять документы в формате HTML, но и используя обычный текстовый редактор, например, Блокнот (notepad.exe), с той только разницей, что для второго способа необходимо знание языка HTML.

Язык HTML (Hyper Text Markup Language — язык разметки гипертекста) — не единственный способ представления информации в World Wide Web, но он является связующим звеном между всеми web-документами. Без языка описания страниц HTML Всемирная Паутина не сможет существовать. Стандарт языка HTML разрабатывался с учетом независимости от аппаратной реализации компьютеров, и, при всей своей простоте, HTML — очень мощный язык. Если просмотреть структуру HTML-файла в простом текстовом редакторе, можно увидеть, кроме текста, множество непонятных слов и символов, заключенных в угловые скобки. Это и есть так называемые теги HTML. По своей сути это операторы языка HTML, которых насчитывается несколько десятков. Каждый оператор заключается в угловые скобки и интерпретируется браузером Интернета. Например, если web-браузер встречается в странице HTML тег **<Center>**, то текст, который следует за этим тегом, будет воспроизведен по центру окна, независимо от того, какая величина диагонали экрана применяемого монитора, какое разрешение установлено в данный момент и каков текущий размер окна программы. Из всего вышесказанного можно сделать вывод, что HTML-файлы являются простыми файлами с текстовой структурой, с той лишь разницей, что, кроме текста, присутствуют еще и теги. Так что можно использовать любую программу, которая умеет сохранять файлы в простом текстовом фор-

мате, только расширение выходного файла надо указать *.htm.

Применение разметочных тегов — это все, что отличает HTML-файлы от простого текстового формата. Разметочные теги широко используются в HTML, и они предоставляют возможность управлять форматированием текста, создавать связи с другими документами, а также встраивать в web-страницы изображения и объекты мультимедиа (звуки, музыку, анимацию, видео). Картинки и все остальное, конечно же, не вставляются в прямом смысле в текст, просто в тексте указаны инструкции для web-браузеров, как эти файлы называются, где их взять, как расположить на экране или каким соответствующим образом их обработать. Кроме того, современные web-документы могут содержать и так называемые активные компоненты, которые по своей сути являются микропрограммами, работающими на компьютере пользователя во время просмотра им web-документа. Короче говоря, разметочные теги являются ключевыми элементами для создания HTML-страниц. При записи тегов регистр символов значения не имеет. Например, тег **<Body>** может быть набран как **<BODY>**, **<body>**, или даже **<BoDy>**. Создавать разметочные теги HTML очень легко. Они состоят из левой открывающейся угловой скобки, собственно имени тега и правой закрывающейся угловой скобки. Для того чтобы текст был выделен полужирным шрифтом, вставьте тег **** перед началом текста и тег **** после окончания выделенного текста. Из этого можно сделать следующий вывод — окончание действия тега отличается от начала действия наличием косой черты — символом слэш (/). Теперь при просмотре web-браузером HTML-файла, текст, расположенный между тегами **** и ****, будет выделен полужирным шрифтом. Практически все теги в HTML имеют начало и окончание своего действия, но есть и исключения из этого правила, например, тег ****.

Теперь, когда вы узнали, как записывать теги, следующим шагом в освоении азов web-дизайна будет то, как устанавливать эти теги вместе, чтобы создать простой HTML-документ.

HTML-документ делится на две составляющие: **<Head>** — заголовок, и **<Body>** — собственно "тело" самой страницы. Раздел заголовка **<Head>** содержит в себе чисто служебную информацию, в которой указывается, например, название документа и кодовая страница. Сам текст документа и все, что будет изображено при просмотре web-браузером, размещено в разделе **<Body>**.

Для удобства создания будущих простых HTML-документов можно использовать шаблон. Вместо того, чтобы начинать каждый раз с чистого листа, создавать документ по этому шаблону гораздо быстрее и удобнее. Для этого нужно открыть шаблон, заменить название, вставить свой текст и сохранить файл под новым именем.

Ниже приводятся описания некоторых наиболее распространенных и часто употребляемых тегов языка HTML:

- **<Html>...</Html>** — эти теги несут информацию о том, что файл имеет структуру HTML-документа, и обозначают начало и конец любого HTML-документа;

- **<Title>...</Title>** — определяет название документа. Слово, помещенное между этими тегами, и есть название документа;

- **<Body>...</Body>** — определяет раздел собственно "тела" документа. Этот раздел включает в себя весь текст документа и теги разметки. Может использоваться с атрибутами **Background="..."** и **Bgcolor="..."**;

- **<A>...** — этот тег при использовании с атрибутом **Href** включает внешнюю гиперссылку на web-документ, а когда используется с атрибутом **Name**, то указывает на внутреннюю гиперссылку в HTML-документе (ссылка по якорю);

- **...** — определяет применяемый шрифт для прилагаемого текста. Используется с атрибутами **Color="..."** и **Face="..."** **Size="..."**;

- **<I>...</I>** — отображает текстовый блок курсивом;

- **<U>...</U>** — отображает текстовый блок подчеркиванием;

- **<Center>...</Center>** — все, находящееся в HTML-

документе между этими тегами, при просмотре будет расположено по центру окна;

- **<P>...</P>** — устанавливает маркеры начала и конца параграфа в HTML-документе. Используется с атрибутом **Align="..."**;

- **...** — используется для упорядочивания списков гиперссылок в HTML-документе;

- **** — вставляет изображение в HTML-документ. Используется с атрибутами **Src="..."** **Width=...** **Height=...** **Alt="..."**.

На этом я закончу свое легкое прикосновение к основам языка HTML, ибо нельзя объять необъятное. Для того чтобы досконально описать все возможности языка HTML, потребовалась бы целая книга. Поэтому я привел лишь самые начальные сведения, которые будут необходимы для того, чтобы иметь представление о языке HTML, и что нужно знать, чтобы начать практически осваивать создание web-страниц.

И в заключение несколько напутственных слов. Интернет стремительно растет и изменяется, и служба World Wide web является его самым быстроразвивающимся сегментом. Это развитие неразрывно связано с языком HTML, с новыми технологиями, такими как апплеты Java и элементы ActiveX. Так что, уважаемые пользователи Интернета, изучайте язык HTML, основы web-дизайна, приобретайте соответствующую литературу. Полученные знания помогут вам создать свои неповторимые home pages, потом их можно будет разместить в Интернете и тем самым заявить о своем существовании всему миру (не страшно?). За профессией web-дизайнера — большое будущее.

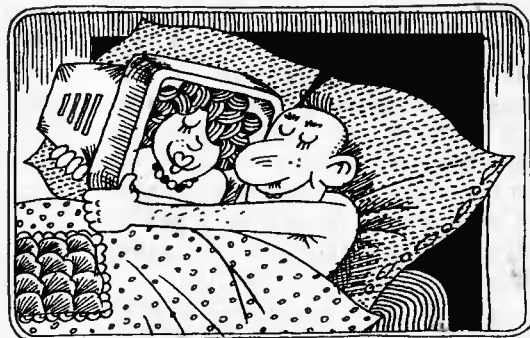


Рисунок О.Попова

Копирование файлов из Интернета

С.ГРИНЧУК,

E-mail: grinchuk@nihe.niks.by

Данная серия статей предназначена прежде всего для тех пользователей сети Интернет, кто уже достаточно уверенно чувствует себя на пространстве всемирной паутины, перешел "на ты" в общении с браузером и хочет несколько расширить свои познания в области Интернет-технологий, с тем чтобы оптимизировать свою работу в сети. Большинство из нас рассматривает Интернет как огромную библиотеку со всевозможной информацией. Поэтому первый и самый важный вопрос, который обычно всех волнует — как эту информацию перекачать на свой компьютер. Конечно, загрузить из Интернета любой файл можно с помощью ставшего уже таким родным и знакомым браузера. Однако это не самый оптимальный путь, особенно в случае больших по размеру файлов. Дело в том, что для загрузки крупного файла может понадобиться слишком много времени, а если произойдет разрыв связи, процедуру копирования придется начинать сначала. В подобных случаях неоценимую помощь вам окажут специальные программы, позволяющие автоматизировать процесс получения файлов из Интернета. Подобные программные продукты часто называют программами загрузки и докачки файлов. Они позволяют определить список файлов (точнее, их адресов в Интернете) для последующего копирования из сети и докачки их в случае обрыва связи. Следует отметить, что адреса могут указывать как на пространство WWW, так и на архивы FTP. Таким образом, если качество вашего подключения к Интернету оставляет желать лучшего, а вы нуждаетесь в быстрой и гарантированной загрузке файлов из сети, то программы загрузки и докачки файлов специально для вас!

В качестве примеров подобного класса программ можно привести GetRight (адрес разработчика в Интернете — <http://www.getright.com>), Net Vampire (<http://www.netvampire.com>) и FlashGet (<http://www.amazesoft.com>), две последних программы имеют возможность перехода к русскоязычному интерфейсу. На практике самой эффективной и удобной в использовании показала себя программа **FlashGet**. Сами разработчики программы называют FlashGet **менеджером загрузки файлов** (Download Manager). Начиная с версии 1.2, FlashGet

— условно-бесплатная программа. Загрузить программу, а также получить более подробную информацию о ней можно на сайте разработчика.

Итак, чем же хороша эта программа? Основное достоинство FlashGet, которое выгодно отличает ее от всех остальных программ подобного рода — возможность автоматически разбивать файлы на части и загружать все части параллельно (в случае, если сервер, с которого производится загрузка, поддерживает докачку и, соответственно, подобную разбивку на части). При этом для каждой части файла открывается отдельное соединение, и в результате ваш канал эксплуатируется гораздо более эффективно, а сама загрузка файлов значительно ускоряется. FlashGet поддерживает разбивку получаемых файлов максимум на 10 частей, к тому же, позволяет выполнять до 8 заданий одновременно. Кроме того, FlashGet поддерживает загрузку по расписанию, что позволяет вам экономить свое время и деньги, закачивая файлы в те часы, когда загруженность канала связи не так велика, либо ниже тариф. Как уже упоминалось выше, FlashGet обладает функцией многоязычности — выбора языка интерфейса программы. Среди поддерживаемых языков, входящих в состав FlashGet — английский, болгарский, венгерский, голландский, греческий, датский, иврит, испанский, итальянский, каталонский, корейский, упрощенный китайский, традиционный китайский, немецкий, норвежский, польский, португальский (brasil), румынский, русский, сербский (латинский), словацкий, словенский, тайский, турецкий, украинский, французский, чешский, шведский, эстонский, японский. Конечно,

это далеко не все возможности FlashGet, однако надеемся, что даже перечисленные достоинства не оставят вас равнодушными, и вы решите вместе с нами более подробно познакомиться с этой чудесной программой.

Начнем мы изучение этого программного продукта с обсуждения вопросов запуска приложения и знакомства с основными элементами интерфейса. Заметим, что рассматривать мы будем FlashGet 1.3.

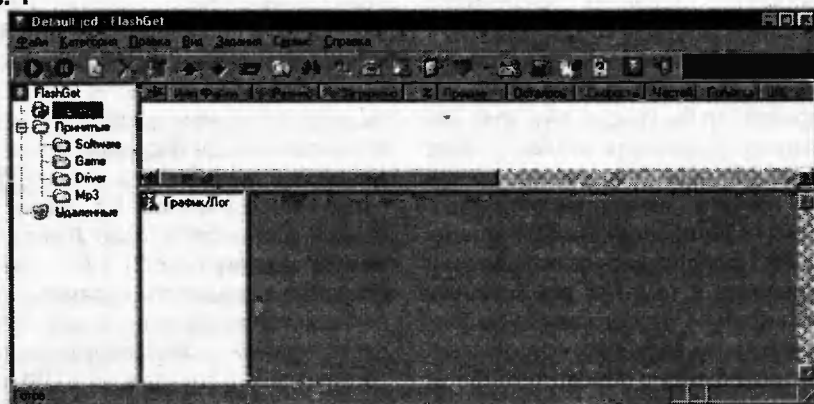
1. Запуск FlashGet. После установки программы на ваш компьютер эта операция выполняется стандартным образом — необходимо последовательно выполнить команды **Пуск/Программы/FlashGet/FlashGet** или же сделать двойной щелчок мышью по соответствующей пиктограмме на рабочем столе MS Windows.

2. Интерфейс FlashGet. Управление основными элементами интерфейса.

Рассмотрим основные элементы интерфейса программы, а также поговорим о том, как управлять отображением этих элементов на экране.

Рабочее окно FlashGet представлено на **рис.1**. В самом верху экрана расположена строка заголовка, где указано название программного продукта и наименование используемой по умолчанию базы данных загрузки **Default.jcd** (вся информация о закачиваемых файлах содержится в файле **Default.jcd**, который и называется базой данных загрузки). Здесь же расположены кнопки управления окном FlashGet: свернуть, восстановить, закрыть. Ниже строки заголовка расположена строка меню, а под ней — панель инструментов. И в самом низу рабочего окна расположена строка

Рис. 1



состояния. Помимо этих, стандартных для любого приложения, элементов интерфейса, рабочее поле окна программы разделено по вертикали на две части. В левой части размещается список категорий, по которым распределяются файлы. Все новые и незаконченные задания на загрузку хранятся в категории **Задания**. Завершенные задания FlashGet размещает в категории **Принятые**. Удаленные задания хранятся в категории **Удаленные**. Для окончательного удаления какого-либо задания на загрузку, его необходимо удалить из категории **Удаленные**. Для каждой категории FlashGet позволяет определить папку на диске вашего компьютера. В этом случае загружаемый файл, отнесенный к определенной категории, будет помещаться в заданную вами папку. Для удобства работы, внутри каждой категории можно создавать дополнительные подкатегории.

В правой части рабочего поля окна отображается содержимое выбранной категории (в данном случае, категории **Задания**). Верхняя половина содержит список заданий, в котором можно просмотреть имя загружаемого файла, его размер, какая часть файла уже загружена, скорость загрузки, адрес файла (URL) и т.д. Настройка столбцов с информацией о заданиях осуществляется с помощью команды **Вид/Столбцы...** Однако следует отметить, что подобная настройка столбцов выполняется только для категории **Задания**. В нижней половине окна отображается более подробная информация о каждом задании загрузки. Например, для категории **Задания** отображается область **График/Лог**, которая показывает визуальную карту загрузки, подробности установления связи с сервером, сообщения об ошибках и другие события. С помощью этой области можно проконтролировать загрузку каждой части файла (Jet) в отдельности. Включение/выключение отображения подробной информации о заданиях производится с помощью команды **Вид/Подробно**.

Мы уже говорили о том, что FlashGet обладает функцией выбора языка интерфейса. Чтобы убедиться в этом, достаточно выполнить команду **Вид (View)/Language**. Для примера попробуйте изменить язык интерфейса программы на английский: **Вид/Language/English**. Прогулявшись по всем меню и убедившись в том, что все команды действительно отображаются на английском языке, вернитесь к русскоязычному интерфейсу с помощью команды

View/Language/Russian.

Для удобства желающие могут включить отображение подписей к кнопкам на панели инструментов. Для этого необходимо выполнить команду **Вид/Панель инструментов** и установить флажок **Отображать подписи к кнопкам**. Однако при этом может оказаться, что далеко не все кнопки поместились на панели инструментов. Выполнив ту же самую команду, вы можете отказаться от отображения подписей. Настройка же самой панели инструментов, определение набора отображаемых кнопок выполняется по команде **Вид/Панель инструментов/Кнопки...**

Хотелось бы отметить, что рассматриваемая нами версия FlashGet (1.3) имеет некоторые возможности по дополнительному оформлению рабочего окна программы. Выполнив команду **Вид/Сетка**, вы перейдете к табличному представлению параметров заданий в области просмотра. А команда **Вид/Оформление** позволяет задать цветовое решение для рабочего окна программы.

Вот и все, что касается настройки основных элементов интерфейса. А сейчас поговорим о том, каким образом добавляются новые задания на загрузку файлов и как они выполняются.

3. Добавление новых заданий загрузки, настройка и выполнение заданий.

Перед началом копирования файлов из Интернета необходимо прописать параметры вашего подключения к сети. Для этого выполните команду **Сервис/Настройки...** или воспользуйтесь кнопкой  на панели инструментов, и на вкладках **Прoxy**, **Дозвон** (в зависимости от способа вашего подключения) укажите параметры подключения к Интернету.

В качестве примера загрузим файл <http://www.nihe.niks.by/rus/courses/computers.shtml>, содержащий информацию о проводимых в РИВШ БГУ курсах повышения квалификации по компьютерным технологиям. Для определения нового задания загрузки выполните команду **Задания/Новое задание** или щелкните по кнопке  на панели инструментов. На экране появится диалоговое окно **Добавить новую загрузку** (рис.2), в котором необходимо определить параметры выполнения задания. В поле **URL** указывается полный адрес копируемого файла. Для нашего случая в поле **URL** вве-

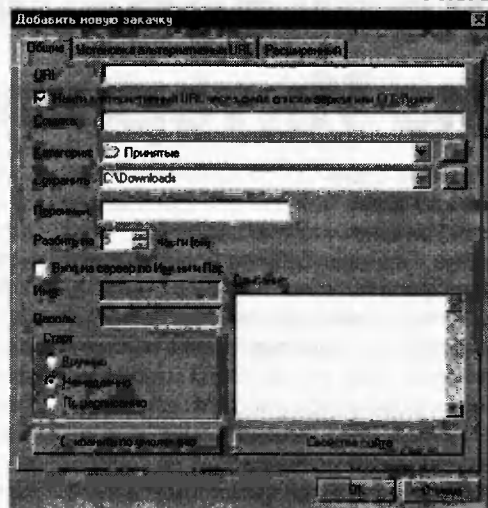


Рис. 2

дите: <http://www.nihe.niks.by/rus/courses/computers.shtml>. Поле **Ссылка** оставьте пустым, с тем чтобы FlashGet автоматически заполнил его за вас (дело в том, что некоторые серверы требуют указания этого параметра для запуска загрузки). Поле **Категория** позволяет определить категорию FlashGet, в которую автоматически переместится файл после выполнения задания. По умолчанию это категория **Принятые**, которую мы и предлагаем вам оставить. В поле **Сохранить** указывается папка на диске вашего компьютера, в которую будет помещен загруженный файл. Для выбора папки щелкните по кнопке , расположенной рядом с полем **Сохранить**, и в качестве папки для принятых файлов выберите свою рабочую папку. Не забудьте подтвердить свой выбор щелчком по кнопке **OK**. Поле **Переимен.** позволяет проинформировать сохранение загружаемого файла под другим именем. Оставьте в этом поле собственное имя загружаемого файла **computers.shtml**. В поле **Разбить на** необходимо указать, на сколько частей FlashGet должен разбить загружаемый файл. Напомним, что подобная автоматическая разбивка файла на части и загрузка всех частей параллельно производится только в том случае, если сервер, с которого копируется файл, поддерживает докачку, и если загружаемый файл достаточно велик, т.к. по умолчанию минимальный размер одной части — 50 Кб. Разработчики программы не рекомендуют разбивать файлы больше чем на 10 частей, т.к. это может привести либо к резкому уменьшению скорости загрузки, либо к разрыву соединения с сервером. В большинстве случаев достаточно разбить загружаемый файл на 3...5 частей. Поэтому в поле **Разбить на** оставьте ва-

риант 5. Раздел **Вход на сервер по Имени и Паролю** используется в том случае, если сервер, с которого производится загрузка, требует идентификации пользователя (ввода имени пользователя и пароля). В большинстве случаев (в том числе, и в нашем) этот раздел заполнять не требуется. В поле **Описание** можно поместить дополнительные комментарии, с тем чтобы по прошествии времени не забыть, что именно за файл загружался. Оставим это поле не заполненным. И последний раздел **Старт** позволяет указать, как именно будет выполняться новое задание. При выборе переключателя **Вручную** FlashGet только добавляет новое задание в список категории **Задания**, но загрузку файла не начинает. Переключатель **Немедленно** позволяет сразу же начать выполнение нового задания, а переключатель **По расписанию** переводит новое задание в ждущий режим для последующей загрузки в запланированное время. Установите в этом разделе переключатель **Вручную** и щелкните по кнопке **ОК** для закрытия диалогового окна **Добавить новую зачатку**.

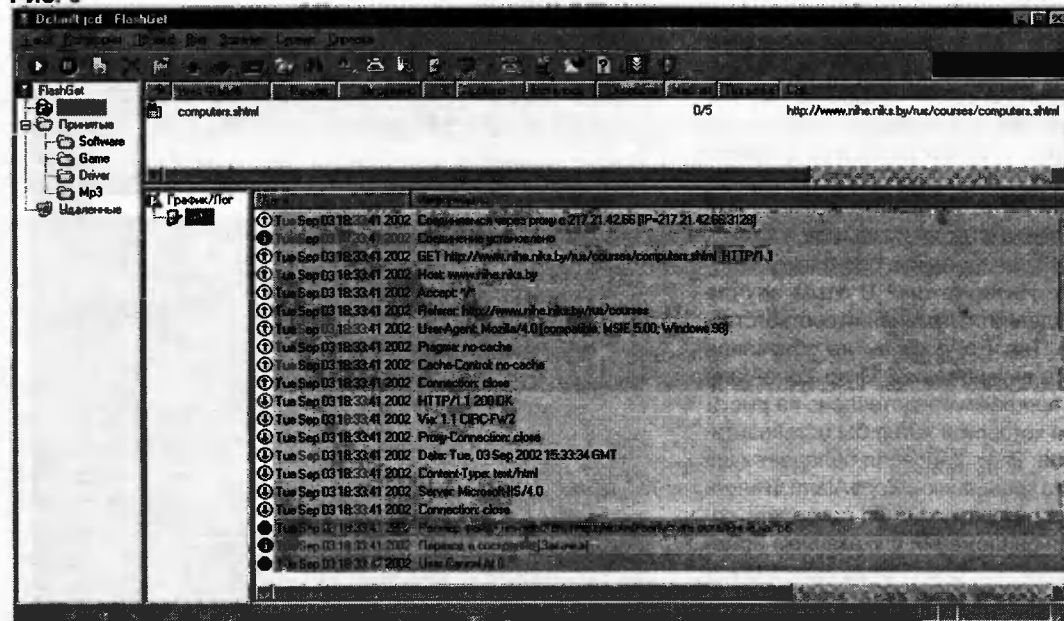
Обратите внимание, что новое задание при этом будет просто добавлено в список категории **Задания**. Чтобы начать выполнение задания, щелчком мыши выделите имя файла в области просмотра параметров задания и выполните команду **Задания/Старт** или же воспользуйтесь кнопкой  на панели инструментов. В любой момент можно остановить выполнение задания, выполнив команду **Задания/Приостановить** или щелкнув по кнопке  на панели инструментов. Поскольку загружаемый нами файл слишком мал (размер

файла ≈9 Кб, а по умолчанию минимальный размер одной части — 50 Кб), разбивка файла на несколько частей не была произведена (в области **График/Лог** присутствует информация только об одной части — **Jet 1**). Для просмотра информации о зачке этой части, необходимо выполнить щелчок мышью по названию части **Jet 1**, и вы получите подробный протокол событий (рис.3). Чтобы продолжить выполнение задания, достаточно еще раз выполнить команду **Задания/Старт** или же воспользоваться кнопкой  на панели инструментов. Когда файл будет полностью загружен, задание из категории **Задания** будет перемещено в категорию **Принятые**. На левой панели перейдите в категорию **Принятые** и выделите загруженный файл **computers.shtml**. При этом в области, расположенной ниже, отобразятся параметры загруженного

Табл. 1

Действие	Реализация
Запуск FlashGet	Пуск/Программы/FlashGet/FlashGet или двойной щелчок мышью по соответствующей пиктограмме на рабочем столе MS Windows
Изменение языка интерфейса программы	Вид (Вид)/Language/Russian
Включение/выключение отображения подписей к кнопкам на панели инструментов	Вид/Панель инструментов/установить/снять флажок Отображать подписи к кнопкам
Настройка панели инструментов	Вид/Панель инструментов/Кнопки...
Настройка области просмотра параметров заданий	Вид/Столбцы.../установить нужные флажки
Включение/выключение отображения подробной информации о заданиях	Вид/установить/снять флажок Подробно
Настройка параметров подключения	Сервис/Настройки... или кнопка  /вкладки Ргоху, Дозвон
Добавление нового задания загрузки	Задания/Новое задание или 
Начать выполнение задания или продолжить выполнение после обрыва связи	Выделить задание в списке заданий категории Задания/Задания/Старт или 
Остановка выполнения задания	Выделить задание в списке заданий категории Задания/Задания/Приостановить или 
Завершение работы с FlashGet	Файл/Выход или 

Рис. 3



это было сделано с целью помешать пользователю случайно закрыть программу во время загрузки. Поэтому чтобы полностью завершить работу с FlashGet, необходимо выполнить команду **Файл/Выход** или воспользоваться кнопкой  на панели инструментов.

На этом наше первое знакомство с программой мы завершаем. Основные операции, освоенные нами на этом занятии, приведены в табл.1.

До новых встреч!
(Продолжение следует)

Выбираем персональный файрволл

Ю.ЛЕВИН,
E-mail: yl@tut.by

Файрволлы или межсетевые экраны обязательно применяются в локальных и корпоративных сетях, имеющих выход в Интернет. Их основное назначение — фильтрация трафика с целью защиты от возможных атак извне, а также ограничение доступа к некоторым ресурсам изнутри защищаемой сети для различных групп пользователей. В этом случае функции межсетевого экрана могут выполняться как аппаратно, так и программно.

Подобные решения существуют также и для индивидуальных пользователей. Это программные средства, позволяющие фильтровать трафик в зависимости от адреса и порта, а также разграничивать доступ к Интернету для различных программ. Такие программы бывают как полностью бесплатные, так и платные. В данной статье мы рассмотрим бесплатные версии программ — ZoneAlarm, Sygate Personal Firewall и Outpost Firewall. Все эти три программы поставляются как в бесплатном варианте, с ограниченными возможностями, так и в платном — с дополнительными возможностями, но, на мой взгляд, даже возможностей бесплатных версий с лихвой хватит для большинства индивидуальных пользователей Интернета.

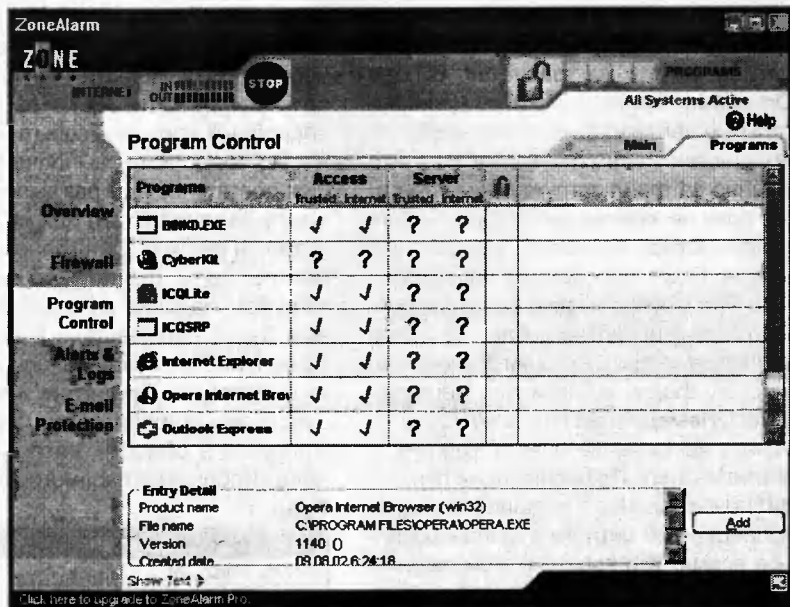
Первая программа, на которой мы остановимся, это **ZoneAlarm**. Она может быть получена с сайта <http://www.zonelabs.com/>, размер дистрибутива — 3,5 Мб (для версии 3.1.291). Данная программа проста в использовании и не требует никакой настройки. После установки на компьютер, ZoneAlarm будет запрашивать разрешение на доступ к удаленным ресурсам для каждой программы, пытающейся это сделать. Информацию о таких программах ZoneAlarm может накапливать, поэтому у пользователя не будет надобности каждый раз разрешать доступ в Интернет для той или иной программы, например, для браузера. Для каждой программы пользователь может либо разрешить, либо запретить доступ в Интернет. Помимо общего контроля над программами, работающими в Интернете, ZoneAlarm позволяет определять, может ли та или иная программа работать как сервер, т.е. отвечать на входящие соединения из Интернета. Такие соединения возможны при использовании активного FTP, а также в ICQ, например, при передаче файлов. Но если ZoneAlarm сообщил, что незнакомая программа ожидает входящего соединения, то это повод задуматься, не попал ли к вам "троянский конь". Многие "троянские кони", а также вирусы способны после попадания на компьютер принимать данные из Интернета или же сами инициировать передачу данных с вашего компьютера. А что, если "троянский конь" или вирус внедрился в программу, которой разрешено работать с Интернетом? В этом случае ZoneAlarm выдаст предупреждение об изменении соответствующего исполняемого файла. Так что, если вы не обновляли версию программы, то будьте внимательны. Окно ZoneAlarm со списком контролируемых программ представлено на рис. 1.

Следующая программа, на которой я хотел бы остановиться — **Sygate Personal Firewall**. Этот файрволл обладает куда большими возможностями по сравнению с ZoneAlarm, и вполне устроит более опытных пользователей. Он позволяет создавать правила доступа для приложений с указанием номеров портов и адреса, к которым это приложение может получить доступ, позволяет записывать активность программ в лог-

файл и многое другое. Но обо всем по порядку.

Размер дистрибутива программы — 5 Мб, он может быть получен с сайта <http://www.sygate.com>. Принцип работы — такой же, что и у ZoneAlarm — при попытке получения програм-

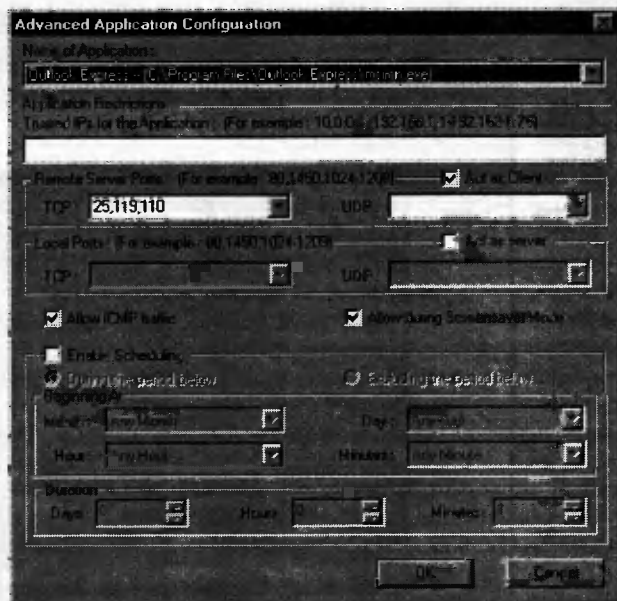
Рис. 1



мой доступа к сети, необходимо дать ей разрешение. По умолчанию программа получает полный доступ, однако при необходимости можно ограничить входящий/исходящий трафик, номер порта, а также срок действия данного правила. Так, на рис. 2 показан диалог создания правил для почтового клиента Outlook Express. Он может работать с почтовыми ящиками по протоколам SMTP и POP3 (порты 25 и 110), а также с группами новостей Usenet по протоколу NNTP (порт 119).

Помимо правил для отдельных приложений, Sygate Personal Firewall позволяет создавать общие правила для системы, т.е. определять ограничения, применяемые ко всем программам, даже если для них не создано никаких собственных правил. К преимуществам персонального файрволла от Sygate я бы отнес наличие графического индикатора общей скорости передачи данных (начиная с 5-й версии) по всем программам. По нему можно в любой момент узнать реальную скорость пере-

Рис. 2

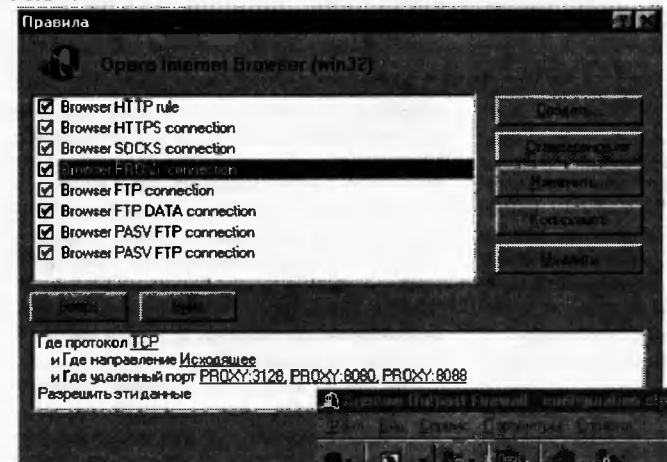


дачи данных через вашего провайдера. Из недостатков можно отметить невозможность сохранения создаваемых правил, так что после переустановки системы или файрволла, правила придется воссоздавать заново.

И наконец, третий файрволл, которому я отдаю предпочтение — **Agnitum Outpost Firewall**. Эта программа сочетает в себе достоинства двух рассмотренных выше программ — легкость настройки ZoneAlarm и, в то же время — гибкость Sygate Firewall. Это достигается, в первую очередь, за счет применения групп предустанавливаемых правил. В отличие от Sygate Firewall'a, здесь каждой программе взначается не одно правило, а несколько групп правил. Каждая группа правил позволяет программе устанавливать соединения через определенные номера портов. Таких групп, в зависимости от назначения программы, может применяться несколько, причем группы можно исключать из использования, не удаляя самих правил! На рис.3 показаны группы правил для Opera Internet Browser.

Выделенная группа правил "Browser PROXY connection" позволяет устанавливать исходящие соединения по протоколу TCP с HTTP-прокси-серверами, имеющими любой адрес, с использованием портов 8080, 3128 и 8088. Как видно, такие настройки позволяют работать с любым прокси-сервером без дополнительной настройки правил. Другие группы правил позволяют работать по протоколам HTTP, FTP и HTTPS, то есть использовать браузер без каких-либо ограничений. И самое главное преимущество Outpost Firewall состоит в том, что все эти многочисленные правила не нужно создавать самому для каждой программы — Outpost Firewall может использовать предустановленные шаблоны для наиболее распространенных интернет-приложений, таких как браузер, менеджер загрузок, почтовый клиент и т.д. Показанные на рис.3 правила сгенерированы автоматически на основе шаблона правил для браузера.

Помимо простоты настройки правил для программ, Outpost Firewall обладает многими другими достоинствами, такими как



просмотр информации по текущим соединениям в реальном времени, а также использование плагинов.

На рис.4 показан общий вид рабочего окна файрволла с отображенными соединениями. По каждому соединению можно просмотреть тип протокола, номера портов, объем переданных и полученных данных, а также скорость их передачи. К сожалению, нет

возможности просмотреть суммарную скорость по нескольким соединениям, что было бы неплохо для программ, устанавливающих сразу несколько соединений.

В стандартную инсталляцию файрволла входят также 5 плагинов.

Плагин "**Реклама**" может выполнять блокировку рекламных баннеров, основываясь как на адресах баннеров, так и на их возможных размерах. В плагине есть список адресов баннеров, однако при необходимости в него можно добавлять адреса тех баннеров, которые по умолчанию не блокировались. А вот блокировкой баннеров по размеру следует пользоваться аккуратно. У меня эта функция вообще отключена, так как она нередко блокировала картинки, по размеру совпадающие с традиционными размерами баннеров, но на самом деле не являющиеся рекламой.

Плагин "**Содержание**" позволяет блокировать загрузку сайтов из составленного пользователем списка. Он отчасти дублирует подобную функцию многих браузеров.

Плагин "**DNS**" выполняет функции DNS-кеша — он накапливает информацию об IP-адресах посещенных хостов и при следующем их посещении использует ее, таким образом избавляя программы от необходимости лишней раз обращаться к DNS-серверу.

Плагин "**Активное содержимое**" позволяет задать правила обработки скриптов, cookies и всплывающих окон при посещении сайтов. Например, запрет использования всплывающих окон может быть полезен при посещении сайтов на бесплатных хостингах, где всплывающие окна нередко используются для показа рекламы. Однако следует отметить, что если в таком окне будет содержаться полезная информация, то ее нельзя будет увидеть.

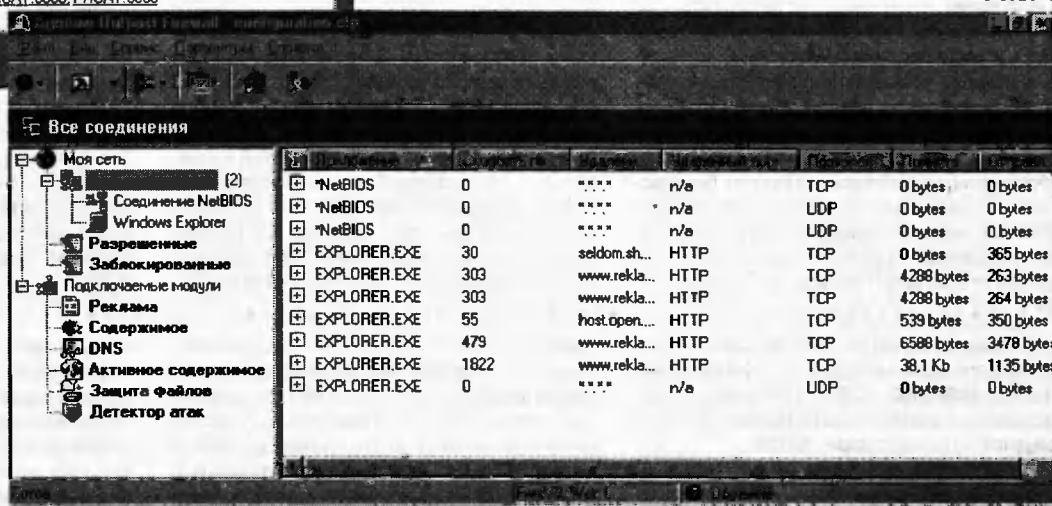
Плагин "**Защита файлов**" позволяет сообщать пользователю о получении по электронной почте писем, содержащих файлы заданных типов, а также автоматически их переименовывать. Это позволяет не допустить запуска вирусов, распространяющихся по почте.

Функция плагина "**Детектор атак**" понятна из его названия. С его помощью можно обнаруживать DoS-атаки либо сканирование портов и своевременно блокировать трафик с атакующего компьютера.

Outpost Firewall может быть получен с сайта www.agnitum.com, размер дистрибутива — 2,5 Мб.

В заключение можно отметить, что все рассмотренные программы по защите компьютера при работе в Интернете по умолчанию включают общие настройки, необходимые для нормального функционирования соединения с провайдером — они позволяют использовать определение IP-адреса для доменных имен, а также выполнять динамическое назначение IP-адреса, применяемое провайдерами. Поэтому они могут начинать работу сразу же после установки. Так что выбирайте файрволл на свой вкус, и безопасной вам работы!

Рис. 4





жит 221 млн. транзисторов (Pentium 4 имеет 55 млн. транзисторов). Тактовые частоты выпускаемых моделей — 900 МГц и 1,2 ГГц. По сравнению с Pentium 4, эти цифры не слишком впечатляют, но, говоря о серверных процессорах (именно таково назначение новой ИМС), необходимо отметить, что



серверы традиционно не гонятся за высокой тактовой частотой процессора, поскольку для них гораздо важнее стабильность работы и отказоустойчивость. Сервер работает 24 часа в сутки годами, выполняя порой одно приложение, которое, собственно, может и не требовать запредельной скорости обработки данных и запредельной же тактовой частоты процессора.

Однако производительность процессора определяется не только (и не столько) его тактовой частотой. Правильно ли считать, что процессор с тактовой частотой 2 ГГц обязательно будет более производительным, чем процессор с частотой 1 ГГц? Критерием оценки производительности процессоров может служить только скорость выполнения ими определенного приложения. Но разным процессорам, в зависимости от их внутренней архитектуры, может потребоваться различное количество машинных тактов для выполнения одной и той же программы. К примеру, процессор с тактовой частотой 1 ГГц может выполнить некоторую программу за 10 тыс. тактов, а процессор с тактовой частотой 2 ГГц — за 40 тыс. тактов. Таким образом, первому процессору потребуется для выполнения программы в два раза меньше времени, чем второму. Следовательно, первый процессор на данном приложении будет более высокопроизводительным, чем второй, несмотря на то что тактовая частота первого процессора меньше, чем второго. Поэтому, говоря о производительности процессоров, следует отчетливо понимать, что она определяется, во-первых, конкретным приложением (насколько хорошо приложение оптимизировано для данного процессора), во-вторых, архитектурой процессора, то есть количеством операций, выполняемых за один цикл, и, в-третьих, тактовой частотой самого процессора.

Говоря о процессоре Intel Itanium 2, следует отметить, что его архитектура кардинально отличается от процессора Intel Pentium 4 или, к примеру, от серверного процессора Intel Xeon. Напомним, за счет чего удалось столь значительно повысить тактовую частоту в процессоре Intel Pentium 4, и почему в процессоре Intel Pentium III максимальная тактовая частота не превышала 1 ГГц.

В любом процессоре выполнение программных команд разделяется на отдельные мелкие шаги, реализуемые в конвейере выполнения инструкций. Чем длиннее конвейер, тем более элементарные операции совершаются при каждом шаге, и тем меньшая производительность приходится на каждый шаг конвейера. Однако именно при длинном конвейере удается значительно

повысить тактовую частоту — чем больше количество стадий, не которые рассчитан конвейер, тем меньше работы приходится на каждый отдельный такт, и там короче этот такт. В процессоре Intel Pentium III длине конвейера составляет 12 стадий, а в процессоре Intel Pentium 4 используется беспрецедентно длинный конвейер в 20 стадий. Именно это обстоятельство и позволяет значительно увеличить тактовую частоту процессора. У длинного конвейера имеются и сильные, и слабые стороны. О сильной стороне уже сказано — это высокая тактовая частота процессора. Что же касается слабых сторон, то это, в первую очередь, длительные задержки при выполнении связанных операций, когда для выполнения одной операции требуется знание результата выполнения другой операции. Следовательно, до тех пор пока не будет завершена одна операция (20 тактов), не может начаться выполнение второй. Впрочем, следует отметить, что в буфере команд всегда найдется несколько инструкций, не зависящих друг от друга, которые можно выполнять параллельно. Учитывая, что в процессорах имеется несколько исполнительных модулей, умеющих работать параллельно, удается избежать простоя при выполнении команд и загрузить конвейер. Второй момент, связанный с длинным конвейером — это алгоритм предсказания переходов в выполнении программного кода. Чем длиннее конвейер, тем важнее становится предсказание того, выполнение какой инструкции понадобится. Ошибка в предсказании перехода процессору с длинным конвейером обходится дорого. Одна ошибка в выборе исполняемых операций в самом начале приводит к тому, что приходится отбрасывать результаты выполнения всех команд в конвейере и начинать все сначала. Для уменьшения вероятности возникновения ошибки в предсказаниях переходов в процессоре Intel Pentium 4 используются изолированные алгоритмы. Кроме того, повышение производительности достигается и за счет алгоритма Advanced Dynamic Execution, допускающего нарушение последовательности выполнения инструкций с целью более плотной загрузки исполнительных модулей. Таким образом, процессор Pentium 4 со своим гиперконвейером позволяет значительно повысить тактовую частоту, но, в то же время, ориентирован на выполнение потоковых операций, способных наиболее эффективно загрузить конвейер процессора.

Микроархитектура процессора Intel Itanium 2 существенно отличается от рассмотренной выше. Прежде всего, это связано с иным на-

значением процессора. Серверный процессор подобного уровня должен быть максимально оптимизирован под параллельное выполнение множества задач. Поэтому процессор Intel Itanium 2 построен на основе так называемой EPIC-архитектуры. EPIC — сокращенное обозначение технологии Explicitly Parallel Instruction Computing (явный параллелизм на уровне команд). Технология EPIC позволяет повысить эффективность параллельной обработки команд, обеспечивая, таким образом, существенное повышение производительности при работе с определенными типами приложений.

Однако сама идея параллелизма несовместима с длинным конвейером — ошибка в предсказании перехода слишком дорого обходится в плане последующего простоя. Отсюда и длина конвейера в процессоре Intel Itanium 2 всего в 8 стадий. Как следствие — невозможность достичь частот, свойственных современным процессорам для ПК. Разработанный на основе 64-разрядной архитектуры, процессор Intel Itanium 2 ориентирован на обработку крупных массивов данных и выполнение транзакций, требовательных к вычислительным ресурсам.

Коротко об архитектуре процессора. Он использует 400-мегагерцовую, 128-разрядную системную шину, пропускная способность которой равна 6,4 Гб/с. Конвейер обработки инструкций имеет длину 8 стадий и состоит из 11 каналов, что позволяет параллельно выполнять несколько задач или нитей одной задачи. Кроме того, в процессоре имеется несколько исполнительных блоков и 328 встроенных регистров (128 основных регистров, 128 регистров для операций с плавающей точкой, 64 предикативных регистра и 8 регистров ветвления). Все это позволяет процессору Intel Itanium 2 выполнять 6 команд за один системный такт. Вот она — основа производительности процессора: 8 команд за такт, 11 исполнительных каналов, 64-разрядная адресация памяти с возможностью адресации в 4 млрд. раз большего объема памяти, чем в процессорах на основе 32-разрядной архитектуры. Все это делает Intel Itanium 2 перспективным процессором на рынке high-end-серверов с 64-разрядной архитектурой. По прогнозам, серверы на базе процессора Intel Itanium 2 продемонстрируют 50-процентный прирост производительности при осуществлении транзакций по сравнению с серверами на базе процессора Sun UltraSparc III, причем стоимость последних окажется на 70% выше. В сравнении с предыдущей версией процессора Intel Itanium, новый процессор показывает 50...100% рост производительности на различных приложениях.

Симбиоз телефона и наладного компьютера предлагает ф. HandSpring. Своими впечатлениями об этом аппарате делится Г. Шенгард в статье "Сотовый компьютер HandSpring Treo 180" (ПЛ: компьютеры, №8/2002, С.32).

Сотовый телефон, который может запоминать до тысячи контактов, уже давно не редкость. Но, видимо, существуют особенно требовательные пользователи, которым мало телефонной базы своего микрорайона. Трубки не только обзаводятся дополнительными объемами памяти, но и постепен-

но осваивают умение проигрывать видеоролики и вовсе декодируют MP3-файлы. В свою очередь, карманные компьютеры тоже перенимают некоторые возможности у сотовых телефонов, правда, такие гибриды часто обладают массой недостатков.

Детище HandSpring, Treo 180, по крайней мере, с первого взгляда вызывает доверие. Небольшие недочеты, однако, быстро себя обнаружили. Удобный защитный флип (так автор называет откидную крышку) с большим окошком позволяет быстро просмотреть SMS-сообщение или определить, от кого по-

ступил звонок, однако сам флип открывается не до конца, всего на 140...150°, из-за чего его легко можно переломить в швыряне. Самое неприятное заключается в том, что внутри флипа находится динамик, совершенно необходимый для телефонного разговора.

Аппарат стал значительно тяжелее и толще классических моделей Palm. Создается впечатление, что дизайнеры и инженеры балластируют на границе понятия "карманный компьютер" — в нагрудный карман летней рубашки Treo 180 лучше все-таки не класть. Второе замечание относится к клавиатуре.



Несмотря на привычную раскладку QWERTY и удобное расположение, она может добавить несколько неприятных минут: клавиши слишком жесткие и чересчур близко расположены друг к другу, к тому же, дизайнеры пожертвовали десятью цифровыми клавишами, из-за чего для набора чисел приходится использовать функциональную кнопку.

Остальные достоинства и недостатки обязаны своим существованием базовым моделям Palm, на основе которого и было разработано это устройство. Например, черно-белый экран с разрешением 160x160 обладает достаточной для комфортной работы контрастностью, и даже при ярком солнечном освещении изображение довольно-таки контрастно.

Функции, относящиеся к телефонной части устройства, полностью перекрывают базовые возможности сотового телефона. Качество приема — не хуже, чем у классических моделей Ericsson, однако до уровня Nokia аппарат пока не дорос. Возможности телефонной книги, SMS и голосовой почты, благодаря клавиатуре, становятся поистине безграничными. Единственная несурзность при использовании Treo 180 в качестве те-

Известно, что производительность компьютера определяется не только тактовой частотой микропроцессора и его архитектурой, но и типом используемой памяти. Поэтому корпорация AMD собирается встроить в свои новые процессоры **блок управления памятью DDR SDRAM**. Об этом идет речь в заметке "Покой им только снится" (ПЛ: компьютеры, №8/2002, С.20).

Осенью AMD планирует выпуск нового поколения микропроцессоров Hammer, одной из особенностей архитектуры которых является встроенный в кристалл контроллер па-

Наконец-то ф. Intel решила вспомнить о скромных пользователях ПК, не готовых сразу выложить достаточно приличную сумму за Pentium 4. Появился его младший собрат — **Celeron**. Об этом сообщает А. Кузнецов в статье "Intel Pentium 4 Celeron — дождался, однеко..." (ПЛ: компьютеры, №8/2002, С.29).

Процессоры Celeron за несколько последних лет завоевали широкую популярность среди пользователей. Под этим названием скрывается целая линия процессоров корпорации Intel, пополнившаяся очередной моделью. Первые процессоры с названием Celeron появились в 1998 г. Компания Intel, уже начав производство Pentium II, решила скорректировать свою маркетинговую политику. Альтернативой этому достаточно дорогому процессору должен был стать его младший собрат, получивший имя Celeron. Отличаться он должен был, в первую очередь, меньшей ценой, что делало его доступным для широкого спектра покупателей. Другим аспектом стала меньшая производительность по сравнению с процессорами Pentium II той же тактовой частоты. Достигалось это следующим образом: аппаратный набор функций в Celeron оставался прежним, но размер кеша второго уровня был уменьшен в четыре раза — 128 Кб против 512. Частота системной шины для этого процессора предусматривалась равной всего лишь 66 МГц.

лефона заключалась в том, что аппарат уверенно принимал сигнал на расстоянии менее полуметра от системного блока компьютера без защитного кожуха, однако изредка "запинался" на улице, хотя возможных источников помех не наблюдалось.

Каких-то претензий к аппарату, как компьютеру, предъявить практически невозможно. "Наладоники" от HandSpring всегда отличались высоким качеством и надежностью. Но, несмотря на немалый объем оперативной памяти и полную поддержку устройством любого программного обеспечения для Palm, разработчики не наградили аппарат богатым набором программного обеспечения. Кроме браузера Blazer, дневника и не слишком-то полезных Expense и City-Time, можно отметить только SMS-, PhoneBook- и SIM-утилиты. Последние программы расширяют возможности телефона.

Остается отметить отличное качество синхронизационного программного обеспечения Palm Desktop и Hot-Sync Manager 3.1. Программы позволяют организовать некое подобие терминала Palm не экране настольного компьютера и практически полностью контролировать наладонник.

мяти DDR SDRAM. Такое решение позволяет несколько быстрее работать с оперативной памятью (правда, о том, какой именно прирост производительности дает встроенный в микропроцессор контроллер памяти, AMD не сообщает). Однако с другой стороны, при изменении используемого типа памяти придется менять архитектуру всего процессора (или, по крайней мере, некоторых его узлов). Ныне самой популярной является DDR, именно под нее и сделан Hammer, но уже в следующем году должно начаться массовое производство чипсетов под новый

Но очень скоро пользователи выяснили, что новые Celeron неплохо разгоняются и способны работать на шине 100 МГц. Их производительность была практически идентична демонстрируемой "полноценными" Pentium II. Происходило это за счет того, что кеш второго уровня у Celeron работал на одной частоте с ядром, против половинной у его старшего собрата. Успех оказался поразительным.

После того как компания Intel начала производить процессоры Pentium III Coppermine, появилась и очередная Celeron, выполненный на его ядре и имеющий уменьшенный наполовину кеш второго уровня. Модель была рассчитана на частоту системной шины 66 МГц, и только начиная с версии 800 МГц версии она стала поддерживать частоту 100 МГц.

Вслед за этой моделью появился процессор Pentium 4, на который Intel сделала основную ставку. И тут обнаружилась некоторая странность — новый продукт с боем начал завоевывать свое место под солнцем, а его упрощенного, более дешевого варианта все нет. Роль Celeron при Pentium 4 была возложена на процессор с ядром Tualatin (технология 0,13 мкм). Решив, по всей видимости, конъюнктурное и временное, т. е. данная модель была уже изначально обречена.

Возник очевидный диссонанс. Все это время Celeron выступал в качестве дешевой альтернативы процессору Pentium. При этом соблюдалось несколько обязательных условий:

Технически Treo 180 выполнен почти идеально, однако оправдать появление этого устройства не так-то легко. Как компьютер, Treo не оказался чем-то новым — пользователи все это уже видели. Зато аппарат стал заметно толще и тяжелее. В качестве сотового телефона Treo тоже не выдерживает конкуренции — даже наличие клавиатуры не дает преимуществ над классическими маленькими и изящными сотовыми трубками. И наконец, цена устройства пока слишком высока, чтобы относиться к Treo 180 как к повседневному рабочему инструменту.

Технические характеристики аппарата:

- процессор — Motorola Dragonball VX 33 МГц;
- память — 16 Мб;
- операционная система — Palm OS 3.5.2H4;
- экран — монохромный, 160x160 точек;
- телефонный стандарт: GSM 900/1800; GSM 900/1900;
- размеры — 110x69x18 мм;
- вес — 153 г;
- цена — 390 USD.

стандарт памяти DDR II. Что же произойдет тогда с Hammer? В AMD успокаивают, дескать, до 2004 г. DDR II популярности не наберет, а через полтора года сам Бог велел делать апгрейд новым микропроцессорам, встраивая в них поддержку DDR II.

Как бы то ни было, даже если с Hammer будет все в порядке, у пользователей проблем прибавится. Разница в архитектуре, по сравнению с Pentium, только возрастет, и однозначно ответить на вопрос, какой из процессоров лучше, пожалуй, уже не сможет никто.

у него было идентичное ядро, поддерживались одинаковые аппаратные функции, например, тот же набор потоковых расширений SSE. Для его работы подходил тот же процессорный разъем и те же наборы системной логики. В случае использования Celeron Tualatin в паре с Pentium 4 эти условия не выполнялись. Необходимы были какие-то шаги, исправляющие сложившуюся ситуацию. Только в этом случае можно было добиться увеличения объема продаж — ведь компания Intel является производителем не только процессоров, но еще и системной логики к ним. А пользователи — люди практичные, и предпочитают приобретать не туловищные решения, а платформы, которые в будущем все же можно будет подвергнуть модернизации с минимальными затратами. Если бы ситуация не изменилась, преимущество получили бы процессоры AMD, устанавливаемые в единый разъем Socket A. Можно было не сомневаться, что компания Intel сделает все возможное, чтобы сохранить свои лидирующие позиции.

Тек и произошло. На свет появился процессор Pentium 4 Celeron, выполненный на ядре Willamette (0,18-мкм технология). Здесь уместно напомнить, что на данный момент производятся два типа процессоров Pentium 4. Первый из них — на ядре Willamette, второй — на ядре Northwood (технология 0,13 мкм). Обе модели имеют одинаковую архитектуру, но отличаются еще и размером кеша второго



уровня. У Willamette он равен 256 Кб, а у Northwood — 512 Кб. Новый Pentium 4 Celeron традиционно имеет уменьшенный объем кеша второго уровня — 128 Кб. Другой его особенностью является меньшая или равная частота системной шины. У Pentium 4 Celeron она составляет 100 МГц. Неоспоримым достоинством нового процессора является поддержка набора SSE 2, который приобрел уже достаточно широкую поддержку

среди разработчиков ПО и де-факто уже стал стандартом. Pentium 4 Celeron предназначен для установки в Socket 478 и в те же материнские платы, что и его старший брат. Единственное, что может потребоваться, так это проверка поддержки BIOS. При необходимости его придется "перешить". Кстати, новый Celeron стоит примерно на треть дешевле настоящего Pentium 4 с аналогичной тактовой частотой.

Основные параметры Intel Pentium 4 Celeron:

- рабочие частоты — 1,7 и 1,8 МГц;
- технологический процесс — 0,18 мкм;
- корпус — FC-PGA2 (478 контактов);
- системная шина — 400 МГц;
- объем кеша первого уровня — 8 Кб данных и 12 000 микроопераций;
- объем кеша второго уровня — 128 Кб;
- микроархитектура — NetBurst.

Компьютер размером с пуговицу и используемый частично как пуговица — это, конечно, то, чего нам всем не хватает. Примерно о такой штучке рассказывает О. Татарников в статье "Wearable PC — компьютеры-коммуникаторы как элемент одежды" (КомпьютерПресс, №8/2002, С.45...46).

Все больше возможностей современных ПК становятся доступными на ходу. Прогресс делает свое дело: "большие" компьютеры вообще уходят в прошлое, ноутбуки становятся все меньше, тоньше и легче, а появившаяся недавно отрасль карманных компьютеров окрепла и быстро развивается. Компьютер больше не ассоциируется с двумя громоздкими белыми коробками на столе и под ним — он стремительно сокращается в размерах.

Большая заслуга в этом принадлежит компании Xubernaut Corp, выпускающей устройства, которые можно считать очередной важной вехой в развитии мобильных технологий. Это тоже компьютеры, но надеваются они непосредственно на тело: на голову — обруч с микромонитором для правого глаза, на пояс — маленькая коробочка, напичканная электроникой, а в кулаке зажимается манипулятор, выполняющий роль мыши. Теперь вы можете идти по улице и одновременно работать за компьютером. Разумеется, легко сказать: "идти и работать" — мало кто может одновременно и читать, и следить за дорогой, не натываясь на встречных прохожих, не врезаясь в автомобили и фоновые столбы. Тем не менее, устройство позволяет оперативно снимать электронную почту, слушать музыку в формате MP3 и постоянно быть в курсе всех мировых событий: узнавать цены на акции, результаты футбольных матчей и прогноз погоды, а в минуту отдыха — остановиться и поиграть в компьютерные игры, прогуляться по Интернету или уточнить маршрут передвижения, загрузив соответствующую карту. И все это можно делать, поглядывая одним глазом на окружающий мир, а другим — на экран компьютерного монитора.

По словам разработчиков, Poma (так называется новейшая версия "головного" компьютера, которая последовала за Mobile Assistant V) идеально подходит для тех, кто живет очень напряженной жизнью и буквально все делает на бегу. Poma заменит и мобильный телефон, и пейджер, и устройства для проигрывания музыкальных файлов, и многое другое.

Обруч с окуляром производства Microvision проецирует прямо на сетчатку глаза изображение, которое субъективно воспринимается как изображение на 15-дюймовом VGA-мониторе с расстояния 45 сантиметров (разрешение 640х480 и углом обзора по диагонали 30 градусов). Обруч весит всего 80 грамм, плотно прижимается к голове, и его можно

носить вместе с очками. Кроме того, экранчик не полностью загорает глаза и не мешает периферийному зрению.

Сам компьютер — это маленькая коробочка, содержащая 32-разрядный RISC-процессор Hitachi SH-4 с частотой 128 МГц, 32 Мб оперативной памяти и слот Compact Flash Type II, который позволяет подключать, например, накопители IBM Microdrive емкостью 1 Гб, беспроводной модем или сетевую карту. Все это управляется Windows CE и весит 280...310 грамм в зависимости от комплектации. Через USB-порт к устройству подсоединяется пульт управления с сенсорным оптическим указателем, который предназначен для простой навигации по экрану.

Хотя эта "игрушка" не дешевая — от 1500 до 2000 USD в зависимости от комплектации, но Xubernaut уверяет, что ее цена вскоре будет значительно снижена. Компания была основана в США в 1990 г., и занимается разработкой "носимых" ПК уже более десяти лет, а в 1995-м увидела свет первая серия носимых компьютеров Mobile Assistant. Ранее, на первых "надеваемых" компьютерах Wearable PC этой компании, использовались процессоры семейства Intel Pentium, которые работали со множеством стандартных операционных систем, таких как Windows 95/98/NT/2000, SCO UNIX, Turbo Linux и др.

Недавно компания Hitachi, известный поставщик процессоров для карманных компьютеров, лицензировала технологию компании Xubernaut, собираясь выпускать надеваемое Интернет-устройство — Wearable Internet Appliance, или WIA, на базе ОС Windows CE, и предложила собственные процессоры для потребительского рынка.

Компания Xubernaut прежде производила устройства в основном для различных корпоративных рынков, а теперь обратила внимание и на рынок карманных устройств для бытового применения. По условиям соглашения, Hitachi будет собирать устройства, а Xubernaut будет их продвигать, продавать и обеспечивать поддержку. Производством других надеваемых компьютеров Xubernaut в промышленных масштабах занимается и компания IBM, которая также является инвестором компании Xubernaut. Кроме того, Xubernaut заключила стратегические альянсы со многими компаниями, владеющими передовыми технологиями, в том числе с Shimadzu, SONY Digital Products, Siemens Nixdorf, Toshiba, SBS, Omron, Ricoh, En Pointe, Nissso Iwai и различными филиалами концерна Mitsubishi.

Подобные устройства пользуются большой популярностью не только у так называемых geeks — людей с гипертрофированной страстью к технике, но и во вполне серьезных компаниях, имеющих большие подразделе-

ния, сотрудники которых работают в полевых условиях, когда требуется свобода рук. Устройства незаменимы, например, для геологов, ремонтников или монтажников сложной аппаратуры, кабельных сетей и коммуникаций (они позволяют пользоваться хранимой информацией, вводить и обрабатывать новые данные непосредственно на месте, то есть там, где обычный ПК, либо даже ноутбук недоступны или неудобны в использовании). Конечно же, разработкой интересуются и военные ведомства. В качестве дополнительных возможностей предлагаются голосовой ввод и управление от команд, подаваемых голосом (распознавание речи обеспечивает программное обеспечение IBM ViaVoice).

Надеваемые компьютеры имеют четыре главных достоинства. Во-первых, это легкие системы, которые вы можете взять с собой, куда бы вы ни направлялись. Это ваш личный мобильный офис, который позволяет вам всегда иметь доступ к любой информации. С этой системой вам не нужны ни справочники, ни записные книжки — все данные всегда при вас. Во-вторых, они имеют простые функции визуализации и управления (в том числе и голосового). Это избавляет вас от необходимости прибегать к помощи манипуляторов и клавиатуры и позволяет параллельно совершать другие действия. В-третьих, это устройство является превосходным коммуникатором, позволяет выходить в Интернет (причем на полноценные сайты), отправлять и получать электронную почту, факсы, то есть вы всегда находитесь на связи и быстро ориентируетесь в любых условиях. В-четвертых, подключение PocketGPS позволяет свободно ориентироваться на незнакомой местности. По сведениям агентства BBC, надеваемые компьютерные системы уже используются армией США для обеспечения связи, навигации и разведки. Специально для этого разработан новый ультрапрочный аппарат: система была сконструирована при участии американской фирмы Second Chance, занимающейся пуленепробиваемыми материалами.

По информации представителей Second Chance, носимый на одежде тактический компьютер предоставляет полицейскому или солдату весьма важные преимущества, и, несомненно, будет находить все более активное применение. Британские исследователи тоже работают над собственными вариантами подобных военных систем.

А для широкого круга пользователей компания Xubernaut собирается предложить такие услуги как дистанционное обучение, музыка, видео, игры, спутниковая навигация, сотовая связь, пейджер, интерактивный электронный банк, мобильная коммерция, биржевая торговля акциями и пр.



Использование динамической памяти в прикладных программах

(Окончание. Начало в №12/2002)

2.2. Использование нескольких динамических массивов вместо одного статического

В турбо-Паскале **невозможно** объявить массив, который требует для своего хранения более 64 Кб оперативной памяти. Это касается объявлений как в разделе VAR, так и в разделе TYPE. Например, если нам потребуется решить задачу о кусках при больших ограничениях на N и M ($1 \leq N, M \leq 200$), то нельзя написать так:

```
type
  TQue = array [1..40000,1..2] of byte;    {Очередь ходов}
  TPQue = ^TQue;
```

```
TMarked = array [1..200,1..200] of boolean; {Пометки на доске}
TPMarked = ^TMarked;
```

так как для хранения массива типа TQue потребовалось бы $40000 \times 2 = 80000$ байтов, а эта величина превышает 64 Кб (напомним, 1 Кб = 1024 байта).

Однако, мы можем хранить в динамической памяти несколько массивов, размер каждого из которых менее 64 Кб, но которые вместе хранят необходимый объем информации. В этом случае несколько усложняется доступ к нужной нам информации. Например:

```
type
  TQue = array [1..20000,1..2] of byte;    {Очередь ходов}
  TPQue = ^TQue;
```

```
TMarked = array [1..200,1..200] of boolean; {Пометки на доске}
TPMarked = ^TMarked;
```

```
var
  PQue1, PQue2 : TPQue;
  PMarked      : TPMarked;
  ...
```

```
begin
  New(PQue1); New(PQue2);
  New(PMarked);
  ...
```

Тогда для обращения к нашей очереди, размещенной в двух динамических массивах PQue1 и PQue2, необходимо модифицировать процедуры Get и Put следующим образом:

```
procedure Put(x,y:integer);    {Занести в очередь}
begin
  Inc(QueEnd);                 {Увеличить количество}
  If QueEnd <= 20000
  then begin
    PQue1^[QueEnd,1] := x;     {Координата по x}
    PQue1^[QueEnd,2] := y;     {Координата по y}
  end;
  else begin
    PQue2^[QueEnd-20000,1] := x; {Координата по x}
    PQue2^[QueEnd-20000,2] := y; {Координата по y}
  end;
  PMarked^[x,y] := true;      {Помечаем использованную}
end;
```

```
procedure Get(var x,y:integer); {Взять из очереди}
begin
  if QueBegin <= 20000
  then begin
    x := PQue1^[QueBegin,1];   {Координата по x}
    y := PQue1^[QueBegin,2];   {Координата по y}
  end
  else begin
    x := PQue2^[QueBegin-20000,1]; {Координата по x}
    y := PQue2^[QueBegin-20000,2]; {Координата по y}
  end;
  Inc(QueBegin);               {Изменить начало}
end;
```

2.3. Пример использования динамической памяти для хранения записей о точке

Объявляем новый тип — запись, хранящую информацию о координатах точки, и указатель на такую запись такого типа:

```
type TPoint = record
  x,y : extended; {Тип Точка — координаты X и Y}
end;
PPoint = ^TPoint; {Тип Указатель на точку}
```

Объявляем переменную типа "Указатель на точку":

```
var
  PP : PPoint; {Переменная типа Указатель на точку}
```

Захватываем динамическую память под точку:

```
New(PP); {Захватить под точку в ОП память}
          {размером 2 поля типа extended,}
          {2*10=20 байтов, и адрес этой}
          {памяти занести в переменную PP}
```

При обращении к полям записи работаем следующим образом:

```
PP^.x := 1.4; {Сформировать координаты точки}
PP^.y := PP^.x + 0.5;
```

По завершении обработки освобождаем захваченную динамическую память:

```
Dispose(PP); {Освободить память, на которую}
              {указывает PP}
```

2.4. Использование ASCIIZ-строк

Как известно, строки, объявляемые в Паскале с помощью типа string, имеют сильно ограниченную длину (не более 255 символов). Развитием строкового типа, который имеет максимальную длину до 65535 символов, является тип pchar. Такая строка не содержит собственной длины в начальном байте, признаком окончания строки является символ с кодом #0.

В Турбо-Паскале имеется модуль STRING, в котором содержится набор разнообразных функций для работы с такими строками. Например, динамическое создание строк — `S := StrNew("abcdef");` и динамическое уничтожение строк — `StrDispose(S);`.



Имеются также функции: "Копирование строк", "Конкатенация строк", "Сравнение строк", "Поиск по строке" и другие. Для получения полной информации о функциях модуля STRING, в среде Турбо-Паскаля наберите `rschar` и нажмите `Ctrl-F1` для получения контекстно-чувствительной помощи.

3. Некоторые сведения о функционировании ЭВМ

Оперативная память компьютера, с точки зрения программиста, может представляться как совокупность смежных байтов, которые последовательно нумеруются от 0 до максимально возможного значения. Один байт может хранить число от 0 до 255 (или от -128 до +127) или символ (русская или латинская буква, цифра, знак операций и т.д.). Последовательности смежных байтов могут использоваться для хранения целых и вещественных чисел, либо символьных строк.

Компилятор — это программа, которая обрабатывает исходный текст вашей программы и заносит результат обработки в оперативную память в виде двух главных компонентов: статической памяти данных и исполняемой части программы. Под статической памятью здесь понимается память, которая выделяется под данные, объявленные в вашей программе в разделе `VAR`. А под исполняемой частью программы понимаются операторы, которые записаны в главной программе между `begin` и `end.`, а также между `begin` и `end`; в процедурах и функциях.

Часть оперативной памяти компьютера может быть занята также и чужими программами. А вот та часть памяти, которая остается свободной после загрузки в память вашей программы, и называется динамической памятью. Именно эта память и захватывается вами с помощью процедур `New`. После вызова в вашей программе процедур `Dispose`, освобождаемая память дополняет список свободных областей динамической памяти.

Единственным средством доступа к динамическим переменным является указатель на место их текущего расположения в ОП — фактически это номер байта оперативной памяти, с которого начинается выделенная область оперативной памяти.

Исторически сложилось, что в персональных компьютерах типа IBM PC указатели областей оперативной памяти представляются в виде `<сегмент>: <смещение>`. В зависимости от типа процессора и режима функционирования операционной системы, абсолютный адрес указателя (реальный соответствующий номер байта оперативной памяти) может вычисляться по различным алгоритмам. Например, для реального режима в DOS это делается по формуле

$$\text{<сегмент>} * 16 + \text{<смещение>}$$

Заключение

Данная статья продолжает серию публикаций [1...11], ориентированных на самообучение программированию и подготовку к областным и республиканским олимпиадам по информатике. Очевидно, что для закрепления данного материала необходимо проверить полученные знания на компьютере как самосто-

ятельным решением поставленных в данном материале задач, так и применением полученных знаний к другим задачам. Полезно также ознакомиться с соответствующими материалами учебных пособий, например, [12...15].

Для проверки полученных знаний автор рекомендует использовать сайт Гомельского государственного университета им. Ф.Скорины <http://dl.gsu.unibel.by> [16].

Литература

1. М.Долинский. Памятка участнику олимпиады по информатике среди школьников. — Радиолучитель. Ваш компьютер, 2000, №1, С.20.
2. М.Долинский. Начинаем программировать самостоятельно. — Радиолучитель. Ваш компьютер, 2000, №№1...6.
3. М.Долинский. Решение задач на тему "Геометрия на плоскости" — Радиолучитель. Ваш компьютер, 2000, №№8, 9.
4. М.Долинский. Разработка программ с использованием определяемых программистом типов данных. — Радиолучитель. Ваш компьютер, 2001, №№1, 3.
5. М.Долинский. Решение задач с помощью очереди и стека. — Радиолучитель. Ваш компьютер, 2001, №№4...6.
6. М.Долинский. Обзор возможностей языка программирования Паскаль. — Радиомир. Ваш компьютер, 2001, №7, С.23.
7. М.Долинский. Алгоритмы генерации комбинаторных объектов. — Радиомир. Ваш компьютер, 2001, №№10, 11.
8. М.Долинский. Некоторые факты из теории чисел. — Радиомир. Ваш компьютер, 2001, №9, С.20.
9. М.Долинский. Использование рекурсивных функций и процедур. — Радиомир. Ваш компьютер, 2001, №№1, 2.
10. М.Долинский. Решение задач с помощью рекуррентных соотношений. — Радиомир. Ваш компьютер, 2002, №№3...6.
11. М.Долинский. Решение задач на графах. — Радиолучитель. Ваш компьютер, 2002, №№7-11.
12. Котов В.М., Волков И.А., Харитонович А.И. Методы алгоритмизации. Учебное пособие для 8-го класса общеобразовательной школы с углубленным изучением информатики. — Мн.: Народная асвета, 1996.
13. Котов В.М., Волков И.А., Лапо А.И. Методы алгоритмизации. Учебное пособие для 9-го класса общеобразовательной школы с углубленным изучением информатики. — Мн.: Народная асвета, 1997.
14. Котов В.М., Мельников О.И. Методы алгоритмизации. Учебное пособие для 10-11-х классов общеобразовательной школы с углубленным изучением информатики. — Мн.: Народная асвета, 2000.
15. Кормен Т., Лейзерсон Т., Ривест Р. Алгоритмы: построение и анализ. — МЦНМО, Москва, 1999, 960 с.
16. А Кан. DL vs. DL: две различные концепции удаленного обучения. — Радиомир. Ваш компьютер, 2002, №11, С.19.



Serrgio /HI-TECH,
E-mail:serrgio@gorki.unibel.by
http://wasm.ru/

НИЗКОУРОВНЕВОЕ ПРОГРАММИРОВАНИЕ

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

Программирование под WIN32

9. Генератор паролей

В тот год, когда Ю'дзын покинул горы,
умел смеяться он и видел свет [он],
хотел больших серьезных перемен [он].
Теперь не хочет, не умеет и не видит...

Скажи, учитель, жив Ю'дзын ли?

Учитель сказал: "ЧТОБ ЛОТОС ЦВЕЛ, ЕМУ НУЖНА ВОДААААА!!"

И ударил его палкой. По голове.

(С) НОМ, portion by Конфуций

#1. В этой главе мы с вами напишем продвинутую программу для генерации паролей. В отличие от тех программ, что мы писали ранее, эта более наворочена и длиннее (а вы чего хотели?), так что будьте бдительны ;).

Не могу не отметить, что DOS'овский прототип программы был взят из книги "Техника и философия хакерских атак" Криса Касперски, а процесс медитирования над win32-версией сопровождался постоянным постукиванием бамбуковой палкой в железных руках Аркадия Белоусова, за что ему большой дзенский сенькс.

Под генерацией паролей подразумевается печать в консольном режиме всех возможных комбинаций символов произвольного "алфавита".

Например, для алфавита, состоящего из трех символов "abc", программа сгенерирует и напечатает следующие пароли: a, b, c, aa, ba, ca, ab, bb, cb, ac, bc, cc, aaa, baa, caa ... cccbcbbaba, aaaccbbaba, baaccbbaba, caaccbbaba... И будет печатать их, пока не наступит одно из нижеследующих событий:

- вам надоела эта безобразия, и вы жмете на "крестик" закрытия окна;

Тут, однако, встает первый, философский вопрос — не является ли закрытие окна консольной программы столь варварским способом нарушением священных принципов Правильного Программирования? Со всей ответственностью за возможные последствия отвечаю: нет, не является! И тут же, на всякий случай, даю себе пути к отступлению: если ВСЕ-ТАКИ является, то хорошо известная каждому сетеву программе ping — очередное доказательство того, что программисты Microsoft этим правилам не следуют, в чем легко убедиться (как, впрочем, и опровергнуть), выполнив команду ping 127.0.0.1 -t.

- число символов пароля превосходит размер буфера;

А вот это уже вопрос математический. Зная число символов в "алфавите", скорость генерации их сочетаний (достав с полки толстый талмуд по комбинаторике), нетрудно подсчитать длину пароля, при превышении которой дальнейшую работу программы стоит признать нецелесообразной ;). Без доказательства приведем такую формулу — при длине буфера N и количестве знаков алфавита B количество разных паролей будет $B \cdot (B^N - 1) / (B - 1)$. Так, при двух знаках (B=2) и длине буфера под 8 знаков (N=8) можно получить 510 паролей. Если

же знаков будет 64, то при длине буфера под 4 знака можно получить более 17 миллионов паролей.

- тепловая смерть Вселенной.

Абстрагируясь от многих вещей, астрофизическую часть этого вопроса можно сформулировать следующим образом — какой размер буфера для двухсимвольного алфавита "01" будет достаточным для того, чтобы программа прекратила работу лишь в момент наступления данного события? Часть квантово-физическая — не приближаем ли мы данное событие, проводя дефрагментацию жесткого диска и тем самым уменьшая коэффициент энтропийности нашей Вселенной? И, наконец, часть экологическая — где взять источник энергии для экспериментальной проверки части астрофизической?

Этот алгоритм можно запросто использовать, например, для реверсирования односторонних функций (что, кстати говоря — альфа и омега криптографии) методом брутфорса (т.е. методом грубой силы). Впоследствии мы еще вернемся к этому вопросу.

#2. Секция данных нашей программы выглядит следующим образом:

```
S_consoleTitle db 'DicMaker+',0
S_alphabet db 'abcde'
szAlphabet = $-S_alphabet
szBuffer = 20
S_password db szBuffer dup(?)
S_key db szBuffer dup(?)
szKey = $-S_key
```

S_alphabet — "алфавит", то есть набор символов, которые будут "участвовать" в генерации пароля. SzAlphabet — его размер (говоря другими словами, число символов). Далее у нас следуют два буфера — S_password для хранения пароля и S_key для хранения "ключа". Учитывая, что ключ и пароль — близнецы-братья, размер буфера под них должен быть одинаковым.

"Что такое пароль, мы знаем, но что подразумевается под термином ключ?" — спросит читатель. И совершенно правильно сделает, что спросит.

Объясняю. Например, для шестисимвольного алфавита "abcdef" мы можем нарисовать следующую таблицу:

Начертание символа	a	b	c	d	e	f
Индекс символа	0	1	2	3	4	5

При генерации паролей в буферах S_password и S_key будут попадать соответственно:

Ключ (байты)	Пароль
00	'a'
01	'b'
02	'c'
03	'd'
04	'e'
05	'f'
00	'aa'
01 00	'ba'
02 00	'ca'
03 00	'da'
04 00	'ea'
05 00	'fa'
00 01	'ab'
...	...
00 00 00	'aaa'
01 00 00	'baa'
02 00 00	'caa'
03 00 00	'daa'
...	...

Как видно из вышеуказанной таблицы, ключ — это своего рода "недоделанный пароль", состоящий не из символов, а



из их "номеров в алфавите". Сначала генерируется именно "полуфабрикат", и только потом он "отображается" на множество "алфавит", и получается "конечный продукт" — пароль.

Однако, даже среди грубиянов (в контексте brute force) обязательно найдется особо грубый ;). Специально для таких мужественных, бескомпромиссных, и, что самое главное, собирающихся экспериментально проверить некоторые теории, изложенные параграфом выше, привожу классический алгоритм генерации паролей:

```
lea esi, S_password
@@nextpassword:
xor ebx, ebx
@@checkover:
inc byte ptr [esi+ebx]
cmp byte ptr [esi+ebx], 'z'
jb @@writepassword
mov byte ptr [esi+ebx], ' '
inc ebx
jmp @@checkover
@@writepassword:
; очередной пароль готов к употреблению
jmp @@nextpassword
```

Медитируем!

#3. А теперь полюбуемся на код программы и попробуем разобраться, как он работает.

Сначала, как и всегда — локальные переменные под хэндл и под "напечатано символов".

```
local houtput :dword
local nwriten :dword
```

Хорошо знакомые по предыдущим программам "подготовительные операции":

```
invoke SetConsoleTitle, addr S_consoleTitle
invoke GetStdHandle, STD_OUTPUT_HANDLE
mov [houtput], eax
```

Теперь, собственно, рабочая часть программы (замрите в ужасе!):

```
mov esi, offset S_key ; (1)
mov edi, offset S_password
mov ecx, esi
```

```
@@keylenloop: ; (2)
cmp ecx, offset S_key+szKey
jae @@break
mov byte ptr [esi], 0
```

```
@@passwordloop: ; (3)
movzx ebx, byte ptr [esi]
mov al, S_alphabet[ebx]
mov [edi], al
```

```
; -|А|----- ; (4)
```

```
push ecx
mov edi, offset S_password
sub ecx, offset S_key-3
mov word ptr [edi+ecx-2], 0A0Dh
invoke WriteConsole, houtput, edi, ecx,
; addr nwriten, NULL
pop ecx
```

```
; -|Б|-----
```

```
mov esi, offset S_key ; (5)
; mov edi, offset S_password
```

```
@@carryloop: ; (6)
```

```
inc byte ptr [esi]
cmp byte ptr [esi], szAlphabet
jb @@passwordloop

mov byte ptr [esi], 0
mov al, S_alphabet[0]
mov [edi], al
inc esi
inc edi
cmp esi, ecx
```

```
jbe @@carryloop

inc ecx
jmp @@keylenloop
@@break:
```

```
invoke ExitProcess, 0 ; ВЫХОД ;)
```

Уф... Что ж, давайте анализировать, и да пребудет с вами долготерпение.

Бряк 1 — загружаем в регистры *esi* и *edi* указатели на буферы ключей и паролей соответственно. Регистр *ecx* используется для подсчета числа разрядов в ключе (оно же — число символов в пароле), с той лишь разницей, что для удобства там лежит смещение от начала буфера ключа. Изначально он инициализируется указателем на первый разряд ключа, что вполне естественно, так как еще не произошло ни одной итерации цикла, и первый разряд является одновременно и старшим, и младшим (на выбор), т. е. единственным ;).

Бряк 2 — сюда переходит управление в том случае, если предполагается увеличение разрядности ключа (она же — длина пароля). Здесь сравнивается указатель на старший разряд ключа с адресом после буфера под ключ. И если "потребности" программы больше, чем "возможности" буфера, перепрыгиваем на *invoke ExitProcess* и благополучно завершаем ее выполнение. А если же подобное безобразие еще не случилось, то инициализируем старший разряд ключа нулем. Собственно, алгоритм начинается отсюда, с "увеличения" ключа до единичной длины.

Бряк 3 — конвертируется старший разряд ключа в соответствующий символ пароля. То есть берется значение разряда ключа, достается соответствующий символ алфавита и записывается в соответствующий разряд пароля. Обратите внимание — меняется только старший разряд пароля, поскольку остальные разряды пароля (которых в начале алгоритма еще нет) к этому моменту должны быть уже инициализированы правильным значением.

Бряк 4 — между А и Б можно вставить любое действие, которое должно произойти со сгенерированным паролем. В данном примере пароль печатается, но ничто не мешает вставить сюда, например, код вычисления хеш-функции пароля и сравнения его с реверсируемым ;). Также обратите внимания, каким "хитрым" способом проводится добавление CRLF в буфер пароля для вывода на печать (с вычислением длины пароля вместе с CRLF) ;).

Бряк 5 — снова берутся указатели. Вследствие особенностей кода между А и Б нам нет необходимости повторно брать указатель на буфер пароля, поэтому эта строка закомментирована. Но, заметьте, не убрана — на тот случай если мы захотим написать между А и Б нечто иное ;). Это разумный компромисс между оптимальностью (когда нет лишних инструкций) и читабельностью/модифицируемостью.

Бряк 6 — увеличивается разряд (разряд, не разрядность!) ключа на единицу. Если не превышено максимальное значение разряда (соответствующее последнему символу "алфавита"), то ключ считается готовым, и значение его разряда конвертируется в символ пароля, то есть идет прыжок на **бряк 3**. Если же максимальное значение разряда превышено, то разряд обнуляется, также "обнуляется" соответствующий разряд пароля и, если не было выхода за пределы позиции старшего разряда (указатель в *ecx*), увеличивается следующий разряд (вам это еще не напомнило длинную многоразрядную арифметику? :). А если выход был — добавляется очередной разряд к ключу (увеличивается его разрядность), то есть идет прыжок на **бряк 1**.

Вот, в общем-то, и вся программа ;).

(Продолжение следует)



Перехват вызовов API в системах семейства Windows

А.ТЕРЁШКИН /NI-TECH,

г. Майкоп,

WWW: <http://wasm.ru/>

1. Введение

Во времена MS DOS ни одна серьезная программа не обходилась без перехватов прерываний — сервисов системы для установки на них своих процедур-обработчиков. Это было совершенно необходимо, например, для обеспечения "псевдо-многозадачности" (por up), реакции на таймер в режиме реального времени, получения расширенной информации об одновременно нажатых пользователем клавишах и т.п. Установка своих обработчиков могла осуществляться даже без ведома системы. Правда, тогда была вероятность того, что DOS выделит память с находящимся в ней обработчиком для какой-либо программы, что чревато крахом системы. Для мелких резидентов это было не страшно, а большие "защищались" несколькими способами — например, маркированием области как используемой DOS в системных целях (установка маркера 80h). Резиденты могли привести систему в нестабильное состояние. Чтобы этого не случилось, разработчики должны были учитывать очень большое число тонкостей и "подводных камней" ОС и принимать соответствующие меры. К тому же, обработчики прерываний писались чаще всего на ЯА (языке ассемблера) в целях экономии памяти и увеличения скорости работы. Отсюда следует, что качественные резиденты — столь необходимые в программировании под DOS механизмы — писали только программисты довольно высокой квалификации.

Системы Win32 претендуют на то, что ни один пользовательский процесс не может нарушить либо повлиять на работу другого или вызвать крах системы. Само понятие "резидент" в Win32 теряет свой смысл, так как каждый процесс работает в своем контексте памяти. Поэтому разработчики этих ОС отказались от подобного механизма, оставив только возможность обработки оконных сообщений (возможно, даже чужого процесса). Обработка оконных сообщений — это единственный метод получения информации о действиях пользователя для GUI-приложений.

Разработчики Win32, конечно, предоставили сервисы ОС для осуществления того, что в DOS могло быть сделано только путем установки резидента. Часть из них реализована через систему оконных сообщений, часть — через сервисы 3-го кольца (user mode), а остальные — в виде низкоуровневых сервисов, которые могут использоваться только драйверами (kernel mode). Т.е. единый в прошлом механизм разделился на совершенно разные по сути подсистемы ОС. Это накладывает большие ограничения на его использование.

К примеру, пользователь NT (Win2k) с привилегиями пользователя или гостя хочет защититься от атак из сети на отказ, нежелательных соединений, либо проверить, не передается ли от него какая-либо информация без его ведома. Это может быть решено путем установки драйвера — sniffера (sniffer) его сетевой карты, т.е. персонального файрволла (firewall). Проблема состоит в том, что он не может установить свой драйвер в систему, так как он не имеет администраторских прав. Это ограничение действительно необходимо — ведь в противном случае даже гость может сделать с системой все что угодно, так как код драйвера исполняется в нулевом кольце. Он, к примеру, может получить администраторские права, поймавшись в коде процесса Winlogon.exe.

Однако эта же задача может быть решена без использования драйверов практически с той же эффективностью.

2. Перехват вызовов API в системах Windows NT и Windows 9X

2.1. Теория

Нельзя утверждать, что адреса функций даже в системных библиотеках (например, Kernel32.dll) не изменяются в зависимости от версии ОС, ее сборки или даже конкретной ситуации. Это происходит из-за того, что предпочитаемая база образа библиотеки (dll preferred imagebase) является константой, которую можно изменять при компиляции. Более того, совсем не обязательно, что dll будет загружена именно по предпочитаемому адресу — этого может не произойти в результате коллизии с другими модулями, динамически выделенной памятью и т.п. Поэтому статический импорт функций происходит по имени модуля и имени функции (либо ее номера — ординала), предоставляемой этим модулем. Загрузчик PE-файла анализирует его таблицу импорта и определяет адреса функций, им импортируемых. В случае, если в таблице импорта указана библиотека, не присутствующая в контексте, происходит ее отображение в требуемый контекст, настройка ее образа, и ситуация рекурсивно повторяется. В результате в требуемом месте определенной секции PE-файла (имеющей атрибут "readable") заполняется массив адресов импортируемых функций. В процессе работы каждый модуль обращается к своему массиву для определения точки входа в какую-либо функцию. Отсюда следует, что для перехвата функций в рамках одного контекста с уже инициализированными модулями, работающими потоками, можно действовать двумя способами.

Первый состоит в следующем. Определяется адрес нашего обработчика перехватываемой функции, расположенного, к примеру, в загруженной в контекст данного процесса библиотеке. Определяется настоящий адрес функции, и производится замещение первых 5 байтов ее кода на "длинный jmp" к нашему обработчику. Для вызова исходного кода функции обработчик должен восстановить прежние байты кода, сделать вызов функции, и после возвращения управления восстановить опкод JMP в начале функции, иначе наш обработчик никогда не получит управление вновь. Этот метод называется "сплайсингом". Рассмотрим его сильные и слабые стороны.

"Плюсы":

- т.к. исправляется только память модуля, в котором производится перехват, наш код будет вызываться в результате вызовов функции как старыми модулями, так и динамически подгружаемыми, т.к. адрес перехватываемой функции не изменился — произведена лишь "врезка" нескольких байтов. При вызове перехватываемой функции наш код обязательно будет вызван;

- будут перехватываться вызовы даже внутри обработанного модуля, т.к. неважно, каким образом точка входа перехватываемой функции получила управление;

- становится элементарной деинсталляция нашего обработчика — достаточно восстановить исходные байты.

"Минусы":

- в системах Win95/98 некоторые системные библиотеки (kernel32, user32 и др.) загружаются в адресное пространство >2 Гб, которое проецируется на все контексты, присут-



ствующие в системе. Это усложняет процесс перехвата их функций, так как содержимое памяти свыше 2 Гб не может быть изменено стандартными документированными API-функциями. Установить разрешение на запись в эту область памяти может функция `_PageModifyPermissions`, вызываемая с помощью `kernel32!VxDCall0` и имеющая номер 1000dh. После установки атрибута "writable" на необходимый регион памяти и осуществления записи в него, изменения произойдут во всех присутствующих контекстах одновременно. Значит, код нашего обработчика должен располагаться по одинаковым адресам во всех контекстах. Это возможно, только если код будет находиться выше 2 Гб — этого можно добиться несколькими способами. Вот самые распространенные из них: непосредственное указание требуемого `imagebase` линкеру динамически загружаемой библиотеки при компиляции (>2 Гб); расположение кода в межсекционном пространстве какого-либо модуля, загруженного выше 2 Гб; выделение памяти, проецируемой на все контексты (например, из файла подкачки) и копирование туда своего кода. Таким образом, наш обработчик будет получать управление в результате вызова перехватываемой функции в любом контексте. Если же необходимо осуществить перехват функции только в определенном контексте, обработчик должен вызывать `GetCurrentProcessId()` для получения сведений о вызывателе. Если же перехват сужается до определенного модуля, то, кроме идентификации текущего процесса, необходим анализ стека для определения вызывающего модуля;

- если наш обработчик получил управление, восстановил исходные байты и занялся чем-то иным, то другой поток в этот момент может вызвать настоящую функцию, т.е. вызов перехвачен не будет — реентерабельность обработчиков не гарантируется даже при правильной организации кода. Эта проблема в общем случае не имеет решения.

Другой метод выглядит так. Определяется точка входа перехватываемой функции. Составляется список модулей, в настоящий момент загруженных в контекст требуемого процесса. Затем перебираются дескрипторы импорта этих модулей в поиске адресов перехватываемой функции. В случае совпадения, этот адрес изменяется на адрес нашего обработчика.

"Плюсы":

- осуществляется реентерабельность нашего обработчика, т.к. код перехватываемой функции не изменяется вообще;
- становится более гибким выбор модуля/процесса, в котором необходимо осуществить перехват. Тело нашего обработчика может находиться по разным адресам в разных контекстах.

"Минусы":

- уже инициализированные модули могут сохранить настоящий адрес функции и впоследствии вызывать именно его, минуя наш обработчик;
- необходимо каким-то образом распространять наши обработчики на динамически загружаемые модули. Это, в свою очередь, можно осуществить несколькими способами:

1. В модуле, содержащем код перехватываемой функции, изменяется таблица экспорта, точнее, относительный виртуальный адрес (RVA) перехватываемой экспортируемой функции. Загрузчик PE будет указывать эти значения (`+imagebase`) в таблицах импорта новых модулей. Также функция `GetProcAddress` будет возвращать адреса наших обработчиков. Этот способ имеет существенный недостаток.

Рассмотрим пример. Пусть мы перехватываем некото-

рую функцию из `kernel32`. Процессом был вызван `GetProcAddress(kernel32_base, &kernel_function_name)` для получения ее адреса. Т.к. `kernel32` загружен во всех контекстах по одному и тому же адресу, то адрес, возвращенный GPA, можно использовать для вызова функции удаленным кодом. Далее процесс выделяет память, к примеру, в контексте своего дочернего процесса, и после этого копирует удаленный код в эту область памяти. После этого он вызывает `CreateRemoteThread` для создания удаленного потока. Но так как мы перехватили одну из вызываемых этим потоком функций, то ее адрес больше не принадлежит региону `kernel32`. Даже если эта функция перехвачена и в дочернем процессе, то нельзя гарантировать однозначность адреса нашего обработчика в обоих контекстах. Т.е. очень вероятно, что при исполнении удаленного потока возникнет ситуация передачи управления не на наш обработчик — например, на неинициализированную страницу, т.е. страницу, не имеющую атрибут "executable", или вообще в чужой код. Любой из этих случаев приведет к GPF, и процесс-мишень будет аварийно завершен.

2. Во избежание ситуации, описанной выше, нельзя перехватывать функции, часто используемые как начальная точка удаленного потока — `LoadLibraryA`, `LoadLibraryW`. Они являются всего лишь переходниками к более мощным функциям — `LoadLibraryExA`, `LoadLibraryExW`. Но их перехват не решит проблему, так как переход к ним в `kernel32` делается коротким — по относительному смещению, не используя таблицу импорта. Но при более детальном изучении этих функций оказывается, что `LoadLibraryExA` сводится к `LoadLibraryExW`, а та, в свою очередь, к недокументированной функции `ntdll!LdrLoadDll`. Ее перехват необходим для решения двух задач — инсталляции перехвата функций на динамически загружаемые библиотеки и возможности установки обработчиков в цепочку.

2.2. Создание и исполнение удаленного кода

Удаленный код — код, исполняемый вне контекста, изначально его содержащего. Чтение-запись процессом памяти другого процесса осуществляется функциями `kernel32!ReadProcessMemory` и, соответственно, `kernel32!WriteProcessMemory`. Синтаксис вызова этих функций идентичен — одним из параметров является хэнгл процесса, над которым производится операция, т.е. процесс "открывается" для каких-либо (определенных) действий. Производится это функцией `kernel32!OpenProcess`.

В операционных системах, поддерживающих систему привилегий (NT/Win2k), возможность успешного открытия процесса зависит от текущего уровня привилегий процесса-исполнителя. В частности, функция `OpenProcess` по отношению к процессу `winlogon.exe` выполнится только при включенной привилегии `SE_DEBUG_PRIVILEGE`. Ею в большинстве случаев обладают только администраторы, так как она позволяет манипулировать всеми без исключения процессами системы, что чревато крахом самой ОС либо ее системы безопасности. Таким образом, возможность чтения-записи памяти чужого процесса ограничена привилегиями пользователя. Так как чтение-запись памяти — краеугольный камень всей идеологии перехватов API, то очевидно, что существует класс задач, невыполнимых пользователем, не имеющим, к примеру, администраторских прав. Однако такой пользователь может применить перехват функций к любому процессу, запущенному (прямо либо косвенно) им самим (`CreateProcess` и аналоги), что в большинстве случаев и требуется.



Рассмотрим варианты внедрения динамически загружаемой библиотеки в чужой контекст. Как известно, после инициализации dll происходит исполнение ее точки входа (как PE-файла) с тремя параметрами. Именно в этой процедуре будет находиться код, осуществляющий перехват функций.

1. Использование стандартной функции SetWindowsHookEx. Эта функция устанавливает хук на оконные сообщения. Она предназначена для отслеживания сообщений окон как своего процесса, так и чужого. В этом случае код обработчика должен находиться в dll. Реализация функции в общих чертах такова:

- она проецирует библиотеку на все контексты, которые удовлетворяют следующим условиям — их потоки имеют в текущий момент GUI-окно, принимающее сообщения от пользователя, а их процессы могут быть успешно открыты пользователем функцией OpenProcess с параметром PROCESS_ALL_ACCESS;

- для каждого из таких потоков она встраивает требуемый обработчик в цепочку существующих. Наиболее интересным моментом является то, что функция автоматически проецирует библиотеку и на новые GUI-процессы — процессы, которых не было в момент ее вызова.

Таким образом, в результате исполнения нижеприведенного кода библиотека myhookingdll будет спроецирована на все оговоренные контексты, и все ее копии получат уведомление DLL_PROCESS_ATTACH:

```
call LoadLibraryA, offset myhookingdll, 0
call GetProcAddress, eax, offset mydummyhook, eax
call SetWindowsHookExA, WH_CBT, eax
```

2. Использование удаленных потоков. В WinNT и Win2k существуют функции для создания удаленных потоков. Из них документирована одна — kernel32!CreateRemoteThread. В качестве параметров к ней передаются хэндл процесса, в

котором будет создан поток, его стартовый адрес (в чужом контексте), аргумент функции потока (ее прототип — DWORD WINAPI ThreadFunc(PVOID pvParam)), начальный размер стека и другие. Самый простой способ загрузки библиотеки в чужой контекст с использованием CreateRemoteThread — это указать стартовый адрес потока как адрес функции LoadLibraryA (или W), и поместить в качестве параметра указатель на имя библиотеки. Для этого нужно проделать следующие шаги:

- открыть чужой процесс функцией OpenProcess как минимум с флагами PROCESS_CREATE_THREAD | PROCESS_VM_WRITE;
- выделить память в чужом контексте для размещения там имени библиотеки с путем (функцией VirtualAllocEx);
- скопировать туда полное имя библиотеки (функцией WriteProcessMemory);
- узнать адрес функции LoadLibraryA (W);
- выполнить CreateRemoteThread с начальным EIP, равным полученному адресу LoadLibrary*;
- закрыть процесс и поток.

Все вышеописанные действия проводит следующий код:

```
call OpenProcess, PROCESS_ALL_ACCESS, 1, PID
xchg eax, ebx
call VirtualAllocEx, ebx, 0, mydllnamesize, MEM_COMMIT, \
PAGE_READWRITE, ebx
call WriteProcessMemory, ebx, eax, \
offset mydll, mydllnamesize, 0, eax, 0, 0
call GetModuleHandleA, offset kernel32
call GetProcAddress, eax, offset LoadLibrary
call CreateRemoteThread, ebx, 0, 0, eax
call WaitForSingleObject, eax, INFINITE, eax
call CloseHandle
call CloseHandle
```

(Продолжение следует)

Исправление набранного в неправильном режиме текста с помощью резидентной программы

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>

Введение

Когда приходится набирать текст, содержащий и русские, и латинские буквы, иногда забываешь переключить режим Рус/Lat, и набираемый текст становится бессмысленным. Бывает, что так же забываешь переключить режим Caps Lock, и вместо прописных букв вводятся строчные и наоборот. Приходится стирать неправильно набранный участок текста и вводить его заново.

Я предлагаю способ решения этой проблемы для MS-DOS с помощью резидентной программы. В этой статье подробно рассмотрен процесс ее разработки.

Постановка задачи

Сначала определимся с терминами. Разрабатываемую программу я буду в дальнейшем называть "резидент", а термин "программа" буду использовать для обозначения программы, в которой вы набираете и исправляете текст. Клавиша (или комбинация с Shift, Alt, Ctrl), предназначенная для активизации резидента, будет в дальнейшем называться "горячей клавишей".

Итак, что нам надо? Пусть мы забыли переключить режим Рус/Lat, и в результате неправильно набрали определенный

участок текста. Когда ошибка замечена, хотелось бы нажать определенную "горячую клавишу" (пусть это будет, скажем, F12), после чего неправильно набранный участок текста окажется исправленным. Точно так же, когда мы забыли включить (или наоборот, выключить) режим Caps Lock, хотелось бы нажать другую "горячую клавишу" (скажем, F11) для исправления неправильно набранного текста.

Возможно, при такой постановке решить задачу и не удастся, придется идти на какие-либо компромиссы. Но это выяснится лишь в процессе разработки.

Начнем с простого...

Пусть у нас имеется строка символов, и известно, что при ее наборе был неправильно установлен режим Caps Lock. Как исправить такую строку?

Если режим Caps Lock выключен, то при нажатии клавиш без Shift вводятся строчные буквы, а с Shift — прописные. Если же этот режим включен, то наоборот, без Shift вводятся прописные буквы, а с Shift — строчные. Таким образом, для исправления строки достаточно заменить в ней прописные буквы на строчные и наоборот.

Идем дальше. Теперь пусть у нас имеется строка симво-



лов, и известно, что при ее наборе был ошибочно включен русский (или латинский) режим. Как исправить такую строку?

При нажатии одних и тех же клавиш в русском и латинском режиме вводятся различные символы. Их соответствия приведено в таблице (для русификатора KeyRus с раскладкой клавиатуры "по умолчанию").

№ строки таблицы	Режим Caps Lock	Режим работы клавиатуры	Вводимые символы
1	Выключен	РУС	абвгдежзийклмнопрстуфхцчшщъыьэюяАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"/'::;
2	Выключен	LAT	f,dult;pbqrkvjghcnea[wxi'o]sm'.zF<DULT;PBQRKVYJGHCNEA(WXIO)SM">Z@#%&^&*~
3	Включен	РУС	АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдежзийклмнопрстуфхцчшщъыьэюя"/'::;
4	Включен	LAT	F,DULT;PBQRKVYJGHCNEA(WXIO)SM'.Zf<duilt;pbqrkvjghcnea(wxi'o)sm">z@#%&^&*~

Отсюда непосредственно следует алгоритм исправления строки. Возможны четыре различных случая:

1. Если при наборе строки был ошибочно включен русский режим, и режим Caps Lock был выключен, то отыскиваем в строке все символы, имеющиеся в строке 1, и заменяем их на соответствующие символы из строки 2.
2. Если был ошибочно включен латинский режим, и режим Caps Lock был выключен, то отыскиваем в строке все символы, имеющиеся в строке 2, и заменяем их на соответствующие символы из строки 1.
3. Если был ошибочно включен русский режим, и режим Caps Lock был включен, то отыскиваем в строке все символы, имеющиеся в строке 3, и заменяем их на соответствующие символы из строки 4.
4. Если был ошибочно включен латинский режим, и режим Caps Lock был включен, то отыскиваем в строке все символы, имеющиеся в строке 4, и заменяем их на соответствующие символы из строки 3.

Как видим, для исправления строки, набранной в неправильном режиме Рус/Lat, надо еще знать, включен или выключен был режим Caps Lock при ее вводе. Если считать, что с момента ввода строки и до ее исправления состояние Caps Lock не изменилось, то достаточно узнать текущее состояние Caps Lock. Как это сделать, мы рассмотрим позднее.

Можно заметить, что строка 1 в таблице отличается от строки 3 только тем, что прописные буквы заменены на строчные, и наоборот. То же самое относится к строкам 2 и 4. Тогда выгоднее для экономии места хранить не все четыре строки, а, например, только первую и вторую, а третью и четвертую получать из них, меняя регистр букв на противоположный. Процедура для изменения регистра букв все равно понадобится для исправления текста, набранного при неправильном состоянии Caps Lock.

Взаимодействие резидента с программой

Набираемый в программе текст хранится в некоторой области памяти. Где именно — "знает" только сама программа. И вот пользователь нажимает "горячую клавишу" для исправления текста. Как резидент "узнает", какой именно текст ему надо исправить?

В самом деле, не выводить же окошко с просьбой пользователю, чтобы он ввел еще раз неправильно набранный текст. Очевидно, что тогда использование резидента потеряет всякий смысл — проще будет исправить текст вручную.

Заметим, что набранный текст также отображается на экране! То есть он находится в видеопамети. И непосредственно после неправильно набранной строки текста стоит курсор! Тогда можно, обращаясь к функциям прерывания BIOS 10h, определить координаты курсора, а затем узнать, какие символы находятся слева от курсора — это и будет исправляемый текст.

Но надо еще определить длину исправляемого участка текста. Не обязательно это будет вся расположенная сле-

ва от курсора часть строки. Может быть, пользователь набрал в неправильном режиме только последние несколько символов. Как быть? Не запрашивать же у пользователя длину неправильно введенной строки?

Определять длину автоматически? А как? Это вопрос непростой. Можно предложить разные способы, но ни один не

может обеспечить 100% надежности. Поэтому я решил, что одно нажатие "горячей клавиши" будет исправлять только один символ текста, находящийся слева от курсора, после чего курсор передвинется на символ влево (если какой-то символ, например, цифру, исправлять не нужно, то курсор просто передвинется влево.) Тогда, последовательно нажимая "горячую клавишу", пользователь сам остановится, когда весь неправильно набранный текст будет исправлен.

При нажатии "горячей клавиши" для исправления текста, набранного в неправильном режиме Рус/Lat, резидент должен еще определить, какой режим был ошибочно включен при наборе текста — русский или латинский. Как это сделать? Если считать, что с момента ввода строки и до ее исправления режим Рус/Lat не переключался, то достаточно знать его текущее состояние. Но сведения о том, какой режим в данный момент включен, хранятся где-то внутри драйвера-русификатора, и где именно — "знает" только он сам. Можно было бы запросить эту информацию у пользователя, но хотелось бы обойтись без этого...

Тогда поступим так — будем просматривать строку от позиции курсора до левого края экрана, пока не попадется буква. Если эта буква — русская, значит, при наборе текста был ошибочно включен русский режим; если латинская — соответственно, латинский режим. Если в строке не оказалось буквы, то считаем, что ошибочно включенным был тот же режим, который был определен при нажатии "горячей клавиши" в прошлый раз.

Обратим внимание на следующее. Если пользователь исправляет строку текста, набранную в неправильном режиме Рус/Lat, нажимая для этого нужное число раз соответствующую "горячую клавишу", то определять, какой режим был ошибочно включен при наборе строки, надо только при первом нажатии "горячей клавиши". Дело в том, что если остались неисправленными только небуквенные символы, а слева остался участок строки, не нуждающийся в исправлении и с другими буквами, то определение режима произойдет неверно.

Например, на экране такая строка (рис.1).

Рис. 1 Это первый участок "nj dnjhjq exfcnjr ■ <- курсор

При первом нажатии "горячей клавиши" первый же символ слева от курсора оказывается латинской буквой. Значит, при наборе был ошибочно включен латинский режим. Последовательные нажатия "горячей клавиши" приводят к исправлению строки (рис.2).

Рис. 2 Это первый участок "то второй участок ■ <- курсор

Осталось исправить только первый символ строки. Если сейчас попробовать снова определить, какой режим был неправильно включен при наборе текста, то первой встреченной буквой окажется русская, и будет сделан вывод, что



был неправильно включен русский режим. Это неверно, и дальнейшее исправление будет неправильным (рис.3).

Рис. 3

Это первый участок Это второй участок
■ <- курсор

Перейдем к рассмотрению следующего вопроса. Пусть после нажатия "горячей клавиши" резидент уже определил, какой символ текста должен быть исправлен, и на какой символ его надо заменить. А что он должен сделать дальше? Выше уже упоминалось, что набираемый в программе текст хранится в некоторой области памяти, известной только самой программе. Так что резидент не может сам исправить текст, а может только передать программе указания по его исправлению.

Как это сделать? А как передает указания по исправлению текста пользователь? С помощью нажатия клавиш, конечно. Как вы стали бы исправлять неправильно набранный символ текста? Сначала нажали бы клавишу BackSpace для стирания этого символа, а затем ввели бы правильный.

Значит, резидент должен "подсунуть" программе информацию о нажимаемых клавишах, как будто их нажимает пользователь. А как это сделать?

В подавляющем большинстве случаев программы используют для опроса клавиатуры прерывания BIOS 16h (может быть, и не обращаясь к нему непосредственно). Следовательно, чтобы "подсовывать" программе нужные коды клавиш, нужно перехватить обработку этого прерывания. Помимо этого, перехват прерывания BIOS 16h нужен еще и для того, чтобы отслеживать факт нажатия пользователем "горячих клавиш".

Листаем справочник — прерывание BIOS 10h

Рассмотрим функции прерывания BIOS 10h, которые понадобятся нам для определения положения курсора и определения кодов символов, отображенных на экране.

При вызове функции ее номер указывается в регистре ah. В [1] упоминается, что после окончания работы вызванной функции содержимое регистров bx, cx, dx, si и bp остается неизменным (очевидно, если эти регистры не используются для возвращения результата).

Для определения координат курсора служит функция 3. Перед ее вызовом в регистре bh надо указать номер текущей страницы видеопамати. После ее вызова в регистре dh будет номер строки, а в dl — номер столбца, где расположен курсор. В регистре cx возвращается размер курсора (он нам не понадобится, но знать, что содержимое cx изменится после вызова функции, необходимо).

Для использования этой функции надо знать номер текущей страницы видеопамати. Чтобы его узнать, используется функция f. После ее вызова нужный нам номер строки будет в регистре bh.

Теперь о том, как определить код символа, если известны его координаты на экране. Для этого можно использовать функцию 8. Перед вызовом в регистре bh надо указать номер текущей страницы видеопамати. После вызова код символа, находящегося в позиции курсора, будет в регистре al.

Понятно, что перед вызовом этой функции надо установить курсор в нужную позицию, а после вызова — вернуть курсор на прежнее место. Для задания нового положения курсора служит функция 2. Перед ее вызовом в регистре bh надо указать номер текущей страницы видеопамати, а в dx — новые координаты курсора (dh — номер строки, dl — номер столбца).

Как видим, использование одной функции влечет за собой необходимость использования другой функции, и так далее. Хотя было бы достаточно всего двух функций — "определить координаты курсора в текущей странице ви-

деопамати" и "определить код символа с заданными координатами в текущей странице видеопамати".

Листаем справочник — прерывание BIOS 16h

Функции этого прерывания предназначены для работы с клавиатурой. Даже если программа пользуется более высокоуровневыми функциями DOS, они в итоге все равно обращаются к этому прерыванию.

Вообще работа с клавиатурой организована так — при нажатии или отпускании какой-либо клавиши происходит аппаратное прерывание int 9. Обработчик этого прерывания заносит скан-коды и ASCII-коды нажимаемых клавиш (далее пару [скан-код, ASCII-код] я буду называть просто кодом клавиши) в специальный буфер (далее — клавиатурный буфер). А когда программа вызывает функцию прерывания 16h для получения кодов нажатых клавиш, обработчик этой функции обращается к клавиатурному буферу.

Здесь мы рассмотрим две из функций прерывания BIOS 16h, которые нам придется использовать и/или перехватывать в резиденте. При вызове функции ее номер указывается в регистре ah.

Функция 0 — чтение кода клавиши. Берет из клавиатурного буфера значение скан-кода и ASCII-кода клавиши, и возвращает их соответственно в ah и al. Если в клавиатурном буфере пусто, функция будет ждать, пока там что-нибудь не появится, т.е. пока пользователь не нажмет клавишу.

Функция 1 — чтение кода клавиши без удаления его из клавиатурного буфера. Если буфер не пуст, флаг z сбрасывается, а прочитанный скан-код и ASCII-код возвращаются соответственно в ah и al. Если в клавиатурном буфере пусто, функция не ждет нажатия клавиши, а при выходе устанавливает флаг z.

Функция 2 — опрос состояния служебных клавиш. Результат возвращается в регистре al. Эта функция понадобится нам, чтобы узнать, включен или выключен режим Caps Lock. Если 6-й бит регистра al равен 0, то режим выключен; если 1 — включен.

Функции 10h и 11h — то же самое, что функции 0 и 1, но с их помощью можно также получить коды дополнительных клавиш, имеющихся на 101-клавишной клавиатуре, и коды некоторых комбинаций клавиш (Alt+Enter и др.). Эти функции, как сказано в [2], присутствуют в BIOS, выпущенном после 15 декабря 1985 г.

Функции 0, 1, 10h, 11h должны быть перехвачены резидентом, чтобы "подсовывать" программе нужные коды клавиш. За одно обращение к функции можно передать программе только один код клавиши, а для исправления неправильно введенного символа надо передать несколько кодов. Следовательно, надо организовать в резиденте специальный буфер (далее — внутренний буфер), помещать оставшиеся коды туда, а при следующих обращениях программы к функциям int 16h для опроса клавиатуры — возвращать коды из этого буфера.

Как уже упоминалось выше, эти функции должны возвращать в ah скан-код клавиши, а в al — ASCII-код. Но, как показала практика (и изучение скан-кодов, возвращаемых русификатором KeyRus), если функция возвращает ASCII-код русской или латинской буквы, то скан-код при этом может быть нулевым. То есть не надо хранить в резиденте таблицу соответствия букв и скан-кодов клавиш, с помощью которых эти буквы вводятся, и не нужна подпрограмма получения скан-кода по ASCII-коду буквы.

Комментарии

От редакции: из-за большого объема листинг программы не приводится, он выложен на web-странице журнала <http://radio-mir.com/> Там же можно взять уже откомпилиро-



ванную резидентную программу.

Резидентная часть программы занимает в памяти меньше килобайта (912 байтов, если быть точным).

Если в используемом вами русификаторе другая раскладка клавиатуры (обычно различие в том, какие клавиши используются для ввода знаков препинания в русском режиме), то следует соответствующим образом изменить строки `rus_chars` и `lat_chars`, определяющие взаимное соответствие символов, вводимых в русском и латинском режимах.

Если BIOS вашего компьютера не поддерживает 101-клавишную клавиатуру (т.е. выпущен раньше 15 декабря 1985 г.), надо заменить вызовы функций 10h и 11h на вызовы функций 0 и 1 соответственно. Соответствующие строки программы отмечены комментарием "!!!".

Как уже упоминалось выше, для исправления текста, набранного при неправильном состоянии Caps Lock, я выбрал клавишу F11, а для исправления текста, набранного при неправильном режиме Рус/Лат, — клавишу F12. Вы можете выбрать другие "горячие клавиши", изменив значения констант `KEY_1` и `KEY_2`. Старший байт константы содержит скан-код клавиши (или комбинации клавиш), а младший байт — ASCII-код.

Полный список пар [скан-код, ASCII-код] для всех клавиш и комбинаций Shift+клавиша, Alt+клавиша, Ctrl+клавиша я не привожу из-за его большого размера. Вместо этого предлагаю воспользоваться приведенной ниже программой, которая при нажатии клавиши (или комбинации клавиш) печатает шестнадцатеричное число, первые две цифры которого — скан-код, а вторые две — ASCII-код. Выход из программы — по нажатию Esc.

```
codeseg segment
    assume cs:codeseg
    assume ds:codeseg

    org 100h

main proc near

    mov ah,10h ;Или mov ah,0 для старого
                ;BIOS.
    int 16h ;Ждем нажатия клавиши.

    cmp ax,011bh ;Если нажата Esc - выходим.
    je exit

    mov bx,offset string
    push bx

    mov cl,4 ;Подготавливаем
    mov ch,ah ;выводимую строку
    shr ch,cl ;в буфере.
    call one_digit
    mov ch,ah
    call one_digit
    mov ch,al
    shr ch,cl
    call one_digit
    mov ch,al
    call one_digit

    pop dx ;Печать строки.
    mov ah,9
    int 21h

    jmp main

exit: ret

main endp

one_digit proc near

    push ax
```

```
        and ch,0fh
        add ch,'0'
        cmp ch,'9'
        jna ml
        add ch,'a'-'0'-10
ml:     mov [bx],ch
        inc bx
        pop ax
        ret

one_digit endp

string db '0000',13,10,'$'

codeseg ends
end main
```

“Бонусы” резидента

Это возможности резидента, которые изначально при его разработке не задумывались, а получились “сами собой”.

Если текущий режим — русский, а надо ввести одну латинскую букву (или наоборот, текущий режим — латинский, а надо ввести одну русскую букву), то может оказаться проще ввести букву, не переключая режим, а затем исправить ее нажатием соответствующей “горячей клавиши”.

Точно так же можно поступить, когда текущий режим — русский, а надо ввести один из символов <, >, [,], {, }, которые вводятся только в латинском режиме, или когда надо ввести один из символов, вместо которых в русском режиме вводятся знаки препинания (для русификатора KeyRus с раскладкой клавиатуры “по умолчанию” это символы @, #, \$, %, ^, &, *). В последнем случае для правильного исправления символа последняя буква в части строки слева от этого символа должна быть русской.

Большое неудобство при наборе текста доставляет то, что знаки препинания вводятся по-разному в русском и латинском режимах. Допустим, вы набираете текст в латинском режиме, а знак препинания ввели так, как он вводится в русском режиме (или наоборот, набираете текст в русском режиме, а знак препинания ввели так, как он вводится в латинском режиме). В этом случае достаточно один раз нажать “горячую клавишу” для исправления текста, набранного в неправильном режиме Рус/Лат, и неверно введенный знак будет исправлен!

“Горячую клавишу” для исправления текста, набранного при неправильном состоянии Caps Lock, можно использовать не только для исправления ошибок, но и просто чтобы заменить в какой-то части текста прописные буквы на строчные или наоборот (например, выделить заголовок прописными буквами). Просто подводим курсор в позицию за последним символом изменяемой строки и нажимаем “горячую клавишу”, пока вся строка не будет обработана.

Можно, разумеется, изменять таким образом не строку, а лишь одну букву. При редактировании текста довольно часто приходится заменять прописную букву на строчную и наоборот — например, при добавлении слова в начало предложения и при удалении первого слова предложения; при разбиении одного предложения на два и при объединении двух предложений в одно.

Литература

1. П.Абель. Язык ассемблера для IBM PC и программирования. — М.: Высшая школа, 1992.
2. А.Фролов, Г.Фролов. Аппаратное обеспечение IBM PC. — М.: Диалог-МИФИ, 1992.
3. А.Фролов, Г.Фролов. Программирование видеоадаптеров CGA, EGA и VGA. — М.: Диалог-МИФИ, 1992.
4. Ф.Пьеро, Ж.-Л.Люкзак, Ф.Рейко и др. Руководство по программированию под управлением MS-DOS. — М.: Радио и связь, 1995.



Сделай сам: Управление Winamp'ом

И.ШИРКО,
E-mail: FDC@tut.by

В этот раз мы поговорим об управлении популярнейшим аудио-плеером, имя которому — Winamp. Благодаря своей "нулевой" цене, поддержке скинов и, что самое главное, плагинов, Winamp установлен на огромном количестве ПК. И этим программным шедевром от Nullsoft, оказывается, можно довольно легко управлять программно. В связи с этим большинство soft-каталогов не обходится без пары-тройки программ управления Winamp'ом. Давайте рассмотрим, что же происходит в недрах таких программ.

Winamp поддерживает большое количество сообщений, которые можно послать его окну при помощи функции

```
SendMessage(  
    HWND:hWnd,           // идентификатор окна  
    Msg:Integer,          // посылаемое сообщение  
    WParam:Integer,       // первый параметр сообщения  
    LParam:Integer        // второй параметр сообщения  
);
```

Имеются два основных типа сообщений (параметр Msg), которые поддерживает Winamp:

- **WM_COMMAND** — служит только для подачи определенных команд Winamp (Play, Stop, Next, Close и т.д.);
- **WM_USER** — используется не только для выполнения определенных действий, но и для определения различной информации (версия, текущая композиция, количество композиций и т.д.).

Табл. 1

Команда	Описание команды
40044	Кнопка "Prev"
40048	Кнопка "Next"
40045	Кнопка "Play"
40046	Кнопка "Pause"
40047	Кнопка "Stop"
40157	Остановиться после текущей композиции
40148	На 5 секунд вперед
40144	На 5 секунд назад
40154	Перейти на первую песню Playlist'a (режим "Shuffle" — случайное проигрывание — должен быть отключен)
40158	Перейти на последнюю песню Playlist'a (режим "Shuffle" должен быть отключен)
40192	Запустить плагин визуализации
40036	Показать/Скрыть эквалайзер
40040	Показать/Скрыть редактор Playlist'ов
40258	Показать/Скрыть главное окно
40298	Показать/Скрыть мини-браузер
40022	Кнопка "Repeat"
40023	Кнопка "Shuffle"
40188	Показать информацию о файле
40058	Увеличить громкость на один процент
40059	Уменьшить громкость на один процент
40001	Закрыть Winamp

В табл.1 приведены основные константы для первого параметра сообщения **WM_COMMAND**. Для их использования запишите процедуру:

```
Procedure WinampCommand (Command:Integer);  
begin  
    //поиск окна Winamp'a  
    WinampHWND:=findwindow('Winamp v1.x',nil);  
    //если поиск успешен,то посылаем сообщение  
    if WinampHWND<>0 then  
        SendMessage(WinampHWND, WM_Command, Command, 0);  
end;
```

Теперь для подачи сообщения типа **WM_USER** нужно выбрать из таблицы понравившуюся константу и пере-

дать ее в качестве параметра процедуре WinampCommand. Пример:

WinampCommand(40044); — переход к предыдущей композиции.

Табл. 2

Id	Data	Пояснения
0	0	Возвращает версию Winamp'a
102	0	Начать проигрывать выбранную в playlist'e композицию
104	0	Возвращает статус проигрывания: 1 — играет, 3 — пауза, иначе остановлен
105	0	Возвращает в миллисекундах позицию проигрывания
105	1	Возвращает в секундах длину композиции
121	n	Выбирает в playlist'e композицию под номером n
122	n	Устанавливает громкость в значение n (от 0 до 255)
123	n	Устанавливает баланс в значение n (от 0 до 255)
125	0	Возвращает позицию текущей песни в playlist'e
126	0	Возвращает samplerate (частоту дискретизации)
126	1	Возвращает bitrate
126	2	Возвращает количество каналов (1 — моно, 2 — стерео)
135	0	Перезагружает Winamp

В табл.2 перечислены основные константы для сообщений **WM_USER**. Для их использования запишите следующую функцию:

```
Function WinampUser (data:Integer, id:Integer):integer;  
begin  
    WinampHWND:=findwindow('Winamp v1.x', nil);  
    if WinampHWND<>0 then  
        result:=SendMessage(WinampHWND, WM_USER, data, id)  
    else result:=-1;  
end;
```

Пример:

WinampUser(1, 105); — возвращает длину текущей композиции в секундах.

Итак, мы научились управлять Winamp'ом, осталось только воплотить наши навыки в небольшую программу. Создайте новый проект, поместите на форму двенадцать кнопок (рис.1) и запишите единственный модуль проекта:

interface

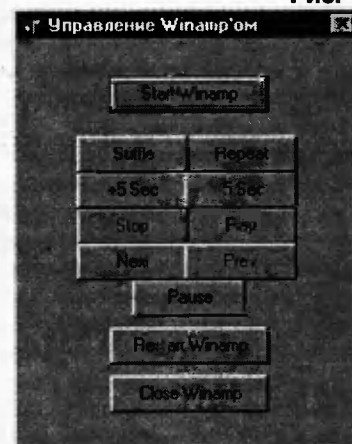
uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, Registry, StdCtrls;

type

```
TForm1 = class(TForm)  
    Button1: TButton;  
    Button2: TButton;  
    Button3: TButton;  
    Button4: TButton;  
    Button5: TButton;  
    Button6: TButton;  
    Button7: TButton;
```

Рис. 1





```
Button8: TButton;
Button9: TButton;
Button10: TButton;
Button11: TButton;
Button12: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  reg:TRegistry;
  WinampHWND:HWND;
implementation

{$R *.DFM}

Function WinampUser(data:Integer; id:Integer):integer;
begin
  WinampHWND:=findwindow('Winamp v1.x', nil);
  if WinampHWND<>0 then
    result:=SendMessage(WinampHWND, WM_USER, data,id)
  else result:=-1;
end;
Procedure WinampCommand(Command:Integer);
begin
  WinampHWND:=findwindow('Winamp v1.x',nil);
  if WinampHWND<>0 then
    SendMessage(WinampHWND,messages.WM_command,Command,0);
end;
//Sanyck Winamp'a
Procedure StartWinamp;
var
  WinampExe:string;
begin
  reg:=TRegistry.create;
  reg.RootKey:=Windows.HKEY_LOCAL_MACHINE;
  reg.OpenKey('Software\CLASSES\Winamp.File\shell\open\command', False);
  WinampExe:=reg.ReadString('');
  reg.Free;
  if WinampExe<>'' then
    begin
      WinampExe:=copy(WinampExe,2,length(WinampExe)-7);
      WinExec(pchar(WinampExe),SW_SHOWNORMAL);
    end;
end;
Procedure Play;
begin
  WinampCommand(40045);
end;
Procedure Stop;
begin
  WinampCommand(40047);
end;
Procedure Pause;
begin
  WinampCommand(40046);
end;
Procedure NextTrack;
begin
  WinampCommand(40048);
end;
Procedure PrevTrack;
begin
  WinampCommand(40044);
end;
{передвинуть текущую позицию на 5 секунд вперед}
Procedure FastForward5Sec;
```

```
begin
  WinampCommand(40148);
end;
{передвинуть текущую позицию на 5 секунд назад}
Procedure FastRewind5Sec;
begin
  WinampCommand(40144);
end;
{перезагрузить Winamp}
Procedure RestartWinamp;
begin
  WinampUser(0, 135);
end;
Procedure CloseWinamp;
begin
  WinampCommand(40001);
end;
Procedure RepeatMode;
begin
  WinampCommand(40022);
end;
Procedure SuffleMode;
begin
  WinampCommand(40023);
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
  StartWinamp;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
  Play;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
  Stop;
end;
procedure TForm1.Button4Click(Sender: TObject);
begin
  Pause;
end;
procedure TForm1.Button5Click(Sender: TObject);
begin
  NextTrack;
end;
procedure TForm1.Button6Click(Sender: TObject);
begin
  PrevTrack;
end;
procedure TForm1.Button7Click(Sender: TObject);
begin
  CloseWinamp;
end;
procedure TForm1.Button8Click(Sender: TObject);
begin
  FastForward5Sec;
end;
procedure TForm1.Button9Click(Sender: TObject);
begin
  FastRewind5Sec;
end;
procedure TForm1.Button10Click(Sender: TObject);
begin
  RestartWinamp;
end;
procedure TForm1.Button11Click(Sender: TObject);
begin
  RepeatMode;
end;
procedure TForm1.Button12Click(Sender: TObject);
begin
  SuffleMode;
end;
end.
```

Разумеется, это только прототип. В одной из следующих статей мы обязательно превратим программу в плагин для Winamp'a и добавим поддержку горячих клавиш. На сегодня все.

Жду ваших предложений по поводу тем следующих статей цикла "Сделай сам".

Адаптер ввода-вывода

В журнале уже описывался эмулятор интерфейса ISA, который позволяет налаживать и ремонтировать различные адаптеры для компьютера [1].

Теперь я хочу предложить вниманию читателей еще одну мою разработку — адаптер ввода-вывода, который подключается к компьютеру через интерфейс ISA и имеет в своем составе регистры ввода и вывода, предназначенные для работы с различными периферийными устройствами (датчиками контроля, кнопками, индикаторами, исполнительными устройствами и т.п.).

Данный адаптер позволит начинающим разработчикам ознакомиться с принципами построения устройств расширения персонального компьютера через интерфейс ISA и, при желании, повторить эту конструкцию для расширения возможностей своего компьютера.

Адаптер позволяет подключить до 32-х устройств ввода и до 24-х устройств вывода. К примеру, при решении задачи охраны помещения, с его помощью можно контролировать до 32-х устройств контроля (датчиков открытия окон и дверей, кнопок пульта управления и т.д.) и управлять 24-мя исполнительными устройствами (сиреной, аварийной лампой, электрозамком и пр.).

Рассмотрим структурную схему адаптера, которая приведена на рис.1. В состав адаптера входят следующие узлы:

- И — интерфейс;
- БД — буфер данных;
- П — поле перемычек;
- И1 — индикатор питания;
- И2 — индикатор обращения;
- СР — схема разрешения;
- РК — регистры контроля;
- РУ — регистры управления;
- ТР — твердотельные реле.

При включении адаптера, схема разрешения "СР" по сигналу сброса от шины ISA принудительно переходит в состояние запрета выходов регистров управления "РУ", и последние переводят выходы управления в третье состояние, что обеспечивает выключение всех исполнительных устройств в начальный момент. Индикатор "И1" индицирует подачу питания на адаптер.

Поле перемычек "П" определяет базовый адрес модуля, при обращении по которому адаптер будет отзываться. При совпадении адреса устройства

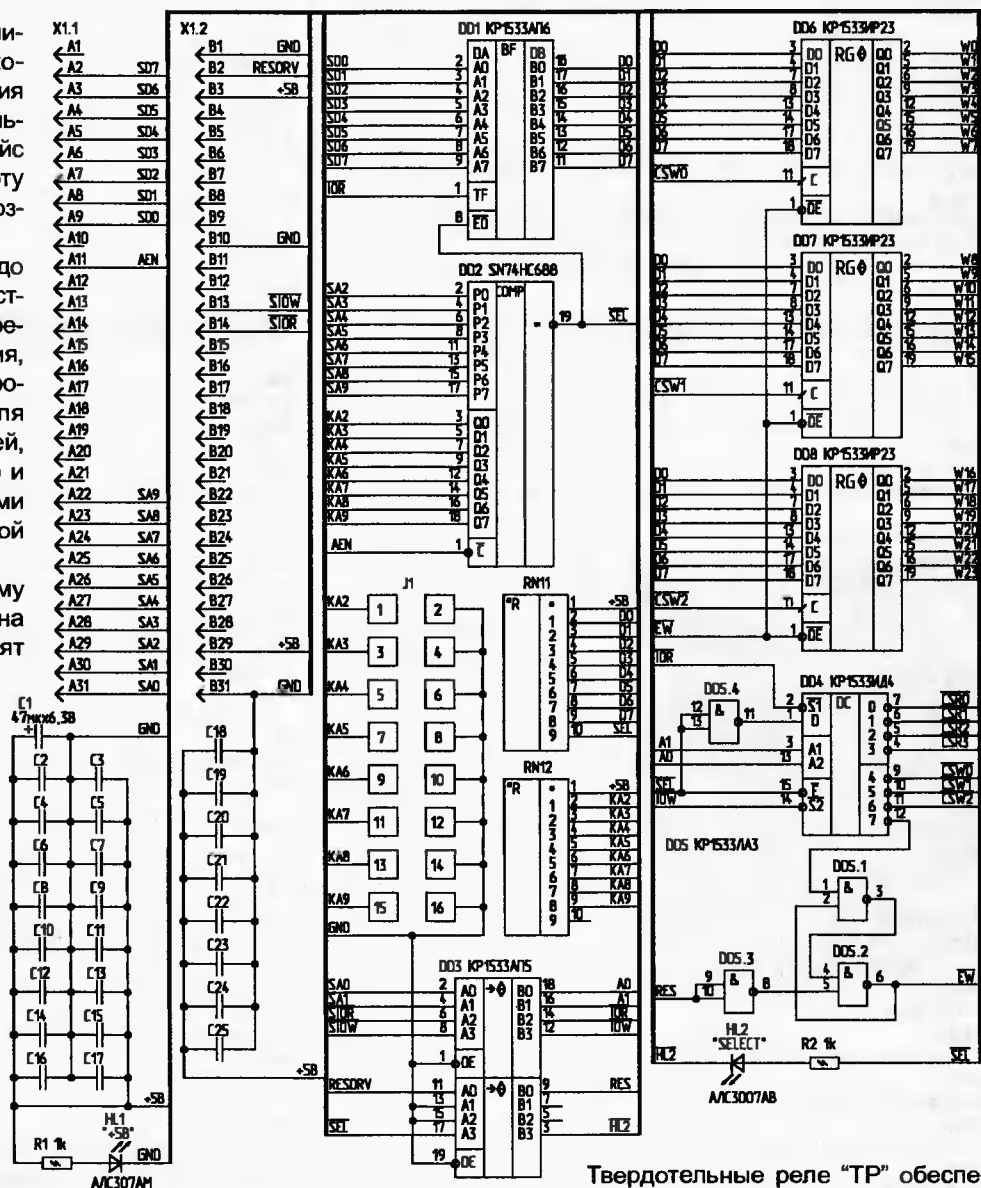
один из синхросигналов "С1"... "С8", которые управляют чтением и записью регистров контроля "РК" и регистров управления "РУ".

Схема разрешения "СР" активизирует выходы регистров управления "РУ" и включает те устройства управления, в управляющие разряды которых были записаны логические нули.

Табл. 1

Разряды адреса	9	8	7	6	5	4	3	2	1	0	
Группа контактов	15, 16	13, 14	11, 12	9, 10	7, 8	5, 6	3, 4	1, 2	-	-	
Базовый адрес	1	0	1	1	1	0	0	0	0	0	2F0h
Состояние группы	P	3	P	P	P	P	3	3	-	-	
Базовый адрес	1	1	0	0	0	0	0	0	0	0	300h
Состояние группы	P	P	3	3	3	3	3	3	-	-	

Примечание: "P" — контакты разомкнуты, "3" — контакты замкнуты.



ввода-вывода на шине ISA с базовым адресом модуля, интерфейс "И" заживает индикатор "И2" и вырабатывает

Твердотельные реле "ТР" обеспечивают согласование логических выходов регистров управления "РУ" с исполнительными устройствами.

Принципиальная электрическая

схема адаптера приведена на рис.2.

Интерфейс "И" выполнен на микросхемах DD2, DD3, DD4, DD5.4 и наборе резисторов RN12. В качестве буфера данных используется микросхема DD1 типа КР1533АП6 и набор резисторов RN11. Поле переключателей выполнено на элементе J1. Индикаторы построены на базе элементов HL1, HL2 и R1, R2. Схема разрешения "CP" выполнена на элементах DD5.1...DD5.3. Функции регистров контроля выполняют микросхемы DD9...DD12 и наборы резисторов RN1...RN4. Функции регистров управления выполняют микросхемы DD6...DD8 и наборы резисторов RN5...RN10. В качестве твердотельных реле применены элементы U1...U12.

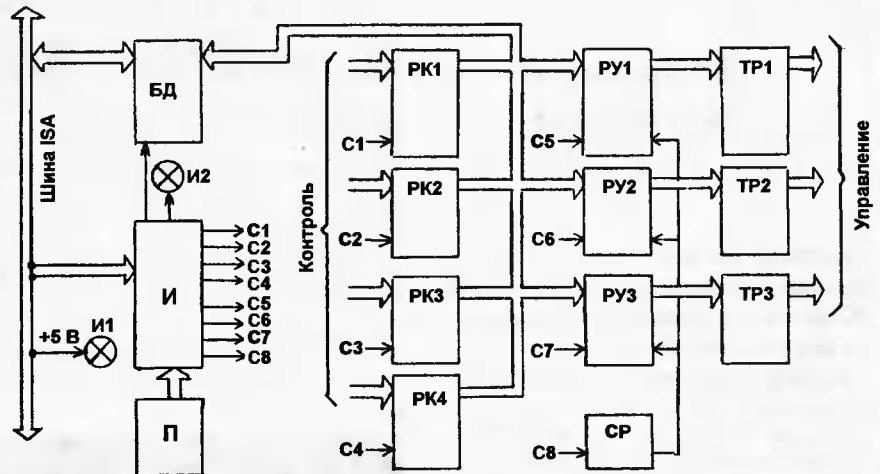
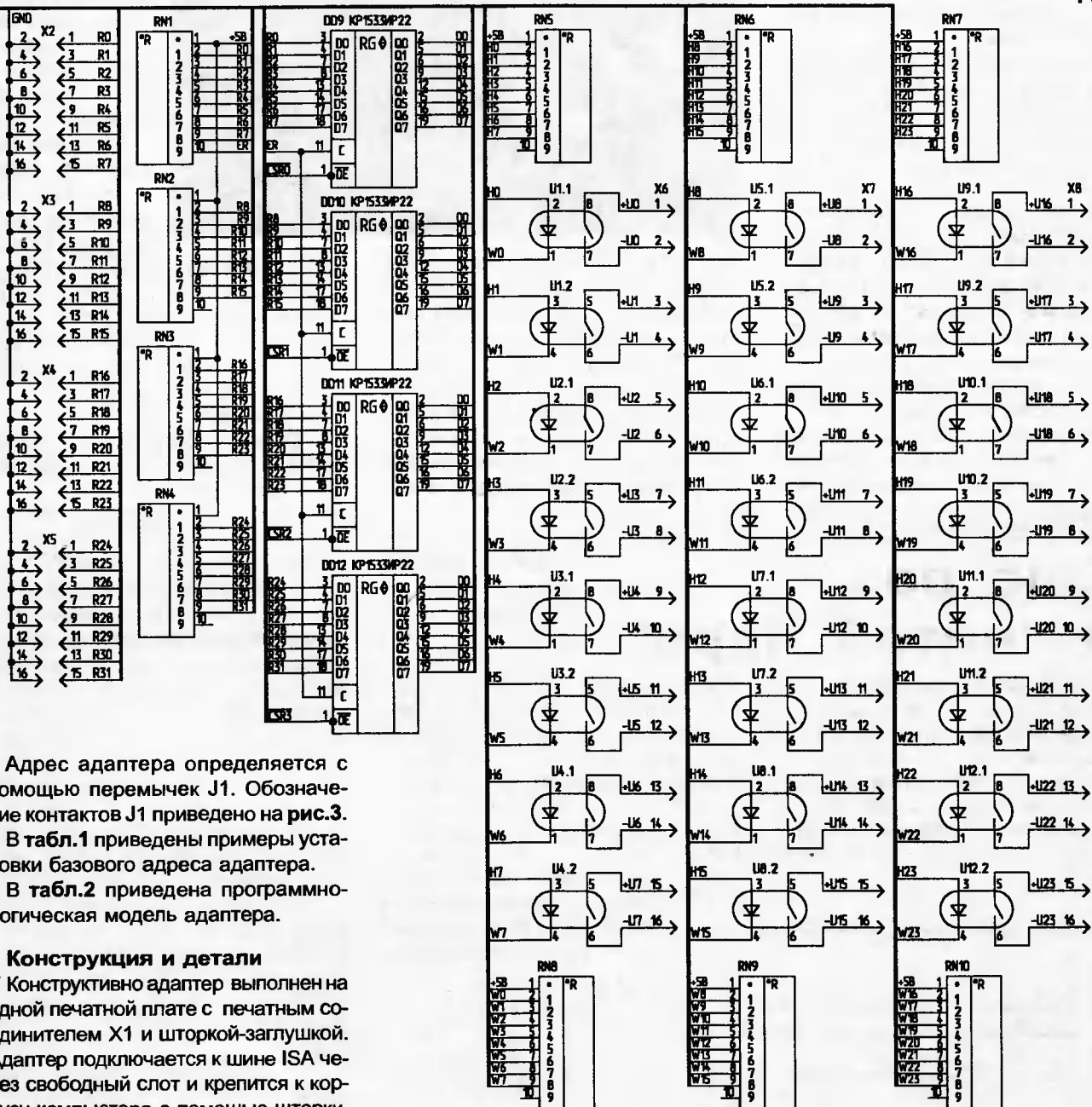


Рис. 2



Адрес адаптера определяется с помощью переключек J1. Обозначение контактов J1 приведено на рис.3.

В табл.1 приведены примеры установки базового адреса адаптера.

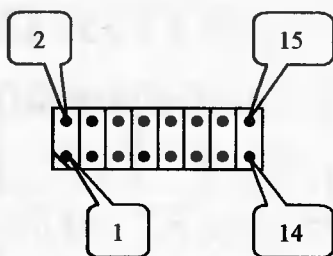
В табл.2 приведена программно-логическая модель адаптера.

Конструкция и детали

Конструктивно адаптер выполнен на одной печатной плате с печатным соединителем X1 и шторкой-заглушкой. Адаптер подключается к шине ISA через свободный слот и крепится к корпусу компьютера с помощью шторки.



Рис. 3



Топология печатной платы была получена в конструкторском пакете PCAD 4.5 и доступна для скачивания на сайте журнала.

В адаптеры были использованы следующие элементы.

Конденсаторы: C1 — электролитический, типа K50-35; C2...C25 — керамические, типа K10-17Б-Н90.

J1 — любые переключки или DIP-переключатель.

Наборы резисторов RN1...RN4, RN11, RN12 — типа HP1-4-9M-0,125, а RN5...RN10 — HP1-4-9M-0,125.

Резисторы R1, R2 — типа C2-23-0,125.

Твердотельные реле U1...U12 — типа 5П14.3Б.

X2...X8 — соединители типа ВН-16.

Вместо микросхем серии КР1533 можно применить микросхемы серии К555 или их зарубежные аналоги. Допускается устанавливать не все регистры контроля и управления и связанные с ними элементы, а только то количество, которое необходимо для решения поставленной задачи.

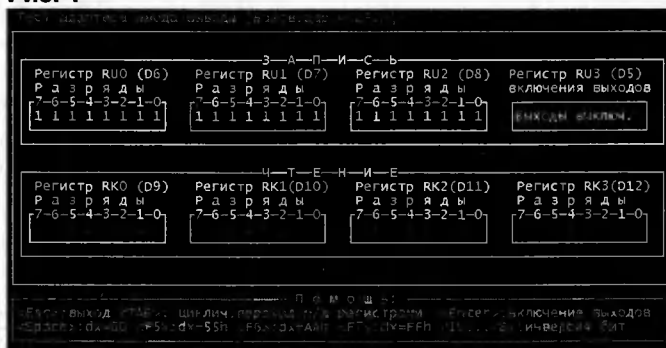
Для проверки данного адаптера мною была написана тестовая программа "test_avv" на языке программирования

Табл. 2

Адрес порта	Операция	Назначение устройства
БА+0	Чтение	Регистр контроля 0
БА+1	Чтение	Регистр контроля 1
БА+2	Чтение	Регистр контроля 2
БА+3	Чтение	Регистр контроля 3
БА+0	Запись	Регистр управления 0
БА+1	Запись	Регистр управления 1
БА+2	Запись	Регистр управления 2
БА+3	Запись	Схема разрешения (запись любого значения по этому адресу переводит выходы всех регистров управления из третьего состояния в активное)

Примечание: БА (базовый адрес) адаптера выбирается пользователем из области свободных адресов компьютера и устанавливается переключками на адаптере. Например, при выборе адреса 2F0h необходимо установить переключки J1 — 1-2, 3-4, 13-14.

Рис. 4



"Си", которая позволяет проверить работоспособность адаптера в составе компьютера и, кроме того, протестировать все подключенные к нему устройства. Программа запускается в режиме DOS и отображает на экране монитора состояние всех регистров адаптера. Внешний вид этого экрана приведен на рис.4. Справка по управлению программой выводится на самом экране. Исход-

ный текст программы с комментариями может послужить пособием для начинающих программистов и разработчиков при написании собственных тестовых и других программ.

От редакции: Исходный текст программы в жур-

нале не приводится из-за значительно-го объема листинга. Он, а также исполняемый код программы тестирования доступны для скачивания на сайте журнала, либо могут быть высланы по предварительному запросу.

Литература

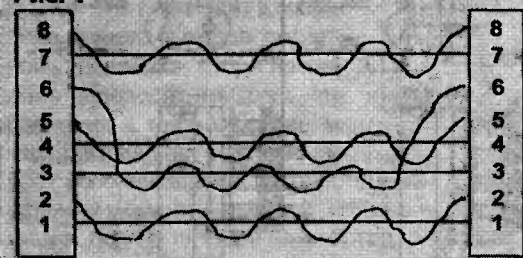
1. О.Вальпа. Эмулятор интерфейса ISA. — Радиомир. Ваш компьютер, 2002, №7, С.33.

ОБМЕН ОПЫТОМ

Еще раз о "витой паре"

Сейчас широко используют локальные сети на "витой паре" — кабеле из 4-х пар проводов, свитых с определенным шагом. Скручивание жил уменьшает емкость линии и повышает эффективную рабочую частоту кабеля. Концы проводов кабеля обжимают в 8-контактных разъемах RJ-45. Жилы

Рис. 1



кабеля расплетают на длину не более 1 см, вставляют в разъем и обжимают. При самостоятельном изготовлении кабеля на "витой паре", концы проводов обжимают, используя подручный инструмент — пассатижи, отвертку и т.п. Жилы в разьеме можно подсоединять в любом порядке, необходимо лишь, чтобы одноименные контакты разъемов прозванивались — такой кабель будет работать. Но при расстоянии между ра-

Рис. 2



бочей станцией и хабом более 10...15 м компьютер перестает "видеть" сеть, или скорость передачи данных резко падает. Поэтому, чтобы добиться максимальной длины сегмента локальной сети (согласно техническим условиям — до 100 м), нужно обеспечить правильную разводку проводников в разъемах [1]. И лучше не экономить, а приобрести специальный инструмент для обжима проводников в разъемах RJ-45, который обеспечит качественную заделку кабеля и надежность соединения.

Схема разводки проводников в прямом кабеле (UTP-порт, RJ-45, "мама"), который применяется чаще всего, приведена на рис.1. На рис.2 показан вид на сетевую карту со стороны подключения кабеля.

Литература

1. М.Гук. Аппаратные средства IBM PC. 2-е изд. — М.: Солон, 2000, С.206.



Музыкальный сопроцессор “Dendy” vs. “ZX-Spectrum”

С.РЮМИК,
г.Чернигов.

Речь пойдет о сравнении возможностей музыкальных сопроцессоров (MC) RP2A03G игровой приставки “Dendy” и AY-3-8910/12 домашнего компьютера “ZX-Spectrum-128” (“ZX”). В пору их расцвета и популярности, в конце 80-х—начале 90-х годов, сравнительный анализ не проводился. Причин несколько, но основная из них — недостаток технической информации о внутреннем устройстве MC.

По прошествии лет, и “Dendy”, и “ZX” перешли в разряд ретро-компьютеров. Появился доступ в Интернет к обилию справочных данных. Теперь можно в спокойной обстановке оценить их музыкальные возможности и выяснить сильные и слабые стороны.

Немного истории

Программируемые генераторы звуков AY-3-8910, AY-3-8912, AY-3-8913 были разработаны американской фирмой General Instrument Corporation (GI, см. рисунок) и выпускались ее подразделением на Тайване. Музыкальные возможности всех трех микросхем одинаковы, различие лишь в количестве дополнительных восьмиразрядных портов, соответственно, два, один и ни одного. На рынке игровых компьютеров фирма GI не была новичком, имея, как минимум, десятилетний стаж работы. Например, в 1976 г. она выпустила в продажу одну из первых общедоступных игровых БИС. С ее помощью можно было сделать телевизионную приставку, содержащую 4 игры типа “Теннис” и 2 игры со световым пистолетом. И все это за 5 долларов!



Небольшой нюанс. Иногда можно встретить искаженное название фирмы GI, в конце которого добавлена буква “s”. Подобная практика берет начало от грамматического правила в английском языке, согласно которому к существительному в притяжа-

тельном падеже приписывается через апостроф буква “s”. Пример правильного выражения: “General Instrument’s AY-3-8910/12”. Если “забыть” про апостроф, то фирма GI превратится в General Instruments, что неверно.

Микросхема AY-3-8910 получила известность в 1983 г., когда она вошла в состав звукового тракта компьютеров серии MSX-1. На протяжении последующих трех лет во всем мире было выпущено около 30 дешевых моделей MSX-совместимых компьютеров. В странах СНГ наиболее известна модель Yamaha-MSX, которой оснащались учебные компьютерные классы.

Сэр Клайв Синклер применил AY-3-8910/12 в “ZX” в конце 1985 г. в надежде на доступность музыкального программного обеспечения MSX, и он не ошибся. Действительно, конвертировать звуковые файлы одной системы в другую оказалось достаточно просто.

На рубеже 90-х годов базовое подразделение по производству микросхем фирмы GI было продано, и на его основе образовалась широко известная ныне фирма Microchip Technology. Музыкальные синтезаторы AY-3-8910/12 не были сняты с производства и еще

какое-то время изготавливались подразделением Microchip в Тайване.

Прототипом MC “Dendy” послужила 40-выводная микросхема RP2A03G, которая в 1983 г. вместе с центральным процессором RP2C02G вошла в состав первых экземпляров игровой приставки “Family Computer” (сокращенно “Famicom”) японской фирмы Nintendo. Судя по одинаковым срокам разработки приставки “Famicom” и компьютеров MSX-1, налицо два конкурирующих варианта введения музыкальных сопроцессоров в игровые системы. В 1985 г. появился американский аналог приставки “Nintendo Entertainment System” (NES), завоевавший популярность во всем мире. Dendy-совместимые приставки по архитектуре больше напоминают “Famicom”, чем NES. В частности, у них совпадает цоколевка разъемов картриджа и джойстиков.

В табл.1 приведены результаты анализа музыкальных возможностей “Dendy” и “ZX”, основанные на сведениях из фирменных Data manual и Datasheet AY-3-8910/12 [1, 2] и информации с интернет-сайта <http://www.parodius.com/~members/>,

Табл. 1

Параметры	AY3-8910/12 ("ZX-SPECTRUM-128")	RP2A03G ("Dendy")
Тактовая частота, МГц	1,7734	1,7734
Число звуковых каналов	3 (A, B, C)	5 (A, B, T, N, PCM)
Число регистров управления	14	18
Диапазон выходных частот, Гц	27...110837 (A, B, C)	54...12315 (A, B); 27...55420 (T)
Разрядность делителя частоты	12	11
Генератор тональных сигналов: - разрядность ЦАП, бит; - шкала уровней	4 Логарифмическая	4 Линейная
Генератор огибающей: - период, мс; - число разновидностей форм	0,14...9460 5	15...2200 3
Генератор шума N: - частота, кГц; - разрядность ПСП, бит	3,5...110,8 -	0,03...447 93; 32767
Свип-генератор ("глиссандо"): - диапазон частот, Гц; - длительность, с	- -	54...f...12315 0,6...28
Канал "цифрового" звука PCM	Внешний	Внутренний



Табл. 2

Регистр "Dendy"		Номер бита (от старшего к младшему)							
Адрес	Функция, канал	D7	D6	D5	D4	D3	D2	D1	D0
4000h	Управление, А	Скважность		Форма		Амплитуда/период			
4001h	"Глиссандо", А	/OE	Скорость изменения				Частота		
4002h	Частота, А	8 младших битов делителя частоты (точный тон)							
4003h	Длительность, А	Длительность одиночного импульса					3 старших бита делителя		
4004h	Управление, В	Скважность		Форма		Амплитуда/период			
4005h	"Глиссандо", В	/OE	Скорость изменения				Частота		
4006h	Частота, В	8 младших битов делителя частоты (точный тон)							
4007h	Длительность, В	Длительность одиночного импульса					3 старших бита делителя		
4008h	Управление, Т	/OE	Длительность одиночного импульса						
400Ah	Частота, Т	8 младших битов делителя частоты (точный тон)							
400Bh	Длительность, Т	Длительность одиночного импульса					3 старших бита дел.		
400Ch	Управление, N	-	-	Форма		Амплитуда/период			
400Eh	Частота, N	ПСП	-	-	-	Частота шума			
400Fh	Длительность, N	Длительность одиночного импульса					-	-	-
4010h	Управление, PCM	Режим		-	-	Частота выборки			
4011h	Амплитуда, PCM	-	7 битов дельта-значений амплитуды						
4012h	Адрес, PCM	8 битов для расчета начала адреса блока данных в ПЗУ							
4013h	Данные, PCM	8 битов для расчета длины блока данных в ПЗУ							
4015h	Общее управление	-	-	-	PCM	N	T	B	A

Примечание: /OE — разрешение режима

Табл. 3

Регистр "ZX-128"		Номер бита (от старшего к младшему)							
Номер	Функция, канал	D7	D6	D5	D4	D3	D2	D1	D0
R0	Точная частота, А	8 младших битов делителя частоты (точный тон)							
R1	Грубая частота, А	-	-	-	-	4 старших бита делителя			
R2	Точная частота, В	8 младших битов делителя частоты (точный тон)							
R3	Грубая частота, В	-	-	-	-	4 старших бита делителя			
R4	Точная частота, С	8 младших битов делителя частоты (точный тон)							
R5	Грубая частота, С	-	-	-	-	4 старших бита делителя			
R6	Частота шума	5 битов делителя частоты шума							
R7	Смеситель	Ю-0	Ю-1	N-C	N-B	N-A	T-C	T-B	T-A
R10	Амплитуда, А	-	-	-	/OE	Амплитуда			
R11	Амплитуда, В	-	-	-	/OE	Амплитуда			
R12	Амплитуда, С	-	-	-	/OE	Амплитуда			
R13	Огибающая точно	8 младших битов делителя частоты (точный тон)							
R14	Огибающая грубо	8 старших битов делителя частоты (грубый тон)							
R15	Форма огибающей	-	-	-	-	Cont.	Att.	Alt.	Hold

Примечания: Ю — порт ввода-вывода; N — шум; Т — тон; /OE — разрешение режима; Cont. — продолжение; Att. — затухание; Alt. — инверсия; Hold — пауза

посвященного NES- и Dendy-совместимым приставкам.

Тактовая частота у обоих MC одинакова и совпадает с точностью до пятого знака. Это не случайно, поскольку "Dendy" и фирменный "ZX" формируют телевизионный сигнал стандарта PAL с цветовой поднесущей 4,43 МГц. Тактовая частота MC получается делением частоты 4,433618 МГц

ровно на 2,5. Принцип формирования звуков у обоих MC также одинаков — деление частоты опорного (тактового) генератора на изменяющиеся во времени коэффициенты. Кстати, идентичность номинала тактовой частоты позволяет утверждать, что для формирования нот в "Dendy" и "ZX" применяются одинаковые коэффициенты деления.

По числу звуковых каналов MC "Dendy" имеет превосходство. В нем, кроме трех обычных тональных каналов (А, В — прямоугольная форма, Т — треугольная форма), присутствуют шумовой (N) и "цифровой" (PCM) каналы. В противовес этому, выходы AY-3-8910/12 (А, В, С) могут быть сконфигурированы на работу в режиме "стерео", в отличие от "Dendy".

Правда, не все знают, что в прародительнице "Dendy", приставке "Famicom", была предусмотрена функция стереозвучания. Оригинальный чип RP2A03G имел два аудио-выхода, в одном из которых смешивались каналы А, В, а в другом — Т, N, PCM. В дальнейшем появились однокристальные модели NES с 82-выводным чипом "NES-on-chip", в котором два стереоканала были объединены в один моноканал. Именно так сейчас устроены все "Dendy"-совместимые приставки.

Количество регистров управления у MC "Dendy" (табл.2) на четыре больше, чем у "ZX" (табл.3), что связано с увеличенным числом звуковых каналов. Оба MC относятся к классу сопроцессоров, то есть они работают автономно от центрального процессора, периодически общаясь с последним при помощи трех шин: адреса, данных, управления.

Диапазон выходных частот и разрядность делителя в канале тональной частоты MC "ZX" больше, чем у "Dendy", однако на практике это преимущество

не используется. Для объяснения причины вспомним, что звуковой ряд фортепиано составляет семь полных и две неполные октавы. Итого 90 нот, имеющих частоты в диапазоне 27,5...4698,6 Гц.

Более важным видится различие в шаге изменения частоты звуков. Вследствие дискретного способа формирования, он будет нелинейным (на низких частотах — мень-



ше, на высоких — больше). Абсолютная величина шага (Δ) рассчитывается по формуле

$$\Delta = F_m * (1/K_1 - 1/K_2) / K_{\text{общ}},$$

где:

- $F_m = 1,7734$ МГц — тактовая частота МС;

- K_1, K_2 — коэффициенты деления для двух подряд идущих нот;

- $K_{\text{общ}}$ — коэффициент счета общего делителя всех частот в МС.

В AY-3-8910/12 значение $K_{\text{общ}} = 16$ для всех каналов — А, В, С. Для "Dendy" в каналах А, В значение $K_{\text{общ}} = 16$, а для канала Т значение $K_{\text{общ}} = 32$. Следовательно, точность воспроизведения высоты нот в каналах А, В, С "ZX" и А, В "Dendy" будет одинакова, а в канале Т — в 2 раза ниже. Можно предположить, что разработчики "Dendy" сознательно приняли такое решение. Канал Т в музыкальных композициях используется исключительно для басовых (низкочастотных) партий, где точность расчета частоты нот достаточно высокая.

Генератор огибающей амплитуды в AY-3-8910/12 наделен более широкими возможностями по всем показателям. В частности, разнообразие форм огибающей позволяет создавать на "ZX" оригинальные спецэффекты и красивые музыкальные пассажи. В "Dendy" круг доступных форм ограничивается ниспадающей пилой (параллелепипедом) и прямоугольником.

Генераторы тональных сигналов в обоих МС используют четырехразрядный ЦАП, хотя для "ZX" существует совместимый МС типа YM2149F ф. Yamaha, имеющий пятиразрядный ЦАП [3]. Шкала уровней громкости в МС "Dendy" — линейная, а в "ZX" — логарифмическая. Насколько это принципиально?

Исследования, приведенные в [3], показывают, что наклон амплитудной характеристики влияет на степень "мягкости" звучания композиций. Если пользоваться музыкальными терминами, то линейная зависимость эквивалентна исполнению мелодии "легато" (плавный переход одного звука в другой), а логарифмическая — "стаккато" (краткое, отрывистое звучание). В первом случае на-

блюдается "мягкое", продолжительное затухание, во втором — "сухое" и быстрое.

Строго говоря, чувствительность слуха человека к громкости звуков носит нелинейный характер, близкий к логарифмическому. Недаром в музыке используют шкалу, состоящую из 7 ступеней громкости от пиано-пианиссимо до форте-фортиссимо с разницей отдачи между ступенями примерно в 10 дБ.

Однако в МС преимущества логарифмической зависимости проявляются только при действительно многоканальном звучании (10 и более каналов) с имитацией динамики реальных музыкальных инструментов. В нашем же случае, в силу объективных причин (малого числа каналов), композиции носят искусственный характер с явно "синтетическими" красками. В таких мелодиях недостатки линейной амплитудной зависимости часто превращаются в ее достоинства...

Генератор шумовых сигналов в МС "Dendy" наделен более широкими возможностями. Он не только выделен в отдельный канал, но и имеет запас по диапазону частот, а также переменную длину псевдослучайной последовательности (ПСП). Подобная гибкость заимствована от игровых автоматов, где шумовые эффекты, например, выстрелы, падения, несут важную смысловую нагрузку. В музыкальных композициях каналу шума часто отводится роль ударных инструментов.

Аналогичные доводы можно привести для **сви́п-генератора** ("глицсандо") МС "Dendy", который не имеет полного эквивалента в "ZX". Эффект "глицсандо" заключается в плавном переборе звуков с постепенным повышением или понижением частоты. Для справки: слово "глицсандо" происходит от французского "glisser" — скользить. Этим эффектом часто пользуются, чтобы симитировать звуки гавайской гитары или сирены.

PCM-канал, он же "цифровой" канал, он же канал с импульсно-кодовой модуляцией (Pulse-Code Modulation). В МС "Dendy" этот канал поддерживается на внутреннем

аппаратном уровне, что упрощает процедуру вывода данных при программировании. Разрядность кодовых слов составляет 7 бит (по некоторым источникам — 6 бит). Каждое слово определяет разность между текущим и последующим значением амплитуды звукового сигнала. В теории информации такой метод называется дельта-импульсным. Соответственно, некоторые авторы в Интернете употребляют названия канала DPCM (Delta PCM) или DMC (Delta Modulation Channel).

Оцифрованные сэмплы музыки или человеческая речь, формируемые в канале PCM, звучат вполне естественно, правда, требуют чрезвычайно большого расхода памяти. Для "Dendy" этот момент важен, поскольку размер ПЗУ картриджа не безграничен. Соответственно, лишь в немногих играх канал PCM используется в полную силу.

В "ZX" ситуация с объемом памяти значительно лучше, благодаря 128 Кб ОЗУ и наличию дискового накопителя. Здесь также используют "цифровой" канал звучания, но он "внешний", без прямой аппаратной поддержки. Для проигрывания "цифровой" музыки AY-3-8910/12 переводят в режим ЦАП, при этом уровни сигнала на выходе сопроцессора устанавливают программно в каждый конкретный момент времени.

На примере канала PCM можно уяснить кардинальную разницу в приемах программирования музыки "Dendy" и "ZX". Для игровой приставки на аппаратном уровне сделано все, чтобы максимально сократить время общения центрального процессора и МС. Во-первых, все регистры управления звуком в "Dendy" отображаются на карту памяти в виде обычных ячеек, следовательно, доступ к ним, в отличие от "ZX", происходит напрямую. Во-вторых, большое количество регистров позволяет двумя-тремя пересылками данных сформировать в "Dendy" такой звук, для извлечения которого в "ZX" требуется целая подпрограмма. В-третьих, длина программного кода музыкальных композиций и сэмплов в "Dendy" получается меньше по объему.



С другой стороны, для "ZX" разработаны мощные музыкальные редакторы, которые позволяют автоматизировать и облегчить труд программиста. И в этом большая заслуга представителей стран СНГ. Как следствие, за последние 15 лет создано много красивых и оригинальных композиций руками "самодеятельных" музыкантов. Разработка музыкальных ZX-интро продолжается и по сей день.

Для "Dendy" редакторов класса "Pro Tracker 3" просто не существует. Например, японский "MKS" не прижился из-за его восточной

специфики подачи информации, американский "NerdTracker-2" имеет ограниченные возможности и нестандартный формат звуковых файлов. Основное направление работы хакеров — поиск музыкальных фрагментов в существующих NES-играх и конвертация их в один из музыкальных форматов. Наиболее интересные образцы Dendy-музыки в формате MIDI можно найти на сайте <http://www.vgmusic.com>.

Подведем итог — музыкальные возможности MC "Dendy" и "ZX" примерно одинаковы. Их недостатки и достоинства взаимно компен-

сируют друг друга, не позволяя отдать кому-либо пальму первенства.

Литература

1. AY-3-8910/8912 programmable sound generator. Data manual. — General Instrument, копия от 30.10.1999, <http://users.rcn.com/carlott/ay3-8910.pdf> (720 Kб).

2. AY-3-8910, AY3-8912, AY-3-8913. Datasheet. — General Instrument, копия от 03.02.1998, http://bulba.at.kz/AY-3-8910_2.rar (526 Kб).

3. Рюмик С. AY3-8910 или YM2149F? — Радиоаматор, 1996, №8, С.26.

Возвращаясь к напечатанному ("РМ. ВК", №5/01, С.35)

Еще об эмуляторах

Если помните, основной идеей моей ранее опубликованной статьи "Эмуляция "ZX-Spectrum": графика" было то, что в эмуляторе изображение можно сделать даже лучше, чем на реальном "ZX-Spectrum". И вот недавно я узнал еще об одном приеме улучшения изображения, реализованном в эмуляторе UnrealSpessy.

На реальном "ZX-Spectrum"

разрешение экрана — 256*192 пиксела. Соответственно, чтобы можно было напечатать 64 символа в строке при 24 строках на экране, размер символа должен составлять 4*8 пикселей. При этом некоторые символы отличаются друг от друга всего одним-двумя пикселями, так что читать текст не очень то удобно. Так вот, эмулятор UnrealSpessy умеет распознавать символы 4*8 (сравнивая изображе-

ние на эмулируемом спектральном экране с хранящимися в памяти образцами символов) и выводить вместо них гораздо более качественные изображения символов — 8*8 или 8*16, за счет того что разрешение графического режима, используемого при эмуляции, больше 256*192. Действительно, очень красивая идея!

И.РОЩИН,
г.Москва.

КОМПЬЮТЕРНЫЕ ХРОНИКИ

Paradox 2002

7-8 сентября в Ростове-на-Дону прошло мультиплатформенное (AMiGA, PC, "ZX-Spectrum") demoparty — Paradox 2002 (<http://scene.org.ru/paradox>). Для "ZX-Spectrum" проводились конкурсы в номинациях Graphics, AY music, Digital music (for covox/soundrive) и Demo. Работ было представлено не очень много — видимо, сказало то, что demoparty прошло всего через две недели после CAFE'2002. Ниже приведена информация о победителях и при-

зерах (в скобках — количество набранных баллов).

Graphics (всего 6 работ): 1 — "Clean" by Gas13 (7,5278), 2 — "YTKU" by Disabler/DPL (7,1154), 3 — "CLOUDS" by Yorzmyey/HOOY-PROGRAM (6,1759).

AY music (всего 13 работ): 1 — "Live from my oldschool party" by Phantom Lord/CMS+PSYHO (6,0635), 2 — "Lovely Soul" by Cj Flexo/GREEN BIT GROUP (5,9258), 3 — "Linda: "Northern Wind" by Silent/DELTA (5,5848).

Digital music (всего 6 работ): 1 — "Melomania" by Vigil/CODERS' ACADEMY (7,4731), 2 — "Element Moderate" by Nikphe/ANARCHIA (5,8821), 3 — "Old remeber remix" by Disabler/DPL (5,7400).

Demo (всего 1 работа) — "INvAsiOn" by O.C.A.

Как обычно, подробную информацию о результатах demoparty, а также сами представленные работы вы сможете найти на крупных спектральных сайтах.



Его величество Пасьянс

А.ГОРЯЧКИН,

г.Кыштым, Челябинской области.

E-mail: anatoliy_goryach@mail.ru

Пасьянсы зародились в XVII веке во Франции. Слово "пасьянс" происходит от слова "patience" — что в переводе с французского означает "терпение". К пасьянсам часто обращались узники, находя в них утешение во время долгого вынужденного одиночества. Сегодня пасьянсы — это не просто увлекательные игры, но и приятная возможность скоротать время и снять усталость после напряженной работы. Пасьянсов существует великое множество, от простейших до головоломных. Они бывают доминошные, карточные, есть и детские пасьянсы. Все они развивают наблюдательность и сообразительность. Интересны в них и процесс раскладывания карт, и конечный результат. В основе некоторых пасьянсов лежат весьма любопытные математические закономерности, некоторые из которых описаны в работах таких известных математиков как Д.Кнут и М.Гарднер.

Логично было бы предположить, что с появлением компьютеров появятся и компьютерные программы, позволяющие играть в пасьянсы. Недаром корпорация Microsoft включила в дистрибутивы операционных систем Windows 9x два карточных пасьянса — "Косынка" и "Солитер". Кстати, слово "солитер" тоже пришло к нам из французского языка и означает "solitaire" — одиночка (тот, кто любит быть один). Все англоязычные пасьянсы так и называются — Solitaire. Существуют компьютерные игры, которые включают в себя целые наборы пасьянсов, например, "Solitaire absolute" (www.wizworks.com) или "Solitaire royale". Из богатого набора игр каждый игрок обязательно для себя что-нибудь выберет. Но без знания правил игры невозможно в нее играть. В этой статье приведены правила игры для 12 наиболее распространенных пасьянсов. Для начала

нужно знать, что во всех карточных играх, в том числе и пасьянсах, туз обозначается литерой A (Ace), король — K (King), дама — Q (Queen), а валет — J (Jack).

Пасьянс "Пирамида"

Варианты названий этой популярной игры: "Могила Тутанхамона", англоязычное — "Pyramid". Свое название пасьянс получил из-за расклада карт в форме пирамиды. Пасьянс раскладывается 52 картами. 28 карт выкладываются в форме пирамиды одна на другую. Остальные 24 помещаются в колоду. Задача — разрушить пирамиду, т.е. удалить из пирамиды все карты. Карты удаляются с игрового поля парами, сумма очков которых составляет 13. Карты удаляются независимо от их масти. Так как "цена" короля равна 13, то он удаляется сам по себе, без пары. Дама имеет 12 очков и удаляется только с тузом, который "весит" одно очко. Валет удаляется с двойкой, десятка — с тройкой, девятка — с четверкой, восьмерка — с пятеркой, семерка — с шестеркой. Если с игрового поля больше нельзя удалить ни одной пары — игра заканчивается. Пасьянс сойдется в том случае, если с игрового поля удалены все карты.

Пасьянс "Гольф"

Англоязычное название этого пасьянса — "Golf". Цель пасьянса состоит в том, чтобы переместить все карты к основанию. Карты перемещаются из расклада в убывающем или в возрастающем порядке независимо от масти. Из расклада могут быть перемещены в основание только свободные карты. Игра начинается после открытия верхней карты в прикупе, которая перемещается в основание. Например, если верхняя карта в основании — четверка, то из расклада на нее может лечь пятерка или тройка. Игра окончена, когда в прикупе больше нет карт, и ни одна карта не может быть

перемещена из расклада. Чтобы пасьянс сошелся, необходимо переместить все карты из расклада.

Пасьянс "Косынка"

Англоязычное название этого пасьянса — "Klondike". Цель игры состоит в том, чтобы сложить четыре стопки карт одной масти от туза до короля. Игровое поле состоит из колоды, семи столбцов карт и дома. Чтобы освободить карты, которые следует переместить в дом, игрок перекладывает их из одного столбца в другой. Существуют упрощенный и сложный варианты этой игры. В упрощенном варианте игры из колоды сдается по одной карте, в сложном — по три. После того как все возможные ходы будут сделаны, открывают одну или три следующие карты из колоды. Игрок имеет доступ к карте, расположенной сверху колоды лицом вверх.

Пасьянс "Кэнфилд"

Англоязычное название этого пасьянса — "Canfield". Цель игры состоит в том, чтобы сложить четыре стопки карт одной масти от туза до короля. Игровое поле состоит из двух колод, четырех столбцов карт и дома. Карты в столбцах укладываются в порядке убывания с чередованием цвета масти. Если один столбец может быть перемещен на другой с соблюдением этого правила, то на освободившееся свободное место может быть перенесена верхняя карта из нижней колоды, вне зависимости от ее "веса" и масти. Как из верхней, так и из нижней колоды карты можно перемещать в дом или в один из столбцов. Из нижней колоды карты сдаются по одной. До тех пор, пока этой карте не будет найдено место в одном из столбцов или в доме, нижняя колода будет недоступна. Из верхней колоды карты сдаются по три. Игрок имеет доступ к карте, расположенной на вершине колоды лицом вверх. После того как из нижней колоды будет снята последняя карта, на свободное место в столбцах может быть перемещен король из верхней колоды.





Пасьянс "Уголки"

Англоязычное название этого пасьянса — "Corners". Цель игры состоит в том, чтобы переместить все карты в дом. Карты раскладываются на игровом поле в виде "креста" из пяти карт или квадрата с отсутствующими "углами". Также, первая карта основания укладывается в верхний левый "угол" квадрата.



Кипы основания — четыре угла ящика. Каждая начальная карта кипы основания в остальных углах должна иметь тот же "вес", что и у исходной карты основания (в левом верхнем углу). В завершённом виде каждая кипа основания будет содержать все 13 карт одной масти. В процессе игры верхнюю карту из колоды игрок может переместить либо в одну из четырех угловых кип (согласно масти по возрастанию) либо в одну из пяти кип "креста" (без учета масти по убыванию или на пустое место). Или, если ничего другого не остается, или по стратегическим причинам, то — в сброс. Верхняя карта сброса тоже является игровой и может быть использована аналогично верхней карте колоды. Кроме верхних карт сброса и колоды, игровыми являются также верхние карты кип "креста". Их можно перемещать либо в одну из четырех угловых кип, либо на одну из кип "креста". Карты угловых кип переместить обратно на одну из кип "креста" не получится — они не являются игровыми. Игра заканчивается, когда кончается колода, и нет возможности передвинуть ни одну из карт "креста" или карту сброса. Пасьянс сошелся, если удалось разложить все карты по угловым кипам.

Пасьянс "Ряды"

Англоязычное название этого пасьянса — "Calculation". Цель игры состоит в том, чтобы переместить все карты в дом. Расклад представляет собой четыре пустых столбца и четыре кипы дома. Кипы основания должны быть построены по определенному порядку независимо от масти:

- в первой кипе (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K);
- во второй кипе (2, 4, 6, 8, 10, Q, A, 3, 5, 7, 9, J, K);
- в третьей кипе (3, 6, 9, Q, 2, 5, 8, J, A, 4, 7, 10, K);
- в четвертой кипе (4, 8, Q, 3, 7, J, 2, 6, 10, A, 5, 9, K).

Карта из колоды во время хода может быть перемещена либо в одну из кип дома, либо на любой столбец в игровом поле. Кроме того, игрок может переместить верхнюю карту одного из столбцов в одну из кип дома. Игра заканчивается, когда кончается колода, и нет возможности переместить одну из верхних карт столбцов в дом. Чтобы выиграть, нужно переместить все карты из колоды и игрового поля в дом, выстраивая каждую кипу согласно вышеуказанным карточным рядам.

Пасьянс "Три перетасовки"

Англоязычные варианты названий этого пасьянса — "Three Shuffles and a Draw", "La Belle Lucie", "Midnight Oil". Цель игры состоит в том, чтобы переместить все карты в дом. Первоначальный расклад представляет собой 16 кип по 3 карты в каждой. Все карты лежат картинкой вверх. Тузы уже перемещены в дом. Карты перемещаются из одной кипы в другую в порядке убывания и согласно масти. Чтобы достичь цели, нужно "освободить" карты, которые можно переместить в дом (в возрастающем порядке и согласно масти). Доступными являются верхние карты каждой кипы. Доступную карту можно переместить в дом или на другую кипу. Есть возможность "перетасовывать" карты расклада дважды, после того как исходный расклад исчерпал себя. Перемещенные в дом карты остаются там после перетасовки. Кроме того, игрок имеет право один раз за игру "взять draw", то есть извлечь одну недоступную карту и переместить ее в одну из кип или в дом. Игра заканчивается, когда все "перетасовки" использованы, "draw взят" и нет возможности переместить ни одну карту. Выигрыш наступает, когда все карты перемещены в дом.



Пасьянс "Рено"

Англоязычное название этого пасьянса — "Reno". Игровое поле состоит из двух колод, четырех столбцов карт и дома. Карты в столбцах укладываются в порядке убывания с чередованием цвета масти. Если один столбец может быть перемещен на другой с соблюдением этого правила, то на освободившееся свободное

место может быть перенесена верхняя карта из нижней колоды, вне зависимости от ее "веса" и масти. Как из верхней, так и из нижней колоды карты можно перемещать в дом или в один из столбцов. Из нижней колоды карты сдаются по одной. До тех пор, пока этой карте не будет найдено место в одном из столбцов или в доме, нижняя колода будет недоступна. Из верхней колоды карты сдаются тоже по одной. Игрок имеет доступ к карте, расположенной сверху колоды картинкой вверх. Доступную карту из верхней колоды можно переложить либо в дом, либо в один из столбцов или (если ничего другого не остается) в сброс. После того как из нижней колоды будет взята последняя карта, на свободное место в столбцах может быть перемещен король из верхней колоды или сброса.

Чтобы переложить одну или несколько карт из одного столбца в другой, укажите на нее и нажмите левую кнопку мыши, а затем укажите туда, куда ее следует переместить, и снова нажмите левую кнопку мыши. Цель игры состоит в том, чтобы сложить четыре стопки карт одной масти от туза до короля.

Пасьянс "Турецкий платок"

Колоду карт в 52 листа раскладывают картинкой вверх на вертикальные девять столбцов по шесть карт. Карты удаляются с игрового поля парами. Цель игры — очистить игровое поле от карт, снимая за один ход по две карты одного достоинства, вне зависимости от ее масти. Игровыми картами являются только нижние, незакрытые карты вертикальных столбцов.

Пасьянс "Свободная ячейка"

Англоязычное название этого пасьянса — "Free Cell". Игровое поле состоит из четырех ячеек дома, четырех свободных ячеек и восьми столбцов карт, расположенных картинкой вверх. Карты перемещают по следующим правилам. В свободную ячейку можно переместить любую карту, находящуюся внизу столбца. В ячейку дома можно переместить любую карту, находящуюся в свободной ячейке или внизу столбца. Карту можно положить на карту той же масти в следующем порядке: двойку на туза, тройку на двойку и т.п. Открытые тузы



автоматически перемещаются в пустые ячейки дома. Карту, находящуюся в свободной ячейке или внизу столбца, можно переместить в низ другого столбца. Карту можно положить на карту другого цвета в следующем порядке: даму на короля, валета на даму и т.п. Цель игры состоит в том, чтобы переместить все карты в дом, используя пустые ячейки как промежуточные. Чтобы выиграть, следует расположить карты по возрастанию в четырех ячейках дома, соответствующих четырем мастям: двойку на туза, тройку на двойку и т.п. Считается (хотя и не доказано), что данный пасьянс сходится при любом раскладе.

Пасьянс “Паганини”

Англоязычные варианты названий этого пасьянса — “Gaps”, “Four Ways”. Колоду карт в 52 листа раскладывают картинкой вверх в 4 горизонтальных ряда по 13 карт в каждом. Все тузы один за другим изымают из расклада и кладут слева от каждого ряда. Цель пасьянса — упорядочить раскладку карт, расположив их в каждом из горизонтальных рядов в масть

за соответствующими тузами в восходящем порядке (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K), используя для перекладки свободные места (окна). В окно можно положить карту, следующую по старшинству за картой, лежащей слева от окна. Например, если слева от окна лежит “9 пик”, то в это окно можно положить лишь “10 пик”. Если окно окажется справа от короля, то туда уже ничего не положишь, окно станет “глухим”. Если пасьянс зашел в тупик (все окна за королями), то есть возможность еще раз перемешать карты, которые к этому моменту еще не упорядочены. Потом можно вновь их разложить по рядам, дополняя число карт в каждом до 13, и оставляя свободным окно после последней из подобранных в масть карт в каждом ряду. Далее, используя свободные окна, завершают перекладку.

Пасьянс “Сэр Томми”

Англоязычное название этого пасьянса — “Sir Tommy”. Это один из старейших пасьянсов среди известных. Он раскладывается одной полной колодой (52 листа). Колоду держат зак-

рытой, поочередно перебирая карты. Когда появляется туз, его откладывают в качестве базовой карты. В некоторых вариантах игр все тузы автоматически перемещаются на свое место. В дальнейшем на тузы собирают масти в восходящей линии до королей. Карты из колоды, не подходящие для размещения на базовых картах, укладывают на один из четырех открытых талонов по выбору. Перекалывать карты из одного талона в другой нельзя. Можно переносить верхние свободные карты с четырех талонов на базовые карты. Если колода исчерпана, а карты из четырех вспомогательных талонов не удалось разместить на базовых картах, то пасьянс сошелся. Не следует класть карту из колоды на вспомогательный талон, в котором уже есть младшая карта данной масти, например даму пик — на талон, в котором уже есть семерка пик, т.к. в этом случае пасьянс гарантированно не сойдется. Напротив, следует стремиться в пределах каждого из талонов собирать масти в нисходящей последовательности. Несмотря на простоту правил, этот пасьянс довольно трудно разложить.

Never Winter Night

А.ВЕНДИЛОВСКИЙ,

г.Минск.

Холодный ветер рвал плащ и старался забраться под капюшон. Ночная мгла скрадывала его фигуру, и равномерный звук шагов затихал среди пустующих домов. Сильная рука сжимала большой витиеватый посох, украшенный рунами и другими странными знаками, которые временами вспыхивали странным бледным огнем, выдавая в его владельце друида. Он шел к большим воротам внутри города...

...История города началась давным-давно, когда эти места были еще дикими, и природа всю хозяйничала на этих просторах. Тогда орды дикарей нападали на путешественников и купцов, что шли из далеких земель к Побережью Мечей и Вратам Балдурса. Путь их был очень опасен. Но в один день группа искателей приключений возвела на перекрестке торговых путей небольшой форт, который стал быстро разрастаться. С каждым новым человеком, приходившим туда, росла его сила. И дикие племена стали получать все больший отпор, а форт быстро раз-

росся в город, который стал оплотом для купцов и путешественников в этой глуши. Богатство и сила царили в нем, крепостные стены были мощны, и уже никто из дикарей не решался попробовать оспорить власть города и крепость стен. Годы летели над крышами города, но сила его не уменьшалась. Однако даже высокие стены и дозорные башни, мечи и магия оказались бессильны перед новым захватчиком.

Откуда она пришла темной беззвездной ночью, не знает никто. И никто уже не сможет сказать, кто стал ее первой жертвой. Она не боялась ничего, с легкостью сражая даже великого воина. Чума... Как лесной пожар, она понеслась по городу, все убивая на своем пути. И взмолились люди своим богам, но священники лишь горестно разводили руками, все их знания и молитвы, настойки и заклинания были бесполезны — вылечить больного они не могли, и Чума получала еще одну жертву. И не только тело — Чума пожирала душу! Днем и ночью, правители и

священники, волшебники и целители искали противоядие, но безуспешно. А город становился все более и более безлюдным, только горы мертвых тел, которые не успевали сжигать на кострах, и запах разлагающейся плоти. Правители перекрыли районы города и выход из него, опасаясь распространения инфекции, но это привело лишь к бунтам, которые были жестоко подавлены. Каждый новый день становился хуже предыдущего. В отчаянии, властители бросили клич о помощи, и многие приключенцы откликнулись на него — кто ради славы, кто по убеждениям, а кто просто ради хороших денег. Они тренировались во дворце и многие уходили на поиски, но мало кто смог вернуться. Но вот стали поступать обнадеживающие вести — удалось найти противоядие! Компоненты его были очень редки, и чтобы их добыть, многие искатели расстались с жизнью. И вот, когда до создания панацеи остался лишь один последний шаг, на замок был совершен набег. Четко рассчитанный удар еще раз доказал, что все об-



рушившиеся в последнее время на город беды насланы реальным врагом. Компоненты пропали, лучшие из бойцов вновь отправились на их поиски.

Мало кто обратил внимание на человека, проходящего через одни из ворот замка, он медленно брел, внимательно осматривая местность перед собой, и каждый его шаг выдавал в нем опытного бойца. Он шел к большим воротам внутри города...

... Огромные, в три человеческих роста, врата глухо загудели от ударов посоха по ним.

— Кого тут бесы носят, Баатор его побери! — на другой стороне появился усталый стражник, в помятой кирасе и с мечом в руках. — Ночь на дворе, никого не велено пропускать! И во... — стражник замолк на полуслове, когда незнакомец вытащил медальон и пропуск. Сплюнув и помянув Баала и прочих милых личностей, стражник отпер врата.

— Как тут обстановка? — перед глазами друида лежала площадь, забаррикадированная сломанными телегами, бревнами и прочим мусором, за которыми скрывались стражники, время от времени стреляя в ночную тьму улиц из арбалетов.

— Просто курорт! — солдат зло усмехнулся, — эти мертвые твари так и прут, из всех углов! Мы едва удерживаем позиции. Зомби не так-то просто убить, а уж мертвых тел тут предостаточно. Мало главного кладбища, так еще и во рвах лежат те, кого съечь не успели. В других районах свои беды, как я слышал, или мертвяки повсюду?

— Нет, мертвяки только здесь? — друид задумался — но там тоже не сладко, начальник тюрьмы сошел с ума и выпустил всех заключенных, да и в других местах не менее весело. Кстати, здесь ведь наверняка есть подземелья или подземные склады, ты можешь показать мне, где конкретно?

— Я? — стражник посмотрел на друида как на законченного сумасшедшего, — сам я туда не сунусь и с тобой никого не пошлю, это верная смерть, а у меня и так людей мало. Если ты со...

— Один он не пойдет. — Легкая тень вышла из сумерек, превращаясь в женскую фигурку с весьма соблазнительными формами. Черный плащ и тихая походка выдавали в девушке профессионального шпиона. Легкая улыбка скользнула по лицу друида.

— Адриэль, а кто сказал тебе, что я пойду один? — рядом с друидом стоял огромный волк, действительно огром-

ный волк, и стражник готов был поклясться, что минуту назад его там не было. Ту, которую друид назвал Адриэль, это явно не удивило.

— Твой волчара конечно неплох, но сможет ли он унюхать ловушки или открыть замок? Я же предупредила тебя, Калгор, один ты не пойдешь, и кстати, я прекрасно знаю данную местность, в отличие от тебя. Воспользовавшись моментом, Адриэль достала зеркальце и стала поправлять невидимый изъяз в прическе. Калгор лишь тихо простонал, вполголоса выругался, помянув некоторых уже забытых богов, и посмотрел на волка, который был явно озадачен таким поворотом событий, но беспрекословно ждал приказаний своего хозяина.

— Ладно, — было видно, что друид сдался, и пошел в сторону главного кладбища.

— Шпионская шлюха! — стражник зло сплюнул. Как все стражники, он на дух не переносил ни гильдию воров, ни их более профессиональных коллег, служащих у правителя. Эти слова не прошли мимо тонкого слуха шпионки, и в темноту с силой полетел маленький стальной шарик. Крик боли показал, что шарик не пролетел мимо цели.

— Вы там долго еще будете любезностями обмениваться? Нам еще предстоит сделать одну работу! — Калгор был раздражен и пропустил момент, когда громадный зомби выскочил из-за угла и с невнятным мычанием бросился на него. Зато волк был на высоте — мгновенный бросок, и голова покойника покатила по мостовой. С гримасой безразличия, волчара слез с трупа и расчихался.

— Вход на склады здесь! — Адриэль повозилась с минуту с замком, оглянувшись по сторонам и сильным ударом ноги вышибла дверь. В ответ на недоуменный взгляд друида она только пожала плечами и нырнула внутрь.

— Какой тут затхлый воздух! — в этот момент в темноте раздались стоны и всхлипывания, которые стали приближаться. Посох друида озарился ярким светом, и они увидели, как со всех сторон огромного складского помещения, заваленного тюками с тканью и прочими подобными товарами, к ним медленно приближаются шесть зомби. Покрытые плесенью и объединенные червями, теряя по ходу куски разлагающегося мяса, они представляли собой ужасное зрелище. В тот же миг с громopodobным рычанием с места сорвался волк, помчавшись к ближай-

шему монстру. Глаза друида вспыхнули колдовским огнем, его руки стали производить пассы в воздухе, а сам воздух засветился от магии. Во вспышке света, под речитатив странных и загадочных слов перед друидом появился огромный барсук.

— Ты бы еще скунса вызвал, — фыркнула Адриэль, швыряя очередной кинжал в толпу. Двое зомби уже лежали на полу без движения, но в этот момент двери в глубине склада распахнулись, и к монстрам подоспела подмога — не менее десятка этих доходяг с ножами и мечами в руках пытались спуститься по лестнице, заходя в тыл к героям.

— Опс! — с рук Калгора сорвались яркие световые шары, отбрасывающие очередного мертвяка. Времени читать заклинание на следующего зомби просто не оставалось, и он изо всех сил огрел того своим посохом. Выхватив свободной рукой меч, друид стал отбиваться от ходячей мертвечины сталью. В это время Адриэль пыталась открыть люк на нижний уровень, но замок был сложной конструкции, и она уже сломала третью отмычку. Поворчав о дилетантах и нелегких условиях трудовых будней, она опять стала аккуратно ковыряться в замочной скважине.

Волк обеспечил друиду секундную передышку, которой тот мгновенно воспользовался. Прокричав несколько слов на древнем наречии, он изо всех сил ударил посохом по земле, и оттуда стали вырастать гибкие как плети корни лозы, оплетающие и хватающие все на своем пути. Несколько зомби сразу оказались опутанными, остальные сильно замедлили свой ход.

— У нас мало времени! — прорычал друид, выдавая тираду на неизвестном для шпионки языке, в которой она смогла понять лишь то, что идет упоминание о чьей-то матери. В этот момент замок сдался. Выхватив из-за пояса свиток, Калгор прочитал с него несколько фраз вслух и изо всех сил швырнул в толпу врагов огненный шарик. Вся троица кубарем скатилась по ступенькам, прежде чем волна огня прокатилась по помещению.

— Просто замечательно! — Адриэль мгновенно приняла боевую стойку, но на этом уровне мертвечиной даже не пахло, а пахло... как-то очень странно. Неприятный запах витал в воздухе, и вокруг было очень много паутины. Возле высохшего трупа лежал клочок бумаги, который друид подобрал и стал внимательно рассматривать.



— Покойнички этот этаж не любят или... — друид напрягся, услышав во тьме странный шорох, от которого шерсть на загривке у волка встала дыбом.

— Не пойми меня превратно, но у меня нездоровое предчувствие, что пора срочно делать ноги! — и словно в подтверждение его слов из темноты выскочило несколько больших пауков.

— Портал! — закричал друид, протыкая ножом одну из тварей, но от удара другой не смог увернуться. Волк бросился защищать хозяина, который с гримасой боли пытался выхватить из кармана своего одеяния бутылочку с эликсиром здоровья. Когда это мерзкое на вкус пошло потекло в горло, он почувствовал, как раны сами собой заживают, и силы вновь возвращаются к нему. С трудом встав на ноги, Калгор повернулся к Адриэль, которая уже практически активировала Камень Возвращения, когда в проходе появился гигантский паук. По высоте он был в рост человека, его фасеточные глаза отражали тусклый свет факелов. Не медля ни секунды, друид выхватил бутылочку с эликсиром жизни и бутылочку с эликсиром скорости, вылил по половине содержимого каждой на пол, а потом смешал остатки. Старым платком он закрыл горловину бутылки и пальцами высек магическую искру, которая повисла в воздухе. Когда вихрь магии за его спиной коснулся его сущности, он позволил искре упасть на импровизированный фитиль и бросил бутылку прямо в паука — взрыв фаерболла показался бы просто детской хлопушкой, по сравнению с грянувшим взрывом. Взрывная волна швырнула портал героев, и они растянулись на мраморных плитах Главного Храма.

— Дзириг Д'Уоррен был бы мной доволен, — пробурчал Калгор, отряхивая со своей одежды огненные искры. Достав из кармана найденный клочок бумаги, он еще раз перечитал написанное на нем и тяжело вздохнул — боюсь, мне придется огорчить наших правителей, врагов у нас намного больше, чем хотелось, и далеко не там, где мы искали. Но ведь это уже по твоей части, Адриэль, разбираться с предателями? — он передал ей клочок письма и побрел к волшебному колодцу, чтобы избавиться от ненужных уже ему вещей и прикупить недостающую амуницию...

Итак, господа геймеры, мы все дождались большого и полновесного хита в жанре RPG. Возрадуйтесь, любите-

ли "Ворот Балдурса" и "Планескейпа" — фирма "BioWare" выпустила очередной шедевр, которым мы будем наслаждаться долгие часы.

Что роднит эту игру с предыдущими, так это огромный и полновесный сюжет, просто наполненный поворотами и интригами, держащий в напряжении до последнего момента. Никакой однозначности, никакой уверенности в том, что в данный момент тебе известна истина. Твои друзья могут тебя предать, а твои враги станут на твою защиту. А если к этому добавить множество добавочных квестов со своими интригами и нюансами, то мы получаем огромный игровой мир, который живет словно и без вас, наполняя все реализмом. У вас имеется определенная свобода в достижении своих целей, как то: подкупить, ограбить, убить, обмануть, уговорить или, как говорят любители подобных игр — "отыграть олдджмент". Хотите быть злым — пожалуйста, добрым — никто не против, благородство — замечательно. Но никто не помешает вам попросить и деньжат за хорошо выполненный квест (ведь каждый наемник должен хорошо оплачиваться). В зависимости от вашего поведения будет меняться и отношение к вам народа, соответственно, и диалоги. Сам этот окружающий мир уже проработан до мелочей за годы существования настольной игры "Подземелья и Драконы" — все время вам будут попадаться ссылки на события, о которых вы могли читать в книгах или даже участвовать в них в других играх. Не проходите мимо всевозможнейших книг и библиотек, читайте свитки, и перед вами откроется удивительный и очень красивый мир, в котором все очень логично и взаимосвязано, который настолько РЕАЛЕН, что вы начнете верить в него с первой секунды.

Говоря о правилах, я должен заметить, что сама игра построена на третьей редакции "ADD". Это несколько меняет устоявшиеся со времен "Ворот Балдурса" представления об RPG. Изменились расы и распределение хитов, но появились и очень интересные особенности.

Во-первых, никто не будет мешать одеть магу или друиду кольчугу либо доспех, если у него хватит сил их носить, либо взять в руки хороший меч (ведь если рассуждать логически, для этого нет никаких физических ограничений!). Естественно, файтера из него толком не получится, и он будет иметь штрафы при использовании вооруже-

ния, но ведь это стало ВОЗМОЖНЫМ! Или можно научить своего мага, например, искусству воровства, главное, чтобы руки были ловкие :). Такой персонаж в своем мультиклассе, конечно, будет ограничен, но новые возможности, открывающиеся при этом, легко перекрывают все недостатки. А это надо учитывать, при том, что на этот раз мы не ходим толпой в шесть человек, как раньше, когда можно было набрать узкоспециализированных персонажей на все случаи жизни. Теперь с вами может быть только один персонаж-помощник, чьими действиями вы толком-то и управлять не сможете (кстати, "компьютерный друг" весьма неплох, и не делает откровенных ляпов, портящих всю игру). Так что подбор персонажа и его специализация — дело довольно тонкое, от которого будет зависеть весь стиль прохождения игры.

Еще одно любопытное нововведение связано с заклинаниями. В процессе сна для каждого из них есть время на восстановление — простые возвращаются практически сразу, на более сложные нужно больше времени, и если вашего мага ночью разбудят нежданные гости, его магический арсенал будет зависеть от времени пребывания во сне.

В завершение, о графике и интерфейсе. Отказавшись в конце концов от спрайта и перейдя на полигоны, программистам удалось создать удивительный по красоте мир, правда, требующий очень мощного процессора, побольше памяти и сильной видеокарты. Можно, конечно, играть и на минимальных уровнях детализации, но это совсем не то, что может показать вам игра. И еще очень много процессорных мощностей уходит на просчет скриптов AI, а как показала практика, на слабых машинах в первую очередь жертвуют "мозгами" персонажей, что отрицательно сказывается на геймплее.

Синтерфейсом особых проблем быть не должно. Правда, остается загадкой, зачем было нужно менять уже привычный интерфейс с "Ворот". Ну да ладно, к этому тоже легко привыкнуть, и особых проблем он не вызывает. Когда к новому интерфейсу привыкаешь, все внимание сразу переходит на сам процесс игры, что и требовалось разработчикам.

Вывод — перед нами очередной шедевр, игра с большой буквы, которые так редко появляются на прилавках магазинов.

Диск с игрой предоставлен интернет-магазином "MediaCraft" www.mediacraft.shop.by



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

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



119454, г. Москва, а/я 37 или
220095, г. Минск-95, а/я 199,
передать по тел. в Минске (017) 221-01-10,
либо через E-mail:

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

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, Амуринский 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 кассетах.

211030, Витебская обл., г.Орша,
ул.Заднепровская, 4/2 — 135, Кучинский Д.М.

Продаю недорогой 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.

Вышло 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@atlasua.net

Бесплатно вышлем каталоги радиосхем, радиодеталей, книг, радиоконструкторов, видео- и аудиокассет, диск с подборкой полезных программ и каталог на CD по радиобиблиотечеству. К заявке на каталог необходимо прикладывать оплаченный конверт с обратным адресом.

422981, Татарстан, г.Чистополь, а/я 248,
Романов А.А. (В.Р.С.)

Куплю контроллер жесткого диска для ПК "Истра 4816.03", программное обеспечение для ПК "Истра-1030M".
453500, Башкортостан, г.Белорецк, а/я 21,

Сафонов С.Н.
Тел. (34792) 4-45-67.

Ищу схему и описание EPSON-совместимого принтера EC-7245 (применялся в КУВТ "КОРВЕТ").
E-mail: zgo@mail.chita.ru

Куплю: игровую приставку "Panasonic 3DO" модели FZ-10; видео-CD-адаптер (68 pin x 1); игровые компакт-диски системы 3DO; джойстики; пистолеты; мышь, можно от других приставок (GoldStar 3DO, Sony 3DO и т.д.) с логотипом 3DO; карту Creative Labs 3DO Blaster и другие ус-

тройства для "Panasonic 3DO".
681000, Хабаровский край,
г.Комсомольск-на-Амуре,
ул.Краснофлотская, 26 — 42, Решетник К.П.

Куплю для ПК "Scorpion ZX-256 turbo" платы GMX, контроллер SMUS, контроллер IBM-клавиатуры и Kempston-mouse с документацией, а также ПО.

681090, Хабаровский край,
Комсомольский р-н, ст.Гурское,
ул.Владивостокская, 4 — 1, Курбатов Н.В.
Тел. 3-91.

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)		
	Радиолобитель. Ваш компьютер	Радиолобитель. КВ и УКВ	Радиолобитель. 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.Минска 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.Гиларовского, д.39 (ст. метро "Проспект Мира" — радиальная);
- г.Москва, ул.Ивана Франко, д.40, к.1, стр. 2 (платф.Рабочий поселок, 15 мин. от Белорусского вокзала);
- г.Москва, ул.Беговая, д.2;
- г.Ярославль, ул.Нахимсона, 12, тел. (0852) 27-57-15.

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

В ООО "ТРЭНТЭКС": 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 "Тулуской" (ст. метро "Тулуская", 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").

А. Вендиловский



newer
winter
night

