

радиомир

11•2004

Ваш компьютер

- "Музыкальные" программы
- Персональные видеоконференции
- Кодирование фильмов в MP4
- Локальные сети под DOS



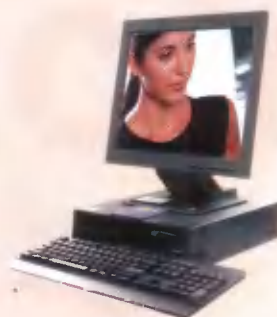


КОМПЬЮТЕРЫ СИТИПРИНТ ДЛЯ ДОМА И ОФИСА

Домашний компьютер: обучение как развлечение!

С новым компьютером Ваши дети смогут быстрее освоить любой предмет, ведь у них будет максимум возможностей для обучения. Компьютер от компании "СитиПринт" на базе процессора **Intel® Pentium® 4 с технологией HT** станет для них настоящим образовательным центром.

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



 **СитиПринт**
КОМПЬЮТЕРЫ И ОРГТЕХНИКА

220006, г.Минск, ул.Полевая 28-7А. Тел./факс (017)2850134.
www.citiprint.by



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,
Анатолий КОРБИТ,
Владимир КУЦЕНКО,
Елена ЛЕВИТМАН,
Николай МЕДВЕДЬ,
Геннадий ПЕЧЕНЬ,
Геннадий ШУЛЬГИН

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

Компьютерная верстка —
Янина БЕЛЬСКАЯ

Техническая графика —
Наталья БЕЛЬСКАЯ,
Александр ГОРАЮТИН,
Мария КАЛАБУХОВА

Оформление обложки —
Наталья БЕЛЬСКАЯ,
Надежда БОГОМОЛОВА

Адреса для писем:
119454, РФ, г.Москва, а/я 37,
факс (095) 432-62-04;
220095, РБ, г.Минск-95, а/я 199
E-mail: rl@radiopage.by
rm@radio-mir.com
WWW: <http://radio-mir.com>

Наши платежные реквизиты:
получатель: ООО "Радиомир Пресс",
ИНН 7729404741, КПП 772901001,
р/с 40702810900010000084
в ООО КБ "Агропромкредит"
ф-л Центральный, г.Москва,
корр. счет 30101810500000000109
БИК 044525109
Адрес банка: 125315, г.Москва,
Ленинградский пр., дом 76, корп. 4

Материалы для публикации принимаются
в рукописном, печатном и электронном
вариантах

Требования к графическим материалам рекламного характера в электронном виде:
CorelDRAW до 10.0, все шрифты в кривых;
bitmaps 300 dpi; TIFF 300 dpi; CMYK.
Приложить печатную копию.

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

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

Подписано к печати 28.09.2004 г.
Формат 60 x 84 1/8.

Печать офсетная. 6 печ. л.
Цена свободная.

© ООО "Радиомир Пресс". Воспроизведение материалов журнала в любом виде без письменного разрешения редакции запрещено. При цитировании ссылка на "Радиомир. Ваш компьютер" обязательна.

Отпечатано в типографии
ЗАО "Красногорская типография".
Заказ № 3059.

радиомир Ваш компьютер

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

№11/ноябрь/2004

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

МУЛЬТИМЕДИА

- А.КАШКАРОВ. "Музыкальные" программы 2
В.КУЦ. Персональные видеоконференции 4

НЕ ТОЛЬКО НОВИЧКУ

- А.ВЕНДИЛОВСКИЙ. Кодирование фильмов в MP4.
Краткий курс молодого бойца 7
С.ГРИНЧУК. Разработка web-сайтов в Microsoft Office FrontPage 2003
Вставка и управление параметрами графических изображений 10
Ю.СИЛКОВИЧ, Б.КИСЕЛЕВ. MATLAB. Интеграция
с элементами MS Office 14
Е.ЛЕБЕДЕВ. Знакомство с PEBrowse Pro Interactive 17
С.РЮМИК. Интернет-калейдоскоп, freeware #12 18

КОММУНИКАЦИИ

- Р.КАРПАЧ. Локальные сети под DOS — стоит ли напрягаться? 20

ДАЙДЖЕСТ 23

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

- А.КОРБИТ. Реализация многопоточных приложений на базе MFC 27
О.ВАЛЬПА. Borland C++ Builder 6 для начинающих 29

ДИАЛОГ ПРОГРАММИСТОВ

- CyberManiac /HI-TECH. Теоретические основы крэкинга 32
А.КАЛЮТОВ. Мозаичные изображения 34
И.РОЩИН. Автоматическое проставление
размеров файлов в HTML-документах 36

РЕЦЕПТЫ

- С.РЮМИК. "PlayStation 2": система внутреннего питания 41
А.БУТОВ. Доработка блока питания компьютера 44

ИГРОТЕКА

- А.ВЕНДИЛОВСКИЙ. Doom 3 45
КОМПЬЮТЕРНЫЕ ХРОНИКИ 47

ДОСКА ОБЪЯВЛЕНИЙ

- Куплю, продам, обменяю 48

Полные листинги некоторых программ расположены по адресу

<http://radio-mir.com>

“Музыкальные” программы

А.КАШКАРОВ,
г.Санкт-Петербург.

В последнее время, наряду с термином “мультимедиа”, получил распространение и другой термин — “музыкальные программы”. Новый термин оказался столь же многозначным, как и его прародитель — им стали называть любые программы, имеющие дело с каким-либо звуком. При этом нередко происходит смешение основных понятий и путаница в принципах работы программ. В результате, одни пользователи не догадываются о некоторых доступных им возможностях, а другие ошибочно возлагают на программу (и компьютер в целом) неоправданные надежды. В этой статье я постараюсь классифицировать методы работы со звуком на персональных компьютерах семейства PC и сделать маленький обзор соответствующих технологий и программных средств.

Audio и MIDI

В современных компьютерах можно выделить две наиболее популярные технологии, имеющие отношение к звуку и музыке:

- Audio (аудио) — наиболее универсальная технология, представляющая произвольный звук как он есть — в виде цифрового представления исходного звукового колебания или звуковой волны (wave), отчего в ряде случаев она именуется wave-технологией. Позволяет работать со звуками любого вида, любой формы и длительности. Звуковая информация обычно хранится в файлах с расширением WAV;

- MIDI — нотно-музыкальная технология регистрации событий, происходящих при игре на электронном инструменте — нажатий клавиш, педалей, воздействий на регуляторы, тумблеры, кнопки и т.п. Последовательность событий образует “электронную партитуру” музыкального произведения — полную программу управления “автоматическим оркестром”. Позволяет точно записать сложное музыкальное произведение, а затем любое число раз исполнить его в точном соответствии с программой.

Audio-технология применяется там, где имеется исходный звуковой сигнал, подлежащий обработке — с ее помощью записывают, обрабатывают и сводят “живые” голосовые партии, речь, шумы, специальные сигналы и т.п. MIDI-технология снискала себе успех в создании музыкальных произведений “с нуля”, посредством только электронных инструментов. При помощи MIDI-системы создаются как музыкальный каркас,

к которому впоследствии будут добавлены голосовые или акустические партии, так и полноценное, законченное музыкальное произведение.

Для применения аудио-технологии достаточно любого звукового адаптера, содержащего АЦП и ЦАП. При этом сложность, качество и цена адаптера совершенно не влияют на принципиальные возможности обработки звука — от адаптера зависит лишь общее качество входного и выходного преобразования, а также сервисные возможности (например, быстрое аппаратное сжатие или фильтрация).

Для использования MIDI-технологии прежде всего нужен электронный музыкальный инструмент, преобразующий последовательность нот и команд управления в звук (обычный или сразу цифровой). Это может быть клавишный синтезатор, звуковой модуль (тон-генератор или синтезатор без клавиатуры), музыкальная карта с аппаратным синтезатором или же программный синтезатор — программа, имитирующая работу реального синтезатора. Соответственно, все возможности, доступные в этой технологии, целиком определяются имеющимся набором MIDI-инструментов.

В Windows каждая технология представлена своим типом звукового устройства. Устройства могут быть реальными (аппаратные адаптеры) и виртуальными (программы-имитаторы, генераторы, фильтры и т.п.). Общение программ с устройствами происходит посредством аудио- и MIDI-портов, которые появляются в системе после установки соответствующих устройств.

Отдельным случаем аудио-портов являются порты DirectSound. Классический (Wave, MME) аудио-порт не гарантирует предельно быстрого вывода звука — при обмене небольшими фрагментами, за счет буферизации и невысокой частоты обращений к адаптеру, возникают значительные (относительно времени звучания самих фрагментов) задержки. Интерфейс DirectSound, входящий в комплект интерфейсов DirectX, дает возможность работать с адаптером с минимальной буферизацией, заметной сокращая задержки. В DirectSound несколько программ используют один порт одновременно, что далеко не всегда возможно при работе с Wave-портами.

Любая программа для работы со звуком на ПК использует в той или иной форме одну из технологий либо несколько сразу. Основные классы программ:

- звуковые процессоры (audio processors);
- звуковые редакторы (audio editors);
- музыкальные редакторы (music/MIDI editors): секвенсоры (sequencers), трекеры (trackers), нотные редакторы (score editors);
- музыкальные процессоры (music/MIDI processors);
- автокомпонаторы (auto composers);
- автоаккомпаниаторы (auto accompaniment generators, jammers);
- распознаватели нот (score recognition software);
- преобразователи форматов (format converters);
- считыватели звуковых дорожек с компакт-дисков (CD rippers/grabbers);
- психоакустические компрессоры (psychoacoustic compressors);
- проигрыватели (players);
- системы для радиовещания и дискотек (delivery systems);
- утилиты и управляющие программы (utility/control software).

Системы записи и сведения, виртуальные синтезаторы и генераторы-анализаторы сигналов также относятся к данной классификации, но это узкоспециализированные программы, поэтому их обзор — это отдельная тема. Многие программы сочетают в себе функции из разных классов. Например, звуковые редакторы и секвенсоры нередко предоставляют также возможности процессоров (обработка в реальном времени), а музыкальные процессоры и автокомпонаторы часто имеют функции секвенсора.

Звуковые процессоры

Программы данного класса имитируют работу типовых устройств обработки звука, применяемых в студийной работе — усилителей, ограничителей, шумоподавителей, компрессоров, эффект-блоков и т.п. Существует три основных типа процессоров:

- *нелинейные (off-line)* — получают сигнал в виде дискового файла, предварительно записанного другими средствами, и записывают результат обработки в другой дисковый файл;
- *сквозные реального времени* — получают сигнал непосредственно с аудиопорта и выводят результат в другой порт;
- *подключаемые модули (plugins)* — получают сигнал от другой программы при помощи специального программного интерфейса (API) и возвращают результат обработки этой же



программе. Фактическим стандартом такого интерфейса стал Microsoft DirectX. Модули обычно поддерживают обработку в реальном времени.

Процессоры первого типа разрабатывались достаточно давно, уступая место процессорам второго типа по мере роста мощности компьютеров. После появления системы DirectX популярные процессоры разрабатываются в этом интерфейсе. Подробнее с работой звуковых процессоров ознакомимся на примере нескольких программ.

DSP/FX Virtual Pack (разработчик — Power Technology, версия — 6.2) — программа имеет набор модулей обработки, которые могут использоваться как посредством интерфейса DirectX, так и непосредственно обрабатывать сигнал, полученный из Wave-файла или аудиопорта. Модули могут использоваться также в качестве расширителей системы многоканальной записи SAW и проигрывателя WinAMP. Алгоритмы перенесены с аппаратной платы эффектов DSP/FX. При наличии в системе плат DSP/FX модули могут использовать процессоры плат, в противном случае вся обработка выполняется программно.

Каждый модуль поддерживает набор пресетов (предустановок) и управление параметрами посредством MIDI-контроллеров. В состав пакета входят следующие модули обработки:

- StudioVerb — параметрический ревербератор с моделированием помещения;
- AcousticVerb — еще один тип ревербератора;
- Optimizer — мягкий пиковый ограничитель и сглаживатель цифровых шумов (dithering);
- Aural Activator — генератор гармоник для восстановления потерянных высокочастотных составляющих;
- Stereo Pitch Shifter — модуль стереофонического изменения высоты звука. Изменяет высоту входного сигнала в соответствии с поступающими MIDI-нотами и перемещением регулятора Pitch Bend. Может использоваться для коррекции неточно выдержанных по высоте нот.
- Multi-Tap Delay — имитатор множественных отражений внутри помещения, от реверберации до длительных эхо;
- Analog Tape Flanger — имитатор аналогового ленточного фленжера;
- Multi-Element Chorus — имитатор хорового эффекта за счет моделирования пространственного разделения звука;
- Parametric EQ — 8-полосный параметрический эквалайзер;
- Auto-Panner — модуль автоматического позиционирования сигнала на стереопанораме;

- Tremolo — модуль внесения амплитудного вибрато (тремоло);

- Widener — расширитель зоны стереоэффекта.

Cylonix Vocoder (разработчик — James J. Clark, версия — 2.00) — в основу 18-полосного вокодера положено то, что речевой аппарат человека работает по разностному принципу — исходный звук голосовых связок, богатый гармониками, подвергается динамическому управлению (резонансному усилению, амплитудной модуляции и фильтрованию) при прохождении через речевой тракт. Принцип работы программы состоит в разложении речевого управляющего сигнала (modulator) на заданное количество частотных полос и анализе динамики в каждой полосе. Полученные в результате анализа сигналы с определенной точностью повторяют динамику работы речевого тракта. Эти сигналы управляют банком частотных фильтров, через которые пропускается богатый гармониками несущий сигнал (carrier). В результате на выходе банка образуется "говорящий" несущий сигнал, как если бы звук подобного тембра издавали сами голосовые связки.

Как несущий, так и управляющий сигнал может быть сгенерирован несколькими способами — самой программой, введен с аудио-порта, либо взят из готового Wave-файла.

Каждый канал вокодера имеет набор регуляторов — уровень, стереопанорама, время затухания, задержка, время и интенсивность эха, уровень дополнительных искажений. Возможно отключение анализа управляющего сигнала — тогда программа работает как обычный 18-полосный эквалайзер.

Звуковые редакторы

Эти программы объединяют в себе функции цифрового магнитофона, звуковой монтажной станции и набора устройств обработки звука (процессоров). Они осуществляют запись, воспроизведение и монтаж (вырезка, вклейка, замена фрагментов фонограммы). Чаще всего имеют набор встроенных и/или подключаемых звуковых процессоров, с помощью которых реализуется сложная обработка записанной фонограммы. Редактор может быть многоканальным, допускающая раздельную запись и обработку нескольких звуковых дорожек с последующим их сведением (микшированием). Ряд редакторов предоставляет звуковые процессоры реального времени, а также средства для исследования сигнала — спектроанализаторы, взвешивающие фильтры и стати-

стические функции:

- запись и воспроизведение звукового сигнала через аудиопорт звукового адаптера (карты);
- чтение и запись звуковых файлов, преимущественно формата RIFF PCM (WAV), а также других распространенных форматов;
- возможность обработки моно- и стереосигнала с разрядностью 8 и 16 и частотой дискретизации до 44100 Гц (нередко — до 24 разрядов и 96 кГц);
- монтаж сигнала (вырезка, вставка, удаление и размножение фрагментов);
- различные способы выделения рабочего участка (selection) при монтаже с выравниванием (snap) по точкам пересечения нуля (zero crossing), по времени (time), по ритмическим долям (beats);
- пометка точек и фрагментов фонограммы с ведением списка таких пометок, из которого можно быстро перейти к помеченному участку, а также поддержка списка участков проигрывания (play list), с помощью которого можно заменить линейное проигрывание на "рваное";
- операции обработки: усиление/ослабление, нормализация (такое масштабирование сигнала, чтобы его амплитуда точно вписалась в заданный диапазон), плавное увеличение/снижение уровня громкости (fade), изменение стереобаланса (pan), сжатие/растяжение динамического диапазона (compress/expand), пороговое пропускание (gate), наложение огибающей (envelope);
- виды эффе́кт-обработки: phaser, flanger, reverb, delay, echo, overdrive, distortion, fuzz и т.п.;
- специальные операции: частотное фильтрование (filter/EQ), изменение высоты (pitch) сигнала или длительности (stretch) участка фонограммы, ослабление шумов (noise, hiss) и щелчков (clicks, pops), формирование музыкальных звуков из участков фонограммы, спектральный анализ всей фонограммы или ее участка;
- преобразование формата сигналов: частоты дискретизации, разрядности отсчета и количества каналов, смешивание стереоканалов в один;
- генерация сигналов различной формы и характеристик — как стационарных, так и с изменяющимися во времени параметрами, а также шумов с различным распределением;
- доступ к зарегистрированным в системе внешним модулям (plugins) обработки звука в стандарте DirectX/ActiveMovie, которым передается для обработки выделенный участок фонограммы;

- синхронизация записи/воспроизведения по MIDI — запуск записи или воспроизведения по внешнему событию, посылка синхронизирующих сообщений другим устройствам (ведущее устройство, master sync), синхронизация по приходящим от других устройств сообщениям (ведомое устройство, slave sync).

Sound Forge (разработчик — Sonic Foundry, версия — 4.5c) — многооконный редактор с поддержкой OLE, видеороликов в формате AVI и дополнительного монитора для их отображения в процессе работы. При работе с файлами в 16-разрядном формате PCM (WAV) есть возможность открыть файл в режиме непосредственного доступа (Direct mode), без промежуточного копирования. Это заметно ускоряет работу, однако лишает возможности сохранить прежнюю версию файла при аварийном завершении.

При монтаже удобно пользоваться функциями Preview Cur/Cursor (прослушивание, каким будет результат вырезки до выполнения самой вырезки) и Pre-roll to Cursor (прослушивание небольшого участка перед курсором). Поддерживается список помеченных участков (regions) и выборочного проигрывания.

В режиме записи возможно создание "пульта дистанционного управления" — Remote Control. При этом основное окно редактора заменяется на небольшое окно пульта, находящееся поверх других окон. Эта функция удобна при записи сигнала, формируемого другой программой, либо устройством, управление которым производится из другой программы.

Возможен прямой сброс данных на диск, минуя системный кэш с обратной записью. Это позволяет избавиться от долгих пауз, когда Windows сбрасывает кэш на диск, останавливая при этом все программы, однако работа диска в таком режиме становится весьма напряженной за счет непрерывного позиционирования.

Редактор может работать с внешними сэмплерами (Akai, E-mu, Kurzweil, Peavey), поддерживающими стандарты MIDI SDS и/или SCSI SMDI. Также поддерживается подготовка сэмплов для ACID — другой программы Sonic Foundry, предназначенной для создания музыки из готовых фрагментов. Генератор сигналов выдает простые периодические сигналы и серии DTMF, а также имеет функцию FM Synthesis — частотно-модуляционный операторный синтез.

При помощи собственного подключаемого модуля Batch Converter со-

здается последовательность операций обработки, которая затем может быть автоматически применена к одному или нескольким файлам. Модуль CD Architect предназначен для формирования и записи звуковых компакт-дисков, Spectrum Analysis служит для спектрального анализа фонограммы, Noise Reduction — для ослабления помех и шумов, Q-Sound — для придания звуку эффекта объемности. Синхронизация по MIDI возможна в обоих режимах — ведущем и ведомом.

Имеется любительская локализация SF 4.5 на русском языке (переведены тексты меню и сообщений). Качество перевода — среднее.

WaveLab (разработчик — Steinberg, последняя выпущенная версия — 2.02) — один из наиболее мощных и удобных современных редакторов. Поддерживает форматы до 24 разрядов и 96 кГц. Предоставляет все необходимые монтажные операции, нормализацию, преобразования динамики, коррекцию высоты/времени звучания. Операции сложной обработки немногочисленны: трехполосный эквалайзер, гармонизатор на 16 голосов (создает дополнительные гармоники основного голоса), качественный Chorus. Основное внимание при обработке уделено поддержке модулей реального времени — собственных, DirectX и VST (от Cubase VST). Для управления модулями сделана специальная панель эффектов (Master Section), в которой можно выбрать до шести модулей одновременно. Для модулей WaveLab/VST отображаются панели управления, стилизованные под аппаратные стоечные блоки. Панели модулей, а также панели инструментов WaveLab (toolbars) могут находиться в любом месте экрана, а не только в окне редактора. Функция Batch Processor позволяет сформировать алгоритм пакетной обработки набора файлов. Поддерживает обмен данными с аппаратными сэмплерами AKAI, Ensoniq, E-mu, Kurzweil, Roland в стандартах SDS/SMDI. Может считывать звуковые дорожки с компакт-дисков. Содержит встроенный CD-рекордер с возможностью задания длительности пауз между дорожками. Имеет несколько стилей и цветовых палитр интерфейса. После выполнения каждой операции отмечается время, затраченное на операцию, с точностью до миллисекунды, что удобно для оценки эффективности.

(Продолжение следует)

В.КУЦ,

E-mail: victor_kootz@mail.ru

В прошлом номере журнала [1] вы познакомились с web-камерами, которые, несмотря на свою многофункциональность, все-таки наиболее часто используются именно для создания персональных видеоконференций. В этой статье мы чуть подробнее остановимся на программном обеспечении и некоторых практических аспектах "превращения" обычных ПК в терминалы видеоконференц-связи. Самый простой тип такой связи выглядит как непосредственное соединение двух компьютеров, оборудованных web-камерами (чаще всего их называют терминалами) — в этом случае необходимо лишь наличие двух базовых комплектов оборудования и установленных на них программных клиентов. Естественно, при этом необходимо иметь возможность соединиться с коллегой либо через компьютерные сети (включая системы радиодоступа), либо по каналам телефонной связи (лучше цифровым).

Почти все клиентские программы могут использовать любое устройство, для которого в системе установлен так называемый capture-драйвер VFW (драйвер видеозахвата, соответствующий API Video For Windows). Этот драйвер обеспечивает приемлемый захват видеопотока стандарта CIF (Common Intermediate Format, основной формат обмена видео в стандарте H.323, размер изображения — 352 x 288 пикселей, кодировка — YUV, разрядность — 8 битов) со скоростью 20...30 кадров в секунду — этого вполне достаточно, чтобы обеспечить получение качественного изображения собеседника. При этом вся нагрузка по обработке видеосигнала (удаление шумов матрицы, компрессия изображения и звука) ложится на центральный процессор, который должен быть достаточно мощным.

Если видеоконференция проводится между абонентами локальной сети, то для видеотерминалов, построенных на базе PC и web-камер, подключаемых к ним посредством интерфейса USB 1.1, обеспечивается вполне приемлемое качество передачи видеопотока формата CIF с частотой смены кадров около 18...20 в секунду совместно с аудиопотоком 8...16 Кбит/с. При этом даже устаревшие сети с пропускной способностью 10 Мбит/с загружаются не более чем на 10%.



Персональные видеоконференции

Для абонентов видеоконференций, проводимых в Интернете, ситуация не столь отрадная. Так, даже самое скоростное модемное соединение (со скоростью до 56 Кбит/с), не в состоянии обеспечить удовлетворительный результат. Кое-какой эффект присутствия еще можно получить, установив минимальный размер изображения 128 x 96 пикселей. При этом, в первую очередь из-за нестабильности аудиопотока, живое общение чаще всего сводится к обмену текстовыми сообщениями, сопровождаемыми "слайд-шоу".

Реально же организовать минимально качественную видеоконференцию по IP-сетям могут лишь владельцы выделенных высокоскоростных цифровых каналов связи, обеспечивающих скорость не менее 512 Кбит/с. Но даже на такой линии для участия в публичных встречах для обеспечения минимально приемлемой частоты смены кадров 12...15 кадров/с размер транслируемого изображения не должен превышать 176x144 пикселей. Низкая эффективность скоростного IP-канала связи объясняется плохим качеством соединения удаленных абонентов, а также значительной задержкой в доставке пакетов данных в реальном масштабе времени, органично присущей всем IP-сетям. К тому же, необходимо учитывать, что количество информации, передаваемой в видеоконференциях стандарта H.323, ощутимо больше, чем в оптимизированных для видеоконференц-связи сетях ISDN (стандарт H.320), которым для достижения "промышленного" качества видеоконференции вполне хватает половины пропускания 256 Кбит/с.

Клиентское ПО для видеоконференций

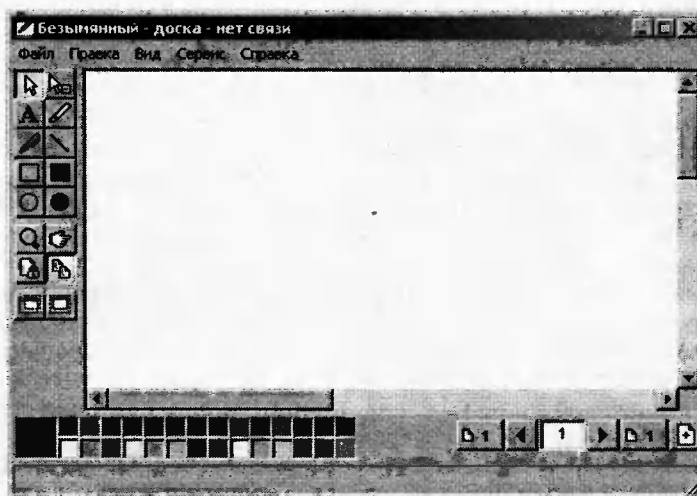
Практически все клиентское ПО для видеоконференций базируется на основе спецификации H.323, включая ряд дополнительных протоколов (например, T.120 — для разделяемых приложений). Для организации видеоконференции на любительском уровне наиболее подходящей будет программа Microsoft NetMeeting, главное достоинство которой — ее доступность, так как она входит в стандартную поставку ОС Windows 2000/XP.

Как следует из названия ("Встреча в сети"), NetMeeting — это нечто большее, чем инструмент для ведения простой беседы (текстом или голосом) через Интернет. С помощью этой программы можно пересылать файлы, проводить видеоконференцию и даже совместно с другими пользователями запускать приложения. Например, вы можете запустить документ Word на своей машине, а ваши респонденты смогут увидеть его и даже редактировать на своем мониторе.

В NetMeeting большое внимание уделено средствам безопасности и защиты предоставляемых услуг, особое внимание уделяется безопасности чата и совместного выполнения приложений. К сожалению, понятие "безопасность" Microsoft трактует довольно однобоко — программа не предоставляет возможности шифрования передаваемых аудио- и видеоданных, что оправдывается суровыми требованиями американских стандартов (конечно, иначе как бы такие пламенные борцы за права "обездоленных" монополий как RIAA могли бы отыскивать себе жертвы?).

Кроме того, NetMeeting может служить для организации многоточечных конференций, когда один из терминалов (Master) выступает в роли видеосервера и отвечает за трансляцию поступающих потоков данных остальным участникам конференции (Slave). В качестве бесплатной системы адресации программа NetMeeting предлагает сервис глобальных директорий Microsoft Internet Directory, что, конечно, не исключает поддержки локальных "призрачников" или прямых звонков.

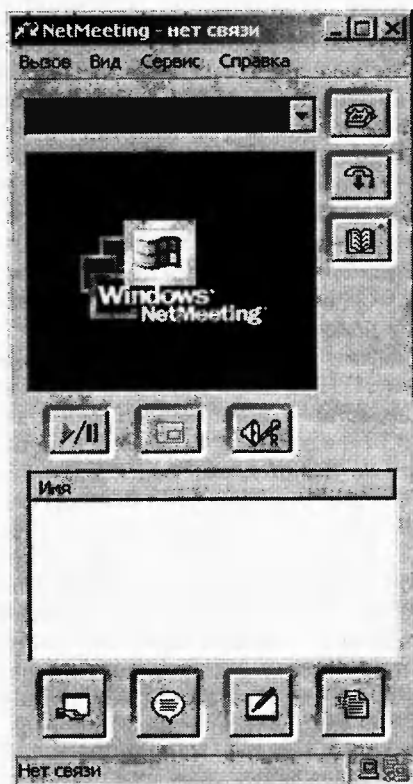
Рис. 1



В программе достаточно хорошо реализован ряд вспомогательных функций. Так, возможность обмена с собеседником графической информацией в интерактивном режиме обеспечивает "белая доска" (WhiteBoard), являющаяся цифровым эквивалентом обычной школьной доски, по которой пишут мелом. При задействовании этого режима у каждого участника открывается дополнительное окно, внешним видом сильно напоминающее графический редактор Paint, со всеми необходимыми для рисования функциями (рис.1). Все ваши действия (рисование, вставка фотографий, графиков и т.п.) тут же будут отображаться на такой же "белой доске" партнера.

Процесс работы с NetMeeting предельно прост и не вызывает затруднений даже у самых что ни есть неподготовленных пользователей. Для установки программы достаточно ввести в консоли команду "conf" (или выполнить Run — conf) и ответить на ряд несложных вопросов в серии конфигурационных диалогов. После установки NetMeeting будет создана учетная запись на одном из специализированных серверов Microsoft. Вам следует выбрать себе псевдоним и пароль для доступа к аккаунту. Если же по каким-либо причинам вы хотите установить соединение без подключения к серверу NetMeeting, тогда вам нужно будет самостоятельно ввести IP-адрес вызываемого абонента. При этом следует учитывать, что большинство провайдеров, предоставляющих услугу Dial-up-доступа в Интернет, выделяют своим клиентам динамические IP-адреса. Так что для того, чтобы к вам можно было обращаться напрямую, минуя удаленный сервер, следует предварительно озаботиться получением у провайдера постоянного IP-адреса.

Рис. 2



Интерфейс NetMeeting продуман до мелочей и не содержит ничего лишнего — четыре ярлыка в нижней части экрана подскажут, что делать (рис.2). Вы можете выбрать между стилем а-ля сотовый телефон, удобным для голосовой связи, или стилем “картинка в картинке” для видеоконференции. Можно также и вовсе уменьшить рабочее окно программы до “размеров” пейджера.

Для того чтобы начать чат, нужно или ввести IP-адрес другого компьютера, или связаться с сервером NetMeeting, чтобы найти собеседника. Для поиска партнера NetMeeting использует соединение с ILS-серверами (Internet Locator Server). Кнопка Directory в левой части экрана откроет для вас список серверов, с которыми можно связаться одним щелчком мыши. Вы можете также пополнить список своими ILS-серверами, нажав на кнопку с изображением развернутой книги (Find Someone in a Directory). Войдя на сервер, для начала разговора достаточно щелкнуть по имени потенциального собеседника (выбрав его из имеющегося списка), который, получив уведомление о вызове, может либо принять его, либо отказаться от разговора, и спустя секунд 10...30 после того как машины установят двустороннюю связь, можно начинать общение. Однако при практическом использовании программы могут встретиться определенные трудности,

и одна из самых существенных — проблемы соединения со многими ILS-серверами. Также после соединения не так уже просто найти подходящего собеседника — NetMeeting имеет всего лишь одну строку для описания абонента, и совершенно отсутствует систематизация по интересам.

Разумеется, NetMeeting — далеко не единственная программа, предоставляющая пользователю возможность полноценного общения в Интернете, более того, сегодня она все чаще замещается более привычными простому пользователю интернет-пейджерами типа MSN Messenger, Windows Messenger (поставляется с XP и содержит лучшее из интерфейса NetMeeting, включая функции видеосвязи), Yahoo Messenger и т.д. Конечно, все эти программы не были изначально предназначены для видеоконференций, ведь их основная цель заключается в передаче и приеме текстовых сообщений. Добавление поддержки видеоконференции было осуществлено при дальнейшем развитии этих программ, так что сегодня они уже могут без особых проблем пересылать звук и видео.

Несколько элементарных советов участникам сетевых видеоконференций

Находясь под прицелом объектива, многие участники видеоконференций испытывают определенный психологический дискомфорт. Чаще всего это связано с запаздыванием реакции собеседника. Например, сплошь и рядом возникает ситуация, в которых человека перебивают, когда он еще не закончил фразу. Кроме того, следует учесть, что камера беспощадно усиливает внешнюю и формальную составляющие разговора, поэтому лучше вести себя максимально сдержанно — не злоупотреблять мимикой, не вертеться и не размахивать руками без особой на то нужды. Также не будет вредным заранее послушать свой голос, записанный на магнитофон — в условиях общения по видео, если вы хотите быть правильно понятым своим собеседником, требуется говорить чуть более медленно и размеренно, чем обычно.

При проведении видеоконференции участникам следует соблюдать ряд несложных правил “хорошего тона”, которые позволят вам почувствовать себя более комфортно во время видеоконференции: — рекомендуется устанавли-

вать низкое разрешение монитора, например 640 x 480 — изображение собеседника в этом случае хорошо выглядит даже при небольших форматах;

— необходимо обеспечить равномерное освещение комнаты с участника-ми, а также придерживаться однотонного стиля одежды — в этом случае камере легче поддерживать автоматический баланс белого, и цвета на дисплее принимающей стороны будут более естественными;

— не стоит использовать встроенный микрофон имеющейся у вас web-камеры (по причинам, изложенным в [1]). Также не следует злоупотреблять функцией отключения аудиопотока при отсутствии сигнала в микрофоне (с целью минимизации информационного трафика ее поддерживают почти все программы) — при этом глотаются начало и окончание фразы, что приводит к возникновению неприятных эффектов.

Вместо заключения

При всех несомненных достоинствах таких симпатичных девайсов как web-камеры, они, честно говоря, пока имеют не очень радужные перспективы выживания на отечественном компьютерном рынке. И главная причина такого положения дел — малое распространение в наших пенатах высокоскоростного доступа в Интернет, наличие которого является обязательным условием для более или менее комфортного визуального общения по Сети, для чего web-камеры, собственно, и предназначены. И только после того как нормальный доступ в Интернет будет доступен для самых широких масс населения, можно будет надеяться на то, что web-камеры сумеют, наконец, завоевать свое место под компьютерным солнцем.

Литература

1. В. Куц. Я тебя вижу, а ты меня? — Радиомир. Ваш компьютер, 2004, №10. С.2.

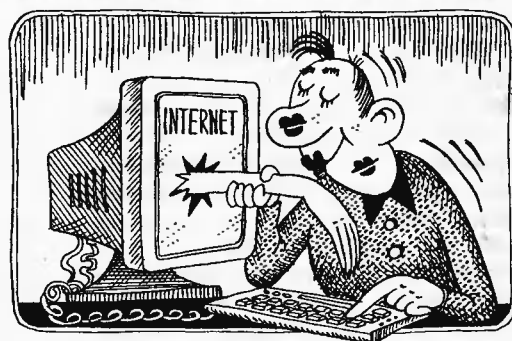


Рисунок О.Попова

Кодирование фильмов в MP4.

Краткий курс молодого бойца

А.ВЕНДИЛОВСКИЙ,
г.Минск.

Количество статей на эту тему в компьютерных журналах и Интернете сейчас, наверное, уже перевалило за тысячу. Читая их, испытываешь стойкое ощущение, что качественно перевести фильм в формат mp4 может даже ребенок. Однако начав заниматься этим практически, внезапно наткнувшись на множество подводных камней, о которых авторы почему-то не упомянули. Начнем с развенчания некоторых мифов, которые появились с того самого момента, когда между любителями DVD- и mp4-фильмов разгорелся спор, чей формат лучше :).

Миф первый — фильмы на DVD всегда потрясающего качества, как картинка, так и звук

Как ни странно, но в этот миф верят практически все, кроме владельцев домашних кинотеатров. Вы удивлены? А ведь DVD — это всего лишь формат сжатия потокового видео, и очень большое значение имеет как качество исходника фильма, так и качество оцифровки. Я неоднократно встречал на фирменных дисках фильмы, в которых попадались артефакты изображения, или качество картинки уступало фильмам в mp4. Если не верите, почитайте журнал "Total DVD" — там очень подробно рассказывают о качестве изображения на DVD-дисках. Если вам попался такой экземплярчик, то как бы вы ни старались, изначальные дефекты никуда не денутся.

Господа пираты тоже отличаются отменным чувством юмора, а карты для видеооцифровки уже спустились по ценам с заоблачных высот, так что "самопальных" оцифровок сейчас пруд пруди. Среди них попадаются и самые жуткие "экранки" (вы не поверите, снятые любительской камерой в кинотеатре!), и снятые с телевизионного экрана (правда, реклама все-таки вырезана). Последние никогда не смогут дотянуться по качеству до настоящего DVD — телевизионный сигнал сильно не дотягивает по плотности информационного потока до реального DVD.

Мифы второй и третий — оцифрованная копия всегда будет намного хуже оригинала, и наоборот, любой фильм можно вставить на один диск без потери качества

И в том, и в другом случае обе стороны несколько перегибают палку. Действительно, при дополнительном сжатии произойдет ухудшение изображения, но при **качественной** оцифровке эти погрешности будут практически

незаметны для человеческого глаза, при условии, что вы будете смотреть фильм на экране монитора или обычного домашнего телевизора! Погрешности техники и человеческого глаза скрывают эти ухудшения. Мы провели специальный тест — посадили перед экранами пять человек, утверждающих, что они являются истинными ценителями качества. Никто из испытуемых не смог верно догадаться, когда я запускал DVD, а когда — оцифровку! То же самое было и со скриншотами из фильмов.

Теперь о третьем мифе — **никогда фильм не может быть качественно оцифрован и помещен на один CD-диск, если его длительность больше 45 минут!** То, что обычно продается на лотках с гордой надписью "DVD-качество", только позорит и покрывает грязью доброе имя mp4. Два-три диска — вот сколько должен занимать фильм!

Миф четвертый — в оцифрованных фильмах нет всех "примочек" DVD!

Опять же, это несколько неверно :). На любой оцифрованный фильм можно наложить настоящий шестиканальный звук прямо с DVD. Достаточно просто вырезать звуковую дорожку и прикрепить ее к готовому фильму, и звук у вас будет точно такой же. Хотите несколько звуковых дорожек — пожалуйста, можете вырезать их все, прикрепить к своему фильму и переключаться по ходу просмотра, сколько вашей душе будет угодно. Субтитры? А в чем проблема? Вырезаем

субтитры и после некоторой обработки вставляем в фильм. Эпизоды? Берем проигрыватель BSPlayer и делаем там закладки, по которым легко можно перейти в необходимый эпизод. Кажется, я ничего не забыл ;).

Теперь от мифов перейдем к кодированию.

Этап первый — перенос фильма на винчестер

Для оптимальной и качественной работы, я советую использовать Windows XP и файловую систему NTFS на винчестере, это поможет избежать проблем с файлами, чьи размеры превышают два гигабайта.

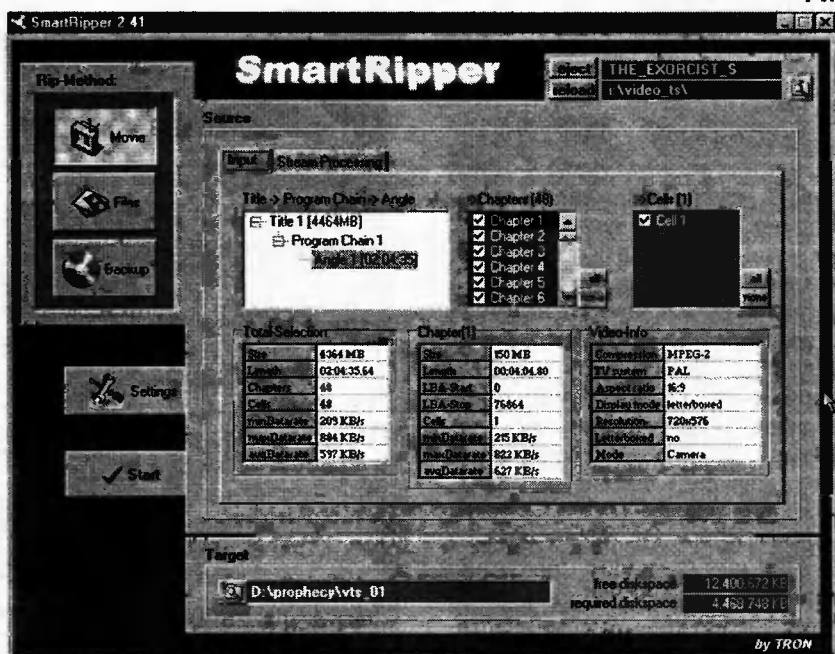
Если вы думаете, что обычное копирование файлов вам тут поможет, то глубоко ошибаетесь — чтобы получить исходный материал, нужно позаботиться о снятии защиты от копирования и зональной защиты DVD-диска. Лучше всего для копирования использовать SmartRipper 2.41 (рис.1).

При запуске данной программы нам предложат три варианта "сдириания":

- Movie — только сам фильм;
- Backup — полное содержимое диска;
- Files — оригинальные vob-файлы.

Нас интересует только первый и второй вариант. Для любопытствующих добавлю, что в закладочке Stream Processing можно выбрать то, что не надо сбрасывать на винт (дополнительную звуковую дорожку, субтитры и т.д.) а в установке Setting — установить

Рис. 1



размер конечного vob-файла. Я всегда стараюсь, чтобы размер плавал в пределах 2...3 Гб.

Внизу видим папочку target — это место, куда будет сброшен фильм. Если окошко внизу справа (свободное пространство на винте) не выделено красным цветом, смело жмем Start и ждем окончания работы.

После окончания процесса мы увидим несколько vob-файлов и файл info.txt. Вот его и откроем в первую очередь. Там мы почерпнем важную информацию:

- формат видео (NTSC, PAL);
- соотношение сторон экрана (16:9, 4:3 и т.п.);
- нумерация и принадлежность звуковых дорожек;
- и т.д.

Подготовка к кодированию

Для продолжения работы необходимо установить следующие программы: VFAPICodec 1.05, DVD2AVI 1.77.3, VirtualDub 1.5.10, NumDub 1.0 rc2, KLCodecPac 2.2 (разумеется, можно использовать и более старые версии программ).

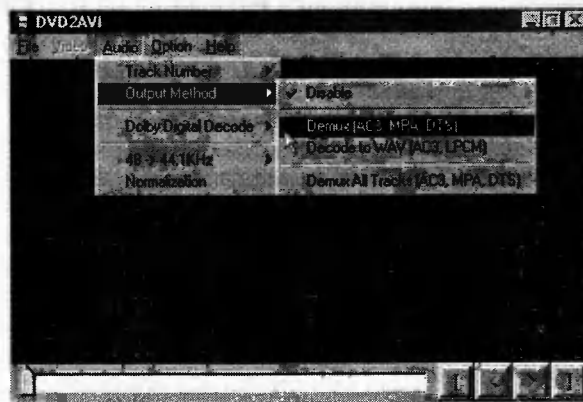
Чтобы сразу избежать вопросов, уточняю. Мы не будем пользоваться пакетами "все в одном". Несмотря на щедрые обещания разработчиков, все эти программы "только раз нажми на кнопку — и получишь готовый фильм" по качеству кодирования могут быть оценены максимум на 4 с минусом, и предназначены для того, для кого качество не играет большой роли. Мы же к таким не относимся.

Запускаем DVD2AVI — для подготовки к фреймсервингу и "вырезания" звуковых дорожек (рис.2).

Жмем File — Open и выбираем первый vob-файл. Если его риппинг прошел без проблем, остальные подключатся сами.

Далее Audio — Dolby digital — Demux (AC3), и в том же меню Audio — Track Number — номер необходимого звукового трека. В нашем случае мы хотим, чтобы звуковая дорожка была исходной многоканальной. Теперь остается только нажать на F4 и придумать название для файла-проекта, опять View, и ожидать, пока DVD2AVI пройдет по всему фильму. Данная программа также позволяет проконвертировать звуковую дорожку в формат wav или подготовить к конвертации лишь выделенный фрагмент, но в нашем случае это не имеет значения. На выходе будет получен файл с расширением .d2v. Программы конвертации не умеют работать с этим файлом, как и с vob-файлами, напрямую. Чтобы исправить эту ситуацию,

Рис. 2



загружаем VFAPICodec (рис.3), затем загружаем в него наш файл и жмем кнопку Convert. Буквально через минуту мы получим avi-файл, в имени которого будет присутствовать последовательность символов d2v. Теперь, если вы еще не установили KLCodecPac, делаем это прямо сейчас. На этом подготовительный этап будет закончен.

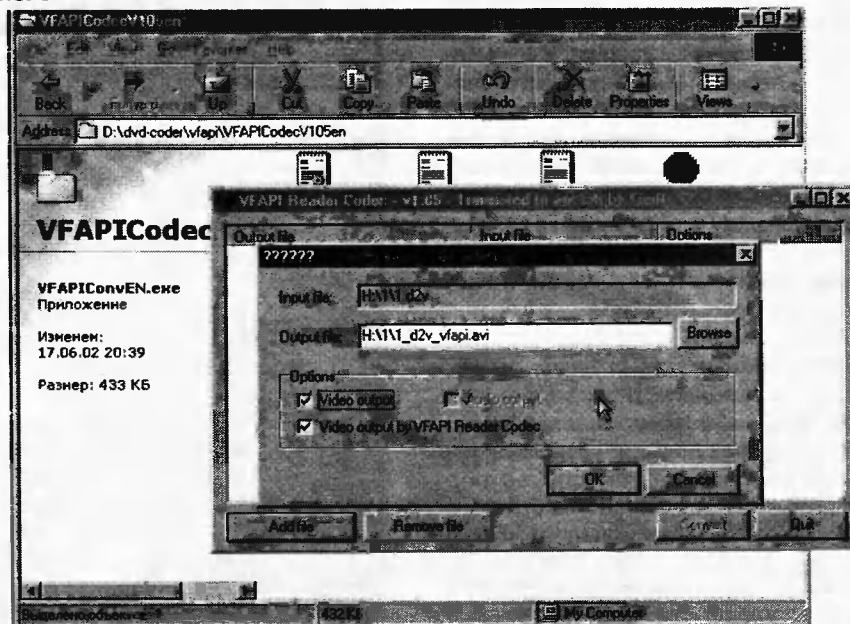
К следующему этапу придется подойти более обстоятельно, при одном взгляде на список кодеков глаза начнут разбегаться, поэтому необходимо предварительно ответить на два вопроса.

1. Где собирается просматриваться данный фильм?

Если только с использованием компьютера, то можете смело выбирать любые кодеки и экспериментировать с теми, которые вам понравятся (если это ваш компьютер, или позаботьтесь, чтобы к фильму прилагался пакет KLCodecPac).

Но если вы будете применять DVD-проигрыватель с поддержкой фильмов mp4, то не надо забывать, что старые

Рис. 3



модели уверенно работают только с 5-й версией DivX.

2. На каком носителе будет находиться фильм?

Если это винчестер, то точный размер конечного файла вас будет интересовать лишь с той точки зрения, поместится ли он весь. Если носитель — матрица DVD-R, то скорее всего — сколько фильмов вы собираетесь на нее поместить. А вот с обычными CD-R несколько сложнее — желательно рассчитать размер,

чтобы все болванки были "залиты" полностью. Бывает очень обидно, когда остается кусок размером с десяток мегабайт, и приходится или отводить под него отдельную болванку, или переделывать всю работу. В Интернете можно встретить множество калькуляторов для расчета размеров фильма, правда, погрешности у них временами очень большие, однако, с приобретением опыта, вы достаточно точно сможете прикинуть "вес" будущего файла "на глазок".

Кодирование

Основными считаются четыре вида кодеков: DivX, XivD, а также Low Motion и Fast Motion. Первые два достаточно универсальны. Я не стану останавливаться на том, какие именно версии этих кодеков лучше, какие хуже. Этот спор длится практически вечно и, немного утихнув, вновь разгорается с выходом каждой новой версии. Последние два — это старейшие из имеющихся кодеков, предназначенные для кодировки фильмов, в которых мало или, наобо-

рот, много активных сцен. Для описания настроек я возьму кодек DivX пятой версии, как наиболее распространенный и аппаратно совместимый. Фильм мы будем кодировать в два прохода, это наиболее оптимальный вариант — в один проход не получишь необходимого качества, и файл будет больших размеров, три, как показывает практика, делать бесполезно.

Открываем VirtualDub (рис.4). Самое первое, что вы увидите — это предупреждение, что используется DivX, и нужно подумать об авторских правах. Смело закрываем окно и продолжаем работать. Загружаем в VirtualDub полученный avi-файл и видим два окошка: в левом — исходный файл, в правом — конечный.

Тут мы опять должны остановиться и подумать о выборе параметров.

Обычно DVD-фильмы записаны в разрешении 720*480, но многие уменьшают разрешение экрана с целью уменьшения размеров файла. Если вас интересует качество, НЕ МЕНЯЙТЕ РАЗРЕШЕНИЕ ЭКРАНА! Рассказы о том, что фильм с разрешением 300 точек выглядит точно так же — чистойшее вранье.

Следующий момент — черные поля широкоформатных фильмов. Я не рекомендую их вырезать по одной-единственной причине, о которой так редко упоминают в статьях — если вы обрежете поля сверху и попытаетесь сохранить пропорции картинки, то вам придется обкорнать экран по бокам или нарушить геометрию и потом удивляться, почему это лица выглядят на экране такими вытянутыми, словно баклажаны? Если DVD-фильмы смотреть широкоформатными не считается зазорным, то и мы не будем создавать проблемы на свою голову.

Если у нас исходный фильм в формате NTSC, то с вероятностью 99,9% можно сказать, что мы заметим странные и неприятные полосы, называемые гребенкой. Это вызвано особенностью построения кадров в данном формате. От "гребенки" мы будем избавляться в самую первую очередь — идем по пути Video — Filters — Add — Deinterlace и выбираем Blend fields together. Чтобы окончательно добить "врага", выбираем Video — Frame Rate — Inverse telecine — Reconstruct adaptive. Если фильм записан в формате PAL, ничего подобного делать не надо!

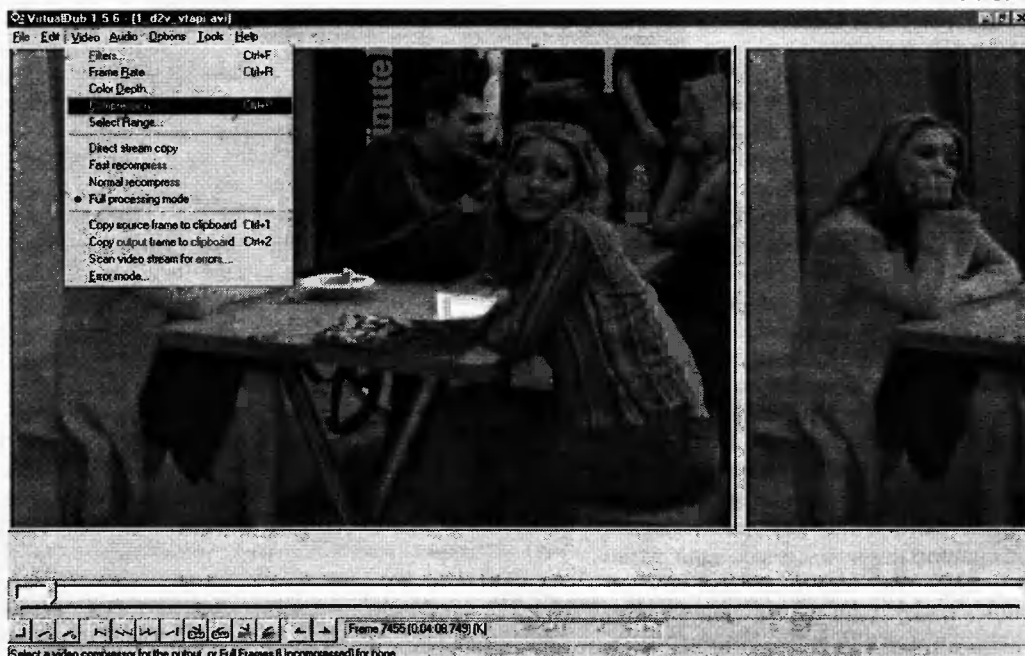


Рис. 4

Вот теперь-то как раз и начнется настройка самого кодека к первому и второму прогону — Video — Compression — DivX — Configure. Переходим в zakładку Profile, снимаем галочку с Choose your profile и устанавливаем галочки напротив Use Quarter Pixel, Use GMC и Use Bidirectional Encoding. Снимать или нет галочку с психовизуального улучшения — это уже дело вкуса. Некоторые утверждают, что качество картинки при этом падает, но вот то, что размер фильма уменьшается — это точно. Установки ключевых кадров — Max keyframe interval — я устанавливаю в диапазоне от 50 до 100 — это очень помогает несколько позже, при окончательной нарезке фильма. Бегунок "Performance — Quality" нужно поместить на максимальную позицию качества. В других версиях кодека это выбирается в выпадающем меню, и там же устанавливается режим Slowest.

На главной странице настроек выбираем 2-pass first pass — двухпроходное кодирование, первый проход. Затем выбирается битрейт. Для хорошего качества он должен быть не менее 1000, для качества, сравнимого с DVD — это 1300....1600. И помните, чем выше насыщенность фильма действием и чем больше в нем мелких деталей (возьмите, к примеру, такие фильмы как "Матрица 2, 3" или "Звездные войны 1, 2"), тем выше должен быть битрейт, чтобы по ходу процесса не потерять детали. Жмем OK и в меню VirtualDub выбираем Save as AVI. Далее придумываем имя для первого файла и ставим галочку внизу, в поле Job (это для того, чтобы первый прогон не запустился немед-

ленно) и вновь возвращаемся в Video — Compression — DivX — Configure. Выбираем 2-pass second pass, сразу же у нас становится активной настройка Bitrate modulation. Если обрабатываемый фильм относится к таким, где мало активных сцен, и действие разворачивается плавно, то лучше поставить бегунок на Low motion 0.2...0.3. Если у вас крутой боевик, смело двигайте в противоположную сторону на 0.3...0.4 единицы. Опять жмем OK и в меню VirtualDub выбираем Save as AVI. Затем придумываем имя для второго файла (оно должно отличаться от имени первого файла!) и ставим галочку внизу, в поле Job.

Далее переходим в File — Job control и видим в поле работы два проекта в режиме ожидания. Отключаем в компьютере все резидентные программы, в том числе и антивирус, и жмем на кнопку Start. Компьютер сам сделает два прохода, по времени это может занять от 6 до 24 часов, в это время к компьютеру лучше даже и не притрагиваться, иначе в фильме будут попадаться весьма неприятные артефакты изображения.

И вот, после длительного ожидания процесс кодирования закончен! Проверим длительность фильма и сравним с исходником, если совпало — отлично. Если нет — начинай все сначала. Промотаем фильм, проверим качество картинки в сценах, где много движения, и в достаточно пассивных сценах. Также проверим на наличие возможных артефактов. Если качество нас устраивает, будем прикреплять звук.

Запускаем NumDub (рис.5) и загружаем конечный файл с фильмом.

Идем в меню *Audio* — *AC3* и указываем нашу вырезанную звуковую дорожку. Если мы хотим прикрепить еще одну, то там же выбираем *Interliving* — *AC3* — *имя второй звуковой дорожки*. В меню *Video* — *Direct stream copy* выбираем *File* — *Save as AVI*, но под именем, иным чем исходный файл. По окончании процесса «вклейки звука» результат проверяем на совпадение картинки и звука. Метод с дорожкой AC3 хорош тем, что разбежки происходят очень редко. Но если подобное случилось, придется запустить процесс заново и, если разбежка постоянная, подкорректировать ее сдвигом на необходимое количество миллисекунд. Если разбежка появляется к концу фильма, то следует установить в меню *Video* — *Frame rate* опцию *Change so video and audio durations match*.

Последний глянец

Готовый фильм мы наверняка хотим подготовить к проигрыванию так, чтобы не приходилось каждый раз настраивать параметры формата изображения и т.п. вручную, а также аккуратно порезать для записи на болванки CD-R.

Опять загружаем фильм в *VirtualDub*, устанавливаем в меню *Video* — *Direct stream copy*, и щелкаем по клавише *Home* (начало эпизода), затем жмем *Ctrl* — *Shift* — *J* и ука-

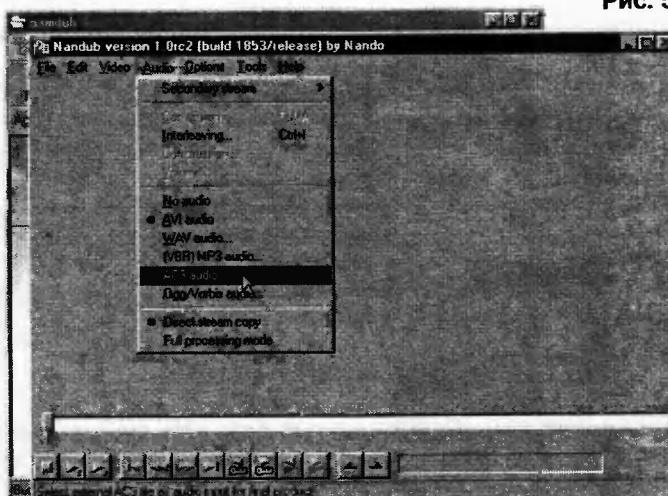


Рис. 5

мента. Снова жмем *Home* — началом второго фрагмента станет окончание первого, опять *Ctrl* — *Shift* — *J* и указываем размер второго эпизода (как сумму размера первого плюс требуемый размер второго). Далее *End* и сохранение. Эту процедуру нужно повторить столько раз, сколько потребуется для нарезки всего фильма. Для каждого эпизода создаем каталог, в который переписываем *BSPlayer*, и начинаем редактировать его *ini*-файл. Вот для примера:

```
[Main]
NumberCDs=2 (на скольких дисках записан фильм)
CD=1 (порядковый номер диска)

[Movie]
Title=Название фильма
Directory=.
FName=имя файла эпизода.avi
AudioName1=Название первой звуковой дорожки (может быть к примеру "русский вариант")
AudioName2=Название второй звуковой дорожки

[Options]
FullScreen=1
Skin=KDEMOD
Lang=Russian
ExitAtEnd=1
Audio=1
AudChan=1
RunHD=0
Aspect=4:3
```

Теперь остается только создать файл *autorun*, и можно смело переписывать фильм на болванки!

Автор благодарит группу "Трейсвандир-Медиа" за консультации в процессе написания статьи.

С.ГРИНЧУК,
E-mail: grinchuk@nihe.niks.by

Разработка web-сайтов в Microsoft Office FrontPage 2003

(Продолжение. Начало в №8-10/2004)

Вставка и управление параметрами графических изображений

Сегодня, как и было обещано, мы поговорим об использовании рисунков на web-страницах. Чаще всего графические изображения выступают в качестве средства оформления web-документов. Например, мы уже знаем, что рисунок может использоваться в качестве фона web-страницы. Кроме того, графические изображения можно добавлять непосредственно на web-страницу для отображения всевозможных иллюстраций. Вот об этой возможности мы и поговорим более подробно, продолжая работу над нашим учебным персональным сайтом myweb2. Поэтому ваша задача — выполнить

1. Запуск Microsoft Office FrontPage 2003 и

2. Открытие web-сайта myweb2, созданного на предыдущих занятиях

3. Вставка графических изображений

В качестве примера украсим домашнюю страницу нашего сайта какой-либо картинкой. Проведем для этого небольшую подготовительную работу. Двойным щелчком мыши откройте в режиме редактирования начальную страницу сайта *index.htm* и создайте пустой абзац после горизонтальной линии (именно сюда мы и будем вставлять рисунок). Далее, как обычно, установите текстовый курсор в новый абзац и выполните команду **Вставка/Рисунок**. Заметьте, что в открывшемся под-

меню содержатся команды, реализующие возможность добавления на страницу рисунков разных типов и из различных источников. Обсудим эти команды более подробно.

Команда **Картинки** позволяет воспользоваться коллекцией картинок от компании Microsoft, содержащей мультимедийные файлы (рисунки, фотографии, звуки, видео), обычно называемые просто клипами, которые можно использовать во всех документах Microsoft Office. Команда **Из файла**, как нетрудно догадаться, позволяет вставить на web-страницу любой заранее подготовленный и сохраненный в виде файла рисунок. После выбора этой команды вам останется

указать, где именно хранится нужное графическое изображение и как называется соответствующий файл. Команда **Со сканера или камеры** позволяет добавлять в документ рисунки, полученные со сканера или цифровой камеры.

Команда **Создать фотоколлекцию** поможет вам при необходимости помещения на web-страницу целого набора фотографий или картинок. Выбрав эту команду, вы должны указать, какие именно рисунки будут демонстрироваться в коллекции, а также выбрать макет, в соответствии с которым указанные рисунки будут располагаться в вашей коллекции. При создании подобной коллекции в нее может быть добавлено любое число фотографий, кроме того, для каждой фотографии можно задать название и описание. Как правило, при создании подобных фотоколлекций используются так называемые **эскизы** — миниатюрное представление рисунка на web-странице, выполненное в виде гиперссылки на полноразмерную версию рисунка. Эскизы являясь средством, позволяющим ускорить загрузку web-страниц, содержащих большое количество рисунков и фотографий.

Ранее мы уже говорили о том, что в Microsoft FrontPage 2003 добавлена возможность вставки объектов Macromedia Flash. И эта замечательная возможность реализована в виде команды **Фильм в формате Flash**. Вам останется только указать местоположение и название Flash-ролика, а всю остальную работу за вас сделает FrontPage.

Команды **Создать рисунок** и **Автофигуры** позволяют поместить на web-страницу рисунок, созданный с помощью панели **Рисование**. На этой панели вы найдете инструменты для создания и оформления всевозможных прямоугольников, овалов, линий, стрелок, элементов блок-схемы и т.д. Команда **Объект WordArt** позволяет помещать на web-страницу текстовые эффекты. Например, для украшения документа можно создать объемную надпись с тенью.

И последняя команда — **Видеозапись** — позволяет вставить на web-страницу видеоклип. Естественно, в этом случае придется указать местоположение и название соответствующего файла.

Итак, добавим на домашнюю страницу сайта графический рисунок из

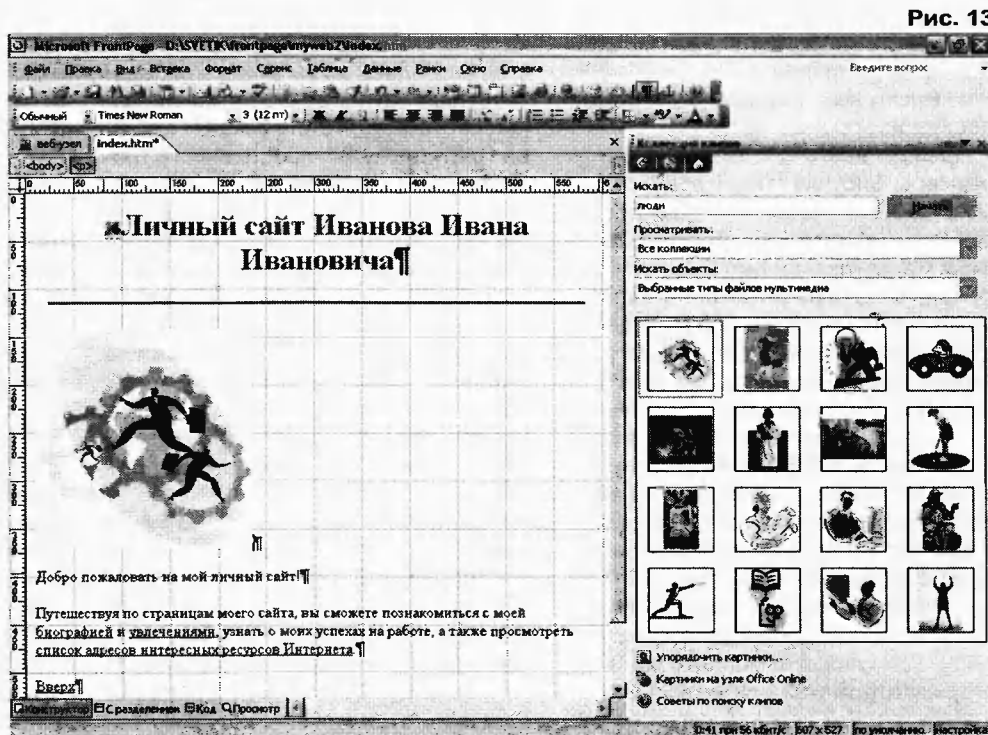


Рис. 13

Коллекции картинок (Microsoft). Установив текстовый курсор в новый пустой абзац и выполнив уже известную нам команду **Вставка/Рисунок/Картинки...**, отобразим на экране область задач **Коллекция клипов** для выбора требуемого рисунка. В поле **Искать** области задач **Коллекция клипов** необходимо ввести слово или фразу, описывающие нужный клип. Например, введите слово **"люди"** (рис.13). В случае необходимости можно ограничить поиск с помощью полей **Просматривать** и **Искать объекты** — поле **Просматривать** позволяет указать, в каких именно коллекциях требуется осуществлять поиск, а поле **Искать объекты** предлагает уточнить требуемый тип мультимедийных файлов. Оставьте в поле **Просматривать** вариант **Все коллекции**, а вот в поле **Искать объекты** уточните, что нас интересуют только графические изображения, для чего раскройте список этого поля и снимите флажки для фильмов и звуков. Вот теперь щелкните по кнопке **Начать**, чтобы приступить к поиску рисунков. Просмотрите результаты поиска и, выбрав понравившийся рисунок, щелкните по нему мышью для вставки на web-страницу. Вот и вся нехитрая технология добавления картинок на web-страницу.

Естественно, в любой момент вы можете удалить ненужные графические изображения. Для этого изображение необходимо выделить и нажать клавишу **Delete** на клавиатуре.

В том случае, если подобный поиск

не дал удовлетворительных результатов, можно просмотреть все имеющиеся клипы в коллекции картинок. Для этого следует в нижней части области задач **Коллекция клипов** выбрать команду **Упорядочить картинки...** При этом вначале может появиться окно **Добавление клипов в коллекцию**, в котором вам предложат провести систематизацию мультимедийных файлов. Чтобы не тратить на это время, щелкните по кнопке **Позже** этого диалогового окна. Вот сейчас вы попадете в окно просмотра Коллекции картинок (Microsoft). Заметьте, что клипы в этом окне разделяются на коллекции. К примеру, **Коллекции Microsoft Office** содержат файлы мультимедиа, поставляемые с приложениями Microsoft Office. Выбрав эту коллекцию, вы можете уточнить интересующую вас тематическую категорию, а дальше приступить к просмотру клипов, входящих в выбранную категорию. Для вставки на web-страницу понравившейся картинки достаточно просто перетащить с помощью мыши выбранный клип в нужное место открытого документа. Не забудьте после этого закрыть окно коллекции картинок.

Если же и этот вариант вам не помог, при наличии подключения к сети Интернет вы можете просмотреть содержимое коллекции картинок на веб-сайте Microsoft Office Online. Для этого предназначена команда **Картинки на узле Office Online** в нижней части области задач **Коллекция клипов**.

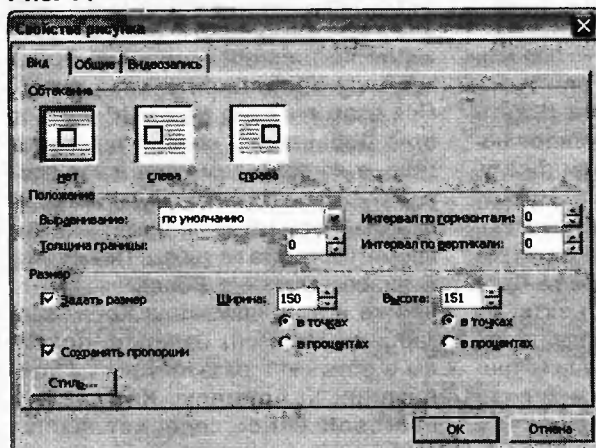
А сейчас поговорим о том, что можно сделать со вставленным на web-страницу рисунком.

4. Настройка параметров графических изображений

Прежде всего следует настроить размеры рисунка. Производится эта операция так же, как и во всех других офисных приложениях. Вначале щелчком мыши необходимо выделить рисунок, а затем, установив курсор мыши на один из маркеров выделения и нажав левую кнопку мыши, растянуть или уменьшить рамку рисунка. Если вы воспользуетесь угловыми маркерами, изменяться будут как высота, так и ширина выделенного объекта, боковые маркеры изменяют либо высоту, либо ширину объекта.

Все остальные параметры графических изображений настраиваются с помощью диалогового окна **Свойства рисунка**. Для вызова этого окна выделите рисунок и выполните команду **Формат/Свойства** или же выполните щелчок правой кнопкой мыши по рисунку и в появившемся контекстном меню выберите **Свойства рисунка...**

Рис. 14



На вкладке **Вид** диалогового окна **Свойства рисунка** (рис.14) находятся параметры, определяющие положение рисунка на web-странице и его размер. В разделе **Обтекание** при необходимости можно задать возможность обтекания рисунка текстом (в нашем случае установлен вариант **нет**, с тем чтобы рисунок остался в пустом текстовом абзаце). С помощью поля **Выравнивание** раздела **Положение** вы можете определить выравнивание текста относительно рисунка (правда, для того чтобы продемонстрировать действие всех значений этого поля, придется рядом с рисунком в том же абзаце ввести какой-либо текст). Поля **Интервал по горизонтали** и **Интервал по вертикали** позволяют указать размер (в пикселах) пу-

стых полей слева и справа от рисунка, а также над и под ним соответственно. Здесь же вы можете определить необходимость отображения границ рисунка. Для этого в поле **Толщина границы** нужно просто ввести требуемое значение ширины рамки рисунка (опять-таки в пикселах). Выше мы уже упоминали об изменении размеров графических изображений с помощью мыши. Однако настроить ширину и высоту рисунка можно и с помощью диалогового окна **Свойства рисунка**. Обратите внимание, что в последнем случае размеры рисунка могут задаваться либо в процентном отношении, либо точно в пикселах (точках).

Перейдите на вкладку **Общие** (рис.15). Здесь находятся параметры определения формата файла графического изображения, а также варианты альтернативного представления для рисунка. В поле **Рисунок** указано местоположение и имя файла добавленного на web-страницу рисунка. Чаще всего картинки из Коллекции картинок (Microsoft) представлены в формате WMF. И здесь следует иметь в виду, что для представления графических изображений на web-страницах используются форматы GIF, JPEG или PNG. Выбор формата для рисунка зависит от многих факторов. Например, формат GIF наилучшим образом подходит для изображений, в которых содержится небольшое количество разных цветов. У этого формата есть несколько преиму-

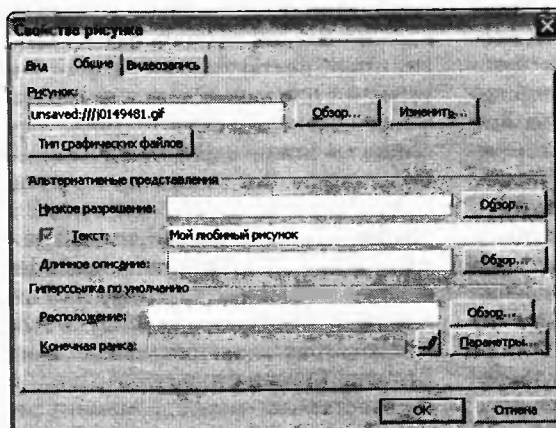
определения прозрачного фона рисунка). Кроме того, для формата GIF можно определить возможность загрузки с чередованием строк вместо стандартной прорисовки рисунка сверху вниз (так называемая чересстрочная загрузка рисунка). В результате рисунок, представленный в формате GIF с чередованием, сначала загружается с низким качеством, а затем качество рисунка постепенно улучшается. Таким образом, пользователь сможет увидеть изображение еще до того, как оно загрузится целиком.

Формат JPEG обычно используется для рисунков высокого качества, содержащих тысячи и миллионы цветов. Этот формат удобен тем, что, изменяя качество рисунка (от 1 до 100), можно управлять степенью сжатия файла и, следовательно, его размером. Понятно, что чем ниже качество, тем меньше размер файла изображения, и тем быстрее он будет загружаться. Кроме того, для этого формата можно определить число последовательных проходов, за которое рисунок отображается при загрузке в браузере, это позволяет создать эффект поэтапного улучшения качества рисунка при загрузке.

Формат PNG с недавних пор является альтернативой форматам GIF и JPEG. Он поддерживает прозрачность для рисунков, содержащих тысячи или миллионы цветов. Однако этот формат пока не очень широко распространен и поддерживается не всеми программами-браузерами.

В результате, поскольку стандартными форматами файлов изображений для web-страниц являются GIF, JPEG и PNG, перед добавлением рисунка на web-страницу необходимо с помощью

Рис. 15. графического редактора произвести преобразование рисунка в требуемый формат. Конечно же, подобная предварительная работа не всегда удобна и требует наличия на вашем персональном компьютере дополнительных программ. Если вы хотите этого избежать, вам на помощь придет все тот же Microsoft FrontPage. Мы уже обратили внимание, что в нашем случае рисунок, вставленный из Коллекции картинок (Micro-



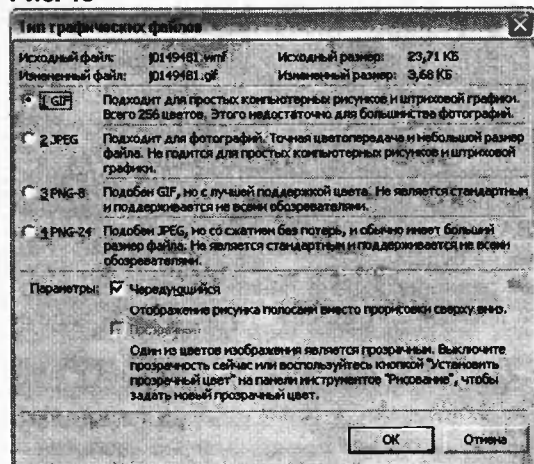
ществ. Во-первых, этот формат позволяет один из цветов изображения сделать прозрачным (согласитесь, это удобно в случае необходимости

soft), представлен в формате WMF. Даже если вы заранее не побеспокоились о преобразовании формата графического изображения, при сохране-



нии web-страницы Microsoft FrontPage предложит вам сохранить рисунок в папке сайта. При этом рисунки, использующие 256 или меньше цветов, будут автоматически преобразованы в формат GIF, а все остальные — в формат JPEG. Согласитесь, это очень удобная особенность программы. В данном случае Microsoft FrontPage частично реализует некоторые функции графического редактора. Если же вы сами хотите определить формат файла графического изображения, в диалоговом окне **Свойства рисунка** на вкладке **Общие** воспользуйтесь кнопкой **Тип графических файлов**. В появившемся диалоговом окне **Тип графических файлов** (рис.16) вы можете не только определить требуемый формат файла и настроить его параметры, но и провести небольшое исследование. Обратите внимание, что в этом окне отображается размер как исходного, так и измененного файла в выбранном вами формате. Таким образом, можно определить, в каком формате и с какими дополнительными параметрами ваш рисунок будет иметь минимальный размер и, следовательно, быстрее открываться при загрузке в браузер. Установите для нашей картинки параметры, указанные на рис.16, и подтвердите выбор параметров с помощью кнопки **ОК**. Обратите внимание, что после этого изменилась запись в поле **Рисунок**.

Рис. 16



На этой же вкладке можно определить варианты альтернативного представления для рисунка. В поле **Текст** раздела **Альтернативные представления** введите: **Мой любимый рисунок**. Этот альтернативный (или еще говорят замещающий) текст будет отображаться при наведении курсора мыши на рисунок в виде текстовой подсказки при просмотре web-страницы в браузере. В том случае, если

пользователь при просмотре вашей страницы в браузере откажется от загрузки рисунков, этот текст будет отображаться в рамке рисунка. Кроме текста, для рисунка можно задать в качестве альтернативы версию этого же рисунка с низким разрешением. Для этого используется поле **Низкое разрешение**. Однако предварительно в графическом редакторе необходимо создать версию рисунка с низким разрешением, не изменяя при этом его ширину и высоту, и сохранить его в виде отдельного файла. Далее в Microsoft FrontPage в поле **Низкое разрешение** диалогового окна **Свойства рисунка** задается отображение версии рисунка с низким разрешением в ходе загрузки оригинала. В этом случае рисунок с низким разрешением предполагается использовать в качестве временной замены для рисунка с высоким разрешением. Понятно, что таким образом время загрузки рисунка-оригинала не будет уменьшено, однако в процессе загрузки приятнее видеть хоть какой-то вариант рисунка вместо пустой области страницы.

Обратите внимание, что на этой же вкладке в разделе **Гиперссылка по умолчанию** для рисунка можно задать еще и ссылку на какой-либо ресурс. При щелчке по кнопке **Обзор...** этого раздела появится уже знакомое нам диалоговое окно создания гиперссылок. Хотя для создания гиперссылки можно воспользоваться и старым алгоритмом — выделить рисунок, затем выполнить команду **Вставка/Гиперссылка...** или воспользоваться кнопкой  панели инструментов **Стандартная** и указать ресурс назначения.

Вот практически и все, что касается настройки параметров графических изображений. Вкладка **Видеозапись** этого диалогового окна предназначена для дополнительной настройки вставленных на web-страницу видеоклипов. Не забудьте подтвердить применение выбранных параметров с помощью кнопки **ОК**.

С помощью инструмента  панели **Форматирование** задайте выравнивание абзаца с рисунком по центру. Сохраните изменения начальной страницы сайта, воспользовавшись инструментом  панели **Стандартная**. Т.к. в сохраняемом документе появился новый внедренный объект, Microsoft FrontPage выведет на экран

диалоговое окно **Сохранение внедренных файлов**. Проследите за тем, чтобы рисунок был помещен в папку **images** нашего сайта, и щелчком по кнопке **ОК** подтвердите сохранение графического изображения в выбранном нами формате.

Нам осталось протестировать web-страницу с помощью программы-браузера. Для этого выполните команду **Файл/Просмотреть в обозревателе** или воспользуйтесь инструментом  панели **Стандартная**. Установив курсор мыши на рисунок, убедитесь в наличии текстовой подсказки. Проконтролируйте внешний вид страницы в режиме отключения графики. Отключите загрузку рисунков (**Сервис/Свойства обозревателя.../вкладка Дополнительно**, в блоке **Мультимедиа** снимите флажок **Отображать рисунки/ОК**), обновите текущую страницу (кнопка **Обновить** на панели инструментов) и убедитесь в том, что альтернативный текст будет отображаться в рамке рисунка. Не забудьте после этого завершить работу с браузером и вернуться в Microsoft FrontPage.

Закройте начальную страницу нашего сайта с помощью команды **Файл/Закрыть** или воспользовавшись кнопкой  в заголовке окна документа. В качестве задания проиллюстрируйте web-страницу **hobby.htm**, посвященную вашим увлечениям, рисунками из Коллекции картинок (Microsoft). Определите размеры рисунков и их положение на web-странице по своему усмотрению, задайте формат файлов графических изображений, определите для них альтернативный текст. Сохраните web-страницу и протестируйте в браузере. Если у вас в наличии есть своя отсканированная фотография, с помощью команды **Вставка/Рисунок/Из файла...** поместите ее на web-страницу **biography.htm**. Настройте параметры фотографии, сохраните web-страницу и просмотрите в браузере.

О дополнительных возможностях по работе с графическими изображениями, предоставляемыми Microsoft FrontPage, мы поговорим в следующем раз. В заключение нашего сегодняшнего занятия выполните

5. Завершение работы с Microsoft Office FrontPage 2003 с помощью команды **Файл/Выход** или кнопки  строки заголовка программы.

(Продолжение следует)



Ю.СИЛКОВИЧ, Б.КИСЕЛЕВ,
г.Минск, БГАТУ.

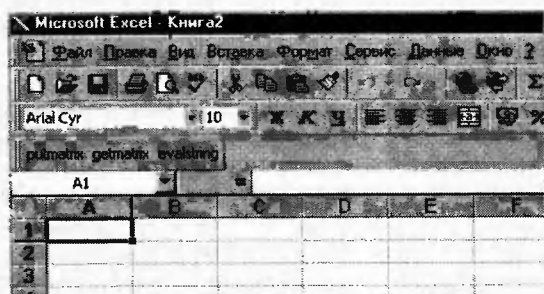
MATLAB. ИНТЕГРАЦИЯ С ЭЛЕМЕНТАМИ MS Office

MATLAB имеет возможность интегрироваться с редактором MS Word и электронными таблицами MS Excel. Первая возможность используется для написания интерактивных документов, так называемых М-книг, которые могут содержать все элементы редактора Word, а также команды MATLAB и результаты их выполнения. При этом команды активизируются прямо из документа (М-книги), и результат помещается также в документ. М-книги — очень удобный инструмент для создания учебных пособий и для написания научно-технических отчетов. Как выполнить интеграцию MATLAB с Word и создать М-книгу, подробно описано в [1]. В новых версиях MATLAB процесс интеграции упростился — при стандартном расположении программного продукта достаточно в командном окне ввести **notebook -setup**, а затем выбрать версию редактора.

В данной статье мы рассмотрим интеграцию MATLAB с электронными таблицами EXCEL [2]. Это позволяет использовать преимущества EXCEL для ввода исходных данных и оформления результатов расчетов, особенно при их больших объемах, когда использование командной строки MATLAB или текстовых файлов менее удобно. Кроме того, значительную часть расчетов можно проводить в электронных таблицах, и только при решении сложных задач (например, построении трехмерных графиков, решении дифференциальных уравнений, решении задач оптимизации большой размерности и т.д.), для которых недостаточно средств EXCEL, использовать всю мощь математического пакета MATLAB. Для интеграции служит программное средство EXCEL Link.

Чтобы установить пакет расширения EXCEL Link, необходимо запустить EXCEL, открыть пункт меню **Сервис/Настройка...**, выбрать кнопку **Обзор...** и в появившемся диалоговом окне указать файл надстройки: **C:\MATLAB6p5\toolbox\exclink\exclink.xls**, где **MATLAB6p5** — имя папки, в которой установлен MATLAB. Установка производится один раз, после этого надстройка запускается автоматически при каждом запуске EXCEL (в дальнейшем ее можно отключать и подключать соответствующим флажком в списке имеющихся надстроек при помощи того же пункта меню). При этом в фоновом режиме запускается командное окно MATLAB, а в EXCEL появляется дополнительная панель инструментов с тремя кнопками (рис.1):

Рис. 1



- **putmatrix** — передача данных в MATLAB. Для этого необходимо выделить требуемый диапазон ячеек и после нажатия данной кнопки в появившемся диалоговом окне ввести имя переменной MATLAB, в которую будут переданы данные. При этом пустые ячейки передаются как нулевые значения;

- **getmatrix** — считывание данных из MATLAB. Для этого необходимо перейти в ячейку, которая будет являться левым верхним углом матрицы, и после нажатия данной кнопки в появившемся диалоговом окне ввести имя переменной MATLAB, из которой будут считаны данные;

- **evalstring** — ввод команды для выполнения в MATLAB точно таким же образом, как если бы она была набрана в его командном окне.

Кроме того, EXCEL Link также имеет 11 функций, которые можно вводить в ячейки наравне со встроенными функциями EXCEL. Основные функции обработки данных практически дублируют кнопки панели инструментов:

- **MLPutMatrix**("имя переменной"; диапазон ячеек) — передача данных из диапазона ячеек в переменную MATLAB;

- **MLAppendMatrix**("имя переменной"; диапазон ячеек) — добавление данных из диапазона ячеек к данным, уже хранящимся в переменной MATLAB. Способ добавления (строки или столбцы) определяется автоматически по размерности добавляемой матрицы. Если можно добавить как строки, так и столбцы, то добавляются строки;

- **MLGetMatrix**("имя переменной"; "ячейка") — считывание данных из переменной MATLAB в диапазон ячеек, левым верхним углом которых является указанная в качестве параметра функции ячейка в текстовом виде;

- **MLEvalString**("команда MATLAB") — выполнение команды MATLAB точно таким же образом, как если бы она была набрана в его командном окне;

- **MLDeleteMatrix**("имя переменной") — удаление переменной из рабочего пространства MATLAB

Отметим, что объединение матриц и удаление их из рабочего пространства можно с таким же успехом произвести и функцией **MLEvalString** с помощью соответствующих команд MATLAB, не используя функции **MLAppendMatrix** и **MLDeleteMatrix**. Таким образом, для расчетов можно обойтись только тремя функциями, соответствующими трем кнопкам панели инструментов. Остальные функции служат для управления запуском MATLAB (при автоматическом запуске они не используются), а также для использования в VBA (Visual Basic for Application), поэтому рассматривать их не будем.

Рассмотрим решение задачи линейного программирования [3] при помощи EXCEL Link. Пусть модель задачи имеет вид:

$$S = (3x_1 + 2x_2)1000 \rightarrow \max, \text{ которое приводим к виду } S = -3x_1 - 2x_2 \rightarrow \min.$$

Ограничения:

$$x_1 + 2x_2 = 6;$$

$$2x_1 + x_2 = 8;$$

$$x_2 - x_1 = 1;$$

$$x_2 = 2.$$

Также заданы условия неотрицательности параметров:

$$x_1 = 0; \quad x_2 = 0.$$

Запускаем EXCEL, в ячейки A2:A3 вводим коэффициенты целевой функции. Выделяем этот диапазон ячеек, нажимаем на панели инструментов кнопку **putmatrix** и в появившемся диалоговом окне вводим имя переменной MATLAB — **S**, в которую передаем матрицу коэффициентов. Аналогичным образом вводим в ячейки A5:B8 и C5:C8 соответственно матрицы коэффициентов и свободных чле-



нов неравенств и передаем в переменные MATLAB **A** и **b**. Нажимаем кнопку **evalstring** и в диалоговом окне вводим команду **lb=zeros(2,1)**, а затем аналогично вводим команду **[x, Sval]=linprog(S, A, b, [], [], lb)**. Для получения результатов переходим, например, в ячейку **A13**, нажимаем кнопку **getmatrix** и в диалоговом окне указываем имя переменной **x**. Далее переходим в ячейку **A16** и с помощью той же кнопки указываем имя переменной **Sval**. В результате в ячейках **A13:A14** получаем решение, а в ячейке **A16** — значение целевой функции. Вместо использования кнопки **evalstring** можно после передачи данных, не закрывая EXCEL, перейти в командное окно MATLAB, выполнить нужные команды и, не закрывая командное окно, вернуться обратно для считывания результатов.

Аналогичную картину можно получить при использовании описанных выше функций EXCEL Link. Для этого в ячейки вводим следующие выражения.

Передача матриц **S, A, b** в MATLAB:

E2 =MLPutMatrix("S";A2:A3);

E5 =MLPutMatrix("A";A5:B8);

E6 =MLPutMatrix("b";C5:C8);

Создание нулевой матрицы нижних границ переменных:

E7 =MLEvalString("lb=zeros(2,1)");

Решение задачи линейного программирования:

E9 =MLEvalString("[x,Sval,exitflag,output,lambda] = linprog(S,A,b,[],[],lb)");

Получение кода завершения расчетов в ячейку **E12**:

E10 =MLGetMatrix("exitflag";"E12");

В зависимости от кода завершения выводим в ячейке **A11** соответствующее текстовое сообщение:

A11 =ЕСЛИ(E12>0;"Решение найдено"; ЕСЛИ(E12=0;"Предельное число итераций"; "Решение не найдено"));

Получаем вычисленные значения переменных и целевой функции:

E13 =MLGetMatrix("x";"A13");

E14 =MLGetMatrix("Sval";"A16");

Так как EXCEL Link позволяет производить обмен только матрицами, а не структурами, которой является переменная **lambda**, то присваиваем значения ее полей **ineqlin** и **lower** соответственно переменным **ineq** и **low** и получаем значения множителей Лагранжа для ограничений неравенствами и нижней границы переменных:

E15 =MLEvalString("ineq=lambda.ineqlin");

E16 =MLGetMatrix("ineq";"C16");

E17 =MLEvalString("low=lambda.lower");

E18 =MLGetMatrix("low";"C13");

Нулевые значения в данных ячейках (кроме **A11**) указывают на то, что функции выполнены без ошибок. Другие значения будут указывать на произошедшую в процессе вычислений ошибку. В результате получим решение, соответствующее табл.1.

Казалось бы, все хорошо считается. Но попробуем изменить исходные данные и посмотрим на результат. Никаких изменений не происходит, результат не пересчитывается. Если же попробовать закрыть файл, а затем повторно его открыть, то мы увидим, что в ряде ячеек с формулами нули изменились на сообщения об ошибках типа **#COMMAND!** или **#NONEXIST!**, которые означают, что возникла ошибка при выполнении команды или попытке получить значение несуществующей переменной. При этом результаты предыдущего расчета не изменятся и останутся теми же.

Дело в том, что EXCEL производит пересчет, основываясь на зависимостях между ячейками. Если, например, мы изменим значение в ячейке **A2**, то будет произведен пере-

Табл. 1

	A	B	C	D	E
1	Коэффициенты целевой функции				
2	-3				0
3	-2				
4	Матрица коэффициентов ограничений		Свободные члены ограничений		
5	1	2	6		0
6	2	1	8		0
7	-1	1	1		0
8	0	1	2		
9					0
10	Результаты решения				
11	Решение найдено		Множители Лагранжа:		
12	Значения переменных		для нижних границ переменных		
13	3,333333333		1,29691E-13		
14	1,333333333		3,07929E-12		
15	Значение целевой функции		для неравенств		
16	-12,66666667		0,333333333		
17			1,333333333		
18			3,66068E-14		
19			1,05649E-12		

счет формулы в ячейке **E2**, которая явно зависит от нее, и в MATLAB будет отправлено новое значение матрицы **S**. Однако функция в ячейке **E9**, которая производит расчет, имеет своим аргументом текстовую константу и в явном виде не зависит от ячейки **E2**, поэтому не пересчитывается. Аналогично не пересчитываются и остальные формулы. При повторном открытии файла EXCEL производит расчет формул в **E9**, которая не зависит от других ячеек, а только потом — в **E2, E5** и **E6**, которые зависят от других ячеек. Поскольку в этом случае расчет в MATLAB производится до передачи туда исходных данных, то мы получаем сообщения об ошибках.

Чтобы разрешить данную проблему, нужно установить связь между этими ячейками искусственным путем. В качестве одного из вариантов такой связи можно предложить использовать функцию ЕСЛИ следующим образом. Расчет по формуле в ячейке **E9** можно производить только после того, как в MATLAB будут переданы все исходные данные, т.е. в ячейках **E2, E5:E7**, отвечающих за эту передачу, будет находиться значение 0. Введем в ячейку **E8** формулу: **=И(E2=0; E5=0; E6=0; E7=0)**. Если все исходные данные были переданы успешно, то в этой ячейке будет находиться значение ИСТИНА, в противном случае — ЛОЖЬ. В ячейке **E9** изменим формулу на следующую:

=ЕСЛИ(E8;MLEvalString("[x,Sval,exitflag,output,lambda] = linprog(S,A,b,[],[],lb)");#ЗНАЧ!);

Теперь формула явно зависит от ячейки **E8**, которая, в свою очередь, зависит от ячеек, передающих в MATLAB исходные данные, поэтому при их изменении выражение в **E9** будет корректно пересчитываться. Если возникнет какая-либо ошибка при передаче данных, наша формула (в **E9**) вместо попытки произвести расчеты с несуществующими переменными будет возвращать значение ошибки EXCEL (**#ЗНАЧ!**). Аналогичным образом установим зависимости и для остальных ячеек с формулами.

Кроме того, EXCEL Link имеет еще одну особенность. Функция **MLGetMatrix** реализована таким образом, что по умолчанию значения принимаемых из MATLAB матриц записываются в ячейки активного листа. При выполнении расчетов с использованием нескольких листов это может приводить к тому, что при пересчете находящиеся на неактивных листах функции **MLGetMatrix** будут записывать полученные матрицы на совершенно другой активный лист рабочей книги, затирая имеющиеся там данные. Для устранения этого в функции **MLGetMatrix** необходимо всегда записывать адрес ячейки с явным указанием имени листа (например, "Лист1!A13", а не просто "A13", если рабочий



лист имеет имя Лист1). Точно таким же образом можно обеспечить вывод результатов на любой лист, например, разместить исходные данные на одном листе, расчетные формулы — на втором, а результаты вывести на третьем листе.

С учетом вышеизложенного изменим расчетные формулы в следующих ячейках:

```
E8 =И(E2=0;E5=0;E6=0;E7=0),
E9 =ЕСЛИ(E8;MLEvalString("[x, Sval, exitflag, output,
lambda] = linprog(S, A, b, [], [], lb)"); #ЗНАЧИ),
E10 =ЕСЛИ(E9=0;MLGetMatrix("exitflag"; "Лист1!E11");
#ЗНАЧИ),
E12 =ЕСЛИ(E10=0;E11; #ЗНАЧИ),
E13 =ЕСЛИ(E9=0;MLGetMatrix("x"; "Лист1!A13");
#ЗНАЧИ),
E14 =ЕСЛИ(E9=0;MLGetMatrix("Sval"; "Лист1!A16");
#ЗНАЧИ),
E15 =ЕСЛИ(E9=0;MLEvalString("Ineq=lambda.ineqlin");
#ЗНАЧИ),
E16 =ЕСЛИ(E15=0;MLGetMatrix("ineq"; "Лист1!C16");
#ЗНАЧИ),
E17 =ЕСЛИ(E9=0;MLEvalString("low=lambda.lower");
#ЗНАЧИ),
E18 =ЕСЛИ(E17=0;MLGetMatrix("low"; "Лист1!C13");
#ЗНАЧИ).
```

Таблица примет иной вид (табл.2).

Табл. 2

	A	B	C	D	E
Кoeffициенты целевой функции					
-3					0
-2					
Матрица коэффициентов ограничений			Свободные члены ограничений		
1	2	6			0
2	1	8			0
-1	1	1			0
0	1	2			ИСТИНА
Результаты решения					0
Решение найдено		Множители Лагранжа:			1
Значения переменных		для нижних границ переменных			1
3,333333333		1,29691E-13			0
1,333333333		3,07929E-12			0
Значение целевой функции		для неравенств			0
-12,66666667		0,333333333			0
		1,333333333			0
		3,66068E-14			0
		1,05649E-12			

Обратите внимание на ячейки E10 и E12. Если бы мы оставили все как прежде, т.е. в E10 написали бы: "...Лист1!E12"..., а в E12 ничего не заносили, то при аварийном завершении программы **linprog** (например, при неправильно сформированных данных) в E12 осталась бы единица, что говорило бы об успешном решении задачи.

Попробуем изменять исходные данные. Как видим, теперь никаких проблем при пересчете не возникает. Конечно, такую простую задачу можно с успехом решить и при помощи поиска решения EXCEL (правда, без столь широкой дополнительной информации о ходе и результатах решения). А если количество переменных больше 200 (максимально возможного в EXCEL)?

Если раздражают маленькие числа (меньше заданной по умолчанию точности (1.e-8)) в множителях Лагранжа, то соответствующие ячейки можно отформатировать так, чтобы в них в подобных ситуациях были нули.

В заключение дадим еще несколько рекомендаций.

1. Все используемые переменные MATLAB находятся в одном и том же рабочем пространстве, поэтому не сле-

дует использовать одни и те же имена переменных для различных целей, если даже они находятся на разных листах EXCEL.

2. Не следует в командах MATLAB изменять значения переменных, передаваемых из EXCEL. В противном случае, при пересчете могут быть использованы их новые значения, что приведет к сообщению об ошибке или неверному результату.

3. При больших объемах вычислений, в зависимости от мощности компьютера, пересчет может занимать более или менее продолжительное время. В таких случаях можно отключить автоматический пересчет и при необходимости производить его нажатием кнопки F9.

4. Не забывайте, что в функции **MLGetMatrix** указывается верхний левый угол, а сама матрица может оказаться достаточно большой и затереть данные, находящиеся в ячейках справа и снизу. Старайтесь правильно оценить размеры возвращаемой матрицы и не размещать там другие данные.

5. Лист EXCEL имеет "всего" 256 столбцов (ограничение в 65536 строк менее актуально). При необходимости ввода большей матрицы ее части можно разместить на нескольких листах, передать их в MATLAB как, например, матрицы A1, A2, A3..., а затем объединить командой **A=[A1 A2 A3 ...]**. Или же, при небольшом количестве строк, ввести ее в транспонированном виде, а затем произвести обратное транспонирование командой MATLAB. Аналогично, не представляет труда разбить большую матрицу результатов на подматрицы соответствующими командами MATLAB и разместить их на нескольких листах EXCEL.

Упражнения

При выполнении упражнений рекомендуем сначала выполнить все операции в EXCEL, а затем попробовать (особенно во втором упражнении) часть операций выполнить в EXCEL, а часть — в командном окне MATLAB.

С помощью предложенных выше средств:

1. Решить упражнения 1 и 2 из работы [3].
2. Построить поверхность [4] $z = x^2 + y^2$ на области $x, y \in [-3; 3]$

Вопросы для самопроверки

1. Как установить пакет расширения EXCEL Link?
2. Какими средствами можно передавать данные из EXCEL в MATLAB и обратно?
3. Какими средствами можно передавать команды из EXCEL для их выполнения в MATLAB?
4. Почему не всегда пересчитываются ячейки EXCEL, связанные с MATLAB, при изменении данных других ячеек?
5. Как обойти проблему вопроса 4?

Литература

1. Киселев Б. "Живые" книги в MATLAB. — Радиомир. Ваш компьютер, 2001, №10, С.4.
2. Ануфриев И.Е. Самоучитель Matlab 5.3/6.x. — СПб.: БХВ-Петербург, 2002.
3. Киселев Б., Силкович Ю. MATLAB. Линейное программирование. — Радиомир. Ваш компьютер, 2004, №№9, 10.
4. Киселев Б. MATLAB. Визуализация результатов вычислений. — Радиомир. Ваш компьютер, 2003, №8, С.4.

Е.ЛЕБЕДЕВ,
г.Череповец.

Знакомство с PEBrowse Pro Interactive

В настоящее время в мире программирования и исследований кода мне известны три отладчика:

- старый добрый SoftICE (www.pimaga.com), который зарекомендовал себя как полнофункциональный и мощный программный продукт для отладки приложений;

- TRW2000 — это аналог SoftICE'a, имеющий ограниченный набор команд, но зато удобный запуск. В настоящее время этот проект "мертв";

- OllyDebug — тоже хороший отладчик, который имеет набор всех необходимых функций для отладки программного обеспечения.

казался слегка неудобным, особенно мешало нагромождение дочерних окон. После SoftICE'a было очень непривычно работать, т.к. то и дело при возникновении ситуации, когда требовалось просмотреть содержимое нужного регистра, хотелось набрать в командной строке — `d eax`.

PEBrowse (PB), как я уже сказал, имеет многооконный интерфейс, состоящий из следующих окон:

- Debug Log — окно истории отладки, в которое отладчик записывает все отладочные события;
- Registers — окно, отображающее состояние регистров;

окне отображаются все ошибки или ошибочные/критические ситуации, возникшие при отладке/работе программы.

Авторы PB добавили в свое детище очень полезную функцию — справочник по командам процессора (OpCodes). Держать опкоды всех команд процессора в голове нереально, поэтому в справочнике опкодов можно посмотреть 16-ричное значение любой команды (например: AND — 22h). Также при вызове контекстного меню на выбранной строке кода, можно получить справку по выделенной команде!

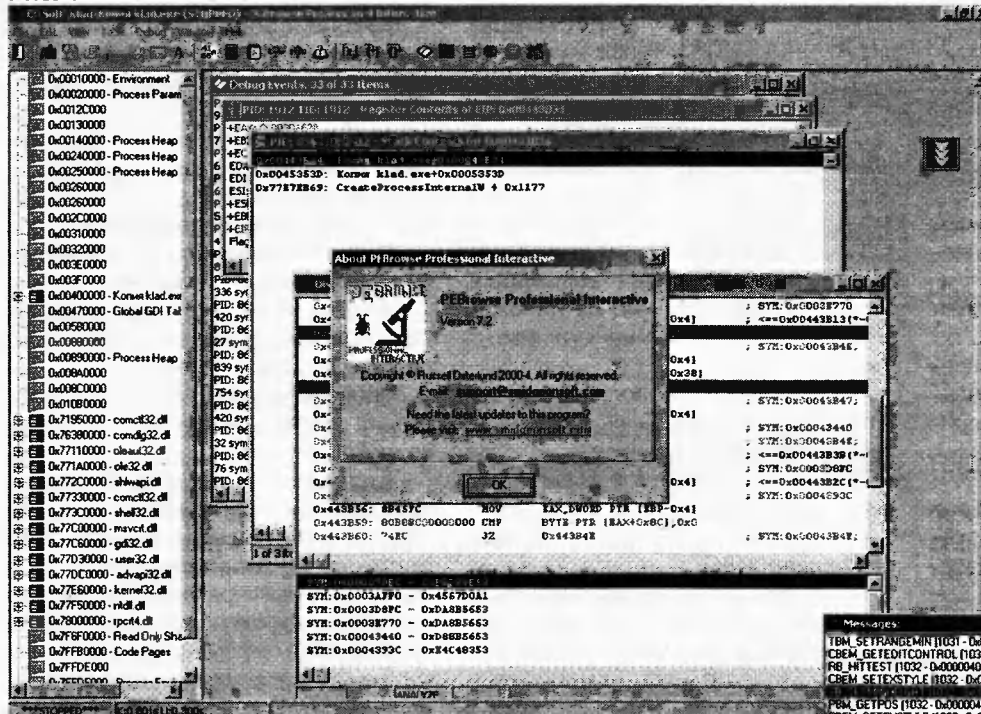
При желании можно посмотреть все сообщения от отлаживаемой программы,

узнать имя объекта, который послал сообщение в десятичном и 16-ричном формате. Для этого требуется включить окно сообщений — Tools/Messages (рис.2).

В строке состояния отображается информация о текущем режиме работы программы (работает/прервана) и комментарии к некоторым командам. Например, когда выполняется команда ветвления (условный переход), в статусной строке отображается состояние данной команды — осуществится условный переход или нет.

PB имеет свой встроенный калькулятор инженерного типа (рис.3), который, помимо стандартных арифметических команд в трех системах счисления,

Рис. 1



Однажды, "ползая" по Всемирной паутине, я наткнулся на сайт www.smidgeonsoft.com, на котором были представлены всевозможные программные продукты. Ознакомившись с содержимым сайта, я уже хотел было уходить, но обратил внимание на программу PEBrowse. Порывав по сайту с целью нахождения описания возможностей и функций программы, я так ничего и не нашел. Пришлось скачать и разбираться самому.

Установка и загрузка программы не составила каких-либо трудностей, и передо мной появился многооконный интерфейс отладчика, представленный на рис.1.

На первый взгляд, интерфейс по-

- Stack — показывает текущее состояние стека;

- Breakpoints — показывает установленные на данный момент точки останова в программе;

- Disassembly — окно с исходным текстом программы на языке Ассемблер.

В окошке, расположенном слева, выводится информация о всех модулях и библиотеках, используемых программой.

PB, в отличие от SoftICE'a, имеет "нормальный" загрузчик, который стабильно работает и останавливает отлаживаемую программу строго на точке входа (Entry Point).

Для контроля корректности работы отлаживаемой программы служит окно Error Codes (Tools/Error Codes). В этом

Рис. 2

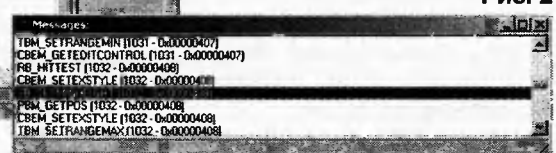
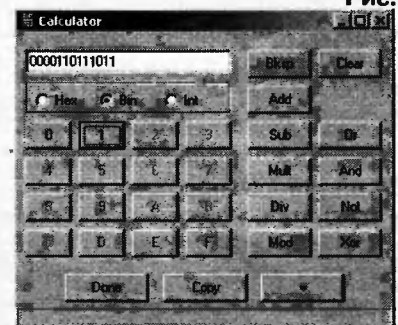


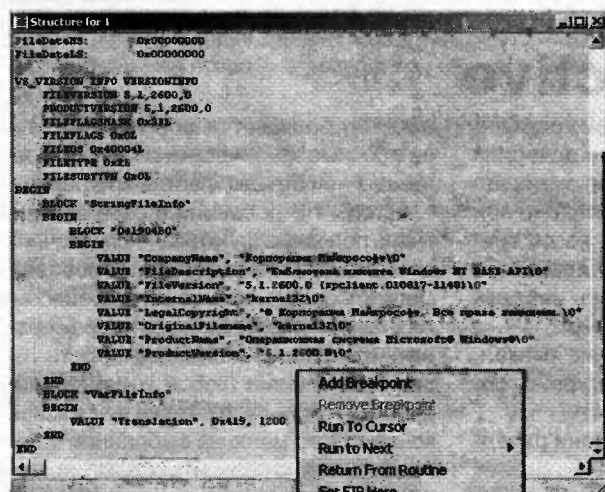
Рис. 3



может выполнять ряд логических (И, ИЛИ) и ассемблерных команд (Add, Sub).

При работе с ASCII-строками может пригодиться окно "Hex Table". Hex Table

Рис. 4



содержит значения всей ASCII-таблицы в ASCII- и Hex-форматах.

После достаточно продолжительного общения с РВ я по достоинству оценил возможность просмотра ресурсов файла прямо из отладчика. В режиме «По умолчанию» ресурсы посмотреть нельзя, а если включить «View/Index Detail», то окно, расположенное слева, заметно преобразится, и, помимо адресов в памяти, станут доступны дополнительные опции. Теперь по каждому адресу памяти можно будет посмотреть соответствующие ему элементы и структуру каждого элемента. Например, структура ресурса «Version» имеет вид, соответствующий рис.4.

Вдобавок, при просмотре любого доступного ресурса мы можем поставить на него точку останова в памяти. Эта возможность делает отладку более тонкой и доступной.

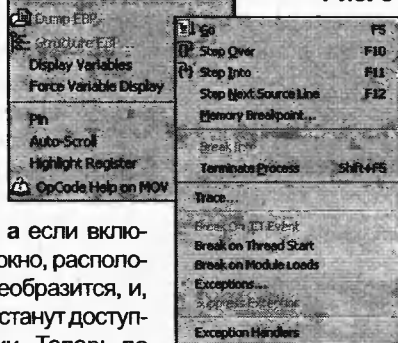
Безусловно, просмотр ресурсов возможности РВ не ограничиваются. Мы можем посмотреть всю таблицу импорта и экспорта файла, атрибуты нужной секции.

Особое место в программе занимает функция «Dump» (сделать «слепок» (снимок) памяти). Если в SoftICE я пользовался только стандартным «дампом», то в РВ функция «дампа» есть везде! Меня это очень удивило и порадовало. Здесь мы можем «дампить» все подряд, любой участок памяти, а также посмотреть его в реальном времени.

Особого внимания заслуживает контекстное меню в окне дизассемблированного кода (рис.5):

Рис. 5

Рис. 6



листинге;

- Run to Next — выполнить код до ближайшей команды «call» или условного/безусловного перехода;
- Go to... — перейти к указанному смещению в коде;
- Auto-Scroll — автопрокрутка кода при исполнении;
- Highlight Register — включить подсветку определенных регистров. Данная функция очень полезна, когда требуется отследить все изменения в нужных регистрах;
- OpCode Help on MOV — в данном случае эта функция позволяет посмотреть

описание команды «MOV» в справочнике опкодов;

Контекстное меню варьируется в зависимости от выбранной строки кода, поэтому существуют еще дополнительные функции:

- Disassembly... — показывает содержимое подпрограммы (call);
- Dump — получить код процедуры (call) из памяти.

Раздел меню «Debug» включает в себя функции управления отладкой программы (рис.6):

- Go — «отпустить» программу в «свободное плавание»;
- Step Over — выполнить команду без захода внутрь. Например, при выполнении команды «call» отладчик просто выполнит ее и не будет заходить внутрь процедуры;
- Step Into — выполнить команду с захождением внутрь;
- Step Next Source Line — непонятная команда, дословно означает «Перейти в пошаговом режиме к следующей строке окна кода»;
- Memory Breakpoint — поставить точку останова на участок памяти;
- Terminate Process — прервать работу отлаживаемой программы;
- Trace... — включить пошаговое выполнение программы до указанного адреса;
- Break on Thread Start — прервать работу программы при старте первой попавшейся «ветви» (thread);
- Break on Module Loads — прервать работу программы при первой загрузке любого модуля;
- Exceptions... — прерваться при возникновении исключений (exceptions). Например, прервать выполнение программы при возникновении ситуации деления на ноль или при переполнении стека.

Подводя итог, хотелось бы отметить, что PEBrowse Pro Interactive — это очень качественный продукт на рынке программного обеспечения, который, надеюсь, в скором времени составит реальную конкуренцию «детищу» фирмы Numega.

С.РЮМИК,
г.Чернигов.

Интернет-калейдоскоп, freeware #12

Мы не столько нуждаемся в помощи друзей,
сколько в уверенности, что мы ее получим.

Демокрит

Мудрое изречение древнегреческого философа, вынесенное в эпитафию, могло бы стать визитной карточкой или девизом Интернета. Действительно, человек, приобщенный к всемирной Сети, чувствует себя гораздо увереннее, спокойнее, защищеннее. Он знает, что в любой момент может получить ответ на вопрос, запросить информацию, подискутировать на

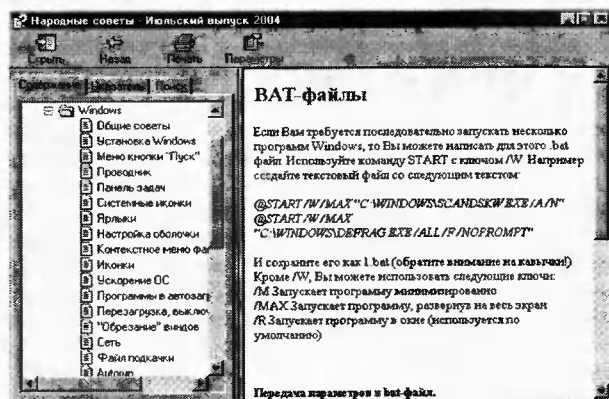
форуме. Другое дело, как отделить «зерна от плевел», чтобы умная голова не стала копилкой многих, но, увы, не совсем нужных знаний...

Народные советы и трюки

Есть категория людей, которые по своему характеру любят «выжимать» из имеющихся у них вещей максимум возможностей. Если это автомобилист, то свой

«Запорожец» он будет постепенно превращать в «Мерседес». Если компьютерщик, то не успокоится, пока его процессор не станет работать на 10% быстрее, чем у друзей. В последнем случае лишними окажутся советы из файла <http://winchanger.narod.ru/sovety.zip>, 183 Кб (авторы И.Чеботарев, А.Климов). Разделы советов: Windows-9x/XP (рис.1), FAR, Internet, Office, Hardware.

Рис. 1



Файл оформлен как справка Windows с расширением *.chm, и обновляется 1 раз в 2 месяца. Небольшой нюанс. Открывая его следует в папке, имя которой не содержит букв кириллицы. Свежая on-line-версия файла доступна на сайте <http://www.webhowto.ru>, справка по реестру Windows — на сайте <http://winchanger.narod.ru>.

Вот один из советов. Предположим, вы хотите еженедельно получать по e-mail копии какой-либо страницы сайта (html, shtml, htm), чтобы следить за обновлениями. Для начала надо послать письмо с пустой темой по адресу www@web2mail.com, дожидаться прихода файла помощи. Далее, на тот же самый e-mail послать письмо с темой, например, subscribe@radio-mir.com weekly. Теперь каждую неделю вам будут присылать заглавную страницу сайта журнала "Радиомир". Служба рассылки ведет себя деликатно, письма присылают только тогда, когда за неделю произошло какое-либо изменение на сайте.

Совет дельный, но на практике в августе 2004 года он не работал! Возможно, летний отпуск, финансовые трудности, временное прекращение работы или даже блокировка адресов с отдельных регионов, например, с Рунета.

Диктанты и тесты

Парадоксально, но факт — количество ошибок у двоечников на диктантах по русскому языку 50 лет назад и сейчас примерно одинаково. Знание "великого и могучего" до сих пор для многих остается камнем преткновения. Исправить положение помогают тренажеры, например, "Словарный диктант" Евгения Банникова (рис.2), <http://www.team9a.web.ug.ru/> (<http://www.listsoft.ru/programs/15883/?fileid=1>, файл slovar.rar, 38 Кб).

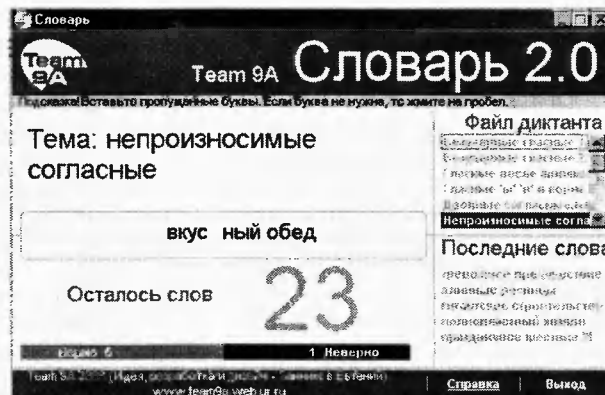
В программе имеются 6 встроенных тестов и возможность создания соб-

ственных диктантов объемом до 500 слов. Для нормальной работы требуется библиотека Runtime среды Visual Basic 6.0 (<http://download.microsoft.com/download/vb60pro/Redist/sp5/WIN98Me/EN-US/vbrun60sp5.exe>, 1019 Кб). Кстати, она может понадобиться и для других программ.

Альтернативный вариант проверки знаний грамматики — созда-

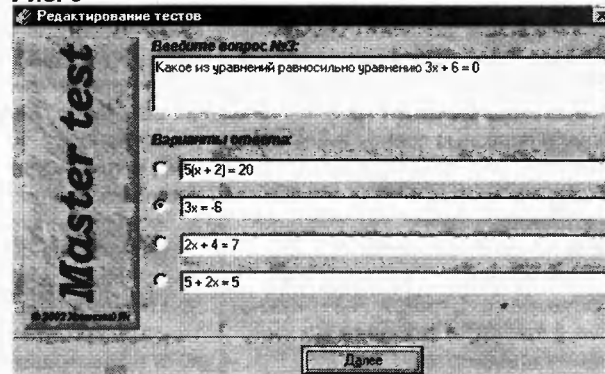
ние теста с ответами на выбор, как в игре "О, счастливчик!". Такую возможность предоставляет программа "Master Test 1.0", автор — Ян Хованский (<http://www.softportal.com/freesoftware/1368/master-test>, файл mt.zip, 171 Кб).

Рис. 2



Это универсальный составитель тестов, который может использоваться для проверки знаний по любым школьным (рис.3) и вузовским пред-

Рис. 3



метам, на викторинах, играх, КВН и даже на собеседованиях при приеме на работу. Количество вопросов — 1...90000, число вариантов ответов — 1...6.

Имеется ограничитель времени обдумывания и возможность сопоставления графического рисунка варианту ответа. Результат показывается в процентном и числовом виде.

Чем измерить талант?

Оказывается, килограммами! Например, один талант аттический равен 26,2 кг, а талант ассирийский — 32,6 кг. Речь идет о старинных мерах веса (греч. talanton — вес), перечень которых содержится в программе "Преобразователь мер", версия 5.1.2, автор — Сергей Зависляк, <http://invandy.nm.ru/soft/measure/> (<ftp://listsoft.ru/pub/10489/measure.zip>, 187 Кб).

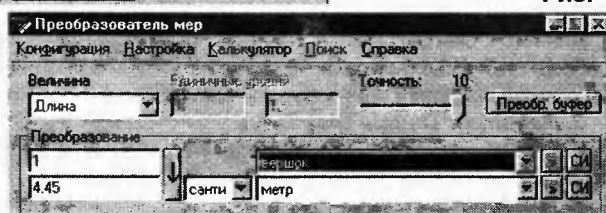
Вся коллекция мер (800 наименований) разбита на 22 категории, в частности, длина, вес, объем, скорость и т.д. Внутри каждой категории можно преобразовать одну меру в любую другую с высокой точностью (рис.4). Наряду с

современной системой СИ, приводятся малоизвестные уникальные меры разных стран мира, например, инглиш, кавальерия, шкалик, мешок. В последнем случае различают мешки бразильские, бельгийские, английские, русские. Характерно, что по объему русский мешок на 4% меньше английского.

Желающие могут присылать описания отсутствующих в программе мер, в частности, в коллекцию пока не вошли такие близкие компьютерщикам названия как флопсы, боды, MIPS.

Ключевые слова для поиска: "Народные советы по Windows", "Справка по реестру", "Словарный диктант", "Master Test", "Преобразователь мер". Перечисленные слова не зря заключены в кавычки — это помогает поисковому серверу находить полное сочетание слов.

Рис. 4





Локальные сети под DOS —

Р.КАРПАЧ,

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

Там чудеса, там леший бродит
и Нортон на винте висит...

стоит ли напрягаться?

На моем сайте www.fdd5-25net в форуме часто встречаются высказывания, на мой взгляд, являющиеся заблуждением, мол, локальная сеть — это преимущество лишь Windows. Хотя известно, что Novell Netware, не являясь графической средой, поддерживает сеть без особых затруднений, но она специально такой создана. Ну а старый, добрый MS-DOS, как он работает в сетях?

Для DOS существует множество программ, самой удобной, на мой взгляд, является MS LAN Manager (www.fdd5-25.net/inet/mslan.zip). О работе локальной сети при помощи этого пакета я и хочу сегодня рассказать.

Но вначале хочу рассказать о том, что привело к написанию данного материала. Дело в том, что у меня уже с полгода как проведена локальная сеть между двумя компьютерами — IBM 286 и Pentium. Мне было интересно узнать, каково это, сидеть в чистом DOS и локальной сети с компьютером, который многие готовы сдать на металлолом. Оказалось, довольно интересно и занимательно, для меня во всяком случае.

Зачем это нужно?

Зачем нужна публикация, посвященная сетям под DOS? Таким риторическим вопросом, в принципе, должен задаться читатель. Уверю вас, что практическая отдача такого технического решения очень велика. В первую очередь, появляется возможность использования старых машин в качестве терминалов и рабочих станций. Ведь, в отличие от своих графических собратьев, DOS не «ест» много ресурсов диска и экономно использует оперативную память. Согласитесь, не всегда нужна трехпроцессорная рабочая станция под управлением последней версии Windows. Тем более, что во многих организациях и даже банках все еще трудится огромное количество x86-компьютеров. Без локальной сети работать очень неудобно — не будешь же бегать каждый раз с дискетой. Купить сейчас ISA-сетевую карту можно всего за 1...2 USD. Так что, мне кажется, что эта статья будет полезна широкому кругу читателей, даже если они безмерно помешаны на Windows, ведь большинство консольных команд, применяемых в DOS, сохранились и в последних версиях творений от Microsoft.

Как и что?

LAN Manager — это не набор красивых графических утилит, а просто множество дополнительных программ и драйверов, работающих в командной строке под DOS. Большинство из них сохраняет свой синтаксис и в Windows 9x, только лишь в последних версиях Windows NT добавлены новые ключи и команды. Так что, если вы до сих пор используете Windows 95/ME, то в этой статье обязательно найдутся полезные советы и для вас.

Вкратце рассмотрим проблемы, с которыми можно столкнуться во время установки LAN Manager.

1. Установка сетевого клиента

Если у вас установлен Windows 3x, то вам следует установить клиент в каталог с оболочкой — так будет проще разбираться с устаревшими драйверами. Строкой PATH нужно включить директорию Windows в загрузку. Проверьте эту строку после того как вы установите сетевой клиент.

2. Программа требует 429 Кб основной памяти

Для того чтобы LAN Manager работал, вы должны иметь 429 Кб свободной стандартной памяти, для этого следует воспользоваться драйвером himem.sys и утилитой Qemmt или ей подобной.

3. Работа с Double Disk Drive

К сожалению, пакет LAN Manager не работает с программой сжатия дисков Vertisoft Systems DoubleDisk.

4. Windows 3.x

Если вы установили LAN Manager и затем уже Windows 3.x,

то программа установки спросит, какой сетевой клиент вам выбрать. Но LAN Manager не появится в списке, так как он новее WIN3.x. Взамен него придется выбирать "LAN Manager 2.1".

5. Использование INTERLNK и INTERSVR

Если эти программы для соединения компьютеров через порты попытаться использовать совместно с LAN Manager, получится довольно неприятная вещь — начинаются сбои не только с сетевым клиентом, но и со всеми настройками сети.

Инсталляция

После того как вы скачали программу и распаковали sfx-архивы, можно начать установку. Как ни парадоксально, LAN Manager понимает только ISA-сетевые карты, для других, например PCI, обязательно нужен соответствующий драйвер. Локальная сеть — вообще штука интересная. Из всех клиентов локальной сети, которые я знаю, только LAN Manager имеет поддержку протокола TCP/IP. Значение адреса обычно задается во время установки. Я проводил локальную сеть между 286/DOS и Pentium/Win98 — особых трудностей не возникло, хотя пришлось помучиться с определенными командами. Жаль, тогда что под рукой не было статьи, подобной данной — времени на настройку сети понадобилось бы намного меньше. Саму установку описывать нет смысла — все происходит быстро и понятно, в лучших традициях Microsoft. Драйвера прописываются в autoexec.bat и config.sys; под Windows не будет "забыт" и system.ini. Но нас должно интересовать не куда они прописались, а что с этим потом делать. Кстати, LAN Manager занимает много основной памяти, поэтому, если не лень, напишите в строках загрузки драйвера в config.sys не "device=", а "devicehigh=" — тогда драгоценные программы будут загружаться в расширенную память, которая находится в диапазоне выше 640 Кб.

Теперь следует окинуть взором возможности команд LAN Manager для работы в сети, их всего две.

Команда Ping

Ping — это утилита для проверки работоспособности сети. Принцип ее работы в общих чертах заключается в послышке сетевому узлу эхо-запроса и ожидании от него эхо-ответа. Каждый узел должен уметь принимать эхо-запросы и возвращать эхо-ответы, разумеется, если он подсоединен к сети и работает.

Отсутствие эхо-ответа от сервера означает, что либо сервер "висит", либо имеется неустраняемое повреждение сети на участке клиент-сервер, обойти которое невозможно. Это свойство делает ping удобным инструментом проверки работоспособности узла и целостности соединения. Впрочем, отрицательный результат работы ping не всегда свидетельствует о наличии какой-либо неисправности. Так, например, самая главная проблема заключается в том, что для применения данной команды необходима установка протокола TCP/IP, а также знание пользователем IP-адреса удаленного компьютера, который вы желаете проверить.

Несмотря на свою простоту, утилита ping понимает достаточно большое количество ключей командной строки, более чем поверхностно описанных в прилагаемой документации. Не удивительно, что многие возможности ping ускользают не только от начинающих, но и от умудренных жизнью пользователей.

Ключ -w используется для задания интервала ожидания эхо-ответа в миллисекундах (по умолчанию — 20 секунд). Если отклик от сервера не будет получен в течение указанного времени, утилита ping сообщит "Превышен интервал ожидания для запроса", намекая на неработоспособность сервера или повреждение сети.

Ключ -n задает количество отправляемых эхо-запросов (по умолчанию — 4). Увеличение количества запросов бывает необходимо для контроля надежности и устойчивости рабо-



ты сервера. Чем выше качество канала, тем меньше разброс по времени ответов.

Ключ **-t** заставляет утилиту ping посылать запросы в бесконечном цикле до ее прерывания нажатием комбинации клавиш **<Ctrl-C>**. **Внимание:** **<Ctrl-Break>** не прерывает процесс, а выводит текущую статистику! Этот ключ очень удобен для ожидания момента пробуждения нехотят зависшего сервера.

Ключ **-l** задает размер посылаемой в эхо-запросе дейтаграммы без учета длины заголовка (28 байтов). Допустимыми являются значения от 0 до 65.500, включительно. По умолчанию размер дейтаграммы составляет 32 байта. Манипулируя этим значением, можно выяснить зависимость: скорость доставки — размер дейтаграммы. Если размер дейтаграммы превысит некоторую критическую величину (определяемую каждым промежуточным узлом самостоятельно), дейтаграмма разрезается на несколько пакетов подходящего размера, каждый из которых добирается до конечной точки маршрута самостоятельно, а на узле назначения они вновь собираются в исходную дейтаграмму.

Ключ **-f** устанавливает на дейтаграмме специальную пометку, запрещающую ее разрезание. Если хотя бы один из промежуточных узлов не может обрабатывать пакеты таких размеров, он "прибивает" дейтаграмму и посылает отправителю уведомление, объясняя причину "смерти" тем, что требуется фрагментация, но установка пометки, ее запрещающая.

Ключ **-i** задает время жизни (сокращенно **TTL — Time To Live**) пакета посылаемых дейтаграмм, измеряемое количеством узлов, которые может посетить пакет (по умолчанию — 128). Каждый промежуточный узел уменьшает значение TTL на единицу, и когда оно достигает нуля, пакет уничтожается с посылкой отправителю соответствующего уведомления. Это позволяет отслеживать маршрут путешествия пакетов, используя ping вместо утилиты tracer.

Ключ **-v** задает значения поля типа службы (**TOS — Type Of Service**). Тип сервиса с помощью некоторых абстрактных параметров указывает предпочтительный вид обслуживания: минимальная задержка, максимальная пропускная способность, максимальная надежность, минимальные задержки на пересылку или обычная, неприоритетная пересылка.

Ключ **-j** задает список узлов для свободной маршрутизации от клиента и аналогичен одноименному ключу утилиты tracer.

Ключ **-k** похож на ключ **-j**, но задает список узлов для жесткой маршрутизации, т.е. пакет передается из рук в руки строго по перечню перечисленных узлов, и ни один из них не может позволить себе воспользоваться услугами "собственного" маршрутизатора для передачи пакета следующему узлу.

Ключ **-a** задает определение имен узлов по их IP-адресам. Так, во всяком случае, сказано в документации. Смысл этого неясен: такое определение и без того происходит автоматически, независимо от наличия (отсутствия) ключа **"-a"**.

Команда Net

Это важная команда для работы в среде DOS. Именно она поможет нам в установке связи, просмотре настроек соединения, покажет статистику, расшарит ресурс или подключит его.

Возможные синтаксисы команды:

NET [CONFIG | HELP | START | STOP | TIME | USE | VIEW].

Рассмотрим использование этой команды подробно.

1. NET CONFIG [SERVER | WORKSTATION] — указывает настройки локальной сети компьютера.

Например, результат выполнения этой команды может быть таким:

Компьютер	\\ROOT
Пользователь	ROOT
Рабочая группа	HOME
Корневая папка	C:\WINDOWS
Версия программы	4.10.1998
Версия системы переадресации	4.00
Команда выполнена успешно.	

2. NET HELP <имя команды> или **NET <имя команды> / HELP** — выводит список всех допустимых команд.

3. NET START — запуск определенных служб. Например:
NET START [BASIC | NWREDIR | WORKSTATION | NETBIND | NETBEUI | NWLINK] [/LIST] [/YES] [/VERBOSE].
Здесь:

- **BASIC** — запуск базовой системы переадресации;
- **NWREDIR** — запуск системы переадресации Microsoft, совместимой с Novell(R);
- **WORKSTATION** — запуск стандартной системы переадресации;
- **NETBIND** — привязка протоколов к драйверам сетевых плат;
- **NETBEUI** — запуск интерфейса NetBIOS;
- **NWLINK** — запуск IPX/SPX-совместимого интерфейса;
- **/LIST** — вывод списка запущенных служб;
- **/YES** — выполнение команды NET START без предварительного запроса данных или подтверждения;
- **/VERBOSE** — вывод сведений о драйверах устройств и службах по мере загрузки.

Для запуска системы переадресации, выбранной при установке, используется команда NET START без параметров. В большинстве случаев использование параметров не требуется.

4. NET STOP [BASIC | NWREDIR | WORKSTATION | NETBEUI | NWLINK] [/YES] — остановка работы служб.

Для остановки системы переадресации, выбранной при установке, используется команда NET STOP без параметров. При этом производится отключение всех общих ресурсов и удаление команд NET из памяти компьютера.

5. NET TIME — синхронизирует время с сервером.

6. NET USE — подключение и отключение общих ресурсов и вывод сведений о подключениях. Возможные синтаксисы:

NET USE [диск: | *] [\\компьютер\папка [пароль | ?]] [/SAVEPW:NO] [/YES] [/NO]

NET USE [порт:] [\\компьютер\принтер [пароль | ?]] [/SAVEPW:NO] [/YES] [/NO]

NET USE диск: | \\компьютер\папка /DELETE [/YES]

NET USE порт: | \\компьютер\принтер /DELETE [/YES]

NET USE * /DELETE [/YES]

NET USE диск: | * /HOME

Здесь:

- **диск** — имя диска, назначаемое общей папке. Эквивалент следующего свободного имени диска — *. При использовании совместно с ключом **/DELETE** производится отключение сразу всех ресурсов;

- **порт** — имя параллельного (LPT) порта, назначаемое общему принтеру;

- **компьютер** — имя компьютера, на котором расположен общий ресурс;

- **папка** — сетевое имя общей папки;

- **пароль** — код для доступа к общему ресурсу (если он имеется);

- **?** — пароль для доступа к ресурсу запрашивается интерактивно. Этот режим может понадобиться лишь в том случае, когда ввод пароля не обязателен;

- **/SAVEPW:NO** — использование этого ключа позволяет предотвратить запись пароля в файл со списком паролей. В этом случае при каждом подключении к ресурсу пароль надо будет вводить заново;

- **/YES** — выполнение команды NET USE без предварительного запроса данных или подтверждения;

- **/DELETE** — отключение общего ресурса;

- **/NO** — выполнение команды NET USE с автоматической выдачей отрицательных (NO) ответов на все запросы, относящиеся к подтверждению действий;

- **/HOME** — подключение к основному каталогу, если сведения о нем имеются в учетной записи LAN Manager или Windows NT;

Команда NET USE без параметров выводит список всех подключенных ресурсов. Для просмотра выводимых сведений с паузами между отдельными экранами используются команды NET USE /? | MORE.

7. NET VIEW — вывод списка компьютеров, входящих в рабочую группу, или списка общих ресурсов на заданном компьютере. Возможный синтаксис:

NET VIEW [\\компьютер] [/YES]

NET VIEW [/WORKGROUP:группа] [/YES].

Здесь:

- **компьютер** — имя компьютера, список общих ресурсов которого следует вывести;

- **/WORKGROUP** — ключ, указывающий на необходимость вывода списка компьютеров, имеющих общие ресурсы, из другой рабочей группы;

- **группа** — имя рабочей группы, список компьютеров которой следует вывести;

- **/YES** — выполнение команды NET VIEW без предварительного запроса данных или подтверждения.

Для вывода полного списка имеющихся общие ресурсы компьютеров из рабочей группы, в которой зарегистрирован данный компьютер, используется команда NET VIEW без параметров.

Практика

После ознакомления с командами я расскажу о работе сети на примере своих IBM 286 и "зверской" машины Celeron 1000. Оба эти компьютера оснащены сетевыми картами REALTEC 8019 (аналог NE 2000). Эта модель представлена в стандартных драйверах LAN MANAGER, так что мне повезло. А вот операционные системы на их бортах разные: на 286-м — MS DOS 6.22, а на Celeron — Windows 98.

Первое, что понадобилось — это кабель, которым я соединил компьютеры. В целях экономии это был обычный кроссовер. Перед этим следует обязательно прозвонить провод и убедиться в его целостности, чтобы потом не пенять на ошибки программы и прочие приятные мелочи! Сделайте это обязательно.

Самое главное в старых компьютерах — это оперативная память. Программы LAN Manager, записываясь в автозагрузку, съедают всю основную память. Поэтому рекомендую все их строки инициализации удалить, все равно потом команда "net start" выполнит загрузку всего необходимого.

Итак, первая команда — **net start netbeui**. Именно при помощи этого протокола я устанавливаю соединение в сети. Почему не TCP/IP? Просто на 286-й машине он не нужен. Если никаких ошибок не возникло (не установлен протокол или не найден файл), должна появиться надпись, что команда выполнена успешно. Это означает, что первый этап прошел хорошо. То же на всякий случай я пишу и в консоли Windows 98. Там тоже все проходит успешно. Если на компьютере установлена Windows 3.x, в файле network.inf в каталоге LAN Manager можно посмотреть конфигурацию локальной сети.

Частичный пример моего файла:

```
NETWORK.INF
[nwsdata]
version="3.1.032"
autoexec_anchor="net start"
config_anchor="workgrp.sys"
default_transport=ms$nwlink
[workgroup]
descr="Workgroup Drivers"
[MultiNet]
lanman="Microsoft LAN Manager",0x200,0,0,mn$lanman
[data]
type=network
version="3.11.060"
autoexec_anchor="net start"
config_anchor="«
transport_hook=ms$ndishlp
defNDIS2_transport=ms$nwlink
[common]
devicesys=ifshlp.sys,@devdir\ifshlp.sys
tsr="net start"
```

```
netstart="net start", "@netdir\net start"
lastdrive=Z
devicesys=protman.dos
devicesys=workgrp.sys
; Protman Install and Protocol Sections
[protman_install]
netdir=1:protman.dos,1:protman.exe
[protman]
drivename=PROTMAN$
param=PRIORITY,,static,MS$NDISHLP
```

Большинство установок, как говорится, "налицо". Пакет LAN Manager, как и любая версия Windows, использует inf-файлы. И при установке пакета на уже готовую оболочку, он частично копирует их в свой, если, конечно, до этого сеть была настроена.

Следующим шагом по установке соединения является команда **net view**. С ее помощью мы узнаем о количестве компьютеров, подключенных в данную рабочую группу. Эту команду я вводил на своей 286-й машине. На другом компьютере, независимо от его модели, должен быть установлен общий доступ к файлам, папкам и принтерам, в противном случае команда ровным счетом ничего не укажет. Если же все будет установлено правильно, то у вас появится надпись с длинной полосой. В ней будут указаны имена компьютеров, их рабочая группа и краткое описание. Когда вы убедитесь, что сеть работает, нужно сделать общедоступным какой-нибудь ресурс. У меня это был раздел D: на диске моего Pentium. Для его подключения я ввел команду **net use f: \\ncld**.

Главное — не запутаться в написании этой, казалось бы, простой команды. Параметр **f:** задает имя диска на моей 286-й машине, а **\\ncld** (имя компьютера)\d (имя общедоступного ресурса) — путь. После таких манипуляций сеть должна работать без проблем, даже если это будет Intel 8088.

Остается банальный вопрос: как в среде DOS сделать общедоступным какой-либо ресурс? Все очень просто, нужно воспользоваться утилитой **share.exe**, которая "прячется" в папке вашего MS-DOS 6.22. Эта утилита устанавливает для ваших жестких и сетевых дисков средства совместного использования и блокировки. Ее синтаксис:

SHARE [/F:память] [/L:блок].

В файле config.sys используется такая запись:

INSTALL=[(диск:)]маршрут\SHARE.EXE [/F:память] [/L:блок]

Параметр **(диск:)]маршрут** задает расположение файла share.exe.

Параметр **/F:память** выделяет на диске память (размер указывается в байтах), которую MS-DOS использует для записи информации о совместном использовании файлов (по умолчанию — 2048). При определении этого объема памяти учитывайте, что каждый открытый файл требует места, достаточного для его маршрута и имени (средняя длина файла и маршрута — 20 байтов). Параметр **/L:блок** задает число одновременно блокируемых файлов (по умолчанию — 20). Обычно share.exe используется в многозадачной среде, где программы совместно используют файлы. Утилита загружает программный код, поддерживающий в таких средах совместное использование и блокировку файлов. После ее установки MS-DOS использует загруженные средства для обеспечения допустимости всех запросов программ на чтение и запись. Например, при обращении двух пользователей к одному файлу, share работает с ним таким образом, что они не смогут записывать в него одновременно.

Несколько советов в заключение

1. Имя группы лучше синхронизировать с именем сервера во время установки LAN Manager.

2. Компьютерам в сети не стоит присваивать неудобные длинные имена.

3. Во время установки LAN Manager не следует забывать об установке всех трех поддерживаемых протоколов, чтобы потом, в случае чего, не возиться, доставляя IPX или TCP/IP.

При написании статьи были использованы материалы сайта CITforum.ru.



ЛИСТАЯ СТРАНИЦЫ

За новыми процессорами ф. Intel последовали ее **новые чипсеты**. Об основных особенностях последних рассказывает С. Пахомов в статье "Новое поколение настольных ПК" (КомпьютерПресс, №7/2004, С.121...125).

Как известно, с июня этого года корпорация Intel перешла на новую маркировку своих процессоров. Теперь в названиях процессоров отсутствует тактовая частота, и все процессоры разбиваются на три модельных ряда: 3xx, 5xx и 7xx.

Процессоры младшей серии 3xx — это процессоры Celeron D с ядром Prescott. Их отличие от других серий процессоров заключается в уменьшенном размере кэша L2, который составляет всего 256 Кб, и в более низкой частоте системной шины FSB, которая в данном случае равна 533 МГц. Процессоры серии 3xx могут размещаться как в новых корпусах LGA775, так и в традиционных корпусах mPGA 478.

Процессоры серии 5xx — это полноценные процессоры для настольных ПК. Все они выполнены на ядре Prescott, устанавливаются в корпуса LGA775, поддерживают технологию Hyper-Threading, имеют кэш L2 размером 1 Мб и поддерживают частоту системной шины 800 МГц.

О процессорах 7xx автор не говорит.

Новый тип корпуса процессора LGA775 имеет принципиальное отличие от всех предыдущих корпусов. Основная его особенность — отсутствие ножек. Вместо них у процессора имеются лишь точечные контактные площадки, а сами контактные ножки теперь размещаются в процессорном разъеме. Такое конструктивное решение, по заявлению компании Intel, позволяет наращивать производительность процессора.

Чипсет Intel 925X Express представляет собой топовую модель чипсета. Он поддерживает процессор Intel Pentium 4 (Prescott), выполненный по 90-нанометровой технологии, а частота системной шины составляет 800 МГц. Соответственно, максимальная пропускная способность системной шины равна 6,4 Гб/с.

Контроллер памяти чипсета (MCH) является двухканальным и поддерживает только новую память DDR2-533/400. При

использовании памяти DDR2-533 в двухканальном режиме максимальная пропускная способность шины памяти составляет 8,5 Гб/с. Всего же контроллер памяти в соответствии со спецификацией поддерживает до 4 Гб памяти и четыре DIMM-слота (по два на каждый канал).

Технология Intel Flex Memory позволяет работать с памятью как в одноканальном, так и в двухканальном режиме. Для реализации двухканального режима работы необходимо только, чтобы в обоих каналах был установлен одинаковый объем памяти. При этом не требуется, чтобы в обоих каналах устанавливалось одинаковое количество модулей памяти. Если же в каналах устанавливается различный объем памяти (всимметричный режим), или используется только один канал памяти, то контроллер будет работать с памятью и в одноканальном режиме.

Другой особенностью северного моста чипсета Intel 925X Express является поддержка новой шины PCI Express x16 для внешней графической карты нового поколения. В сравнении с шиной AGP 8x, обеспечивающей общую пропускную способность 2 Гб/с, новая шина PCI Express x16 имеет пропускную способность 8 Гб/с, то есть в четыре раза больше. Кроме того, в шине AGP 8x канал для чтения и записи данных используется совместно, и шина оптимизирована для одиночных операций. В шине PCI Express x16 применяются выделенные каналы для чтения и записи, то есть шина оптимизирована для одновременных операций чтения/записи. В результате шина PCI Express x16 допускает высокопроизводительный вывод нескольких потоков видео и графики.

Северный мост чипсета Intel 925X Express (чип NG9225X) связан по высокоскоростному интерфейсу DMI с концентратором ввода-вывода (южный мост) ICH6. Всего существуют три типа южного моста: ICH6R (чип 82801FR), ICH6W (чип 80801FW) и ICH6RW (чип 80801RW).

Новая версия южного моста имеет ряд существенных преимуществ по сравнению с предыдущей версией ICH5, и прежде всего это:

- новый 8-канальный звук Intel High Definition Audio (Intel HD Audio);
- восемь портов USB 2.0;

- шина PCI Express x1;
- шина PCI;
- четыре SATA-канала;
- один PATA-канал;
- технология Intel Wireless Connect;
- сетевой гигабитный интерфейс Intel GbE.

Далее автор дает краткую характеристику этих особенностей нового чипсета.

Чипсеты семейства Intel 915 P/G/GV Express по своим функциональным возможностям во многом напоминают чипсет Intel 925X Express. Во всех чипсетах используется один и тот же южный мост (ICH6R, ICH6W или ICH6RW), и отличие заключается лишь в северном мосте. Для чипсета Intel 915G Express — это микросхема NG82915G, для Intel 915P Express — микросхема NG82915P, а для Intel 915GV Express — микросхема NG82915GV.

Все чипсеты данного семейства поддерживают процессор Intel Pentium 4 с частотой системной шины 800/533 МГц. Кроме того, все они поддерживают и память DDR2-533/400, и память DDR400/333. Причем на плате могут быть слоты для установки памяти либо DDR2, либо DDR. Если же на плате имеются слоты для установки модулей памяти как DDR2, так и DDR, то возможно использование только одного типа памяти. Память DDR2-533/400 можно использовать только при частоте системной шины 800 МГц, а память DDR400 — при частоте системной шины как 800 МГц, так и 533 МГц. Память DDR333 поддерживается только при частоте системной шины 533 МГц. Контроллер памяти во всех чипсетах является двухканальным и поддерживает до двух DIMM-слотов на каждый канал памяти. При этом всего можно установить до 4 Гб памяти.

В чипсете Intel 915P Express нет интегрированного графического ядра, и для установки видеокарты используется шина PCI Express x16. Чипсет Intel 915G Express содержит интегрированное графическое ядро Intel Graphics Media Accelerator 900, но, в то же время, поддерживается и шина PCI Express x16 для установки внешнего графического решения. Чипсет Intel 915GV Express отличается от Intel 915G Express лишь отсутствием шины PCI Express x16, что предопределяет использование данного чипсета в материнских платах, ориентированных на офисные ПК.

О сенсорных экранах рассказывает А. Юдов в статье "Прикоснуться к прогрессу" (CHIP, №7/2004, С.68...71).

В настоящее время используются несколько типов сенсорных экранов — резистивные, емкостные, инфракрасные и на поверхностных акустических волнах. Все они имеют свои плюсы и минусы и применяются для решения различных задач. Наибольшее распространение получила резистивная технология. Принцип работы этих экранов основывается на измерении электрического сопротивления части системы в момент прикосновения. Различают 4-, 5- и 8-проводные варианты резистивных экранов. Все они имеют многослойную структуру — при касании какие-то слои совмещаются, и измерение возникших в

результате токов позволяет определить точку касания.

В экранах с четырьмя и восемью проводками два слоя используются для измерения двух координат. В верхнем слое создается градиент тока, а в нижнем слое при их соприкосновении производится измерение сопротивления получившейся системы. Для нахождения второй координаты роли слоев меняются — в нижнем создается градиент тока, а в верхнем измеряется значение сопротивления. Экраны с восемью контактами имеют по одному дополнительному калибровочному контакту на каждом крае слоя, что позволяет достичь большей точности за счет компенсации температурных колебаний сопротивления выводов и цепей.

Пятипроводные сенсорные экраны, разработанные фирмой Elo Touch Systems, работают сходным образом. Различие состоит лишь в том, что обе координаты измеряются на нижнем слое. Это позволяет им продолжать работу при сильно загрязненном или поврежденном верхнем слое.

Экраны с четырехпроводной технологией дешевле и потребляют меньшую мощность. Они отлично работают в смартфонах, КПК и других устройствах, имеющих небольшой экран. Восемипроводные экраны, к примеру, используются во многих моделях Tablet PC.

Вторая, весьма распространенная разновидность сенсорных экранов — емкостные. При прикосновении к такому экрану изменяется емкость какой-то части систе-



мы, что дает возможность, измеряя мгновенные токи, локализовать точку касания. Подобным образом работают емкостные сенсорные экраны фирмы MicroTouch. Используется и альтернативная технология — Near Field Imaging (модуляция ближнего поля), предлагаемая фирмой Dynargo. Конструкция таких экранов не предусматривает никаких механических движений — прикосновение определенным образом влияет на электростатическое поле внутри внешнего слоя, что позволяет вычислить координаты точки касания.



Аналоговый вычислительный машин (АВМ) посвящена статья М.Белоуса "Вуду без цифры" (Подводная лодка, №7/2004, С.96...99).

В этих машинах реальные задачи решаются "по аналогии". Это возможно потому, что в природе процессы описываются относительно небольшим числом закономерностей. Различаются только размерности величин и коэффициенты, а уравнений, связывающих их воедино, совсем немного. На практике, например, широко используется подобие электрических и механических явлений.

Однако во многих случаях такое устройство создается для решения одной конкретной задачи. Даже если на готовой модели можно рассматривать другой аналогичный процесс, необходим перерасчет коэффициентов подобия, что само по себе может оказаться нетривиальной проблемой. Зато, в отличие от цифрового компьютера, аналоговый практически не требует программирования — сам способ его построения и функционирования уже является готовой программой.

Простейшей АВМ можно считать логарифмическую линейку, использование которой началось около 1600 года. Ее "аналоговость" заключается в известном свойстве степенных функций: если две величины с одинаковыми основаниями и разными степенями перемножаются, то показатели их степеней складываются. На подвижной и неподвижной полосках счетной линейки деления наносятся в логарифмическом масштабе, поэтому умножение чисел преобразуется в сложение. Кроме умножения, логарифмическая линейка позволяла выполнять и другие действия, вплоть до извлечения корней и возведения в степень.

Успешно освоив работу со счетными линейками, математики пришли к созданию номограмм — заранее рассчитанных графиков функций. По этим тщательно выполненным рисункам, зная значение аргумента (или даже нескольких аргументов), можно было, не проводя утомительных вычислений, быстро получить результат. Особенно полезным такое нововведение стало на флоте — начиная с конца XVIII века номограммы для определения места судна по данным секстанта, компаса и хронометра включаются в практические руководства по навигации.

Быстро совершенствовавшееся инженерное искусство требовало, однако, более универсального устройства для расчетов. Особенно важно было научиться

Экраны на поверхностных акустических волнах (ПАВ) используются в основном в киосках. Главное их преимущество — вандалоустойчивость и прочность. На рабочей поверхности экрана создается акустическая волна, которая искажается при прикосновении. Это дает возможность, сравнивая выходной сигнал с шаблонами, записанными в памяти контроллера, определить точку касания. Разрешающая способность экранов ПАВ довольно низка, однако для киосков это не критично. Разработчиком технологии является фирма Elo Touch Systems.

определять площадь фигуры, образованной произвольной замкнутой кривой (то, что мы сейчас называем определенным интегралом). И в 1814 г. такой прибор, названный планиметром, был предложен английским ученым Дж.Германом. Планиметр представлял собой систему реек и веревочных тяг и был достаточно сложным в обращении, хотя и заметно облегчал вычисления. Через сорок лет немец А.Амслер усовершенствовал это устройство. Он предложил систему с прокатываемым вдоль "интегрируемой" кривой роликом. Позднее на базе этой разработки англичанин Дж.Томсон создал планиметр следующего поколения — фрикционный интегратор. Практически одновременно с Дж.Томсоном, в 1878 г., проект механического вычислителя площади, вычерченной на плоскости замкнутой кривой, был предложен польским математиком Абданк-Абанкевичем.

В 1904 г. известный русский кораблестроитель Алексей Николаевич Крылов построил первую в мире механическую машину для решения дифференциальных уравнений. Более универсальная механическая интегрирующая машина "Интерграф" была создана в 1927 г. американцем Ванневаром Бушем. В 1942 г. он модернизировал ее, сделав электромеханической. Двумя русскими учеными, Н.Павловским в 1918 г. и Н.Минорским в 1936 г., был предложен метод электрогидродинамических аналогий. С.Гершгорин в 1927 г. разработал теорию построения сеточных моделей — тех самых электрических устройств, поведение которых можно было уподоблять различным исследуемым явлениям, контролируя и с уверенной точностью регистрируя их. Дело было за воплощением этих идей в металл — и решающим шагом здесь стало создание в 1942 г. Бертраном Расселом (США) решающего усилителя постоянного тока. Именно он и явился непосредственным прародителем большинства последовавших за ним, в том числе и современных, аналоговых вычислительных машин.

В 1936 г. советский инженер И.Брук предложил свою схему механического интегратора и электрического расчетного устройства для моделирования параметров энергетических систем (в основном, токовых потерь при передаче электричества по проводам и перераспределении нагрузки на подстанциях). На основе этой работы в 1947 г. была создана АВМ ЭДА — "Электродинамический Анализатор". Машина успешно решала определенные

Инфракрасные сенсорные экраны обладают еще меньшей разрешающей способностью, чем ПАВ-экраны, и являются самыми дорогостоящими, однако имеют ряд принципиальных преимуществ. Во-первых, такие экраны абсолютно прозрачны и пропускают 100% света. Во-вторых, они имеют практически неограниченный ресурс работы. Все эти преимущества следуют из того факта, что инфракрасный сенсорный экран является чистой оптической системой — он представляет собой рамку, создающую сетку инфракрасных лучей прямо перед поверхностью дисплея.

группы дифференциальных уравнений вплоть до 20-го порядка, причем практически в реальном масштабе времени, по мере задания исходных условий.

В начале 50-х годов прошлого века начали использоваться еще неуклюжие, но крайне перспективные универсальные цифровые ЭВМ. Такие компьютеры хороши именно своей универсальностью — если нужно решить новую задачу, их достаточно просто перепрограммировать. Но такое достоинство имеет и оборотную сторону — в условиях, когда вычислительная задача остается неизменной, аналоговые машины часто оказываются эффективнее и дешевле цифровых. И количество таких задач достаточно велико.

На первое место для АВМ, как и для вычислительной техники вообще, исторически встали приложения военного характера. Это прежде всего управление артиллерийским, в особенности зенитным, а потом и ракетным огнем. Таблицы артиллерийских стрельб — зависимость дальности полета снаряда от наклона ствола, погодных условий, силы порохового заряда и т.п. — составлялись уже начиная с XVII века. Во время второй мировой с ее массированным применением орудий точная и быстрая корректировка огня стала важнейшей вычислительной задачей. Не случайно одними из первых среди отечественных АВМ стали широко применяться ПУАЗО — механические и электромеханические устройства для управления зенитной стрельбой. Наводчик орудия, снабженного ПУАЗО, не тратил лишнего времени на утомительный поиск нужных столбцов и строчек в таблице стрельб — он просто наводился по прицельной рамке и с уверенностью знал, что зенитный снаряд взорвется именно в той точке, куда он предполагает его направить.

После войны выяснилось, что поистине уникальной областью применения АВМ оказались разнообразнейшие задачи управления. С точки зрения математики, управление — это анализ текущего состояния системы, определение его отклонения от идеального (ожидаемого) и выработка рекомендаций по приближению системы к этому ожидаемому состоянию. Подобие законов природы, о котором упоминалось выше, во множестве случаев распространяется и на менее формализованные области — так, уравнениям экономической эффективности производства, в зависимости от вложенных сил и средств, совсем не сложно найти аналогичные физические процессы. АВМ, таким образом, успешно



применяются в составе соответствующих экспертных систем — как консультационных, так и собственно управляющих.

Написанная человеком программа может содержать скрытую ошибку, универсальный компьютер может зависнуть — но аналоговая экспертная система работает идеально не потому, что ее поведение корректно задано, а потому, что иначе она работать просто не может; таковы физические законы ее функционирования. Конечно, от поломки не гарантирована и АВМ, но, по крайней мере, вероятность такого события для нее меньше, чем для цифрового аппаратно-программного комплекса.

Еще один значительный плюс узкой специализированности АВМ — в доступных ей скоростях работы. Физический процесс, аналогичный моделируемому, протекает в ней в реальном масштабе времени — каким бы сложным он ни представлялся с точки зрения математического описания. В итоге оснащенный АВМ экспертный системам становится доступной такая область управления как опережающий анализ — на аналоговой модели можно за сравнительно короткое время "проиграть" различные ситуации, определить реакцию системы на самые разные комбинации внешних воздействий. Современные АВМ способны "прокручивать" сотни и тысячи циклов моделирования в секунду. Цифровые ЭВМ, работающие с виртуальным математическим описанием процесса, могут конкурировать с ними лишь при наличии серьезнейшей вычислительной мощи и, соответственно, неоправданной дороговизне самих расчетов. Больше того, на базе АВМ во множестве отраслей возможно создание самонастраивающихся систем оптимизации — когда на основе опе-

режающего анализа аналоговый вычислитель определяет оптимальный на текущем шаге набор входных параметров, автоматически передает их на управляющие узлы, а затем снова проводит опережающий анализ следующего порядка — с каждым шагом находя все более совершенные условия функционирования системы.

А в такой области как многократное решение стохастических дифференциальных уравнений (метод Монте-Карло), аналоговым вычислителям вообще нет равных — если, опять-таки, принимать во внимание эффективность поиска решения. Анализ методом Монте-Карло часто применяется в практических инженерных задачах, а на цифровых ЭВМ его осуществление осложняется не только трудностями реализации алгоритма, но и серьезными проблемами с генерацией по-настоящему случайных числовых последовательностей. То, что в привычных нам компьютерах называется "генератором случайных чисел", на самом деле производит лишь псевдослучайные ряды — слишком уж детерминирован процесс двоичных вычислений.

Как ни хороши в областях своего применения АВМ, есть у них чрезвычайно серьезный "врожденный" недостаток — сама их аналоговость. Мельчайший элемент современной цифровой ЭВМ — транзистор — настолько мал и унифицирован, что для серьезной работы их требуются сотни миллионов. Зато — полностью идентичных и не слишком трудоемких в изготовлении (если разделить, скажем, стоимость процессора на число содержащихся в нем транзисторов, цена за каждый окажется смехотворно малой). В аналоговом же компьютере точность и надежность каждого элемента — мерило качественной рабо-

ты всей системы в целом. Те же электро-механические приборы ПУАЗО, безусловно необходимые в бою, в эксплуатации были достаточно капризными, а в изготовлении — достаточно дорогими. Требовалось с величайшей точностью выдерживать все технологические нормы, привлекать труд высококвалифицированных мастеров. И значит, вне военного применения, где цена все же стоит на втором месте после эффективности, электромеханические АВМ себя не всегда оправдывали.

Другая особенность аналоговых вычислителей — их ограниченная точность. Какой бы ни была АВМ — гидравлической, пневматической, электромеханической или электронной — она воспроизводит некий физический процесс, а измерения его характеристик доступны лишь с некоторой точностью, не говоря уж о точности реализации самого этого процесса (ошибки сопряжения механических деталей, теплотечи в электрических схемах и т.п.). В итоге, погрешность работы аналоговых вычислителей, решающих дифференциальные уравнения десятого, скажем, порядка, достигает нескольких процентов. Таким образом, эффективность АВМ оказывается естественным образом ограниченной — пока обеспечиваемая аналоговыми моделями точность удовлетворительна, они выгоднее цифровых; при повышении же требований именно к точности компьютерам на бинарной логике не находится равных.

Современные АВМ далеко ушли от логарифмической линейки. Чаще всего они используются сейчас в составе гибридных систем, содержащих и цифровые ЭВМ, поскольку для решения некоторых задач аналоговые "логические элементы" оказываются неизмеримо эффективнее цифровых.

Если у вас есть цифровой фотоаппарат, то, **быть может, стоит приобрести и струйный фотоприпртер?** Ответить на этот вопрос вам поможет статья С.Асмакова "Струйная фотопечать: реалии и перспективы" (*КомпьютерПресс*, №7/2004, С.108...110).

По мере роста популярности цифровой фотографии все больше изображений, полученных цифровыми камерами, в дальнейшем используются исключительно в электронном виде — для просмотра на экране ПК или телевизора (при помощи бытовых DVD-плееров), пересылки по E-mail и т.п. Поэтому значительное увеличение количества снятых кадров, обусловленное переходом от пленочной фототехники к цифровой, не может гарантировать пропорционального роста объемов печати фотографий.

Серьезную конкуренцию персональным фотоприпртерам составляют специализированные фотолаборатории, предлагающие услуги по печати изображений с цифровых носителей. К настоящему времени сети подобных лабораторий функционируют уже во многих крупных городах России.

С точки зрения конечных пользователей, фотолаборатории имеют как минимум два важных преимущества по сравнению с фотоприпртерами. Во-первых, они позволяют

получать отпечатки с более низкими затратами, и, во-вторых, пользователю в этом случае не требуется приобретать оборудование (припртер) и тратить время и средства на его техническое обслуживание, приобретение и обновление программного обеспечения и пр. В общем-то, даже не обязательно иметь дома компьютер — изображения в лабораторию можно передать и на сменном носителе, используемом в фотоаппарате. Таким образом, для широкого круга фотолюбителей, не нуждающихся в дополнительной обработке полученных снимков на ПК, обращение в фотолабораторию — более предпочтительный вариант по сравнению с приобретением собственного фотоприпртера.

Однако существенными достоинствами фотоприпртера являются гибкость и оперативность. Имея в своем распоряжении этот аппарат, можно быстро и не выходя из дома получить отпечатки в любое время суток. Это особенно удобно в том случае, когда возникает потребность оперативно напечатать одну или несколько фотографий — например, для пришедших в гости друзей или родственников.

Практически все современные модели фотоприпртеров позволяют печатать фотографии без подключения к ПК — непосредственно с цифровых камер либо со

сменных носителей. Это значительно упрощает процесс получения отпечатков и, кроме того, дает возможность брать малогабаритные фотоприпртеры с собой — например, на дачу или в длительную поездку.

Однако, помимо функциональных возможностей и удобства, существует еще один важный фактор — цена. Более высокая (по сравнению с затратами на печать цифровых снимков в фотолаборатории) стоимость получаемых отпечатков является, пожалуй, наиболее серьезным сдерживающим фактором для пользователей, потенциально готовых приобрести струйный фотоприпртер.

В настоящий момент реальная стоимость фотографического отпечатка формата 10х15 см на дешевых моделях струйных фотоприпртеров составляет (в зависимости от сюжета изображения и типа используемой бумаги) от 40 до 60 центов, то есть примерно 11...17 руб. (при условии использования оригинальных чернил и фотобумаги). Для сравнения: цены на печать фотокарточек такого же формата с цифровых носителей в специализированных фотолабораториях колеблются в пределах от 5 до 10 руб.

Стоимость отпечатка струйного припртера складывается главным образом из затрат на приобретение расходных материалов — чернил (картриджей) и специальной



бумаги. Для производителей струйных принтеров продажа расходных материалов является основной статьей дохода. Поэтому вполне закономерно, что при значительном снижении стоимости самих принтеров цены на расходные материалы практически не изменяются.

Что касается качества, то этому показателю отпечатки современных струйных фотоприпринтеров (естественно, при использовании оригинальных чернил и специальной бумаги) во многих случаях выигрывают по сравнению с карточками, сделанными в мини-лабораториях по традиционной химической технологии (причем это касается как визуальных характеристик изображения, так и его долговечности).

Несмотря на уже достигнутые успехи, работа над дальнейшим совершенствованием технологий струйной печати продолжается. Так, компания Hewlett-Packard во второй половине прошлого года представила новый вариант своей технологии фотореалистичной печати HP PhotoREt Pro, использующей печать в восемь красок. К "классическому" набору, включающему чернила голубого, пурпурного, желтого, светло-голубого, светло-пурпурного и черного цветов, были добавлены два оттенка нейтрально-серого (серый и светло-серый). Использование нейтрально-серых чернил при печати монохромных изображений позволяет добиться идеального баланса серого во всем диапазоне оттенков. Что касается цветных изображений, то ис-

пользование дополнительных чернил позволило расширить цветовой охват (до 79,2 млн. оттенков) и обеспечить более точное воспроизведение тональных переходов в близких по цвету оттенках. Первым принтером, в котором была использована система HP PhotoREt Pro, стал Photosmart 7960.

В начале нынешнего года компания Canon представила восьмикрасочную систему печати ChromaPLUS. К традиционному набору добавлены чернила красного и зеленого цветов. Кроме того, в некоторых принтерах Canon (в частности, в i990) используется упрощенная версия ChromaPLUS (без зеленого красителя).

Но, пожалуй, наиболее радикальные шаги в области совершенствования систем струйной печати предприняла компания EPSON. Ее специалисты разработали абсолютно новую систему полностью пигментных чернил UltraChrome Hi-Gloss, причем с весьма непривычной цветовой раскладкой. В ней используются чернила голубого, пурпурного, желтого, красного и синего цветов в сочетании с двумя разновидностями черного (оптимизированными соответственно для глянцевых и для матовых носителей). Помимо семи цветов, имеются и прозрачные чернила (так называемый глянec), которые используются для формирования глянцевой поверхности на тех областях изображения, где не нанесены какие-либо чернила (то есть на абсолютно белых участках). Данная сис-

тема была впервые реализована в фотопринтере Stylus Photo R800.

Спустя некоторое время в компактном фотоприинтере PictureMate была использована упрощенная версия этой системы. В ней также используются полностью пигментные чернила UltraChrome Hi-Gloss, однако базовых цветов всего шесть: голубой, пурпурный, желтый, красный, синий и черный.

Автор отмечает, что, наряду с неоспоримыми достоинствами, пигментные чернила UltraChrome Hi-Gloss имеют и определенные недостатки. В частности, принтеры с такой системой печати более требовательны к типу используемых носителей. Оптимальный носитель для пигментных чернил — гладкая глянцевая бумага. На бумаге с матовой поверхностью отпечатки, сделанные чернилами UltraChrome Hi-Gloss, все-таки уступают по качеству аналогичным изображениям, отпечатанным чернилами на водорастворимой основе.

Стоит отметить, что в настоящее время новые разновидности систем печати используются главным образом в довольно дорогих моделях фотопринтеров, а удельный вес этих устройств в парке эксплуатируемых печатающих устройств невелик. Намерены ли производители внедрять их в бюджетных моделях — пока неясно. Скорее всего, ответить на этот вопрос можно будет уже через два-три месяца, когда компании анонсируют новые принтеры в ходе традиционного осеннего обновления своего модельного ряда.

Если вы приобрели новый качественный монитор, но качество изображения на нем далеко от того, что вы ожидали, следует обратить пристальное внимание на используемую видеокарту. **Как видеокарта формирует видеосигнал и, следовательно, влияет на качество изображения**, рассказывается в статье "Прозрачность резкости" (CHIP, №7/2004, С.62...67).

На ЭЛТ-мониторы и подключенные через аналоговый вход ЖК-мониторы видеосигнал всегда попадает одинаковым образом: вначале цифровые данные поступают в кадровый буфер видеокарты, а затем, пройдя через RAMDAC (Random Access Memory Digital to Analog Converter — цифро-аналоговый преобразователь, имеющий локальную память), они преобразуются в аналоговый сигнал и передаются далее на вход монитора.

Качество RAMDAC в большинстве случаев оценивается в виде его тактовой частоты, т.е. максимальной скорости обработки пикселей, измеряемой в мегагерцах. Считается, что теоретически при работе на частоте 350 МГц RAMDAC может выдавать на экран монитора 350 млн. пикселей в секунду. На самом деле RAMDAC, работающий с такой скоростью, нужно было бы обозначить числом 175 МГц, поскольку частота вывода графических элементов изображения на экран дисплея равна половине рабочей частоты RAMDAC.

При разрешении монитора 1600x1200 пикселей и частоте обновления экрана 85 Гц, за одну секунду передается около 163 млн. пикселей (1600x1200x85). К этому необхо-

димо добавить еще примерно 35% точек, не воспринимаемых глазом. Вот почему, для того чтобы добиться желаемой частоты обновления при указанном разрешении, RAMDAC должен работать с частотой около 230 МГц. Все современные видеокарты имеют RAMDAC, работающий на частотах от 350 до 400 МГц, поэтому у них есть достаточный "запас мощности".

На участке видеотракта от RAMDAC до выхода видеокарты располагаются низкочастотные фильтры, которые в идеале должны пропускать сигналы с частотой, не выше граничной для RAMDAC. Все, что находится выше этого диапазона и представлено в форме высокочастотных помех, должно поглощаться фильтрами.

На практике НЧ-фильтры работают не так хорошо, как хотелось бы. С одной стороны, они глушат часть полезного сигнала, а с другой — пропускают часть высокочастотных помех. Избирательная способность НЧ-фильтра зависит от так называемого порядка. Чем выше порядок фильтра, тем выше его качество, но, соответственно, выше стоимость его компонентов. Большинство производителей не дают информации о порядке своих фильтров, но известно, что в большинстве видеокарт используются фильтры третьего порядка, и только Matrox устанавливает на свои видеокарты Parhelia фильтры пятого порядка. Таким образом, главными факторами, определяющими качество видеосигнала на видеовыходе, являются сам RAMDAC и качество следующего за ним НЧ-фильтра, которое напрямую зависит от используемых при его изготовлении ком-

понентов. Разумеется, влияние оказывают разводка платы и качество электрических соединений. Вот почему бывает так, что принципиально одинаковые, созданные на одном и том же чипе видеокарты выдают изображение различного качества.

TFT-мониторы также чувствительны к качеству поступающего на их вход видеосигнала. В принципе, из-за специфики устройства матрицы, у ЖК-дисплеев нет настоящих проблем с резкостью изображения, однако и они не в состоянии сотворить чудо, если видеосигнал при быстрой смене черных и белых пикселей не достигает нужного уровня интенсивности. В этом случае и на плоском мониторе мы увидим недостаточно контрастное и не совсем резкое изображение.

Еще один важный критерий для ЖК-мониторов — точность, с которой видеокарта соблюдает заданный пиксельный растр. Если частота вывода графических элементов изображения на экран дисплея колеблется слишком сильно, то аналого-цифровой преобразователь (АЦП) монитора не в состоянии корректно синхронизировать изображение. Следствием этого будет дрожание и мерцание картинки.

Самое простое техническое решение, позволяющее повысить качество картинки — передача изображения с видеокарты на монитор в цифровом виде через разъем DVI. Однако на практике по-прежнему чаще всего применяются аналоговые разъемы, так как большинство мониторов не имеют цифрового входа. Да и среди видеокарт далеко не все оснащены соответствующим выходом.



Реализация многопоточных приложений на базе MFC

А.КОРБИТ,
г.Минск, БГУИР.

Прежде чем рассматривать данную реализацию, вспомним основные понятия.

Процесс (process) — это экземпляр выполняемой в данный момент времени программы (копия программы, загруженной в память, при этом каждому процессу выделяется 4 Гб адресного пространства). Код и данные динамических библиотек, если они используются, также размещаются в адресном пространстве потока. Кроме того, процессу принадлежат такие ресурсы как файлы, динамические области памяти и потоки, которые уничтожаются при его уничтожении.

Поток (thread) — это последовательность исполнения текущего кода внутри процесса, который запускается в его адресном пространстве и использует ресурсы, выделенные процессу.

Любой процесс имеет как минимум один поток, получивший название "первичный" (primary thread). 32-х разрядная ОС Windows является средой многозадачной и позволяет создавать "вторичные" потоки (secondary threads). Вторичные потоки повышают эффективность приложения, так как позволяют решать несколько разных задач одновременно. При этом потокам устанавливаются приоритеты, в зависимости от которых планировщик потоков регулирует выделение времени для каждого потока. Это означает, что потоки обладают свойством "вытесняющей" многозадачности — каждому потоку выделяется для его выполнения в соответствии с его приоритетом "квант времени". По истечении данного кванта времени выполнение процесса прерывается, и вызывается процедура распределения квантов времени. Эта процедура при отсутствии более приоритетных компонентов может выделить квант времени этому же потоку. Каждый поток имеет собственный набор регистров процессора и собственный стек, т.е. под потоком можно понимать одну из ветвей исполнения процесса.

В библиотеке MFC определены два типа потоков — рабочие потоки (worker threads) и потоки пользовательского интерфейса (user interface threads). Такое разделение присуще только MFC.

Потоки пользовательского интерфейса способны принимать и обрабатывать сообщения пользователя, т.е. могут обслуживать ввод пользователя. В них реализован цикл обработки событий и сообщений, возникающих при взаимодействии пользователя и приложения. Примером такого потока является первичный поток приложения, представляемый классом приложения, производным от CWinApp. Вторичные потоки пользовательского интерфейса также способны обеспечивать взаимодействие с приложением, не снижая скорость выполнения приложения.

На рабочие потоки возлагается решение фоновых задач, которые не требуют пользовательского ввода. Т.е. рабочие потоки обеспечивают дополнительные пути выполнения задачи внутри интерфейсного потока, например, фоновые вычисления. Выполнение таких задач, как правило, занимает много времени, и пользователь имеет возможность параллельно продолжать работу с приложением.

Все потоки в приложениях на базе MFC — объекты класса CWinThread или производного от него. Это справедливо и для первичного потока приложения, так как класс CWinApp, отвечающий за его создание, является прямым

наследником класса CWinThread. Работа с этим классом позволяет в полной мере использовать возможности библиотеки. Класс CWinThread поддерживает хранение контекста потока в дескрипторе потока m_hThread, что является обязательным при работе с MFC.

Создание рабочего потока

Для создания рабочего потока необходимо выполнить две операции — создать исполняющую функцию, т.е. написать код управляющей функции, в котором указываются все действия для данного потока, и запустить поток.

Для создания рабочего потока необходимо вызвать глобальную функцию **AfxBeginThread**, которая создает и инициализирует новый объект класса CWinThread, запускает его на выполнение и возвращает его адрес пользователю. У этой функции два обязательных параметра:

- адрес исполняющей функции;
 - значение аргумента исполняющей функции;
- и четыре — с умалчиваемыми значениями:
- приоритет потока (по умолчанию — "нормальный", возможные значения приоритета потока задаются в описании функции ::SetThreadPriority);
 - предполагаемый размер стека (по умолчанию он равен 0 — такой же, как и у вызывающего потока);
 - вид начала выполнения (по умолчанию равен 0 — нормальное начало выполнения);
 - атрибут безопасности потока (по умолчанию равен 0 — такой же, как и у вызывающего).

Исполняющая функция, на которую указывает первый параметр **AfxBeginThread**, должна быть описана в виде **UINT ID(LPVOID pParam);**

где:

- **ID** — ее идентификатор;
- **pParam** — параметр — 32-разрядная величина, значение которой совпадает со вторым параметром функции **AfxBeginThread**, используемой для создания данного потока. Исполняющая функция может интерпретировать это значение любым способом, например, как скалярную величину, а может и вообще игнорировать.

Завершая свою работу, функция возвращает либо значение типа **UINT** — причину завершения, либо нулевое значение — нормальное завершение работы. Поток завершается самостоятельно при достижении в функции потока оператора **return** или явным вызовом функции **AfxEndThread()**.

Создание интерфейсных потоков

Интерфейсные потоки используются, когда необходимо обрабатывать сообщения независимо от работы других потоков. Каждое приложение на базе MFC уже имеет главный (основной) поток. [Главный поток приложения (производный от класса CWinApp) запускается вместе с приложением и завершается вместе с ним.

Для создания интерфейсного потока необходимо выполнить следующие действия:

- создать класс на базе класса CWinThread, например, CMyThrd. При этом при описании и реализации этого класса необходимо использовать макросы **DECLARE_DYNCREATE** и **IMPLEMENT_DYNCREATE**;

- перегрузить часть функций этого класса. Обязательно перегрузить метод **InitInstance()**. Он производит инициа-



лизацию экземпляра потока, т.е. вызывается при его создании и должен обязательно возвращать TRUE, если поток готов к запуску. В этом методе инициализируются переменные, выделяется память, создаются окна, а переменной `m_pMainWnd` класса `CWinThread` присваивается указатель на созданное окно;

- для освобождения памяти или других каких-то действий по окончании потока необходимо перегрузить метод `ExitInstance()`. Метод `OnIdle()` перегружается для выполнения потока при пустой очереди сообщений;

- для создания интерфейсного потока необходимо вызвать функцию, и в качестве аргумента передать ей имя класса, производного от `CWinThread`. Например: `pMyThrd=(CMyThrd *)AfxBeginThread(RUNTIME_CLASS(CMyThrd))`.

Исполняющая функция потока `CWinThread::Run()` содержит цикл обработки сообщений, и обычно не перегружается, так как данный цикл фактически является функцией потока. Этот поток по умолчанию используется MFC в качестве диспетчера получаемых сообщений. А вот для того, чтобы перехватывать сообщения перед тем как диспетчер сообщений отправит их функциям `TranslateMessage` и `DispatchMessage`, необходимо перегрузить метод `PreTranslateMessage()`.

Схема сообщений может быть реализована в классе потока, так же как и в любом классе, производном от `CCmdTarget`. Также имеется возможность воспользоваться специальным макросом `ON_WM_MESSAGE` для обработки сообщений, предназначенных только потоку, а не всему окну. Отправлять сообщения непосредственно потоку может метод `CWinThread::PostThreadMessage()`. Обработываются такие сообщения методом `CWinThread::HandleMessage()`.

Завершение работы потока

Обычно поток завершается, когда заканчивается выполняемая им функция. Эта функция завершает работу при получении сообщения `WM_QUIT`. Досрочное завершение потоков в любой момент времени осуществляется вызовом функции `AfxEndThread()`.

Синхронизация потоков

Введение потоков потребовало создания специального механизма, получившего название "синхронизация". Одной из основных причин использования синхронизации является тот факт, что два или больше потока могут одновременно требовать доступа к так называемому разделяемому ресурсу, который реально не может быть предоставлен сразу нескольким потокам. Поэтому если предоставить ресурс конкретному потоку, остальные должны быть приостановлены. В Win32 для синхронизации потоков создана специальная система. В MFC механизм синхронизации поддерживается базовым классом `CSystemObject`, производными от него классами `CCriticalSection`, `CEvent`, `CMutex`, `CSemaphore`, а также двумя вспомогательными классами `CSingleLock` и `CMultiLock`.

Проиллюстрируем процесс многозадачности на простом примере. При помощи AppWizard создадим проект — диалоговое окно с кнопкой "Выполнить". Щелчок на этой кнопке одновременно запускает два процесса. В течение некоторого интервала времени они будут выполняться, а затем практически одновременно завершатся, и появившиеся `MessageBox` сообщат об этом.

1. Запустите AppWizard и выберите **File\New** в закладке **Projects**. Установите **Projects name** — **"Many"**, **Location** — **"путь к рабочему каталогу"**, **MFC AppWizard.exe** и подтвердите операции, нажав **OK**. Далее **a type of application** установите **"Dialog based"**, **\Finish** и еще раз **OK**.

2. Уберите надпись **"TODO:..."**, нанесите на диалоговое окно кнопку **Button** с названием **"Выполнить"**, запустите на выполнение и получите окно, как показано на рис.1.

Теперь "оживим" наш проект.

3. Делаем щелчок правой кнопкой мыши в любом месте диалоговой панели. Выбираем **ClassWizard**. В закладке **Message Maps**, в **Object Ids** устанавливаем **IDC_BUTTON1** и увязываем с **Messages** — **BN_CLICKED** и щелчок на **Add Function...** Подтверждаем внесенные изменения — **OK**. Далее делаем щелчок на **Edit code** и вместо фразы **// TODO: Add your control notification handler code here** создаем в файле **"ManyDlg.cpp"** следующий обработчик щелчка левой кнопкой на кнопке:

```
void CManyDlg::OnButton1()
{
    AfxBeginThread(proc1, this);
    AfxBeginThread(proc2, this);
    AfxMessageBox("Threads");
}
```

Основу кода составляет функция **AfxBeginThread**, которая, как вы уже знаете, и запускает процесс. Первый обязательный параметр — это указатель на функцию потока (в данном случае — `proc1` и `proc2`). Второй параметр устанавливает приоритет выполнения функции, и если явно указать — `this`, то функции потоков будут иметь такой же приоритет, что и приложение. Таким образом, обработчик `OnButton1()` запустит эти две функции в виде потоков, а сам продолжит свое выполнение, сообщив об их запуске при помощи **AfxMessageBox**.

4. Перед обработчиком добавляем тексты исполняющих функций (процедур потока), которые и определяют наполнение двух потоков, т.е. дополнение к объектам класса **CString str** с начальными значениями — сообщениями о завершении потоков **"End of Process1,2="**. Для потока 1 это первые пять букв латинского алфавита, для потока 2 — последние пять букв:

```
UINT proc1(LPVOID Param)
{
    CString str="End of Process1="; char c;
    for(c='a';c<='e';c++)
    {
        str+=c;
        Sleep(1000);
    }
    AfxMessageBox(str);
    return 0;
}

UINT proc2(LPVOID Param)
{
    CString str="End of Process2="; char c;
    for(c='v';c<='z';c++)
    {
        str+=c;
        Sleep(1000);
    }
    AfxMessageBox(str);
    return 0;
}
```

В самое начало файла нужно добавить прототипы этих двух функций, например, так:

```
// CAboutDlg dialog used for App About
UINT proc1(LPVOID Param);
UINT proc2(LPVOID Param);
class CAboutDlg : public CDialog
{
    ...
}
```

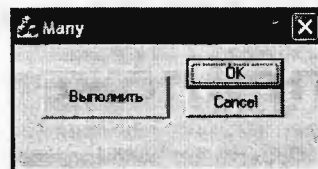


Рис. 1



Таким образом, суть двух потоков состоит в следующем — добавить объекту-строке str класса CString, сообщаемому о завершении потоков, в первом случае — пять первых букв латинского алфавита, и пять последних — во втором случае.

5. Теперь отобразим выполнение главного потока — реакцию на щелчок левой кнопкой мыши на диалоговом окне. Для этого в файле "ManyDlg.cpp" в карту сообщений класса CManyDICg, производного от CDialog, добавляем макрос ON_WM_LBUTTONDOWN(). В конец файла "ManyDlg.cpp" добавляем сам обработчик:

```
void CManyDlg::OnLButtonDown(UINT nFlags, CPoint point)
{
    AfxMessageBox("Main Process");
    MessageBeep(-1);
}
```

В файл "ManyDlg.h" добавляем декларацию обработчика:

```
afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
//))AFX_MSG
DECLARE_MESSAGE_MAP()
};
```

Рис. 2

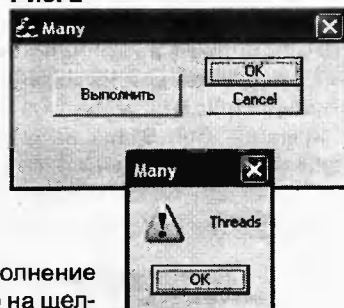
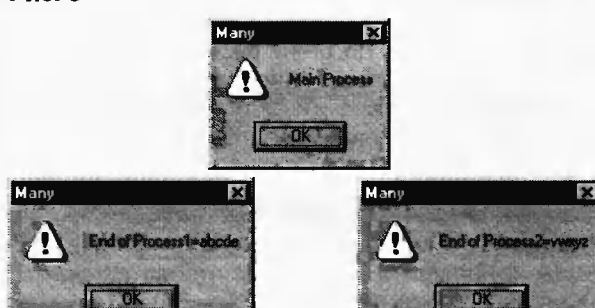


Рис. 3



Работа

Запускаем проект, щелкаем левой клавишей мыши на кнопке "Выполнить" и получаем сообщения, как показано на рис.2. Закрываем окно "Threads", щелкаем левой кнопкой мыши на поверхности окна Many, получив тем самым сообщение о выполнении основного процесса (со звуковым сигналом). И после этого, благодаря задержке в выполнении функций Proc1 и Proc 2, через некоторый временной интервал появляются сообщения о завершении работы этих функций (рис.3).

Литература

1. Н.Секунов. Самоучитель Visual C++ 6.0. — СПб.: BHV-Санкт-Петербург, 2000.
2. С.Глушаков и др. Программирование на Visual C++ 6.0. — Харьков, ФОЛИО, 2002.

Borland C++ Builder 6 для начинающих

(Продолжение. Начало в №№7-10/2004)

О.ВАЛЬПА,

E-mail: sandh@narod.ru

Теперь рассмотрим инструменты отладчика, которые могут нам при отладке программ.

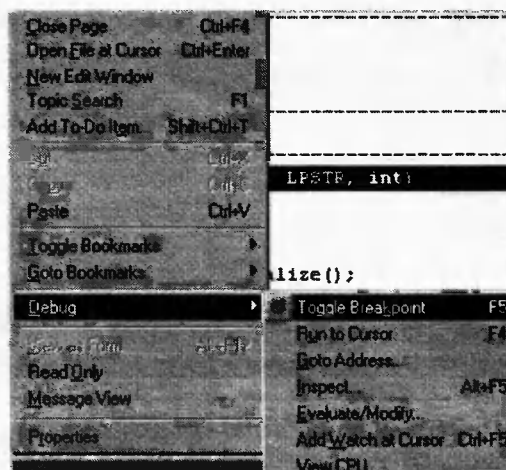
Инструменты отладчика

Основными инструментами отладки любой программы являются команды выполнения программы по шагам и установки точек останова. Естественно, что такие команды имеются и в среде разработки Borland C++ Builder 6. Команды пошагового выполнения программы вызываются из главного меню в группе Run или с помощью "горячих" клавиш. Список этих команд с соответствием "горячим" клавишам и описанием назначения приведен в таблице.

Команда	"Горячие" клавиши	Назначение
Step Over	F8	Пошаговое выполнение программы без захода в вызываемые функции и подпрограммы
Trace Into	F7	Пошаговое выполнение программы с заходом в вызываемые функции и подпрограммы
Trace to Next Source Line	Shift+F7	Перейти к следующей исполняемой строке команды
Run to Cursor	F4	Выполнить команды до места расположения курсора
Run Until Return	Shift+F8	Выполнить команды до выхода из выполняемой функции и остановиться на операторе, следующем за этой функцией
Show Execution Point	-	Переместить курсор на следующую исполняемую строку команды

Команды установки точек останова и некоторые другие вызываются из контекстного меню инспектора кода через раскрывающийся список команды Debug (рис.7), а также с помощью "горячих" клавиш.

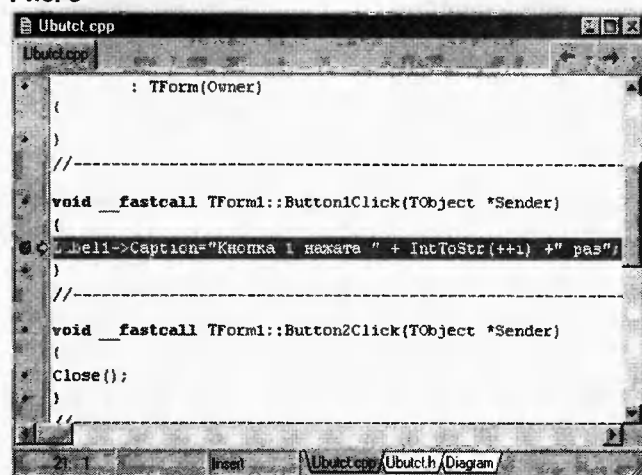
Рис. 7



Попробуйте применить эти команды на практике, используя нашу программу. Поместите курсор в инспекторе объектов на строку Label1->Caption="Кнопка 1 нажата " + IntToStr(++i) + " раз"; и нажмите клавишу F5. При этом строка окрасится в красный цвет, что свидетельствует о том, что на ней установили точку останова. Теперь при выполнении программы произойдет автоматический останов

на этой строке. Проверьте это, выполнив команду **Run** с помощью клавиши **F9**. Программа запустится, и на экране появится ее форма. Но стоит щелкнуть левой кнопкой мыши по кнопке с названием **Кнопка 1**, как программа остановится, и откроется окно инспектора кода (рис.8) с зеленой стрелкой напротив нашей строки.

Рис. 8



Теперь, нажимая клавишу **F8** на клавиатуре, можно по шагам продолжить выполнение программы. Для прерывания программы необходимо нажать комбинацию клавиш **Ctrl+F2**. Продолжить выполнение программы можно с помощью клавиши **F9**, а затем для ее завершения щелкнуть по кнопке **Выход** в окне программы. Попробуйте на практике выполнить и другие команды отладчика, чтобы понять их назначение.

Помимо команд отладки, среда разработки позволяет просмотреть переменные, используемые в программе, состояние регистров процессора, процессы, происходящие в программе, и т.п. Для этих целей используются окна отладчика.

Окна отладчика

Доступ к окнам отладчика производится через пункт главного меню **View→Debug Windows**. При этом откроется окно (рис.9), в котором расположены кнопки открытия окон отладчика.

Команда **Breakpoints**

Рис. 10



открывает окно, в котором отображаются и редактируются точки останова (рис.10).

Из этого окна через контекстное меню, вызываемое щелчком правой кнопки мыши, можно осуществлять добавление, удаление и редактировать точек останова (рис.11).

Рис. 9

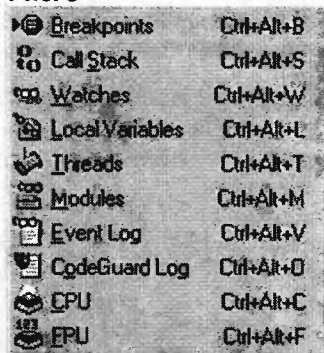
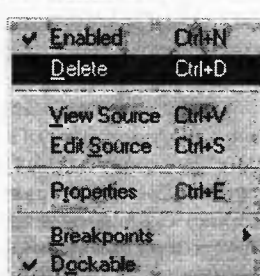


Рис. 11



Для вызова данной команды отладчика можно использовать "горячие" клавиши **Ctrl+Alt+B**. Остальные команды можно вызывать аналогично. Комбинация "горячих" клавиш для каждой из команд изображена правее самой команды (рис.9).

Команда **Call Stack** вызывает окно просмотра стека программы (рис.12). Данное окно позволяет увидеть вызовы функций, производимые в программе в момент ее отладки.

Рис. 12

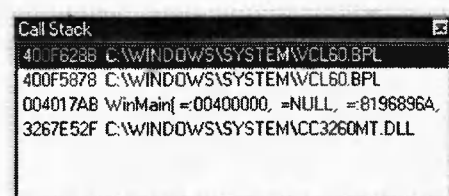
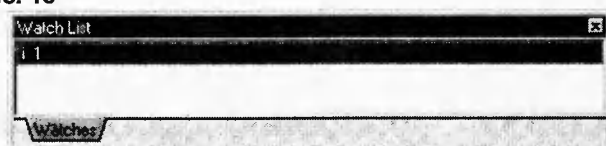


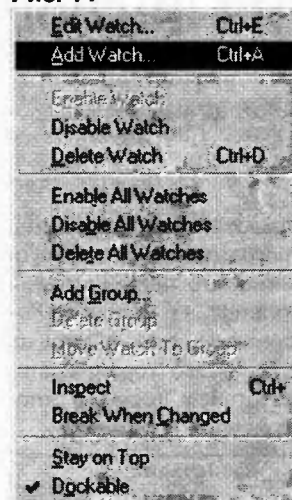
Рис. 13



Команда **Watches** открывает окно переменных (рис.13), в котором можно просматривать переменные, используемые в программе во время ее работы.

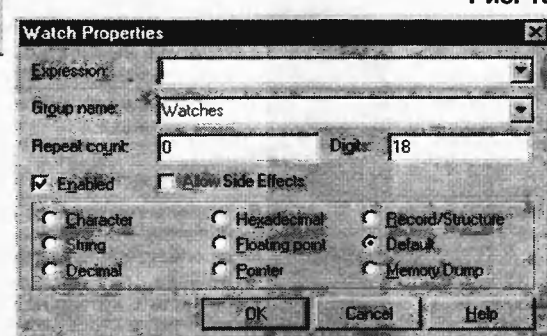
Из этого окна щелчком правой кнопки мыши можно вызвать контекстное меню (рис.14), с помощью которого появляется возможность редактировать, добавлять, активизировать, делать невидимыми и удалять переменные из окна отображения переменных.

Рис. 14



Окно редактирования свойств переменных **Watch Properties** (рис.15) вызывается из контекстного меню командой **Edit Watch** или с помощью клавиш **Ctrl+E**. Оно позволяет выбрать отображаемые переменные (**Expression**), разбить переменные на группы (**Group name**), задать количество элементов массива для отображения (**Repeat count**), ограничить разрядность отображаемых значений (**Digits**), разрешить отображение переменной (**Enabled**), разрешить отображение вы-

Рис. 15

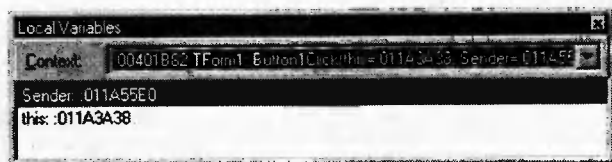




ражений (**Allow Side Effects**) и выбрать один из девяти форматов отображения переменных.

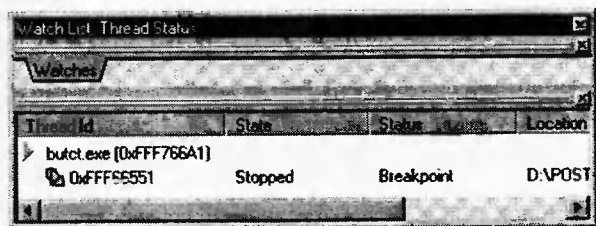
Команда **Local Variables** показывает текущие локальные переменные в отладочном режиме работы (рис.16).

Рис. 16



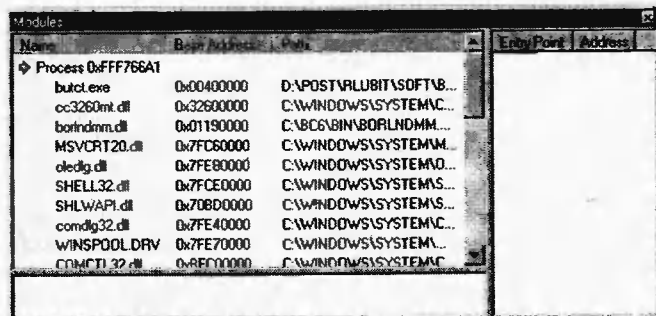
Команда **Threads** отображает окно процессов, происходящих в режиме отладки программы (рис.17). В частности, в нем можно увидеть имя исполняемой программы, состояние процесса отладки и адрес точки останова.

Рис. 17



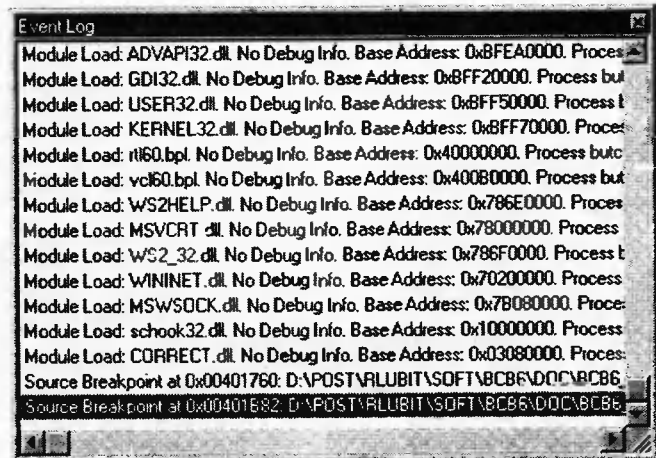
Команда **Modules** вызывает окно, в котором можно просмотреть все программные модули, их адреса и точки входа (рис.18).

Рис. 18



Команда **Event Log** отображает окно регистрации событий программы (рис.19).

Рис. 19



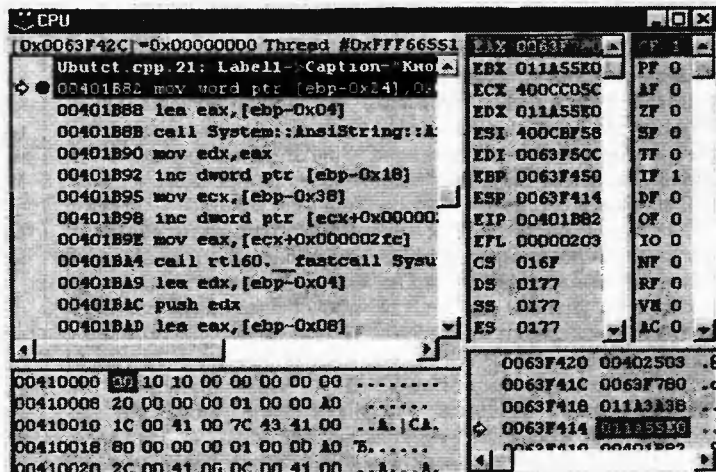
Команда **Code Guard Log** вызывает окно сообщений (рис.20), сгенерированных специальной утилитой отладки CodeGuard.

Рис. 20



С помощью команды **CPU** вызывается окно, в котором отображается работа программы в машинных кодах и содержимое регистров процессора (рис.21). Если программа не запущена, данная и следующая команды недоступны.

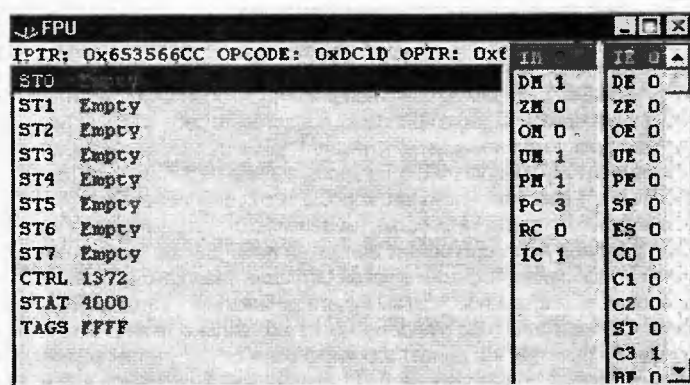
Рис. 21



Наконец, команда **FPU** вызывает окно для просмотра состояния математического сопроцессора компьютера (рис.22).

Для тренировки и запоминания я настоятельно рекомендую читателям с помощью "горячих" клавиш поочередно открыть эти окна во время пошагового выполнения программы.

Рис. 22



На веб-странице журнала в приложении к статьеложены файлы проекта рассмотренной в качестве примера программы.

Литература

1. Подбельский В.В. ЯЗЫК СИ++. — М.: Финансы и статистика, 2003.

(Продолжение следует)



ИССЛЕДОВАНИЕ ПРОГРАММ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КРЭКИНГА

(Продолжение. Начало в №№8-12/2003, 1-10/2004)

Слишком хорошо — тоже нехорошо

Как вы уже успели убедиться, обращение с брейкпойнтами — занятие далеко не такое простое, как может показаться с первого взгляда. В предыдущей главе мы разбирались в том, почему точки останова иногда не делают того, что мы от них ожидаем, значительная же часть этой главы посвящена противоположной проблеме — что делать, если точки останова срабатывают слишком часто. После всего рассказанного в предыдущей главе, такая ситуация может показаться вам невероятной, но представьте себе, такое тоже бывает.

Если вы входите в число поклонников отладчика SoftICE, то не могли не заметить одну милую особенность этого инструмента — если вы поставили брейкпойнт непосредственно на системную функцию, SoftICE будет «всплывать» при каждой попытке исполнить эту функцию, независимо от того, внутри какого процесса функция была вызвана. Такое поведение отладчика, несомненно, бывает полезным при отладке драйверов, хук-процедур и самой операционной системы. Но наша беда (или счастье — это уж с какой стороны посмотреть) в том, что мы отлаживаем «обычную» программу, и постоянные выпадения в отладчик по совершенно неинтересным поводам — это далеко не то, о чем мы мечтали. Вот если бы удалось сделать так, чтобы брейкпойнты работали внутри только одного процесса... Те, кто начал знакомиться с SoftICE по пакету NuMega DriverStudio 2.6, скорее всего, не увидят в этом никакой проблемы: BPX <имя функции> IF PID=<ID_нужного_процесса> — и дело в шляпе. Набрав это «заклинание», они, в общем-то, будут полностью правы, ибо это вполне хороший и самый короткий путь к цели. И я никогда не скажу дурного слова о тех, кто последует этим путем. Однако этот путь — далеко не единственный из возможных, и если вам интересны иные способы решения проблемы постановки брейкпойнта — добро пожаловать в музей истории SoftICE.

Мое общение с этим отладчиком началось с версии 3.23 для Windows 9x, случайно обнаруженной на свежем «хакерском» диске. Пусть по нынешним меркам тот СофтАйс совершенно не производит впечатления — тогда это был шедевр! Впрочем, этот шедевр был не без недостатков, самым главным из которых была его «неразборчивость» при срабатывании брейкпойнтов. Как я уже отмечал, поставить брейкпойнт на функцию WinAPI, срабатывающий исключительно внутри нужного процесса (по-научному брейкпойнты с такими свойствами называются «address-context sensitive», они же «контекстно-зависимые»), напрямую было невозможно. Чтобы почувствовать, насколько серьезной была проблема, представьте себе следующую картину: бряк, поставленный на функции чтения из реестра (которая называется RegQueryValue[Ex], надеюсь, вы еще не забыли мой «поминальник»), работает настолько хорошо, что отлавливает абсолютно все попытки чтения из реестра, независимо от того, выполняет их отлаживаемая вами программа или какая-либо другая. Любые попытки осмысленной отладки в такой ситуации заведомо обречены на провал, единственное, чем вы будете заниматься — это нажиманием клавиш Ctrl-D, ибо Windows настолько сильно любит читать данные из реестра, что делает это много раз в секунду (если хотите своими глазами посмотреть на эту странную любовь, RegMon вам в этом поможет). Так что же, поставив бряк на RegQueryValueEx и получить удовлетворительный результат совсем никак невозможно? Как бы не так! Если нельзя, но очень хочется, то все-таки можно.

Начнем с небольшого, но очень важного определения. **Адресный контекст** (он же контекст процесса) — это все виртуальное адресное пространство, выделенное данному процессу. «Военная хитрость», при помощи которой мы «проагрегидим» контекстно-независимый брейкпойнт до контекстно-зависимого, основана на том факте, что хотя SoftICE и игнорирует контекст в момент срабатывания контекстно-независимых брейкпойнтов (в число которых входят и брейкпойнты на функции WinAPI), сам текущий адресный контекст от этого никуда не исчезает. И потому в момент

срабатывания условного брейкпойнта отладчик вполне способен прочитать любые данные из существующего в этот момент виртуального адресного пространства. Проще говоря, если брейкпойнт сработал внутри программы X, то отладчик сможет «увидеть» адресное пространство программы X вместе со всеми данными, содержащимися в этом пространстве, а заодно и содержимое всех регистров, каким оно было в момент срабатывания брейкпойнта. А если отладчик «видит» все адресное пространство процесса и способен читать исполняемый код программы, значит, можно попытаться идентифицировать процессы по особенностям их исполняемого кода» выглядит весьма прозаично — нужно забраться при помощи шестнадцатеричного редактора внутрь секции кода (впрочем, для идентификации можно использовать и любые другие заведомо не изменяющиеся при работе программы данные), выдернуть оттуда первый попавшийся DWORD (назовем его **My_DWORD**) и запомнить виртуальный адрес (**My_Addr** соответственно), по которому этот DWORD находился. Дальнейшие операции, выполняемые уже в отладчике, ненамного сложнее — BPX имя_функции WinAPI IF (*My_Addr)==My_DWORD. Все.

Думаю, с пониманием того, что делает эта команда, ни у кого сложностей не возникло — мы соорудили условную точку останова, которая в момент срабатывания проверяет, не лежит ли по адресу **My_Addr** двойное слово, равное **My_DWORD**. И если такое двойное слово по нужному адресу обнаруживается, то отладчик приостанавливает исполнение программы. Поскольку случайно встретить пару программ, у которых в секции кода по одним и тем же виртуальным адресам находились бы одинаковые DWORD'ы, вряд ли возможно, такой простейший способ различения процессов в абсолютном большинстве случаев отлично срабатывает. Сложности могут возникнуть только в двух случаях: при отладке упакованных программ и если нужно параллельно отлаживать два экземпляра («экземпляр» следует понимать как «instance») одного и того же приложения — поскольку код обоих процессов идентичен, в общем случае различить их по содержимому адресного пространства затруднительно.

В настоящее время применение описанного выше метода для определения, в контексте какого из процессов сработал брейкпойнт, в общем-то, не требуется — в современных версиях SoftICE (вообще, сам отладчик SoftICE, особенности работы с ним и различия между разными его версиями — это отдельная большая тема, которая будет закрыта не раньше, чем прекратится развитие самого SoftICE) эта проблема решается безо всяких ухищрений. Однако поскольку особенности отдельных инструментов имеют весьма слабое отношение к теоретическим вопросам крэкинга, которым посвящена данная работа, я в дальнейшем не буду акцентировать внимание на этих особенностях — разобравшись в предлагаемом материале, вы сами найдете способ с максимальной эффективностью использовать эти особенности. Для того же, чтобы избежать путаницы при изложении материала этой главы, мы будем считать, что все брейкпойнты, о которых ниже пойдет речь, являются контекстно-зависимыми, то есть работают исключительно в рамках того процесса, в котором они установлены. Таким образом, мы «уравняем в правах» SoftICE и отладчики третьего кольца защиты («ring3 debuggers»), к которым, в частности, относится OllyDebug.

Сам по себе создание условного брейкпойнта, срабатывающего исключительно при появлении по некоему адресу известного значения — прием весьма часто употребляемый и полезный во множестве ситуаций. Самое известное из применений этой техники — «остановить программу, если переменная равна некоему значению» — мы рассматривать не будем по причине его крайней банальности, такие вещи вы способны проделывать самостоятельно, и, возможно, даже с закрытыми глазами. Куда менее очевидна возможность использования брейкпойнтов, для того чтобы «притормозить» упакованную программу сразу после начала ее исполнения. Возможно, вы



уже встречали в статьях по крэкингу сокращение "OEP", которое расшифровывается как Original Entry Point, "оригинальная ("истинная") точка входа". Возможно, вы также читали о том, что распаковка скаченных исполняемых файлов включает в себя поиск адреса этой самой OEP. Если же вы ничего такого еще не читали и не встречали, то вам необходимо запомнить следующие базовые сведения:

1. Entry point — это точка, с которой начинается исполнение программы после загрузки. Адрес точки входа в Win32-приложении хранится в PE-заголовке исполняемого файла.

2. Абсолютное большинство упаковщиков и навесных защит изменяют значение адреса точки входа таким образом, чтобы управление передавалось распаковщику или защитному модулю соответственно.

3. Original entry point — это entry point исполняемого файла до того, как файл был сжат/зашифрован, и значение точки входа было модифицировано упаковщиком.

4. Для успешной распаковки программы в общем случае требуется узнать адрес OEP, остановить программу в тот момент, когда распаковщик передаст управление на OEP и в этот момент снять дампы со всех секций процесса. Также довольно часто после снятия дампа приходится восстанавливать таблицу импорта, в некоторых случаях может потребоваться ручная правка параметров секций.

Один из приемов отыскания OEP заключается в следующем — нужно поставить брейкпоинты на функции WinAPI таким образом, чтобы один из них сработал заведомо недалеко от точки входа в программу. Выяснив, откуда был произведен этот первый вызов WinAPI'шной функции, вы сделаете первый (и самый важный) шаг на пути к адресу OEP. После этого обратной трассировкой в уме или каким-либо иным методом вы будете "раскручивать" программу, как бы заставляя ее исполняться "задом наперед", и таким образом шаг за шагом подбираться к OEP. Идея, лежащая в основе этого метода поиска OEP, крайне проста — современные компиляторы устроены так, что в коде абсолютного большинства программ неподалеку от точки входа содержится, как минимум, один вызов функции WinAPI (обычно такой функцией является GetModuleHandle или GetCommandLine).

Другой важный момент состоит в том, что практически каждый компилятор генерирует в начале программы специфический ("начальный" (startup) код, который производит некие действия по инициализации данных, необходимых для работы стандартных библиотек. Причем, этот код у разных компиляторов (а часто и у разных версий одного и того же компилятора или даже при разных опциях компиляции) имеет достаточно специфический и легко узнаваемый вид, и повлиять на содержание этого startup-кода разработчик программы обычно не может (или может, но в ограниченных пределах). Поэтому, чтобы вам было легче искать OEP, я настоятельно рекомендую изучить, какой код помещают в начало программы наиболее распространенные компиляторы — так вы быстро научитесь определять OEP исключительно по внешнему виду кода, находящегося в окрестностях первого вызова функции WinAPI.

Однако как отличить вызов нужной нам функции из только что распакованной программы от всех прочих вызовов этой же функции, которые могут выполняться в процессе работы распаковщика? Как раз для этого вы можете воспользоваться проверкой значения, находящегося по некому известному адресу внутри кода программы (обозначим этот адрес буквой А). Извлечь нужное значение дампером из запущенной программы обычно не составляет никакой сложности. Поскольку известные мне упаковщики распаковывают код в направлении от меньших адресов к большим (и с достаточной точностью можно предположить, что так работает большинство распаковщиков), лучше выбирать эталонное значение ближе к концу программы. После того как такое значение у вас будет, нужно лишь поставить условную точку останова на нужную функцию, а в качестве дополнительного условия указать равенство значения по адресу А эталонному значению, которое вы извлекли из распакованной программы. В результате ваш брейкпойнт сработает не раньше, чем будет распакована соответствующая часть программы, так что довольно велика вероятность того, что, продолжая отладку в пошаговом режиме, вы вернетесь в код программы, а не в недра распаковщика.

Наблюдая за работой запакованных программ, вы могли заметить, что после распаковки они всегда располагаются в памяти по одним и тем же адресам. Применив против такой программы ка-

кую-нибудь утилиту вроде ProcDump, вы даже можете прочитать параметры секций программы, в частности — начальный адрес и размер. После этого вполне естественным кажется вопрос: "А что если проверять, принадлежит ли адрес возврата, лежащий на стеке, промежутку адресов, в котором расположен код программы?" И вопрос этот отнюдь не праздный. Дело в том, что нередки упаковщики и навесные защиты, в которых блок, выполняющий дешифровку и распаковку кода, располагается в области, не пересекающейся с той, в которой в итоге будет расположен исполняемый код программы. Поэтому, поставив брейкпойнт на функцию и в качестве условия указав что-нибудь вроде ([esp]>401000) && ([esp]<501000), вы добьетесь того, чтобы ваш брейкпойнт активизировался лишь в том случае, если при выходе из функции предполагается возврат в код отлаживаемой программы. Данный метод, так же как и описанный выше, может использоваться для поиска OEP через обратную трассировку от первого вызова функции WinAPI. Однако этим возможности условных точек останова, проверяющих адрес возврата, отнюдь не ограничиваются.

Очень часто возникает необходимость посмотреть параметры вызова некой часто используемой функции в тех случаях, когда вызов этой функции был сделан из одной или нескольких точек, и при этом проигнорировать все прочие вызовы (которых может быть очень много). Если решать задачу "в лоб", то необходимо найти все ссылки на данную функцию и установить по брейкпойнту на каждый интересующий нас вызов. Обычно такой подход вполне удовлетворителен, но мы не будем искать легких путей и посмотрим, как ту же задачу можно решить при помощи одного-единственного брейкпойнта. Вы уже знаете, что для распознавания, откуда был сделан вызов функции, можно использовать адрес возврата, лежащий на вершине стека, и поэтому без труда напишете соответствующее условие для брейкпойнта. Такое условие может выглядеть следующим образом:

```
([esp]==ret_addr1) || ([esp]==ret_addr2) || ...
```

где ret_addr1 и ret_addr2 — адреса возврата. Все это достаточно очевидно, и вы можете задать вопрос, зачем нужно было изобретать очередную велосипед, если традиционный подход дает ничуть не худшие результаты? Первая причина — "человеческий фактор", поскольку работать с большим количеством точек останова не всегда удобно даже в насквозь визуальном OllyDebug, а уж "рулить" десятком-другим брейкпойнтов в SoftICE — занятие, как называется, на любителя. Так что, если есть возможность значительно уменьшить число брейкпойнтов и облегчить себе жизнь, почему бы этой возможностью не воспользоваться? Кроме того, перед вами может встать задача, обратная вышеприведенной — отслеживать все вызовы процедуры, за исключением нескольких. И в этом случае сформировать строку с условием вида

```
([esp]!=ret_addr1) && ([esp]!=ret_addr2) && ...
```

в которой ret_addr1, ret_addr2 — адреса возврата, при которых точка останова не должна срабатывать, представляется гораздо более простым, чем искать в коде программы все подзвучиваемые ссылки на интересующую вас процедуру и "обвешивать" их брейкпойнтами.

Однако более важным представляется применение условных брейкпойнтов для обнаружения неочевидных вызовов функций. Я уже рассказывал в предыдущей главе о приемах, при помощи которых разработчик защиты может вызывать процедуры неявным образом, из-за чего такие обращения к процедурам становятся "невидимы" для дизассемблеров и прочих инструментов анализа "мертвого кода". В качестве дополнительного средства маскировки авторами защит могут параллельно использоваться как обычные вызовы процедур, так и неочевидные; я встречал такое в приложении к MessageBoxA и к функции чтения из реестра, но вообще эта техника может быть применена к любой достаточно часто вызываемой процедуре, используемой в защитных механизмах. В результате, в дизассемблированном коде мы увидим несколько ничем не примечательных явных вызовов, но, скорее всего, не заметим самого интересного. Поставив "обычный" брейкпойнт на вызываемую функцию, мы можем столкнуться с тем, что эта функция вызывается десятки раз, и поэтому тоже не сможем определить, задействована ли эта функция в защитном механизме — интересные нас неявные вызовы потеряются среди десятков и сотен вызовов явных. Вот если бы был какой-нибудь способ отделить явные вызовы от неявных...

(Продолжение следует)

А.КАЛЮТОВ,
E-mail: k90012@hotmail.com

Мозаичные изображения

Мозаичные изображения получили свое название из-за того, что они, подобно мозаике, состояются из множества различных изображений меньшего размера. Мозаичными являются, например, карты больших участков поверхности Земли, полученные со спутников или самолетов. Хотя существуют широкоугольные фотокамеры (к примеру, камеры типа "рыбий глаз", дающие угол обзора 180°), но они имеют свои ограничения, и, кроме того, довольно сильно искажают изображение. Вообще говоря, потребность в сборке одной большой картинке из множества малых возникает всякий раз, когда линейные или угловые размеры интересующего объекта превышают возможности камеры, сканера и т.п.

После сбора материала необходимо "склеить" результирующее изображение из множества перекрывающихся кусков. Для человека эта проблема не представляет особой сложности, но при попытке алгоритмизации процесса склейки возникают сложности.

Во-первых, перекрывающиеся участки отдельных кусков, как правило, не являются абсолютно идентичными, поэтому обычные алгоритмы поиска одинаковых последовательностей пикселей здесь не подойдут.

Во-вторых, уже для изображений средних размеров вычислительная сложность алгоритма с простым перебором всех возможных взаимоположений становится слишком велика. Например, для двух фотографий размером 600x400 в худшем случае (т.е. когда искомое положение окажется последним) придется проверить примерно $(399+400) \cdot 600 = 479400$ взаимных положений изображений. Поскольку в каждом участке перекрытия в среднем используется 1/4 (600x400) пикселей, очевиден порядок числа операций, которые нужно выполнить для каждой склейки. А ведь изображения могут быть большими по размеру, и их может быть много...

Сформулируем задачу с точки зрения математики. Т.к. изображение представляет собой просто последовательность троек чисел RGB, определяющих для каждой точки 24-bit цвет, величина

$$D_{12}^i = |R_1^i - R_2^i| + |G_1^i - G_2^i| + |B_1^i - B_2^i|$$

определяет различия в цвете i -й точки двух изображений, пронумерованных соответственно 1 и 2. Тогда для участка перекрытия из N пикселей величина $D = \frac{1}{N} \sum D_{12}^i$ (сумма по всем точкам этого участка) означает *среднюю разницу* двух изображений. Таким образом, необходимо найти такое взаимное положение двух изображений, при котором достигается минимальное значение величины D . Очевидно, чтобы склеить всю картинку, такую операцию следует повторить для любых двух соседних элементов изображений.

Алгоритм быстрого поиска такого положения заключается в следующем.

На первом этапе из пары исходных картинок генерируется последовательность уменьшающихся изображений

с коэффициентом уменьшения, равным двум. Для получения оптимального результата достаточно, чтобы линейные размеры минимального изображения в получившейся последовательности были порядка 70...80 пикселей. Такое значение размера выбрано из соображений баланса быстродействия и качества результата. Увеличивая эту величину, тем самым можно улучшать качество склейки изображения. Время вычислений при этом квадратично увеличивается.

На втором этапе осуществляется поиск начального наилучшего взаимного положения двух минимальных изображений, полученных на предыдущем этапе. Этот поиск производится прямым перебором всех возможных положений и вычислением для каждого участка перекрытия средней разницы D .

На последующих этапах происходит уточнение наилучшего положения. Т.к. каждое последующее изображение в последовательности в два раза больше предыдущего, достаточно проверить примерно 25 позиций относительно наилучшего положения, найденного на предыдущем этапе (рис.1).

Описанный алгоритм требует существенно (на порядок и более) меньше вычислений, чем прямой перебор. Его недостатком является отсутствие гарантии по определению действительно оптимального положения двух изображений — элементов мозаики. Иными словами, в отличие от алгоритма полного перебора, существует ненулевая (хотя и достаточно малая) вероятность ошибки.

Рассмотрим некоторые технические моменты создания программы. При реализации алгоритма удобно использовать объектно-ориентированное программирование, в частности, следующий класс:

```
class CPicture{
public:
    unsigned char Level;
    float difference; // средняя разница
    int x1size,y1size,x2size,y2size; //размеры изображений
    int xbest,ybest; // координаты наилучшего положения
    BYTE* IM1; // указатели на исходную пару картинок
    BYTE* IM2; //
    BYTE* im1; // уменьшенная в 2 раза картинка
    BYTE* im2; //
    CPicture* Nextpic;
    CPicture* Previouspic;

    void GetReduceImage(int xbs, int ybs, char ni);
    float CulcLDifference(int xi, int yi);
    CPicture(int xs1,int ys1,int xs2,int ys2, BYTE* pl1, BYTE* pl2);
    ~CPicture();
};
// функция вычисления средней разницы (см. текст статьи)
float CPicture::CulcLDifference(int xi,int yi){
    int xlength, ylength,fs=0;
    int x1r,y1r,x2r,y2r;
    BYTE* p1; BYTE* p2;

    x1r=xi;    y1r=(yi<=0)?0:yi;
    x2r=0;    y2r=(yi>=0)?0:-yi;
    xlength=3*min(x1size-x1r,x2size);
    ylength=min(y1size-y1r,y2size-y2r);
    p1=im1+(y1r*x1size+x1r)*3;
```



Рис. 1

```
p2=im2+(y2r*x2size+x2r)*3;
for (int k=0;k<ylength;k++){
    for (int l=0;l<xlength;l++)    fs+=abs(p1[l]-p2[l]);
    p1+=3*xlsize;
    p2+=3*x2size;
}
return ((float) fs/(xlength*ylength));
}
// конструктор класса
CPicture::CPicture(int xs1,int ys1,
    int xs2,int ys2, BYTE* p11, BYTE* p12){
    IM1=p11;IM2=p12;
    im1=(BYTE*)malloc(3*(xs1/2)*(ys1/2));
    im2=(BYTE*)malloc(3*(xs2/2)*(ys2/2));
    GetReduceImage(xs1,ys1,1);
    GetReduceImage(xs2,ys2,2);
    xbest=xlsize/2;
    ybest=ylsize/2;
    difference=1000;
};
// деструктор
CPicture::~CPicture(){
    delete im1;
    delete im2;
};
// функция уменьшения изображения : здесь следует использовать
// алгоритм качественного уменьшения - с усреднением пикселей
// последний аргумент - выбор изображения (ni=1 - первое,...)
void CPicture::GetReduceImage(int xbs, int ybs, char ni){
    int count=0;
    BYTE* p1=(ni==1)?IM1:IM2;p2,p3=(ni==1)?im1:im2;
    BYTE* pfirst=p1;
    unsigned int tr,tg,tb;
    for (int y=0;y<ybs/2-2;y++){
        p1=pfirst+6*xbs*y;
        p2=p1+3*xbs;
        for (int x=0;x<xbs/2;x++){
            tr=(p1++)+(p2++);
            tg=(p1++)+(p2++);
            tb=(p1++)+(p2++);
            tr+=*(p1++)+(p2++);
            tg+=*(p1++)+(p2++);
            tb+=*(p1++)+(p2++);
            p3[count++]=(tr/4);
        }
    }
}
```

```
for (int y=0;y<ybs/2-2;y++){
    p1=pfirst+6*xbs*y;
    p2=p1+3*xbs;
    for (int x=0;x<xbs/2;x++){
        tr=(p1++)+(p2++);
        tg=(p1++)+(p2++);
        tb=(p1++)+(p2++);
        tr+=*(p1++)+(p2++);
        tg+=*(p1++)+(p2++);
        tb+=*(p1++)+(p2++);
        p3[count++]=(tr/4);
    }
}
```

Рис. 2

```
p3[count++]=(tr/4);
p3[count++]=(tg/4);
}
}
if (ni == 1){
    xlsize=xbs/2;ylsize=ybs/2;
}else{
    x2size=xbs/2;y2size=ybs/2;
}
}
```

При использовании этого класса, например, операция генерирования иерархической последовательности изображений выполняется следующим простым кодом:

```
// pBuf1, pBuf2 - указатели на исходную пару изображений
// с размерами соответственно xlYl и x2Y2
CPicture* Curpict= new CPicture(x1, y1, x2, y2, pBuf1, pBuf2);
Curpict->Level=0;
Curpict->Previouspic=Curpict;
// создание последовательности уменьшающихся изображений
// (получится своеобразный аналог русской матрешки)
while (minimum(Curpict->xlsize, Curpict->ylsize)>80){
    Curpict->Nextpic=new CPicture (Curpict->xlsize, Curpict->ylsize,
    Curpict->x2size, Curpict->y2size, Curpict->im1, Curpict->im2);
    Curpict->Nextpic->Previouspic=Curpict;
    Curpict=Curpict->Nextpic;
    Curpict->Level=Curpict->Previouspic->Level+1;
}
```

Полный текст программы, склеивающей изображение из произвольного числа кусков произвольных размеров, можно взять на сайте <http://radio-mir.com/> в приложении к статье или обратившись по моему e-mail.

И в заключение, пару слов о результатах тестирования. В качестве эталонной использовалась программа RapaVue Image Assembler v.2.02 [1], в которой есть функция «сшивки» (stitching) изображений. С этой же программой поставляется набор демонстрационных файлов-изображений. Результат склейки из четырех отдельных частей — картинка 1039x1345x24 bit — показан на рис.2,





время работы программы — 9 с, для PanaVue Image Assembler при той же конфигурации системы потребовалось порядка 40 с.

Недостаток описанного выше алгоритма — возможная ошибка при определении положения склеиваемых картинок — является общим для всех программ с полной автоматизацией процесса склейки. Отдельные изображения могут быть получены не в одинаковых условиях (например, фотографии ландшафта при меняющемся освещении), иметь зашумленность и т.д. Все это усложняет задачу склейки. Таким образом, для дальнейшей работы больше всего подходят изображения, полученные по возможности в идентичных условиях. Это означает прежде всего одинаковую ориентацию всех составляющих изображений относительно некоторой оси. Цифровые фотографии должны быть сделаны на достаточном удалении от объекта съемки для уменьшения искажений из-за перспективы.

Некоторые программы для создания мозаичных мегапиксельных изображений имеют интерактивный интерфейс, и взаимное положение отдельных элементов-изображений определяется пользователем вручную, путем

отыскания так называемых *контрольных точек* — изображений одной и той же точки или объекта. Эти точки играют роль “точек сшивания”, которые жестко задают положение двух соседних изображений друг относительно друга. Очевидно, для каждой пары изображений достаточно указать две контрольные точки. Одна из программ, в которой реализован этот метод — Panorama Tools [2], разработчик — Helmut Dersch. Это набор достаточно мощных утилит, свободно распространяющийся по лицензии GNU GPL. Для удобства работы есть визуальная оболочка — PTAssembler [3].

В Интернете можно найти сравнительно много ресурсов по рассматриваемой теме. По адресу <http://graphicssoft.about.com/od/panorama/> находится небольшая коллекция ссылок на соответствующее ПО, технические статьи и галереи мегапиксельных изображений.

Ссылки

1. <http://www.panavue.com/products/index.htm>
2. <http://www.tawbaware.com/ptasmblr.htm>
3. <http://www.tawbaware.com/maxlyons/pano12ml.htm>

Автоматическое проставление размеров файлов в HTML-документах

И. РОЩИН,

г. Москва,

ZXNet: 500:95/462.53

E-mail: bestview@mtu-net.ru

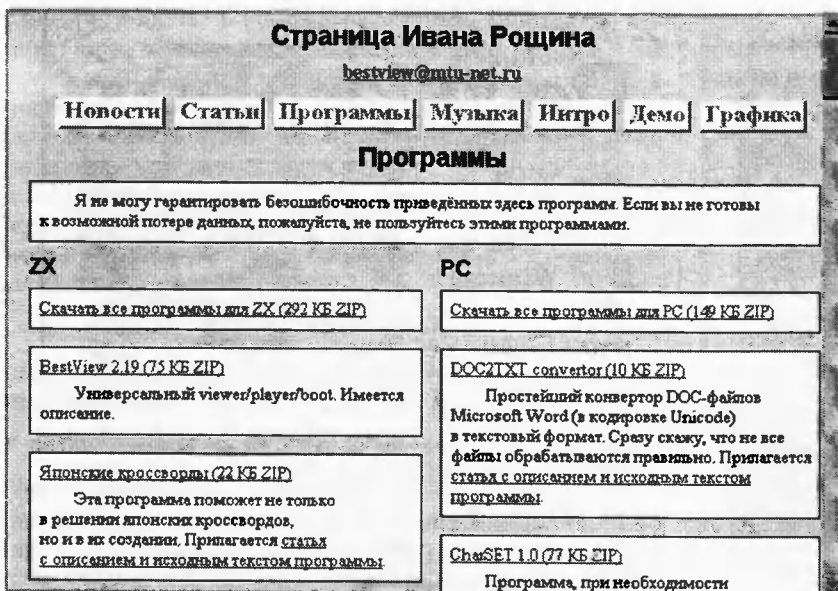
WWW: <http://www.ivr.da.ru>

Введение

При выкладывании на своей web-странице zip-архивов я указываю их длину в килобайтах (рис. 1, 2), так как эта информация может пригодиться посетителям (например, чтобы оценить время загрузки архива).

Значительного количества архивов, я оценил объем предстоящей работы по проставлению их новых размеров и задумался об автоматизации этого процесса. В результате была написана программа, о которой я и хочу рассказать в этой статье.

Рис. 1



Но то, что удобно для посетителей страницы, не всегда удобно для ее создателя :-). Каждый раз при добавлении нового архива или при изменении уже существующего мне приходилось вычислять его размер в килобайтах и добавлять эти сведения в HTML-документ, содержащий ссылку на архив. И вот однажды, после изменения зна-

Постановка задачи

На жестком диске моего компьютера находится локальная копия web-страницы, с которой я работаю. Эта копия включает в себя и все zip-архивы, размер которых требуется указывать. В HTML-документах я использую относительные адреса в ссылках на эти архивы, так что ссылки работают независимо от того, где расположена web-страница: у меня на жестком диске или в Интернете.

В HTML-документах я указываю размер архива либо в тексте ссылки (рис. 1), либо в атрибуте “title” ссылки (тогда размер отображается при наведении курсора на ссылку — рис. 2). Случаю, когда размер архива указан в тексте ссылки, соответствует примерно такой фрагмент HTML-документа (этот фрагмент взят как раз из документа, изображенного на рис. 1):

```
<p class="ms"><a href="all zx.zip">Скачать все программы для ZX (292 KB ZIP)</a></p>
```

Рис. 2 2.03.2004. Выпущена очередная версия BestView — 2.19. Отличия от версии 2.18:

 Скачать (75 KB ZIP)



Случаю, когда размер архива указан в атрибуте "title", соответствует примерно такой фрагмент (этот фрагмент взят как раз из документа, часть которого изображена на рис.2):

```
<p><b>2.03.2004.</b> Выложена <a href="prog/bv_2_19.zip"
title="Скачать (75&nbsp;КБ&nbsp;ZIP)">очередная версия
BestView&nbsp;&#151; 2.19</a>.</p>
```

Случаю, когда я добавляю новую ссылку на zip-архив, размер которого требуется указывать, соответствует фрагмент, аналогичный первому или второму из рассмотренных выше, за исключением того, что размер архива в нем не указан, то есть вместо, например, строки "(75 КБ ZIP)" в документе содержится просто "(КБ ZIP)".

В моих HTML-документах во всех трех перечисленных выше случаях (и только в них) после адреса ссылки на zip-архив, размер которого требуется указывать, но до адреса следующей ссылки (если она есть), обязательно содержится строка вида "(* КБ ZIP)", где "*" — ни одной, одна или несколько десятичных цифр. Таким образом, по этому признаку при обработке HTML-документа можно отличать ссылки, для которых надо проставлять размер (назовем такие ссылки "нужными"), от других ссылок.

От программы требуется, чтобы она просматривала указанный HTML-документ и для каждой "нужной" ссылки определяла размер соответствующего архива в килобайтах и подставляла этот размер вместо "***" в находящуюся после этой ссылки строку вида "(* КБ ZIP)".

Описание алгоритма

Исходя из сказанного выше, можно предложить такой алгоритм: просматриваем обрабатываемый HTML-файл от начала к концу; если встретили строку 'href="' (или 'HREF="' — регистр букв не имеет значения) — значит, далее (до следующего символа "\"") следует адрес ссылки (пока неизвестно, "нужная" это ссылка или нет), и этот адрес надо запомнить. Если встретили строку вида "(* КБ ZIP)", где "*" — ни одной, одна или несколько десятичных цифр, значит, последняя встретившаяся ссылка (ее адрес запомнен) — "нужная". Преобразуем адрес этой ссылки в полное имя соответствующего файла (zip-архива) на жестком диске. Определяем размер этого файла в байтах, затем вычисляем размер в килобайтах, округляя до ближайшего целого числа (но если длина файла — от 1 до 511 байтов включительно, то устанавливаем длину в килобайтах, равной 1, а не 0, так как надпись "0 КБ" будет выглядеть несколько странно, все-таки длина файла не равна нулю). После этого подставляем размер в килобайтах вместо "***".

Замечу, что в программе используется несколько модифицированный, более легкий для реализации вариант вышеописанного алгоритма — вместо непосредственного изменения исходного HTML-файла программа формирует временный файл, а потом уже копирует его содержимое в исходный файл.

Еще одна особенность реализации: при описании алгоритма упоминается о поиске в файле строки вида "(* КБ ZIP)", но так как эта строка содержит русские буквы, возникает сложность — в различных кодировках коды русских букв различны, а кодировка обрабатываемого HTML-файла заранее не известна. Можно

было, конечно, ограничиться какой-то одной кодировкой (у меня, например, в локальной копии web-страницы все HTML-файлы хранятся в кодировке Windows-1251), но я предпочел другой способ — при поиске в файле вышеуказанной строки русским буквам могут соответствовать любые символы. Таким образом, можно обрабатывать HTML-файлы в любой кодировке вида "один символ — один байт" (например, DOS, Windows-1251, KOI8-R, ISO 8859-5). Более того, даже если вы решите сделать версию web-страницы на английском языке, и вместо русского сокращения "КБ" в HTML-документах будет английское "KB", программа все равно будет работать правильно! Хотя при разработке программы такая задача и не ставилась :-).

Текст программы

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

#include <io.h>
#include <stdio.h>
#include <dir.h>
#include <ctype.h>
#include <string.h>

/* ===== Описание функций ===== */

int cmp_str (FILE* f, char* s);
long def_len (char* href, char* cur_path);
void chdir_1 (char* path);
FILE* fopen_1 (char* name, char* mode);
int fgetc_1 (FILE* f);
void ungetc_1 (int c, FILE* f);
void fputc_1 (int c, FILE* f);
void fputs_1 (char* s, FILE* f);
void fclose_1 (FILE* f);

/* ===== Пошла сама программа ===== */

#define MAX_HREF_LEN 255 /* Максимальная длина адреса ссылки.*/
#define MAX_PATH_LEN 128 /* Максимальная длина пути к файлу. */

int code_k, code_b; /* Коды русских букв "К" и "Б" в
кодировке обрабатываемого HTML-файла
(определяются автоматически). */

int main (int argc, char *argv[])
{
FILE *f_src; /* Исходный файл */
FILE *f_dst; /* Формируемый файл */
int href_found=0; /* Признак "в исходном файле уже
обнаружена хотя бы одна ссылка" */
char href[MAX_HREF_LEN+1]; /* Адрес (т.е. значение атрибута
"href") последней встреченной в
исходном файле ссылки */
int href_overflow; /* Признак "длина адреса последней
встреченной в исходном файле
ссылки оказалась больше максимальной"
(т.е. больше MAX_HREF_LEN) */
long l,l_kb; /* Длина zip-архива, на который указывает
ссылка (l — длина в байтах,
l_kb — в килобайтах) */
char cur_path[MAX_PATH_LEN+1]; /* Путь к обрабатываемому
HTML-файлу */
char *temp_name="$tmp"; /* Имя формируемого временного файла */
int i,c;

/* ===== Подготовительные действия ===== */

printf ("\n===== \n");
printf ("!Add_KB — автоматическое проставление размеров;\n");
```



```

printf ("          файлов в HTML-документах.      \n");
printf ("      (с) Иван Рошин, Москва, 22.04.2004    \n");
printf ("      E-mail: bestview@mtu-net.ru              \n");
printf ("      WWW: http://www.ivr.da.ru                 \n");
printf ("L=====n\n");

if (argc!=2) {printf ("Syntax: add_kb.exe
filename.htm\n\n");
    exit(1);}

if (getcwd(cur_path,MAX_PATH_LEN)==NULL)
{printf ("Ошибка при определении пути к HTML-файлу!\n\n");
    exit(1);}

f_src=fopen_1(argv[1],"rb");
f_dst=fopen_1(temp_name,"wb");

/* ===== Главный цикл ===== */

main_loop:

/* Если в исходном файле с текущей позиции находится строка
'href=', то считаем, что обнаружена ссылка. Устанавливаем
признак href_found. Считываем идущий за 'href=' адрес
ссылки, пока не встретим символ '"', и помещаем этот адрес
в переменную href. При этом копируем в выходной (временный)
файл символы, считанные из исходного файла. Если окажется,
что адрес ссылки длиннее MAX_HREF_LEN, то устанавливаем
признак href_overflow (нельзя сразу же прерывать выполнение
программы с выдачей сообщения о слишком длинном адресе
ссылки, так как эта ссылка вполне может и не относиться к
файлу, размер которого требуется указывать) */

if (cmp_str(f_src,"href=\"")=1)
{
    href_found=1;
    i=0;                /* Текущая длина адреса ссылки */
    href_overflow=0;
    fputs_1 ("href=\"",f_dst);
    while ((c=fgetc_1(f_src))!='"')
    {
        if (c==EOF) goto end_loop; /* Если конец файла - выходим
                                    из главного цикла. */
        if (i<MAX_HREF_LEN) {href[i]=c; href[i+1]=0; i++;}
        else href_overflow=1;

        fputc_1(c,f_dst);
    }
    fputc_1('"',f_dst);
    goto main_loop; /* Возвращаемся к началу главного цикла */
}

/* Если в исходном файле с текущей позиции находится строка
вида "(*&nbsp;?&nbsp;ZIP)" (о том, что в этой строке
обозначают символы "*" и "?", можно узнать из описания
функции cmp_str), значит, надо определить длину файла
(zip-архива), соответствующего последней встреченной ссылке,
и подставить ее в формируемый файл вместо "*" */

if (cmp_str(f_src,"(*&nbsp;?&nbsp;ZIP)")=1)
{
    if (href_found==0)
    {
        printf ("Ошибка: в исходном файле встречена строка \n");
        printf ("вида \"(*&nbsp;KB&nbsp;ZIP)\", но нет      \n");
        printf ("ссылки, предшествующей этой строке!    \n\n");
        exit(1);
    }

    printf ("%s\n",href); /* Печатаем на экране адрес ссылки */

    if (href_overflow==1)
    {
        printf ("Ошибка: в исходном файле встречена строка \n");
        printf ("вида \"(*&nbsp;KB&nbsp;ZIP)\", но адрес      \n");
        printf ("ссылки, предшествующей этой строке,      \n");
        printf ("оказался слишком длинным!                \n\n");
        exit(1);
    }
}

```

```

l=def_len(href,cur_path); /* Узнали длину файла,
                           соответствующего ссылке */

/* Теперь преобразуем полученную длину файла в байтах в длину
в килобайтах, округляя до ближайшего целого числа. Но если
длина файла - от 1 до 511 байтов включительно, то
устанавливаем длину в килобайтах равной 1, а не 0,
так как "0 KB" будет выглядеть несколько странно: все-таки
длина файла не равна 0 */

if ((l>=1)&&(l<=511)) l_kb=1;
else l_kb=(l+512)/1024;

fputc_1 (' ',f_dst);
if (fprintf(f_dst,"%ld",l_kb)==EOF)
{printf ("\nОшибка записи!\n\n"); exit(1);}
fputs_1 ("&nbsp;",f_dst);
fputc_1 (code_k,f_dst);
fputc_1 (code_b,f_dst);
fputs_1 ("&nbsp;ZIP",f_dst);
goto main_loop; /* Возвращаемся к началу главного цикла */
}

/* Если в исходном файле с текущей позиции нет ни строки
'href=', ни строки вида "(*&nbsp;?&nbsp;ZIP)", то, если
еще не достигнут конец исходного файла, просто копируем
символ из исходного файла в формируемый временный файл
и переходим к началу главного цикла. */

c=fgetc_1(f_src);
if (c!=EOF) {fputc_1(c,f_dst); goto main_loop;}

/* ===== Выполнение главного цикла закончено ===== */

end_loop:
fclose_1 (f_src);
fclose_1 (f_dst);

/* Исходный файл успешно обработан. Теперь открываем
сформированный временный файл для чтения, а исходный
файл - для записи, и копируем содержимое временного файла
в исходный. Затем удаляем временный файл. */

chdir_1 (cur_path);
f_src=fopen_1(temp_name,"rb");
f_dst=fopen_1(argv[1],"wb");
while((c=fgetc(f_src))!=EOF) fputc (c,f_dst);
fclose_1 (f_src);
fclose_1 (f_dst);
unlink (temp_name);

printf ("\nAll OK!\n\n");
exit (0);
}

/* =====
Функция def_len - определение длины файла по адресу
ссылки на этот файл.
Вход: href - адрес ссылки на файл,
      cur_path - путь к обрабатываемому HTML-файлу.
Выход: длина (в байтах) файла, на который указывает ссылка.
===== */

long def_len (char* href, char* cur_path)
{
    char temp_path[MAX_PATH_LEN+1]; /* Здесь будем формировать
                                     путь к файлу, на который
                                     указывает ссылка */

    int i;
    long l;
    FILE *f_tmp;

    strcpy (temp_path,cur_path); /* Первоначальное значение
                                     пути - это путь к обрабатывае-
                                     мому HTML-файлу */

```



```

/* Для более удобного выполнения последующих действий приводим
путь к виду без '\' в конце. Если путь указывает не на
корневой каталог (тогда его длина больше трех символов), то
это условие уже выполняется, иначе (путь вида "C:\",
длина — три символа) отбрасываем последний символ пути */

if (strlen(temp_path)==3) temp_path[2]=0;

while (strchr(href,'/')!=NULL) /* Пока в адресе ссылки
еще есть "/" */
{
/* Если первые три символа href — «../», то переходим на
уровень вверх: отрезаем эти три символа от href и отрезаем
от temp_path все символы, начиная с последнего "\" и до конца */
if (memcmp(href,"../",3)==0)
{
href=href+3;
if (strlen(temp_path)==2) /* Если уже в корневом каталоге */
{
printf ("Ошибка: в обрабатываемой ссылке слишком много \n");
printf ("переходов на уровень вверх! \n\n");
exit (1);
}
*strchr(temp_path,'\\')=0;
continue;
}

/* Иначе начало href до первого "/" — это имя каталога, в
который надо перейти. Отрезаем это имя от href и добавляем
к temp_path */

i=strchr(href,'/')->href; /* Длина имени каталога */
if ((strlen(temp_path)+1+i)>MAX_PATH_LEN)
{printf ("Ошибка: слишком длинный путь к файлу!\n\n");
exit (1);}
strncat(temp_path,"\\",1);
strncat(temp_path,href,i);
href=href+1+i;
}

/* Теперь в href осталось только имя файла, а путь к файлу
сформирован в temp_path. Если этот путь указывает на
корневой каталог (тогда его длина — два символа),
добавляем "\" в конце */

if (strlen(temp_path)==2) {temp_path[2]='\\'; temp_path[3]=0;}

/* Осталось определить длину этого файла */

chdir_1 (temp_path);
f_tmp=fopen_1(href,"rb");
l=filelength(fileno(f_tmp));
fclose_1(f_tmp);
return (l);
}

/* =====
Функция cmp_str — проверка совпадения (без учета регистра
букв) строки s и того, что находится в файле f с текущей
позиции. Символу "*" в строке соответствуют ни одной,
одна или несколько десятичных цифр в файле. Символу "?" в
строке соответствуют любые два символа в файле.
При совпадении функция возвращает 1, а позиция в файле
указывает на первый символ после строки; если в строке был
символ "?", то коды соответствующих ему двух символов в
файле сохраняются в глобальных переменных code_k и code_b.
При несовпадении функция возвращает 0, позиция в файле
не меняется.
===== */

int cmp_str (FILE* f, char* s)
{
fpos_t pos;

```

```

int c,i;
int k,b;
int result=1;

fgetpos (f,&pos); /* Запомнили текущую позицию в файле */
for (i=0;i<strlen(s);i++)
{
if (s[i]!='*')
{
while (isdigit(c=fgetc_1(f))); /* Пропускаем все цифры */
if (c!=EOF) ungetc_1(c,f);
continue;
}
if (s[i]=='?')
{
k=fgetc_1(f); if (k==EOF) (result=0; break);
b=fgetc_1(f); if (b==EOF) (result=0; break);
continue;
}
if (tolower(fgetc_1(f))!=tolower(s[i])) (result=0; break);
}
if (result==0) fsetpos (f,&pos); /* Восстановили позицию в
файле */

else {code_k=k; code_b=b;}
return (result);
}

/* =====
Аналоги некоторых функций для работы с файлами (chdir, fgetc,
ungetc, fputc, fputs, fopen, fclose). При ошибке прерывают
выполнение программы, печатая соответствующее сообщение;
таким образом, после вызова этих функций не надо проверять,
была ли ошибка.
===== */

void chdir_1 (char* path)
{
if (chdir(path)!=0)
{printf ("Ошибка при переходе в каталог \"%s\"\n\n",path);
exit (1);}
}

int fgetc_1 (FILE* f)
{
int c;
c=fgetc(f); if (ferror(f)) {printf ("Ошибка чтения!\n\n");
exit (1);}

return (c);
}

void ungetc_1 (int c, FILE* f)
{
if (ungetc(c,f)==EOF)
{printf ("Ошибка при вызове функции \"ungetc\"\n\n");
exit (1);}
}

void fputc_1 (int c, FILE* f)
{
if (fputc(c,f)==EOF) {printf ("Ошибка записи!\n\n");
exit (1);}
}

void fputs_1 (char* s, FILE* f)
{
if (fputs(s,f)==EOF) {printf ("Ошибка записи!\n\n");
exit (1);}
}

FILE* fopen_1 (char* name, char* mode)
{
FILE* f;
char* s;
char* s1="чтение";
char* s2="запись";

```

```

f=fopen(name,mode);
if (f==NULL)
{
    if (mode[0]!='r') s=s1; else s=s2;
    printf ("nОшибка открытия файла \"%s\" на %s!n",name,s);
    exit (1);
}
return (f);
}

void fclose_1 (FILE* f)
{
    if (fclose(f)==EOF) {printf ("nОшибка закрытия файла!n");
        exit (1);}
}

```

Исполняемый файл программы (для DOS) вы можете скачать с сайта журнала или с моей web-страницы.

Руководство пользователя

Использовать программу, конечно, имеет смысл только в случае, если у вас такая же ситуация, как описано в разделе "Постановка задачи". Также обратите внимание на следующие ограничения программы:

- адреса файлов в "нужных" ссылках (т.е. значения атрибута "href") обязательно должны быть в двойных кавычках (""). Да, в [1] указано, что значения атрибутов могут быть и в одиночных кавычках ('), а в некоторых случаях кавычки не обязательны, но мне было проще реализовать поддержку только двойных кавычек :-);

- программа поддерживает только относительные адреса файлов в "нужных" ссылках, причем только такие адреса, у которых в начале ни одной, одна или несколько последовательностей "../" и/или "имя каталога", а затем — имя файла;

- так как это DOS-программа, она не поддерживает длинные имена файлов и каталогов в адресах файлов в "нужных" ссылках;

- программа не поддерживает &-последовательности (entities) в адресах файлов в "нужных" ссылках. Впрочем, из символов, которые допустимы в имени файла или каталога в DOS (как указано в [3], это латинские буквы, цифры и символы -, _, \$, #, &, @, !, %, (,), {, }, ", ~, ^), обязательной замене на &-последовательность подлежит лишь символ "&" (он должен быть заменен на "&"). Из-за одного символа я не стал усложнять программу, тем более, что этот символ не так часто встречается в именах файлов и каталогов (на моей web-странице — вообще ни разу).

- программа не поддерживает символы, заданные в виде %xx (где xx — шестнадцатеричный код символа) в адресах файлов в "нужных" ссылках.

При запуске программы необходимо указать в командной строке имя обрабатываемого HTML-файла (он должен находиться в текущем каталоге):

```
add_kb.exe filename.htm
```

В процессе работы программа выводит адреса встретившихся в этом файле "нужных" ссылок. После успешной обработки файла программа печатает сообщение "All OK!" и заканчивает работу.

Как уже упоминалось в разделе "Описание алгорит-

ма", в процессе обработки HTML-файла программа не изменяет исходный файл сразу же, а записывает результат во временный файл. Таким образом, если на этапе создания временного файла выполнение программы будет прервано из-за какой-либо ошибки (скажем, если в одной из "нужных" ссылок был указан неправильный адрес, и соответствующий файл, длину которого надо было определить, не был найден), то исходный HTML-файл останется неизменным. После того как исходный HTML-файл полностью обработан и временный файл успешно сформирован, программа копирует содержимое временного файла в исходный файл и после успешного завершения копирования удаляет временный файл. Таким образом, если при копировании выполнение программы будет прервано из-за какой-либо ошибки, то результат работы программы останется на диске во временном файле (его имя — "\$.tmp").

Программа возвращает операционной системе код завершения: 0 — в случае удачного завершения и 1 — при какой-либо ошибке. Сообщение об ошибке также выводится на экран. Если программа была вызвана из bat-файла или из другой программы, код завершения может быть в них проанализирован, и, в зависимости от его значения, могут быть предприняты какие-либо действия.

Кстати, помимо выполнения основной задачи, еще одно полезное свойство программы заключается в том, что она обнаруживает ситуацию, когда в "нужной" ссылке адрес zip-архива указан неправильно, так что ему не соответствует никакой файл. В этом случае выполнение программы будет прервано с выдачей сообщения о невозможности открытия соответствующего файла (при этом обрабатываемый HTML-документ останется неизменным). У меня самого так обнаружилась неправильность пары ссылок :-).

Источники

1. HTML 4.01 Specification. — <http://www.w3.org/TR/1999/REC-html401-19991224/>
2. T.Berners-Lee, R.Fielding, L.Masinter. Uniform Resource Identifiers (URI): Generic Syntax (RFC 2396). — <http://www.ietf.org/rfc/rfc2396.txt>
3. В.Э.Фигурнов. IBM PC для пользователя. Изд. 5-е, испр. и доп. — М.: Финансы и статистика, 1994.

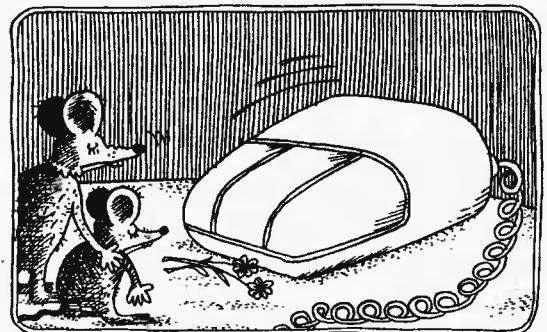


Рисунок О.Попова



“PlayStation 2”:

система внутреннего питания

С.ПЮМИК,
г.Чернигов.

Сложные системы имеют тенденцию противопоставлять себя своим же функциям. Принцип Шателлера

Говорят, что по сложности системы внутреннего питания можно судить о функциональной сложности всего устройства. Если это так, то игровая приставка PS2 относится к одной из высших категорий. Судите сами, система питания ее процессорной платы формирует 15 каналов напряжений и содержит четверть общего числа микросхем.

В PS2 имеются разнородные структурные блоки, выполняющие логически законченные задачи. Каждый из них требует оптимального питания. Например, для джойстиков необходимы напряжения 3,5 и 8 В, для Rambus ОЗУ — 1,8 и 2,5 В, для центрального процессора EE и графического синтезатора GS — 1,6 В. Более того, для улучшения линейности выходных каскадов видеокодера и аудио-ЦАП применяется не обычное напряжение 3,5 В, а повышенное до 5 В.

По величине напряжения питания можно оценить уровень технологии изготовления микросхем. Чем меньше питающее напряжение, тем сложнее технология и выше плотность упаковки транзисторов. Самые малые размеры имеют транзисторы в СБИС EE и GS. При технологических нормах проектирования 0,18 мкм барьерный слой их затворов состоит всего лишь из нескольких сотен атомов!

На рис.1 приведена функциональная схема системы внутреннего питания процессорной платы PS2 модели SCPH-30004R (версия 6). Без понимания взаимодействия ее составных частей нельзя приступать к ремонту. Условные названия микросхем, разъемов, цепей питания соответствуют [1].

Основу системы питания составляют стабилизаторы постоянного напряжения понижающего типа. Два из них, G1 и G4, являются ключевыми с широтно-импульсным (ШИ) регулированием, остальные — аналоговые, компенсационного типа.

Ключи K1...K4 выполнены на мощных МДП-транзисторах (MOSFET). Управле-

ние ими, а также разрешение работы стабилизаторов G1, G4, G8, G9 осуществляется системным контроллером IC001.

SMD-предохранители PS1...PS13 — плавкие, TH1...TH3 — самовосстанавливающиеся (“полисвич”).

После включения PS2 в сеть 220 В, от платы внешнего питания поступает напряжение 12 В (12V1). Первым запускается стабилизатор G3, который подает питание на микросхему IC001. Пока ни одна из кнопок на лицевой панели PS2 не нажата, микросхема IC001 запрещает работу остальных стабилизаторов и переводит ключи K1...K4 в разомкнутое состояние. После нажатия любой кнопки, микросхема IC001 открывает ключи K1, K4 и разрешает работу стабилизаторов. Ключи K2, K3 замыкаются лишь при подключении к разъему

“PCMCIA” внешнего устройства, например, жесткого диска.

На рис.2 приведены схемы стабилизаторов G1, G4. Выполнены они идентично, немного отличаясь в номиналах элементов и точках подключения резисторов. Здесь и далее на схемах для упрощения не показаны керамические и танталовые конденсаторы разной емкости, включенные параллельно питанию. На процессорной плате их около 200 штук, например, одну только цепь 1V61 (питание EE) блокируют 68 конденсаторов!

Процессом стабилизации выходных напряжений управляют ШИ-контроллеры: IC601 MB3889 ф. Fujitsu и IC604 “4818” ф. Texas Instruments, работающие на частоте 300 кГц. Каждый из них содержит два независимых канала с выходными МДП-ключами Q604...Q609. Ключи работают синхронно во времени, в связи с чем такие преобразователи называют ключевыми синхронными.

Процесс преобразования иллюстрирует рис.3. В первом полупериоде под воздействием импульсов от ШИ-контроллера замыкается ключ Q604.1 и размыкается Q604.2, происходит накопление энергии в катушке индуктивности L603. Во втором полупериоде ключ Q604.1 размыкается, а Q604.2 замыкается, энергия из катушки индуктивности L603 передается в нагрузку 5V03.

Конденсатор C636 сглаживает пульсации выходного напряжения, а фильтр L602-C603 уменьшает коммутационные помехи во входной цепи. Делитель R601...R603 служит элементом обратной связи, по которому ШИ-контроллер регулирует длительность импульсов управления. Выходное напряжение $U_{\text{вых}}$ (5V03) определяется по формуле $U_{\text{вых}}[B] = V_{\text{ref}} * (1 + (R601 + R602) / R603)$, где $V_{\text{ref}} = 1,23 \text{ В}$ — опорное напряжение микросхемы IC601.

Особенность схемы, приведенной на рис.3 — ключ Q604.2. В обычных ключевых (не синхронных) стабилизаторах вместо него ставят диод, который во втором полупериоде открывается и бесполезно рассеивает довольно

Рис. 1

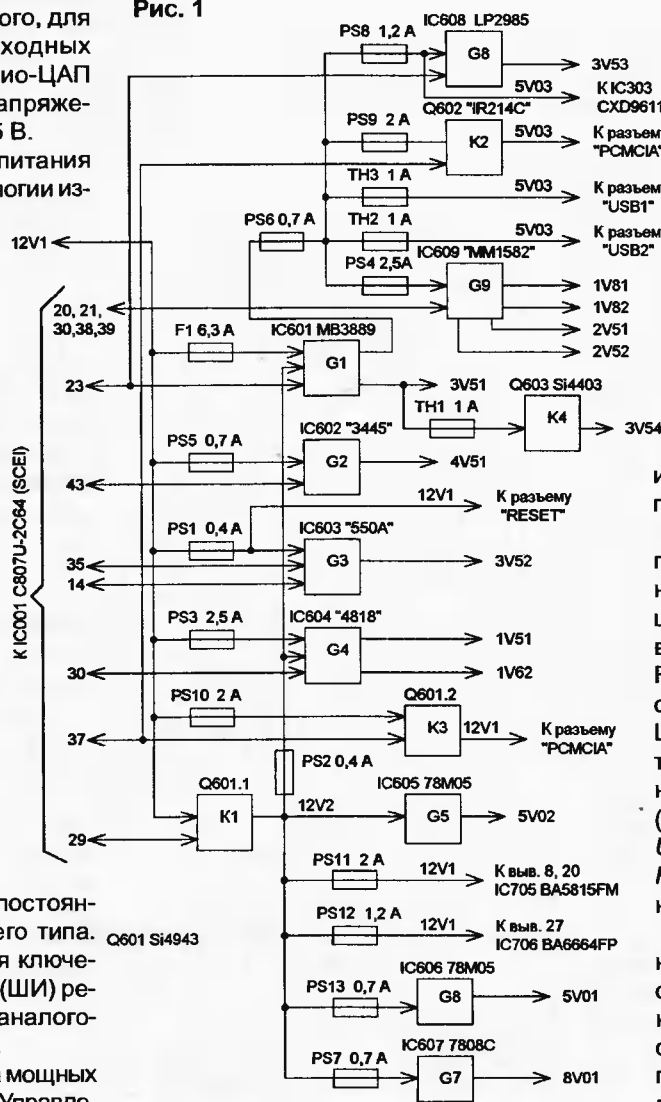
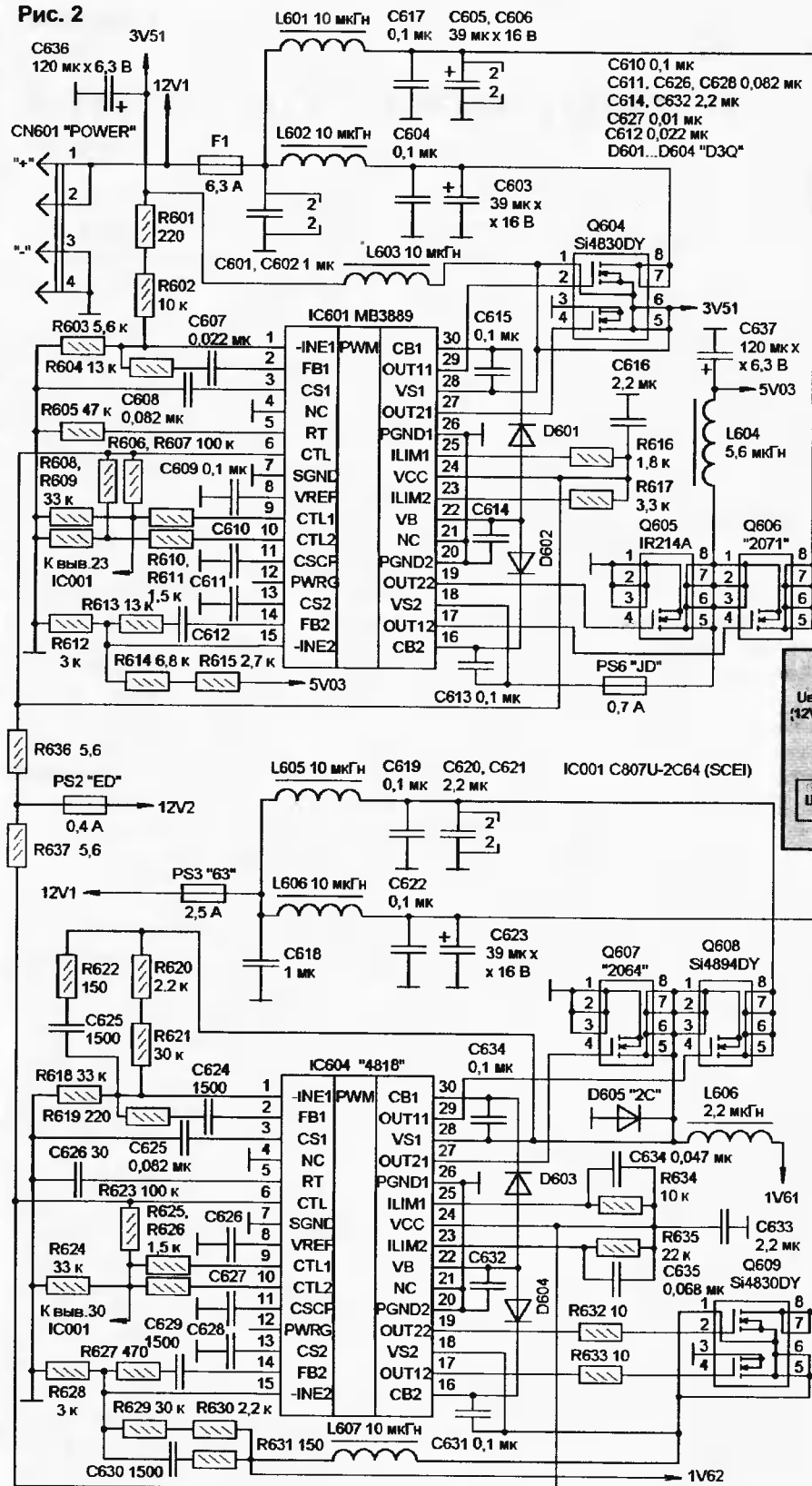




Рис. 2

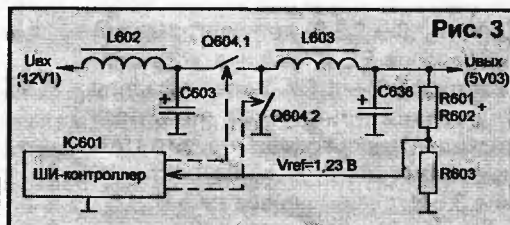


чтобы ключи никогда не были открыты одновременно.

Параметры применяемых в PS2 ключей ф. Vishay Siliconix приведены в табл. 1.

Параметры IC601 MB3889: рабочая частота — $F=100...500$ кГц — определяется формулой $F[kГц]=14100/R605[kОм]$; $U_{вх}=5,5...18$ В; $I_{вых}=120$ мА; $P=1,4$ Вт; имеется встроенная защита от перегрузки по току и напряжению; корпус — TSSOP-30. Благодаря высокой частоте преобразования, катушки индуктивности и конденсаторы фильтра имеют малые габариты, ключи не требуют радиаторов. Два канала преобразования выходной мощностью 15...20 Вт занимают на двусторонней плате место всего лишь 4 x 4 см!

Выходные напряжения можно подрегулировать резисторами R601, R615, R620, R630. Все они имеют небольшой номинал (по сравнению с последовательно включенными резисторами



R602, R614, R621, R629), что позволяет в сумме подобрать точное сопротивление.

Судя по паспортной мощности потребления EE и GS, рабочие токи нагрузок стабилизаторов достигают 6...7 А. Это немудрено, ведь EE содержит 10,5 млн., а GS — 43 млн. транзисторов. Для отвода тепла от 540-выводного корпуса EE и от 387-выводного корпуса GS применяют массивные алюминиевые радиаторы, прикрученные винтами к шасси.

Кроме того, предусмотрено общее воздушное охлаждение радиаторов при помощи вентилятора M1, который подключается к плате через разъем CN602 (рис. 4). Вентилятор управляется элементами Q610, IC602. Транзистор Q610 вместе с усилителем разбаланса IC602 образуют регулируемый стабилизатор напряжения. Резистивный делитель R641, R642 устанавливает его порог. Если напряжение, поступающее от системного контроллера IC001 на вывод 1 IC602, превысит порог, то транзистор Q610 открывается, запуская вентилятор. И наоборот, если напряжение ниже порога, то транзистор будет закрыт, и вентилятор прекратит работу.

большую мощность. Ключ Q604.2 обладает гораздо меньшим сопротивлением в открытом состоянии (доли ома), в связи с чем потери мощности минимальны. Как следствие, стабилизатор имеет высокий КПД (90...94%), но необходимо аппаратно следить за тем,

Табл. 1

Название элемента	Тип канала	Число ключей	$V_{см}, В$	$R_{см}, Ом$	$I_{с}, А$	Фирменная технология
Si4830DY	N	2	30	0,03	7,5	LITTLE FOOT
Si4894DY	N	1	30	0,018	12,5	TrenchFET
Si4943DY	P	2	20	0,03	8,4	TrenchFET
Si4403DY	P	1	20	0,032	9	TrenchFET

Рис. 4

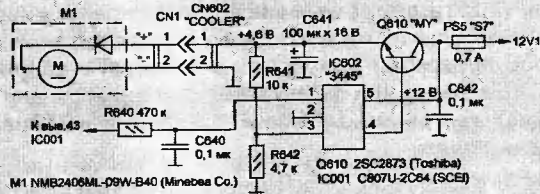
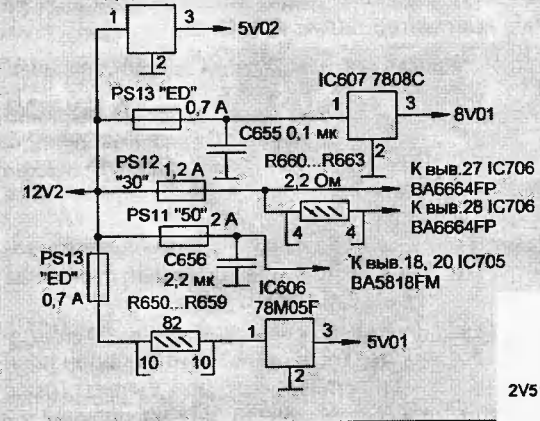


Рис. 5



Вентилятор M1 китайской фирмы Minebea рассчитан на постоянное напряжение 7 В и ток 0,29 А. В целях снижения шума он эксплуатируется при пониженном до 4,6 В напряжении.

На рис.5 приведены схемы стабилизаторов G5, G6, G7. Это стандартные трехвыводные микросхемы серии 78xx с фиксированными выходными напряжениями 5 и 8 В. Ток нагрузки IC605, IC606 — 0,5 А, IC607 — 1 А, имеется защита от КЗ по выходу, корпус — КТР-3.

Напряжение 8V01 поступает на разъем для джойстиков. Поскольку джойстики для PS2 и "PlayStation" (PSX) совместимы между собой, то дискуссия о том, можно ли подавать на джойстики PSX повышенное напряжение, окончена. Судя по напряжению стабилизации IC603 (8 В) и величине питания PSone (7,5 В), джойстики PSX и PS2 рассчитаны не на точное, а на "плавающее" напряжение в диапазоне 7,5...8 В.

На рис.6 приведены схемы трех ключевых каскадов на транзисторах Q601...Q603, а также стабилизаторов напряжения на микросхемах IC603, IC608, IC609.

Микросхема IC608 LP2985IM5-3,5 (ф. National Semiconductor) — это малошумящий, отключаемый внешним сигналом стабилизатор, имеющий очень низкую разность напряжений между входом и выходом (Low-Noise Ultra Low-Dropout). Типичное значение падения напряжения на нем составляет 3 мВ на холостом ходу и 350 мВ при токе нагрузки 150 мА. Напряжение стабилизации $U_{ВЫХ} = 3,5 \text{ В} \pm 1\%$, $U_{ВХ} = 3,8...16 \text{ В}$, имеется защита от коротких замыканий в нагрузке.

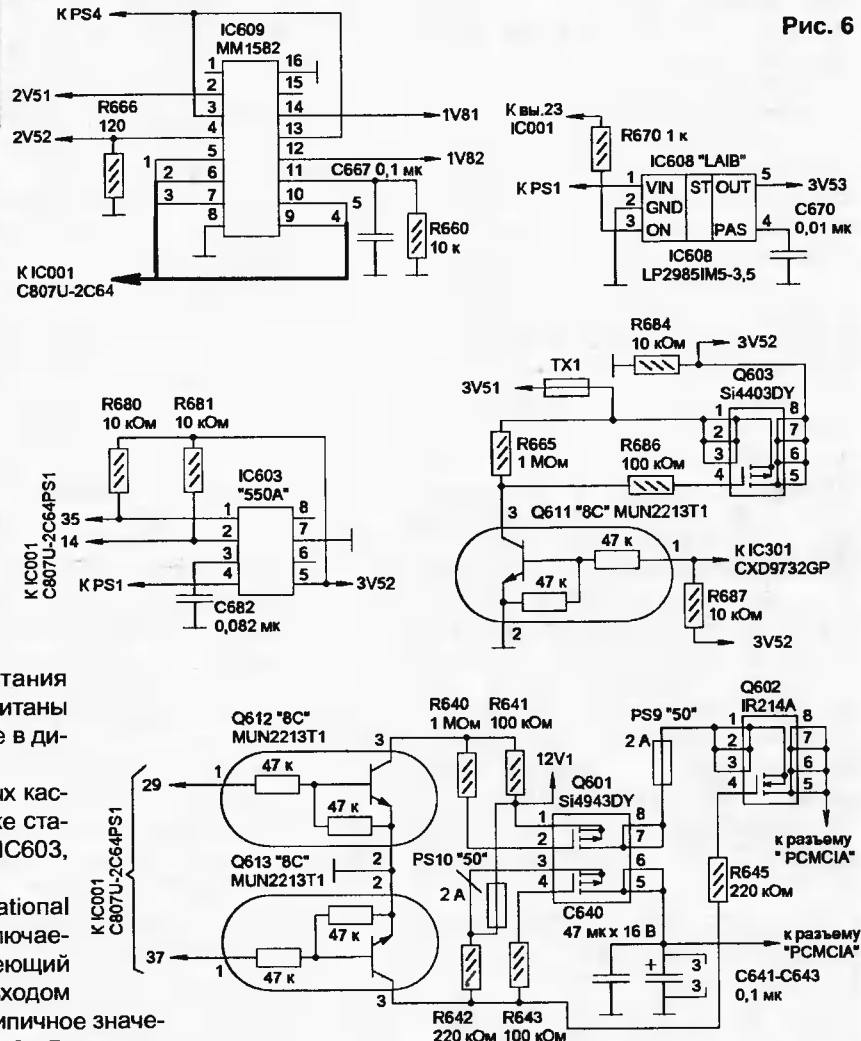
Неисправности канала внешнего питания

Опытные ремонтники знают, что больше половины всех неисправностей современных радиоэлектронных уст-

Табл. 2

Обозначение	Маркировка	Тип	Фирма	Напряжение, В	Защищаемые узлы
Плавкие предохранители					
PS1	ED	FCC16	Kamaya	12	Система управления
PS2	ED	FCC16	Kamaya	12	ЕЕ, GS, СБИС
PS3	63	CCP2E63	KOA	12	ЕЕ, GS (1,6 В)
PS4	63	CCP2E63	KOA	5	RDRAM, GS, IC301
PS5	S7	ICP-S0.7	Rohm	12	Вентилятор
PS6	JD	FCC16	Kamaya	5	RDRAM, GS, DVD/CD
PS7	S7	ICP-S0.7	Rohm	12	Двигатели джойстиков
PS8	30	CCP2E30	KOA	5	Канал чтения DVD/CD
PS9	50	CCP2E50	KOA	5	Разъем PCMCIA (5 В)
PS10	50	CCP2E50	KOA	12	Разъем PCMCIA (12 В)
PS11	50	CCP2E50	KOA	12	Каретка DVD/CD
PS12	30	CCP2E30	KOA	12	Канал вращения диска
PS13	S7	ICP-S0.7	Rohm	12	Лазерная головка
Самовосстанавливающиеся предохранители					
TH1	1X1	Mini SMD100	Raychem	3,5	Джойстики, карты памяти
TH2, TH3	1X1	Mini SMD100	Raychem	5	Разъемы USB1, USB2
F1	6.3A	CCF1NTE6.3	-	12	RDRAM, GS, DVD/CD

Рис. 6



ройств приходится на канал питания. Система внутреннего питания процессорной платы PS2 является "подозреваемым номер один", независимо от характера неисправности. При ремонте всегда следует иметь перед глазами функциональную схему системы питания (рис.1) и структурную схему процессорной платы [1].

В табл.2 приведен список предохранителей, находящихся



на процессорной плате, с указанием узлов, которые они защищают. При ремонте предохранители обычно заменяют отрезком тонкой проволоки диаметром 0,1...0,2 мм, поскольку фирменные предохранители для поверхностного монтажа дефицитны и дорогостоящи.

Микросхемы стабилизаторов серии 78M05 и 7808C выпускают многие изготовители, например, ON Semiconductor, Rohm, STMicroelectronics. В крайнем случае, для замены или перепроверки подойдут КР142ЕН5А, КР142ЕН8А, но с формовкой выводов по месту и максимальным укорочением их по длине.

Поиск неисправностей начинают с прозвонки всех пре-

дохранителей. Типичные значения сопротивления — не более 1 Ом (для самовосстанавливающихся предохранителей допускается до 3...7 Ом). Далее подают питание и вольтметром проверяют выходные напряжения стабилизаторов. Заниженные или завышенные значения являются информацией к размышлению и соответствующим ремонтным действиям.

Литература

1. Рюмик С. "PlayStation 2": процессорная плата. — Радиодир. Ваш компьютер, 2004, №9, С.38.

Доработка блока питания компьютера

А.БУТОВ,

с.Курба, Ярославской области,

E-mail: butoff@yandex.ru

При эксплуатации персонального настольного компьютера в процессе его работы без резервного источника питания был обнаружен такой эффект — при кратковременном (не более 80...120 мс) пропадании электроэнергии в сети, например, во время сильного ветра, в работе компьютера происходил сбой. Компьютер не успевал выключиться, но даже такого небольшого перебоя в электроснабжении оказывалось достаточно, для того чтобы система производила перезагрузку. Следует отметить, что другие электроприборы с импульсными источниками питания, например, телевизор и видеоманитофон, продолжали работать без сбоев. Предположение о том, что емкость оксидных конденсаторов в цепи сетевого выпрямителя блока питания, которые фильтруют выпрямленное напряжение, имеет малую величину, получило подтверждение после снятия защитного кожуха блока питания.

Системный блок компьютера был собран в корпусе "Meditower CX-2858 ATX 2.03", блок питания которого маркировался как трехсотваттный. Измеренная потребляемая от сети переменного тока 220 В мощность в режиме максимальной загрузки процессора и остальных узлов компьютера достигала 180...210 Вт. При этом на осциллограмме выходного напряжения

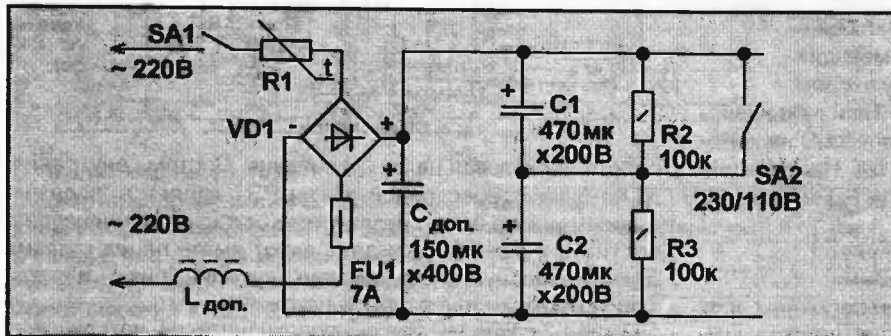
+3,3 В на фоне шумов была заметна небольшая огибающая частотой 100 Гц (удвоенной частоты сети). Было принято решение увеличить емкость фильтрующих конденсаторов, что, по замыслу, должно было повысить устойчивость системы к кратковременным сбоям в сети электроснабжения. На рисунке приводится упрощенный фрагмент узла выпрямителя импульсного блока питания, установленного в упомянутый системный блок. В незначительных вариациях этот узел будет примерно одинаковым во всех блоках питания стационарных компьютеров. Как видно, для напряжения питания 220 В суммарная емкость последовательно включенных конденсаторов C1, C2 составляет около 235 мкФ, что очень мало как для эффективной фильтрации выпрямленного напряжения, так и для долговременного надежного функционирования самих фильтрующих конденсаторов. Для улучшения качества фильтрации выпрямленного напряжения был установлен дополнительный оксидный конденсатор 150 мкФ x 400 В фирмы "Samsung", обозначенный на рисунке как C_{доп.} Этот конденсатор был смонтирован внутри корпуса блока питания, к которому был прикреплен металлической скобой с применением изолирующих прокладок из мягкой стеклоткани. Кон-

денсатор следует устанавливать так, чтобы он не препятствовал прокачиваемому потоку воздуха.

Кроме описанной выше доработки блока питания, был установлен вспомогательный сетевой фильтр (дроссель), обозначенный на рисунке как L_{доп.} Этот дроссель наматывается на ферритовом кольце K38x24x7, и содержит 45 витков провода МГТФ сечением по меди 0,44 мм. Дроссель устанавливается рядом с сетевым выключателем, расположенным на задней стенке системного блока.

Также было принято решение о замене установленного в блоке питания вентилятора "MJ802012" на более мощный и дорогостоящий двухскоростной "Thermaltake A1214". Этот вентилятор оборудован встроенным термодатчиком, и при температуре окружающего воздуха менее 30...35°C имеет частоту вращения 2900 об/мин, при более теплом воздухе скорость вращения крыльчатки возрастает до 5000 об/мин. Черный провод — это "минус" напряжения питания кулера, красный — "плюс", желтый провод предназначен для контроля частоты вращения, и в данном случае не используется. После замены вентилятора блока питания, уменьшился не только нагрев этого блока, но и на 5°C снизилась температура процессора, что объясняется повышением эффективности системы охлаждения всего системного блока, содержащего в моем случае в общей сложности 7 вентиляторов.

Прежде чем браться за предложенные в этом материале доработки, нужно быть уверенным не только в своих знаниях и навыках, но и четко представлять, к каким последствиям могут привести неумелые или небрежные манипуляции. При выполнении всех схемотехнических работ дорабатываемое оборудование должно быть полностью обесточено.





А.ВЕНДИЛОВСКИЙ,
г.Минск.

Doom 3

Космический челнок вынырнул из черноты космического пространства и устремился вниз, навстречу огненно-рыжей планете. Очень скоро разряженная атмосфера Марса стала разогревать корпус корабля, который с каждой секундой терял скорость, неумолимо падая на поверхность планеты. В самом конце салона, не отрывая глаз от иллюминатора и не обращая никакого внимания на рекламную болтовню телевизоров, сидел сержант. Он был еще молод, но шрамы и боевые награды на груди говорили, что



ему многое удалось повидать в этой жизни. Рядом с ним никто не сидел — один взгляд холодных голубых глаз заставлял любого непрошеного собеседника теряться. Он просто хотел побыть один, и он вспоминал...

— Ты должен стрелять, когда я прикажу и куда я прикажу! — молодой лейтенант был просто вне себя от ярости. Или ты закончишь так же, как твой отец!

— Там просто толпа безоружных, испуганных гражданских! Там женщины и дети, и среди них нет солдат!

— Я сказал!... — лейтенант, с налитыми кровью от бешенства глазами, уже вытаскивал пистолет.

— А не поцеловал бы ты меня в — сержант, одним прыжком одолел пространство между ними и схватил руку, уже направляющую на него пистолет. Они оба покатались по апрельской грязи, но опыт боевых операций в такие моменты решает все. Он подмял лейтенанта под себя, четким и сильным ударом вышиб ему несколько передних зубов, а потом, не обращая внимания на крики, выламывая пальцы противника, вырвал у него пистолет. И со всей силы опустил его рукоятку на лицо лейтенанта, превращая его в кровавую кашу из плоти и хрящей, потом еще и еще раз, пока крик жертвы не превратился в жалобный скулеж...

Он тяжело вздохнул, упоминания об отце очень часто выводили его из себя. Знал ли он отца? Он помнил лишь его безумие, странные рассказы и ощущение ужаса, что всегда присутствовало рядом с ним. И особенно тайна... То, что произошло двадцать лет назад на Фобосе и Деймосе, так и осталось для него загадкой, но он знал только одно — там случилось что-то настолько страшное, что все верхушки власти боялись даже

вспоминать об этом. Рассказывали, что несколько тысяч человек погибли мучительной смертью, и многие определенно кивали на его отца как на человека, совершившего это. Другие намекали, что это был единственный выход, и не вмешайся тогда отец в происходящее, все могло зайти намного дальше... Думаю, они побаивались его, но и старались сохранить — всю оставшуюся жизнь отец прожил на маленьком острове в Тихом

Океане, который был его персональным сумасшедшим домом и тюрьмой.

«... За нападение на старшего по званию и причинение тяжелых физических повреждений, Трибунал присуждает три года штрафного батальона. Но

принимая во внимание послужной список обвиняемого, его профессиональную квалификацию, а также рекомендации Высшего Командования Земли, данное наказание заменится переводом в охранную службу марсианского отделения Корпорации, где обвиняемый будет находиться до получения последующих указаний!»

Он криво усмехнулся, можно было подумать, что его помиловали — на самом деле, это была каторга похуже штрафбата, от тоски там даже тараканы дохли, а рассказы о последствиях радиации и вдыхания марсианской пыли передавались из уст в уста. Ирония судьбы, а ведь отец точно так же загремел сюда, когда началась вся эта кавария, только, кажется, его командир погиб...

Корабль встряхнуло, и пассажиры, не торопясь, двинулись к выходу. Приемный зал был маленький, а сама база находилась глубоко под землей, в прорытых в скалах туннелях, словно здесь славно потрудились стая бешеных кротов. Корпорация была одной из самых могущественных финансовых структур, и очень многое, что поддерживало жизнь на Земле, имело клеймо этой компании. Но теперь ее взгляд обратился к звездам, и космические базы стали расти как грибы после дождя.

База гудела как растревоженный улей — всюду сновали техники, перетаскивая на пневмотележках какое-то загадочное оборудование, люди в белых халатах не отходили от экранов мониторов, был слышен сильный шум. Но первое, что бросилось в глаза — огромное число охранников, даже для большого марсианского го-

рода. Они были повсюду, они были насто-роже, и они не напоминали ожиревших тывков, которые попадались ему на Земле — это были профессиональные бойцы.

Особо осматриваться вновь прибывшему не дали, едва он доложил командиру о прибытии, как его тут же отправили искать какого-то безумного ученого, потерявшегося в подземельях старой энергостанции. Пока он добирался до шлюза, успел узнать, что именно сегодня проводят какой-то очередной, но очень важный эксперимент, поэтому нервы у всех были на пределе.

... Пока он спешно перебирался из главного корпуса к развалинам старой энергостанции, Марс не один раз плюнул в лицо горстью оранжево-ржавого песка. Пришлось выскочить на поверхность, но даже с кислородной маской он ощущал, как тяжело дышать. Двери шлюза открылись с трудом — время и эрозия сделали свое дело. Вокруг было темно, под ногами путались какие-то трубы и коробки, а вся конструкция, казалось, рухнет от малейшего чиха. Но впереди было немного света от работающих мониторов и была слышна непонятная возня. Не желая тянуть кота за хвост, он включил свет.

Когда свет высветил его лицо, у маленького человечка в белом халате вырвался крик ужаса, тот смотрел на сержанта, словно увидел приведение, и повторил имя. Имя отца?

— Вы знали моего отца? — Ученый смотрел на него еще более удивленно и... испуганно. Потом он вцепился ему в рукав, словно от этого зависела его жизнь, и судорожно, без остановок, проглатывая слова, стал говорить.



— Ты его сын! Невероятно! Ты должен их остановить! Сейчас же! Пока еще есть время! Твой отец остановил их тогда, но сейчас они возьмут реванш, и всем нам придет конец! Всадники Апокалипсиса уже стоят за стенами этой базы в ожидании приказа! Ты ведь должен знать, ведь он ДОЛЖЕН БЫЛ ТЕБЕ ЭТО РАССКАЗАТЬ! Иначе...

Сержант изо всех сил встряхнул говорившего, вкладывая в этот рывок все раздражение и злость. — Кого остановил мой отец? Откуда вы его знали, и что вообще тут происходит, черт возьми?

Ученый поник, устал от сев на перевер-



нутый ящик. На экране монитора мигало табло обратного отсчета.

— Значит, он ничего не рассказал... Он был охранником, как и ты, Корпорация исследовала возможности телепортации. Это был способ мгновенного переноса грузов и людей на гигантские расстояния. Казалось, что после этого звезды падают к нашим ногам. Одна база располагалась на Фобосе, другая на Деймосе. Их остатки ты мог видеть из челнока. Никто не задумывался, где находился предмет, пока перемещался из одной точки в другую, было придумано множество теорий, только ни одна из них не была верной. Пра-



вильный ответ был — Ад! Самый настоящий, который намного страшнее, чем тот, что описан в Библии, но это мы поняли лишь позднее. Множество людей прошло телепортер, а потом они стали пропадать... Когда при переходе пропало сразу сто двадцать человек, Корпорация забила тревогу и попыталась приостановить эксперименты с переходом людей, тогда телепортеры заработали вновь, но только на прием. Мы проложили для Ада дорогу, и он этим незамедлительно воспользовался! Демоны уничтожали все и вся, а те из сотрудников, кто проходили телепортер, тоже изменились. Это был чистый ужас, крики умирающих и звуки стрельбы. Я висел на орбите в космическом челноке, а твой отец стоял возле входа в лабораторию, когда все это началось. А через пятнадцать минут все было кончено, людей осталось только двое — он и я. Твой отец вошел на базу и в одиночку устроил им то, что не смогла сделать вся охрана. Но эти твари уже успели перебросить десант на Землю, и твой отец рискнул шагнуть в телепортатор, чтобы остановить их там. Путешествие в Преисподнюю и то, что он увидел во тьме, навсегда изменили его. Но эти придурки не поверили ни одному нашему слову! Они решили, что это было помешательство, а возможно, что на нас напала неизвестная раса, тоже имеющая телепортаторы. Они долго готовились, и вот вторая попытка, но она может оказаться еще хуже прежней. Ты должен остановить их! Ты...

В этот момент поверхность пола ощутимо вздрогнула, на экране застыли нули. Пару секунд ничего не происходило, а

потом отключился свет. Но за несколько секунд до этого сержант успел разглядеть, как голову охранника одним махом откусила тварь весьма поганого вида. Внутри его все сжалось, он почувствовал, что что-то, сдерживаемое им всю жизнь, вырывается на свободу, он словно услышал слова отца: "Когда ты почувствуешь холод, всегда падай на землю и стреляй во все, что движется, иначе эти головы дотянутся до тебя..." — голос отца доносился через пелену времен.

Бросившись на пол, он заметил, как что-то круглое и огненное пронеслось рядом с ним и ударило в ученого. Он готов был поклясться, что это напоминало череп. Несчастный, отброшенный сверхъестественной силой в дальний угол, завизжал и ненадолго затих, но буквально через пару секунд он вновь поднялся, и издавая непонятный воющий хрип, двинулся на сержанта. За это короткое время его облик невероятно изменился — глаза закатились, были видны только налитые кровью белки, кожа стала серой и покрылась страшными рубцами, из которых сочилась мерзкая на вид жидкость. С трудом шагая, скуля и роняя слюни, как бешенный зверь, монстр попытался ударить

сержанта. Преодолевая нахлынувшее чувство безразличия и отвращения, перемешанное со страхом, тот выстрелил. И в следующий миг тошнота нахлынула с новой силой — пуля попала в голову, снесла часть черепа, выплеснув серое вещество мозгов на стенку, челюсть зомби была полностью вырвана, а один глаз висел на стебельке нервов, как экзотический фрукт, покачивающийся из стороны в сторону. Но зомби упрямо продолжал наступать, словно и не заметил повреждений. Сержант выстрелил еще и еще раз, пули отрывали куски разлагающейся плоти, пока не оторвали голову, перебив позвончик. Тело мертвеца вздрогнуло и рассыпалось красным пылающим пеп-



лом. Сержант медленно поднялся, в его уши ворвалась какофония боя, ведущегося в главном здании, крики ужаса, перекрывающиеся непонятным рычанием и автоматной стрельбой... Но он в этот момент был очень далеко — все, что он знал и во что верил, только что перевернулось

с ног на голову, стабильный и прочный реальный мир уступил место ненавязчивому скольжению в бездну, а безумные рассказы отца стали истиной, более не нуждающейся в доказательствах. Он посмотрел на мониторы, переключаясь с одной работающей камеры на другую. По коридором ходили мертвецы, у некоторых отсутствовали части тела, кругом царил разгром, свет практически отсутствовал, а пол и стены были залиты кровью и ошметками разорванной плоти. На одном экране бежала парочка охранников, забрала их шлемов были разбиты, и из-под них текла слезь, а глаза горели ярко-красным огнем верности Преисподней. С другого экрана безучастно скалился в камеру мертвый солдат — что-то или кто-то вырвал ему лицо, оставив только окровавленный череп. В темноте некто на двух ногах раскидывал огненные шары. На сержанта накатило абсолютное спокойствие, не оставив и малейшей тени сомнения в душе. Спокойно перезарядив пистолет, он засунул его в кобуру, достал винчестер. Еще раз проверил амуницию и двинулся ко входу.

— Не скучайте, засранцы, со мной вам тоже не повезло! — и, раскурив сигару, шагнул вперед...

Когда эта статья попадет вам на глаза, в эту игру переиграют практически все, и все журналы с газетами пропоят оду великопеленной игре. Я тоже снимаю шляпу перед разработчиками и хочу сказать всем вам, что вы... не видели этой игры! Не считайте меня ненормальным, я говорю не о бета-версиях или о чем-то подобном. Я хочу сказать, что вы не видели игру такой, какой ее сделали и видели создатели, и она просто не могла произвести на вас необходимое впечатление. Наверное, сейчас у вас появилась мысль, что я тихо сбрендил, но давайте перейдем к примерам. Представим, что вы запустили игру на маленьком черно-белом мониторе, а звук перевели на спикер. "Да это же полная фигня, и никакого кайфа!" — воскликните вы, а я с вами полностью соглашусь. А потом замечу, что 89% игроков видели ее именно в таком варианте. А все потому, что системные требования, приведенные в конце этой статьи — это минимум, на котором игра запустится (т.е. запустится вообще, а не отправится в край Потерянных Кластеров). Мой 2500 Бартон с 512 Мб оперативки и Радеоном 9600ХТ с 128 Мб на борту оказались СЛАБЫМИ, по определению. Мне удалось найти некоторые старые статьи Дж.Кармака, где весьма откровенно говорилось, что самые сильные топовые конфигурации

РС на сегодняшний день будут по определению лишь ХОРОШЕЙ или более или менее оптимальной конфигурацией. И что атмосферой игры смогут проникнуться лишь те, кто будет использовать по крайней мере ЧЕТЫРЕ колонки (и хорошую звуковую карту). Я это проверил, подру-



бив новую звуковую систему и одолжив на сутки Radeon 9800 XT, и с сожалением хочу сказать, что Кармак прав. Повторилась история первого DOOMa, когда необходимые компьютеры у игроков появились лишь через несколько лет после выхода игры. Не ошибусь, предположив, что в ближайшие несколько лет, после проведения очередных апгрейдов, вы будете потрясены возможностями этой игры.

Движок игры потрясает своим уровнем детализации и своей реалистичностью. Он не оставляет сомнений, что эпоха фотореалистических игр уже не за горами. От одного взгляда на скрин с монстрятником, когда видишь столь мелкие детали, может бросить в дрожь. Игра света и тени... Советую полюбоваться, как свет разбивается вращающемся пропеллером, и как качественно прописаны динамические тени. А свет от фонарика заслуживает отдельных аплодисментов — полюбуйте, как он ТОЧНО отображается на предметах! Походки персонажей самые что ни на есть плавные, без буратиновских дерганий — четко продуманная скелетная анимация, ранее использовавшаяся лишь в голливудских блокбастерах. Впрочем, от голливудских фильмов игра ушла недалеко. Недаром ее сравнивают с интерактивным фильмом ужасов. Были игры, в которых изредка становилось страшно. Были игры, в которых становилось страшно время от времени. Но я не помню ни одной игры, где страшно ВСЕГДА. Она изводит своим постоянным напряжением, и в этом ее прелесть, каждая поставленная сцена, где нам навстречу выплывает очередное новое создание Ада, или кровавые видения персонажа... Недаром Кармак заяв-

лял, что постарается перенести в игру атмосферу фильмов "Чужие" и "Живые мертвецы". А звук! Он просто завораживает, каждому шороху ты будешь внимать, ожидая подвоха. Тут я должен заметить, что звуковой фон очень богат, зависит от местоположения игрока и очень точно позиционирован. Это очень важно, так как почти все время вам придется провести в темных и полутемных коридорах, старательно наводящих депрессию (слабонервным лучше сразу записаться к психиатру), и чтобы понять, есть ли еще кто-то в комнате, кроме вас, и откуда бежит противник, ориентироваться придется только на звук. Это смогут оценить лишь владельцы звуковых систем 5.1.

Теперь информация для ветеранов всех Думов. Теоретически оружие и монстрятник перекочевали без изменений, но на практике любой монстр из первой части на фоне последней версии выглядит не страшнее Микки Мауса. У многих прокатится слеза умиления от встречи с Импом или Бароном Ада (после того как эти твари получают свою порцию свинца!). Но новых чудовищ тоже более чем достаточно (некоторые из них выглядят так ужасно, что мои ночные кошмары просто отдыхают!). Как и прежде, винчестер — оружие пролетариата, на близком расстоянии он смертоносно убоен. Автомат быстро расстреливает обойму, пистолет не более чем добивалка, если патроны более серьезного оружия тратить не хочется, а ракетлаунчеры BFG будут попадаться нечасто. Единственно, мне еще не попадалась на глаза бензопила, и я все-таки залезу в хелпы, чтобы узнать, найду или нет я этот любимый предмет всех думеров? А! Движка не при-

нес революционных веяний, но скрипты использованы очень грамотно — десантники прячутся за ящиками, противник старается использовать все свои сильные стороны и лишний раз не лезть на рожон. Особенно это чувствуешь, если эта тварь засядет в конце длинного коридора и не спеша поливает тебя свинцом, а ты, прыгая из стороны в сторону, ощущаешь себя уткой в тире. Легкого прохождения не будет!

Есть хиты, есть Хиты, а есть игры, которые перешагивают этот рубеж и на долгие годы становятся примером, как должны делаться игры. И рад сообщить, что сейчас перед нами именно такая игра!

Minimum System Requirements
English version of Microsoft(r) Windows(r) 2000/XP

Pentium(r) 4 1.5 GHz or Athlon(tm) XP 1500+ processor

384 MB RAM

8x Speed CD-ROM drive (1200 KB/sec sustained transfer rate) and latest drivers

2.2GB of uncompressed free hard disk space (plus 400MB for Windows(r) swap file)

100% DirectX(r) 9.0b compatible 16-bit sound card and latest drivers

100% Windows(r) 2000/XP compatible mouse, keyboard and latest drivers

DirectX(r) 9.0b (included)

3D hardware Accelerator Card Required

— 100% DirectX(r) 9.0b compatible 64MB

Hardware Accelerated video card and the latest drivers or ATI(r) Radeon 8500 (9000, 9200, 9500, 9600, 9700, 9800), or Nvidia(r) GeForce 3/Ti (4MX, 4/Ti, FX, 6800) series

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

КОМПЬЮТЕРНЫЕ ХРОНИКИ

"Звездное наследие"

Давным-давно, в 1995 г., творческой группой STEP была издана игра для "ZX-Spectrum" — "Звездное наследие". Текстовый квест с отличной графикой и музыкой, с захватывающим фантастическим сюжетом — одним словом, игра была великолепной!

Вряд ли найдется спектрумист, не слышавший о ней. Немало часов провел за прохождением "Звездного наследия" и автор этих строк :-).

И вот, теперь стало известно, что готовится к выпуску (в первом квартале 2005 г.) PC-версия этой знаменитой игры. Авторы обещают, что игра будет интересной даже для тех, кто уже знаком со спектрумовской версией. Подробнее читайте на сайте разработчика — <http://www.stepgames.ru>.

P.S. Кстати, на demoparty Chaos Constructions 2004 прошла вызвавшая большой интерес зрителей презентация демо-версии "Звездного наследия", на которой было продемонстрировано прохождение первых локаций игры.

Antique Toy 2004

Подведены итоги Antique Toy 2004 — виртуального конкурса музыки и графики для "ZX-Spectrum". Регламент проведения этого конкурса: организаторы сначала объявляют правила и принимают работы от участников, а затем выпускают сборники с работами, распространяемые в сетях, и происходит голосование, принять участие в котором может каждый желающий.

Как общее количество конкурсных работ (58), так и количество проголосовавших (105) в этот раз были заметно больше, чем на прошлом Antique Toy. Ниже приведены краткие результаты — первые три места в каждой из номинаций (в скобках указывается количество баллов, набранное работой).

Mainstream music (всего 22 работы):
1 — "Konami" by C-jeff/Brainwave/Mimic/AY Riders (627), 2 — "Der Flaum" by Key-je/Triebkraft (599), 3 — "Under A Clear Bluemoon" by Gasman/Hooy-Program/AY Riders (586).

Progressive music (всего 11 работ): 1 —

"Against The Terror" by Back Noise and Gibson/USG (627), 2 — "The Soul Is Alive" by Ik/Tiraspol Art Group (621), 3 — "Freeze Soul" by Moran/Cyberpunks Unity (572).

Cover music (всего 12 работ): 1 — "Heroquesters" by n1k-o/skrju (622), 2 — "Dancing Queen" by Gasman/Hooy-Program/AY Riders (578), 3 — "It's Come From The Dark" by Black Fox (573).

Traditional graphics (всего 7 работ): 1 — "Finality" by Deadie/Horrorsoft/CPU (830), 2 — "Sprengstoff" by Raynoal/Mayhem/Psycho (656), 3 — "Alien Ate It" by The Sentinel (522).

Alternative graphics (всего 6 работ): 1 — "Friendship" by Exocel/Just For Fun/Spaceballs/Checkpoint (631), 2 — "Kinder Tripman" by Moran/Cyberpunks Unity (538), 3 — "Satan Claus" by Yerzmyey/Hooy-Program/AY Riders (416).

Полная версия результатов, а также сами конкурсные работы доступны на сайте Scenery (<http://scenery.natm.ru>).

Подготовил И.Рощин.



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

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

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

Вышлю по электронной почте копии статей из радиолобительских журналов прошлых лет (60...90 гг.). Подробный каталог в письме. Андрей.
E-mail: oldcopy@yandex.ru

Санитарка терапевтического отделения с благодарностью примет в дар любой компьютер б/у.

247703, Гомельская обл., Калининский р-н, д. Бобровичи, 144 — 6, Жевлакова В.Г.

Меняю диски с "Лонгхорн", "КьюЭнИкс РТП6", "Инсерт", "ОС X", 128000 адресов электронной почты, "Sybase PowerDesigner 9.5" на б/у системный блок Р-133.
E-mail: vianst2004@mail.ru.

Продам недорого импортный контроллер для общей шины ("Электроника 60") с двумя винчестерами; приборы Ц-315, Ц-435, Ц4317М, Ц20-05, ТЛ4М, ПНТ-59, Ц4312, ГНЧР-2, ГЦТ-03 (PAL), ГИР, цифровую шкалу; однокассетную магнитоу. 140033, Московская обл., пос. Малаховка, Быховское шоссе, 45 — 32, Кутузов Е.В.

Куплю плату ISA WinRadio за символическую цену.
E-mail: rogojinandrey@mail.ru

Продаю коллекцию CD для "Sega Dreamcast" и "Panasonic 3DO" (различные эмуляторы и программы, много игр и фильмов), кабель для подключения "Dreamcast" к компьютерному монитору. Вышлю по почте. Могу выслать список имеющихся дисков (вложите в письмо конверт с обратным адресом).

442150, Пензенская обл. г. Нижний Ломов, а/я 58, Сергей.

Продам-куплю программы "ZX-Spectrum" в среде TR-DOS и ОС CP/M. Предлагаю каталог с дискетой.

632383, г. Куйбышев, НСО, 1 — 20 — 45, Третьяков А.А.

Меняю: мультиметр цифровой МИЦ-01, компьютер БК-01, видеоконсоль ЛН-441 с ОС-5-01, твелекмеру КТП-63, клавиатуру герконовую ЕС 9003 на радиостанции "Ангара", Р-143 или аналогичные. 399540, Липецкая обл., пос. Тербуны, ул. Первомайская, 13, Харитонов В.П.
Тел. 8-274-2-29-29, с 19.00.

Продаю тюнер winradio WR-1000 (частота до 1,3 ГГц, CW, SSW, AM, FMN) в виде платы в компьютер.

Тел. 8-029-644-88-90.
E-mail: winradio@mail.ru

Ищу процессор ATHLON с разъемом SLOT-A.
E-mail: santinal@yandex.ru

Куплю запрограммированное ПЗУ для "ZX-Spectrum 128 к" с TR-DOS версии 5.03, схему "ZX-Spectrum 128 к" с музыкальным синтезатором на MC YM2149F и с контроллером ИГМД.
617833, Пермская обл., г. Чернушка, ул. Дружбы, 15, Хатмуллин И.Х.

Подписные индексы:

Радиомир

- для жителей России и стран СНГ (кроме Беларуси): 48996 — подписка по каталогу Агентства "Роспечать" (72370 — годовая), 25785 — подписка по Объединенному каталогу "Пресса России" с адресной рассылкой (электронный адрес подписки в INTERNET - www.presssafe.ru);
- для жителей Беларуси: 48996, 489962 (для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

Радиомир. КВ и УКВ

- для жителей России и стран СНГ (кроме Беларуси): 48924 — подписка по каталогу Агентства "Роспечать", (71545 — годовая);
- для жителей Беларуси: 48924, 489242 (для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

Радиомир. Ваш компьютер

- для жителей России и стран СНГ (кроме Беларуси): 48925 — подписка по каталогу Агентства "Роспечать" (45995 — годовая);
- для жителей Беларуси: 48925, 489252 (для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.

Кроме того, предприятия регионов России, а также ближнего и дальнего зарубежья могут оформить подписку на журналы "Радиомир", "Радиомир. КВ и УКВ", "Радиомир. Ваш компьютер" через ООО "Корпоративная Почта" по телефонам: (095) 953-92-62, 953-92-02, 953-93-20.

Получить подписку или заказать имеющиеся в наличии отдельные номера журналов можно из редакции. Текущие цены приведены в таблице. Наложным платежом журналы не высылаются.

Год	Можно заказать следующие номера журналов			Расценки на 1 экз. (с учетом пересылки)		
	Радиолобитель	Радиолобитель. Ваш компьютер	Радиолобитель. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2000	3 — 6, 8 — 12	3 — 6, 10 — 12	6, 7, 9, 11, 12	20	800	6
2001	4 — 5	2, 5, 6	2, 3, 5, 6	25	900	6
	Радиомир					
	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2001	7, 9, 10, 12	7, 9 — 12	7, 8, 10, 11	30	1000	7
2002	2 — 12	1 — 12	1 — 3, 5 — 12	35	1500	7
2003	1 — 12	1 — 12	1, 5 — 8, 10 — 12	40	1800	8
2004	1 — 6, 8 — 12	1 — 6, 8 — 12	1 — 12	45	2100	9
2005	1 — 12	1 — 12	1 — 12	45	3000	11

Наши платежные реквизиты:

- для жителей России и стран СНГ (кроме Беларуси) —
получатель: ООО "Радиомир Пресс", ИНН 7729404741, КПП 772901001,
р/с 40702810900010000084 а ООО КБ "Агропромкредит", ф-л Центральный, г. Москва,
корр. счет 30101810500000000109, БИК 044525109
Адрес банка: 125315, г. Москва, Ленинградский пр-т, дом 76, корп. 4;

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

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

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

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

Приобретение отдельных номеров журналов

В РОССИИ:

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

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

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

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

В ООО "Альянс-Книга": 115533, г. Москва, ул. Нагатинская, д. 6, эт. 1, оф. VII (ст. метро "Нагатинская"), тел. (095) 258-91-94, 258-91-95, 258-91-96, 111-05-88, 111-63-28. E-mail: abook@mail.ru, abook@inbox.ru
На радиорынках: "Царицыно" (место 13/А), "Митино" (ряд 1, контейнер 17 и место Т-8).

На УКРАИНЕ:

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

В КАЗАХСТАНЕ:

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

В БЕЛАРУСИ:

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



КОМПЬЮТЕРЫ СИТИПРИНТ ДЛЯ ДОМА И ОФИСА

**Компьютер — это инвестиция
в развитие Вашего бизнеса!**

Приобретая в компании “СитиПринт” компьютер на базе процессора Intel® Pentium® 4 с технологией HT, вы инвестируете в свое будущее.

Новый компьютер предоставит необходимую поддержку для Вашей автоматизированной системы бухгалтерского учета, позволит оперативно создавать и рассылать электронные письма клиентам и поставщикам, а также изготавливать профессиональные презентации.

**Подарите Вашему бухгалтеру точность
современных технологий.**



 **СитиПринт**
КОМПЬЮТЕРЫ И ОФТЕХНИКА

220006, г. Минск, ул. Полесья 28-7А. Тел./факс (017)2850134.
www.citiprint.by





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

7 способов отправки сообщений!

Стационарные охранные системы для дома и офиса

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



am® Атлант Моторс



Спутниковые автомобильные охранные системы

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