

радиомир

11 • 2005

Ваш компьютер

- Локальная сеть у вас дома
- Утечка памяти
- Низкоуровневый доступ к дискете
- Восстановление поврежденной BIOS





КОМПЬЮТЕРЫ **СИТИПРИНТ** ДЛЯ ДОМА И ОФИСА

**Ваши сотрудники способны на большее?
Позвольте им доказать это**

Компьютер от компании "СитиПринт" на базе процессора **Intel® Pentium® 4** с технологией **HT** поможет Вашим сотрудникам выполнять больше работы за меньшее время.



 **СитиПринт**
КОМПЬЮТЕРЫ И ОРГТЕХНИКА

220006, г.Минск, ул.Полевая 28-7А.
Тел./факс (017)2850134, 2218728.
www.citiprint.by

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

Главный редактор
Ольга СТРЫЖАНКОВА
Зам. гл. редактора
Иван БЕЛЬСКИЙ

Редколлегия:
Сергей ДРОЗДОВСКИЙ,
Владимир КУЦЕНКО,
Елена ЛЕВИТМАН,
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,
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: 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 рек-
ламного 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 публи-
куемой информации несут ответственность
рекламодатели и авторы. Мнение редакции
не всегда совпадает с мнениями авторов

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

Подписано к печати 27.09.2005 г.
Формат 60 x 84 1/8.
Печать офсетная. 6 печ. л.
Цена свободная.

© ООО "Радиомир Пресс". Воспроизведение матери-
алов журнала в любом виде без письменного разре-
шения редакции запрещено. При цитировании ссыл-
ки на "Радиомир. Ваш компьютер" обязательна.

Отпечатано в типографии
ООО "Красногорская типография".
Заказ № 2738.

радиомир Ваш компьютер

ЕЖЕМЕСЯЧНЫЙ МАССОВЫЙ ЖУРНАЛ

№11/ноябрь/2005

ЧИТАЙТЕ В НОМЕРЕ

ВОКРУГ ПК

НЕ ТОЛЬКО НОВИЧКУ

А.ГРИНЧУК. Windows XP для пользователя	
Работа в локальной сети	2
Е.ШАКЕЛЬ. Эффективная работа с табличным процессором Microsoft Excel XP	
Создание пользовательских диаграмм	5
ST_НАТАВИШ. Моддинг. Наука, искусство или изыски XXI века?	9
В.КУЦ. Acronis TrueImage 8.0	11
С.РЮМИК. Интернет-калейдоскоп, freeware #24	15

КОММУНИКАЦИИ

Б.ШИЛЬНИКОВ. Локальная сеть у вас дома	17
А.КАШКАРОВ. Чудо XX века: реальность и перспективы	19

ДАЙДЖЕСТ

ПРОГРАММЫ И АЛГОРИТМЫ

УРОКИ ПРОГРАММИРОВАНИЯ

О.ВАЛЬПА. Borland C++ Builder 6 для начинающих	25
А.КОРБИТ, Т.КРИВОНОСОВА. Поиск и сортировки: сортировка обменом	26

ДИАЛОГ ПРОГРАММИСТОВ

Р.ФАСИХОВ. Написание плагинов для перехватчика API-функций	28
В.КАРЦЕВ. Управление потоком сообщений в приложениях, использующих библиотеку MFC	30
И.РОЩИН. Программа для поиска и замены	33
А.КОРБИТ. Утечка памяти	34
С.РУСАКОВ. Низкоуровневый доступ к диску в Win 2000/XP	36
Е.БОНДАРЕНКО. Алгоритм решения японских кроссвордов	38

РЕЦЕПТЫ

В.ФЕДОРОВ. Простое восстановление поврежденной BIOS компьютера	41
Возвращаясь к напечатанному №6/2005, С.37. А.АНИСКОВИЧ. Как преобразить свой компьютер	42
А.КАШКАРОВ. Простое автоматическое включение колонок для ПК	44

МИР 8 БИТ

Я.ОЧАКОВСКИЙ. Системные переменные	45
--	----

ДОСКА ОБЪЯВЛЕНИЙ

Куплю, продам, обменяю	47
------------------------------	----

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

А.ГРИНЧУК,

E-mail: gav@nihe.niks.by

Windows XP для пользователя

Работа в локальной сети

Красиво выживать не запретишь.
Джон Рембо

Сейчас осталось не так уж и много компьютеров, работающих "сами по себе", и домашние, и офисные компьютеры часто подключены к сети. Это не только дань моде — работа в сети качественно отличается от индивидуальной. Конечно, преимуществ больше. Можно совместно использовать документы, что позволяет переходить с машины на машину, не таская за собой самые необходимые файлы. Также наличие сети помогает решить проблему дискового пространства: рядом может найтись компьютер, на котором размер винчестера побольше. Дополнительно можно упомянуть совместное использование принтеров и сканеров и, не так уже актуальное, совместное использование CD-ROM. Нельзя обойти стороной и набравшие популярность сетевые игры. За все это приходится расплачиваться некоторой утратой безопасности и надежности. Локальная сеть в большей степени уязвима перед атаками компьютерных вирусов. Также более проблематичной становится защита информации. Увы, построение и настройка локальной сети — задача еще более сложная и многогранная, чем настройка операционной системы. Поэтому ограничимся здесь основными приемами "выживания" в сети.

Прежде всего, стоит упомянуть, что компьютеры в сети подразделяются на две категории: серверные (управляющие сетью) и клиентские, предназначенные для обычной работы. Сами сети бывают двух типов. Первый вариант — одноранговые, когда каждая машина выполняет как клиентские, так и серверные функции. Преимуществом является простота создания, недостатком — нестабильность работы. Поэтому в одноранговую сеть объединяют не более 5 — 7 компьютеров. Для более масштабной работы создаются иерархические сети. В сетях такого типа один

из компьютеров выделяется для выполнения исключительно серверных функций, остальные — сугубо для клиентских. Благодаря такой специализации удается добиться большей надежности и эффективности, хотя фактически приходится жертвовать одной машиной. Поэтому иерархические сети стоит создавать, когда используется большое количество клиентских машин. Но в этом случае возникает новая проблема — в десятках компьютеров легко запутаться. Для облегчения навигации компьютеры часто (но не

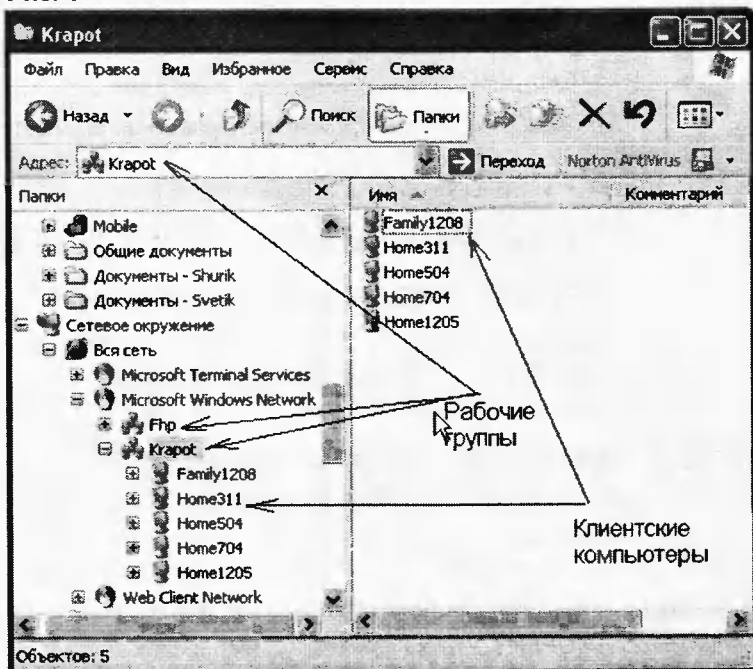
ответствующие аппаратные средства и программное обеспечение. Итак, в первую очередь нужна сетевая карта (сетевая плата, сетевой адаптер — все это синонимы) и физическое подключение к сети, чаще всего осуществляемое с помощью стандартного сетевого кабеля. А вот что касается программного обеспечения, то здесь удастся обойтись без лишних хлопот — все, что нужно для успешной работы, уже входит в состав Windows XP.

Вполне возможно, что вам понадобится доступ к папке с документами, находящейся на другом компьютере, или же вы захотите поделиться своими документами. Для этого:

- на компьютере (внимание!), на котором расположена нужная папка, запускается Проводник и открывается папка;
- по папке производится щелчок правой клавишей мыши и выбирается пункт меню **Общий доступ и безопасность** (рис.2);
- устанавливается флажок **Открыть общий доступ** (рис.3).

Если вы хотите предоставить возможность полноценной работы с файлами, то необходимо **Разрешить изменение файлов по сети**. И, наконец, имя папки на ло-

Рис. 1



всегда) объединяют в рабочие группы. Эта система напоминает структуру файлов и папок (рис.1), но за одним исключением — если в папку можно вкладывать другие папки, то к рабочим группам такая операция неприменима. Зато доступ к сети можно получить прямо из Проводника — **Сетевое окружение\Вся сеть\Microsoft Windows Network\Рабочая группа\Клиент**.

Что же нужно для того, чтобы ваш компьютер мог работать в сети? Стандартный ответ на подобного рода вопросы звучит так — нужны соот-

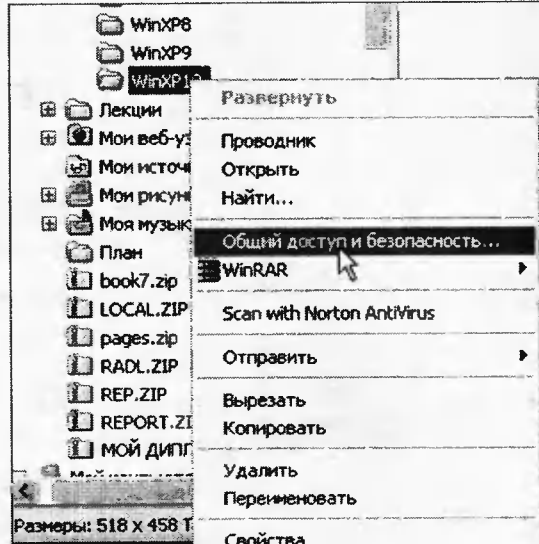


Рис. 2

Рис. 3

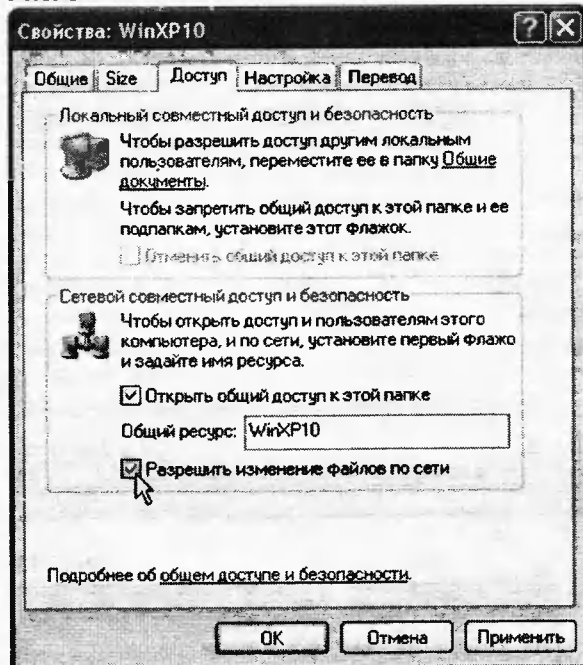


Рис. 5

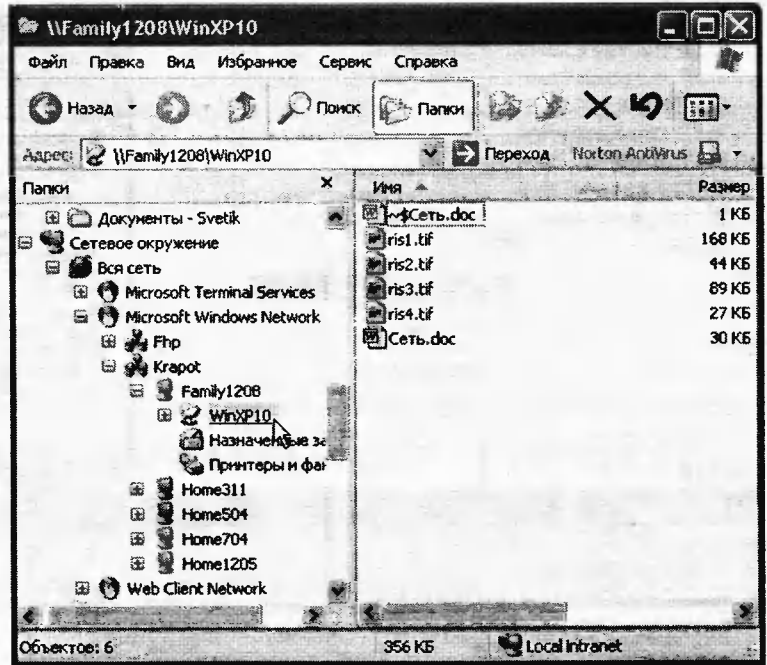
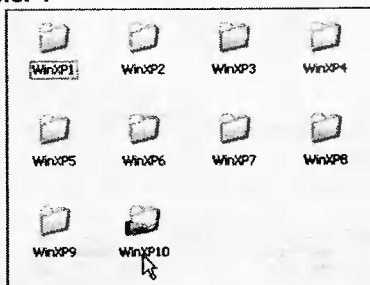


Рис. 4



кальном компьютере и в сети может не совпадать. Сетевое имя указывается в поле **Общий ресурс** (рис.3). Общую папку легко отличить от остальных по внешнему виду — значок папки теперь сопровождается рисунком руки, удерживающей папку (рис.4). Изъятие папки из общего доступа достаточно очевидно: щелчок правой клавишей по папке **Общий доступ и безопасность** / **Снять флажок Открыть общий доступ** (рис.3).

После всех манипуляций папку можно наблюдать из сети (рис.5). Обратите внимание, что при «взгляде из сети» не содержится никакой информации о том, где находится папка (диск размещения, родительская папка и т.д.).

Однако навигация в локальной сети часто сопровождается задержками в ра-

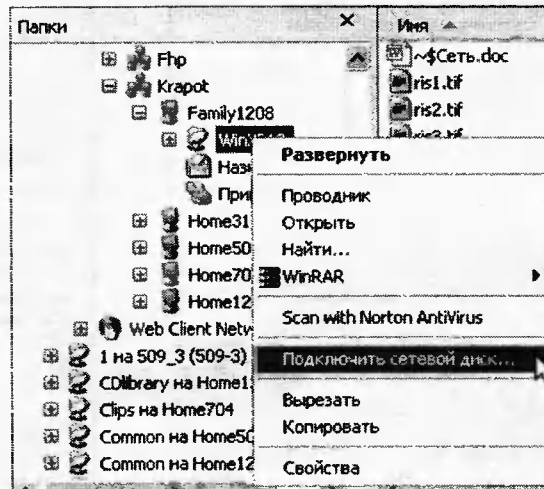


Рис. 6

— достаточно один раз найти в сети нужную папку, щелкнуть правой клавишей мыши и выбрать команду **Подключить сетевой диск** (рис.6). Появляется возможность «обмануть» компьютер — эта папка будет восприниматься как еще один диск на вашей машине. Только предварительно придется указать имя диска (рис.7), а выбор опции **Восстанавливать при входе в систему** позволит автоматически подключать диск при загрузке операционной системы. Результаты труда можно проверить в Проводнике — в общем списке дисков появится новый элемент со значком, немного отличающимся от стандартного (рис.8). Есть и возможность избавиться

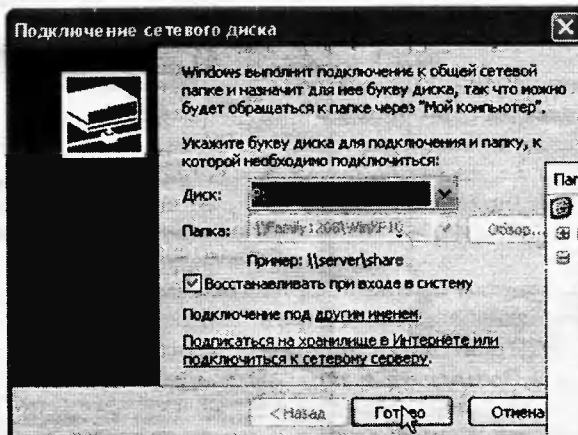
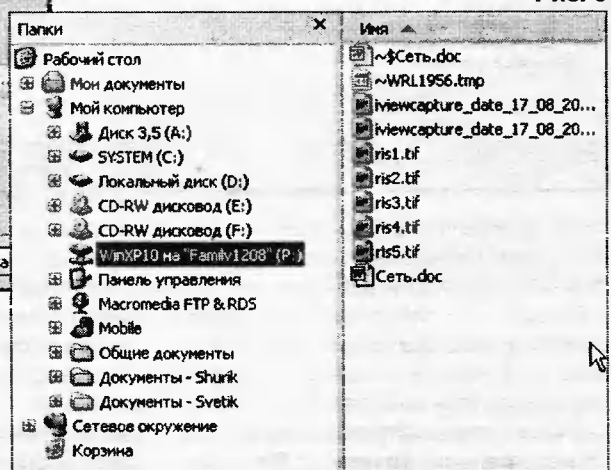


Рис. 7

воднике — в общем списке дисков появится новый элемент со значком, немного отличающимся от стандартного (рис.8). Есть и возможность избавиться

Рис. 8



боте — приходится тратить драгоценные секунды на ожидание отклика сервера. Да и чаще всего постоянно нужна одна и та же папка. Выход есть

Рис. 9

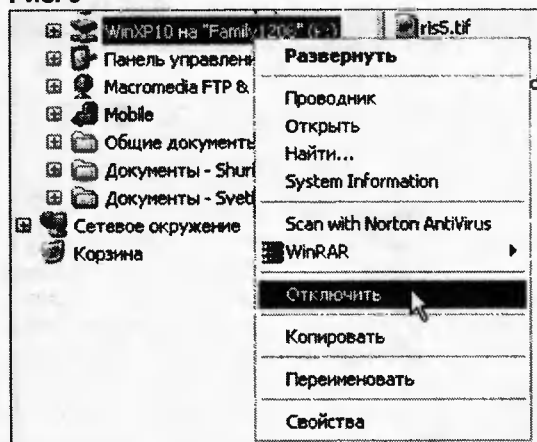


Рис. 10

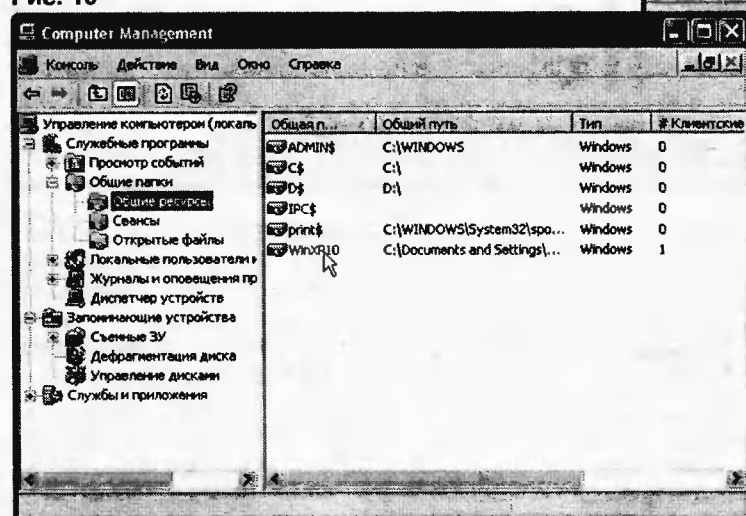
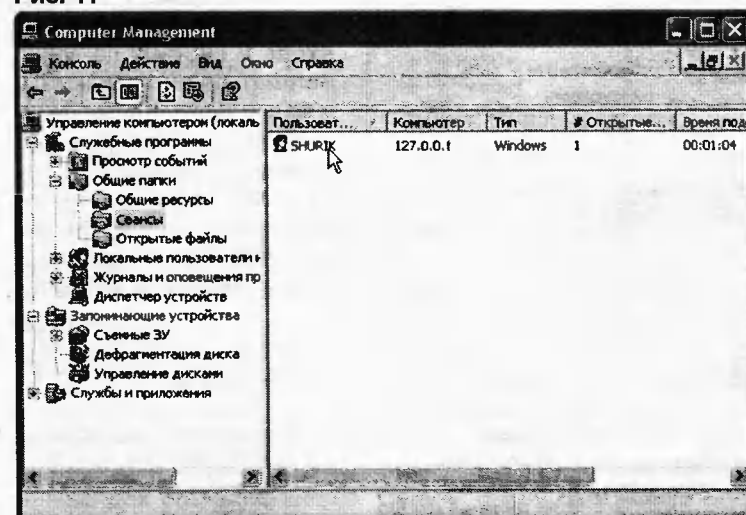


Рис. 11



ся от ненужного подключения — щелчок правой клавишей по сетевому диску и команда **Отключить** (рис.9).

Естественно, работа с использованием сети становится более удобной, однако это удобство — палка о двух концах. Приходится контролировать открытые папки и просматриваемые другими пользователями документы. Вот толь-

ко копание в Проводнике с целью найти все "папки с ручками" — дело неблагодарное. Необходимые сведения можно получить, пройдясь по маршруту **Пуск\Панель управления\Администрирование\Управление компьютером\Общие папки**. Спря-

Рис. 12

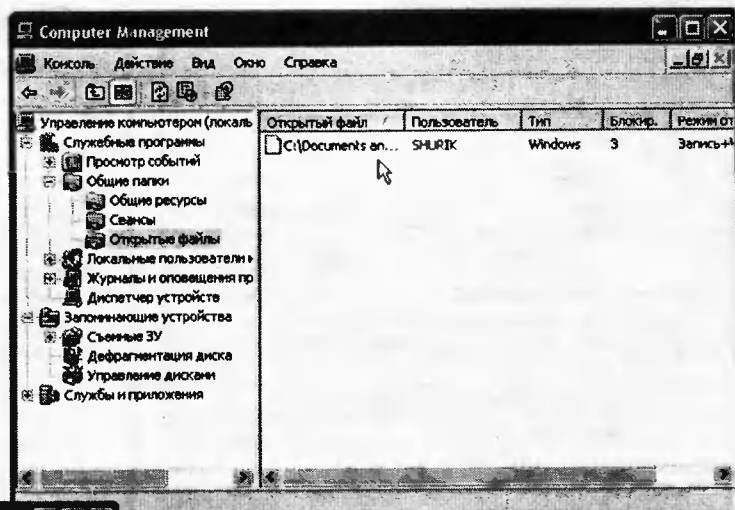


Рис. 13

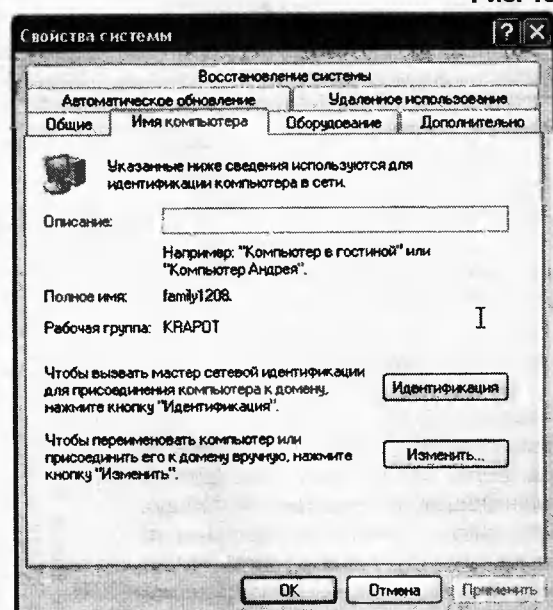
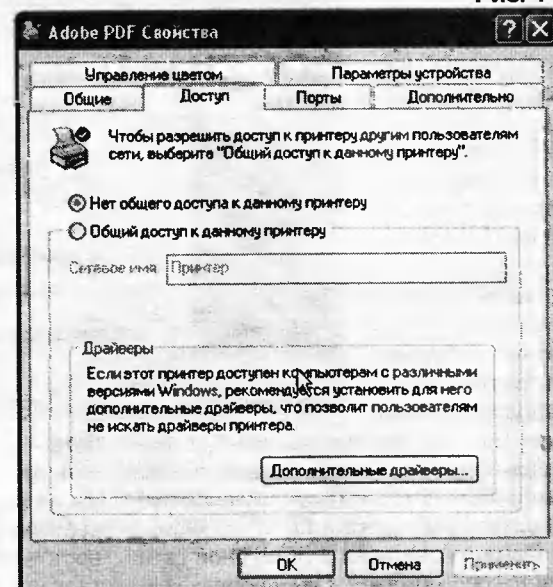


Рис. 14





тано далековато, зато есть все, что необходимо. В разделе **Общие ресурсы** (рис.10) содержится информация обо всех открытых в общий доступ объектах, в том числе и полные пути к папкам. **Сеансы** (рис.11) отображают список подключенных пользователей, а в разделе **Открытые файлы** (рис.12) можно узнать детали о том, кто и что конкретно просматривает. Отключить зарвавшегося соседа можно щелчком правой клавишей мыши и выбором команды **Отключить...**

Порой в тупик ставит и такой воп-

рос — как же узнать имя компьютера, на котором в данный момент работаешь? Ответ можно найти в свойствах системы: **Пуск\Панель управления\Система**. На вкладке **Имя компьютера** указывается как само имя, так и название рабочей группы (рис.13).

К панели управления приходится обращаться достаточно часто. Например, понадобится она и при предоставлении принтеров в общий доступ: **Пуск\Панель управления\Принтеры и факсы**. Далее, по аналогии — щелчок правой клавишей

Общий доступ\Общий доступ к данному принтеру (рис.14).

Почти все внешние устройства подключаются по схожему сценарию. Можно вспомнить и CD-ROM, и Flash-карту. Если присмотреться внимательнее, то в последние годы все основные достижения в компьютерной технике, да и в развитии Windows связаны именно с сетевыми технологиями. Так что смелее используйте сеть, но только будьте морально готовы к тому, что, помимо глюков программ, придется разбираться и с коллизиями в сети.

Е.ШАКЕЛЬ,

г.Минск, РИВШ БГУ,

E-mail: shakel@nihe.niks.by

Эффективная работа с табличным процессором Microsoft Excel XP

(Продолжение. Начало в №№1-8,10/2005)

Создание пользовательских диаграмм

Наш сегодняшний урок я предлагаю начать с довольно интересного приема, который, возможно, заинтересует не только тех, кто еще учится работать с табличным процессором Microsoft Excel XP, но и тех, кто считает себя в некоторой степени специалистом. Позвольте спросить, что вы делаете в том случае, если руководство поручило вам выполнить глобальный отчет, иллюстрированный огромным количеством диаграмм? Используйте готовые образцы? А если вас попросили использовать оригинальное оформление? Оформляете вручную. Но это очень долго и очень неудобно. Однако выход из сложившейся ситуации есть — это создание собственных нестандартных типов диаграмм, которые вы в дальнейшем сможете использовать совместно с другими пользователями. Например, если один и тот же заголовок и цветовое оформление должны сохраняться во всех диаграммах, то можно создать диаграмму с этим заголовком, выбрать подходящее цветовое решение, сохранить ее как нестандартный пользовательский тип, а затем использовать его в качестве шаблона для создания всех последующих диаграмм.

Используем данный прием. Начнем как всегда с загрузки приложения Microsoft Excel XP и открытия рабочей книги **Учебный_файл_1.xls**. Наш предыдущий урок был посвящен приемам создания, редактирования и форматирования диаграмм. В результате были построены две диаграммы. Оформите

диаграмму "Количество единиц" в соответствии с рис.41, если вы не сделали это на прошлом уроке, или сохраните свое оформление.

Рис. 41



Затем:

- выделите ее;
- в меню **Диаграмма** выполните команду **Тип диаграммы**;
- переключитесь на вкладку **Нестандартные**;
- в группе **Вывести** установите переключатель **дополнительные**;
- щелкните по кнопке **Добавить** и введите название и описание нового типа диаграммы (рис.42)

- щелчок по кнопке **ОК** завершит сохранение данной диаграммы в качестве шаблона.

А теперь используем свой шаблон для иллюстрации ряда данных "Общая стоимость":

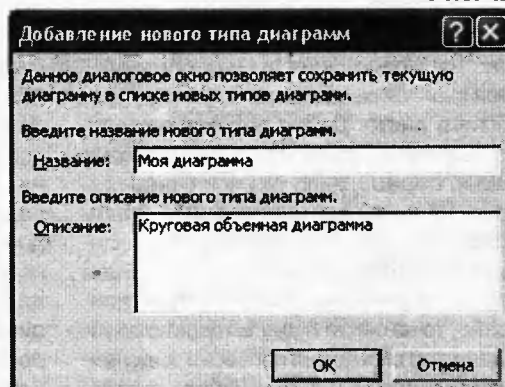
- выделим одновременно два диапазона ячеек — A1:A7 и D1:D7;

- выполним команду **Диаграмма меню Вставка** или воспользуемся инструментом  на панели **Стандартная**;

- на вкладке **Нестандартные** установим переключатель **дополнительные** и выберем тип **Моя диаграмма**;

- пользуясь кнопкой **Далее>**, последовательно пройдем по остальным шагам мастера, выбирая нужные параметры. Диаграмма готова. Оформление ее в точности соответствует сохраненному шаблону.

Рис. 42



Как подготовить рабочую книгу к печати

Итак, дорогой читатель, если ты регулярно следишь за нашими публикациями, выполняешь предлагаемые са-



мостоятельные задания и придерживаясь рекомендаций автора, то, очевидно, у тебя имеется уже практически готовая рабочая книга, содержащая набор рабочих листов с расчетными таблицами и диаграммами. Остались буквально последние штрихи для того, чтобы твоя работа выглядела безукоризненно не только на экране компьютера, но и в печатном виде.

Прежде чем печатать, хотелось бы убедиться в том, что, во-первых, в созданных нами таблицах нет ошибок, во-вторых, что печатный вариант наших стараний будет настолько же хорош, как и электронный. А для этого, рассмотрим процедуры проверки орфографии, предварительный просмотр и установку параметров страницы.

Проверка орфографии

Проверка электронной таблицы на наличие орфографических ошибок является важной составляющей редактирования рабочего листа, поскольку, в отличие от текстового процессора Microsoft Word, эта процедура не является автоматической. Проверку орфографии можно осуществлять для текущего рабочего листа, рабочей книги целиком, а при необходимости — и для произвольного фрагмента ячеек. В первом случае достаточно находиться на том рабочем листе, который нуждается в проверке, во втором — должны быть выделены ярлыки всех рабочих листов рабочей книги, в третьем — нужный диапазон ячеек. Отметим, что выделить ярлыки всех рабочих листов одновременно можно, щелкая по ним мышью при нажатой клавише **Ctrl**, либо при помощи команды **Выделить все листы** контекстного меню, которое появится у вас на экране, если вы щелкните правой кнопкой мыши по ярлыку любого рабочего листа. Далее необходимо воспользоваться командой **Орфография** меню **Сервис**, либо кнопкой **Орфография** на панели инструментов **Стандартная**, либо функциональной клавишей **F7** верхнего ряда клавиатуры. Если орфографических ошибок не будет найдено, то на экран будет выведено сообщение об окончании проверки выделенной области, листа или набора листов. Если же будут найдены ошибки или неизвестные слова, то общение пользователя с компьютером продолжится в окне диалога **Проверка орфографии** (рис.43). Здесь в области **Нет в словаре** указывается слово, которое счи-

Рис. 43

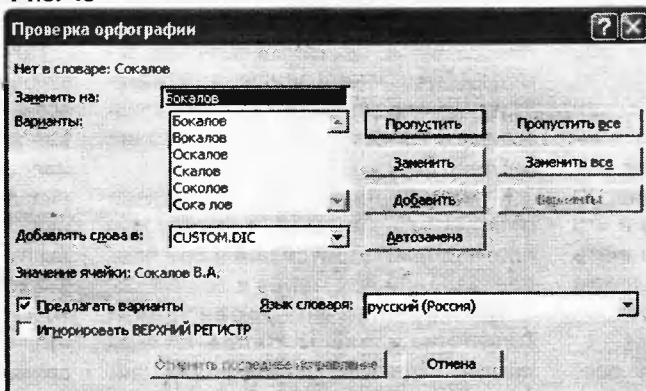
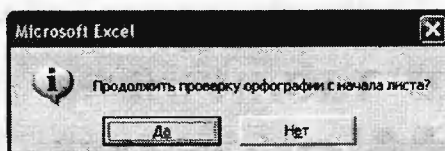


Рис. 44



тается ошибочным, в поле **Варианты** — возможные варианты его замены, если таковые существуют. Заметим, что при проверке орфографии используется встроенный словарь, поэтому не удивляйтесь, что, кроме слов, которые действительно содержат ошибки, вы увидите и слова, написанные правильно, но не содержащиеся в словаре. Их программа тоже считает ошибочными. Выбрав щелчком мыши нужный вариант для замены или впечатав его с клавиатуры в поле **Заменить на**, щелкните по кнопке **Заменить** для замены только данной ошибки или **Заменить все** для автоматического исправления аналогичных ошибок далее в электронной таблице. В случае, если предлагается исправить правильно написанное слово, вы можете пропустить его в исходном виде, щелкнув по кнопке **Пропустить**, либо пропустить аналогичные слова во всем тексте, воспользовавшись кнопкой **Пропустить все**, либо добавить данное слово в словарь пользователя при помощи кнопки **Добавить** для того, чтобы в дальнейшем при проверке других файлов это слово не распознавалось как ошибочное. После того как вы благополучно добрались до конца рабочего листа, рабочей книги или выделенной области и исправили все ошибки, на экране возникнет окно диалога с информацией об окончании проверки. Щелкните в нем по кнопке **OK** и вы выйдете из режима проверки. Если вы осуществляете проверку орфографии текущего рабочего листа, то возможна и другая ситуация — в процессе проверки вы увидите диалоговое окно (рис.44). Не удивляйтесь, просто проверка орфографии начинается с той

ячейки, в которой находится табличный курсор, а дойдя до конца рабочего листа, программа предлагает проверить ту часть, которая осталась не затронутой. Щелкните по кнопке **Да**, если хотите сделать это.

Предварительный просмотр

А что же дальше? Возможно, многие из вас скажут, что все приготовления закончены и можно печатать. На самом деле это не так. Настало время проверить, как же будут выглядеть результаты нашего творчества на бумаге.

Необходимо помнить, что экранное представление таблицы, как, впрочем, и диаграммы, зачастую значительно отличается от того, которое получается при выводе на печать. Это объясняется тем, что рабочий лист, имеющий значительные размеры, разбивается на фрагменты, размер которых определяется размерами печатного листа. Кроме того, печать некоторых элементов рабочего листа, таких как заголовки строк и столбцов, границы ячеек или примечания к ячейкам, определяется самим пользователем. С другой стороны, такие компоненты как номера страниц, колонтитулы или поля, к присутствию которых на печатной странице мы привыкли, также не являются обязательными и не задаются автоматически.

Для этого, чтобы проверить, как будет выглядеть ваша таблица или диаграмма на бумаге, используется режим предварительного просмотра. Перейдите на рабочий лист **Заработная плата** и выполните команду **Предварительный просмотр** меню **Файл**, либо воспользуйтесь инструментом **Предварительный просмотр** панели инструментов **Стандартная**. Данный режим исключает всякую возможность редактирования и форматирования электронной таблицы, за исключением изменения ширины ее столбцов, но позволяет задавать и просматривать параметры печатной страницы. Ряд кнопок в окне предварительного просмотра предусматривает опции просмотра, настройки параметров для печатаемых листов, параметров принтера и печати (табл.17).

Мы рекомендуем вам последовательно просмотреть действие каждой из кнопок. При нажатии кнопки **Разметка страницы** вы выйдете из режима предварительного просмотра и будете просматривать электронную

Табл. 17. Режим предварительного просмотра

Вид кнопки	Действие
	Просмотр следующей страницы в многостраничных рабочих листах. Если выделено несколько листов, то по достижении последней страницы текущего листа при нажатии кнопки Далее отображается первая страница следующего выделенного листа
	Просмотр предыдущей страницы в многостраничных рабочих листах. Если выделено несколько листов, то по достижении первой страницы текущего листа при нажатии кнопки Назад отображается последняя страница предыдущего выделенного листа
	Переключение между полноэкранным режимом просмотра и режимом увеличения. Перейти из полноэкранный режима просмотра в режим увеличения и обратно можно также щелчком мыши по любой области листа
	В окне диалогов Печать можно установить параметры принтера и печати, а затем напечатать выделенный лист
	В окне диалогов Параметры страницы можно установить параметры страницы для печатаемых листов
	Включение/отключение маркеров, которые можно перетаскивать мышью для настройки полей страницы, полей верхнего и нижнего колонтитулов и ширины столбцов
	Переход в режим разметки страницы. В данном режиме отображаются границы печатных страниц. Можно изменить область печати, нажав мышью на границе и переместив ее в другое место
	Выход из режима предварительного просмотра и возвращение к обычному режиму просмотра активного листа
	Открытие окна справки Microsoft Excel для раздела Окно предварительного просмотра

таблицу в режиме разметки. Если после этого вы снова войдете в режим предварительного просмотра, то вместо кнопки **Разметка страницы** увидите кнопку **Обычный режим**, которая вернет вас к исходному режиму отображения таблицы.

Установка параметров страницы.

Итак, если вас полностью удовлетворили результаты предварительного просмотра, то можно приступать к печати. Но, чаще всего, это не так. Поэтому перед выводом на печать таблицы или диаграммы необходимо сначала произвести установку параметров страницы: настроить размеры полей печатной страницы и расстояния от края бумаги до верхнего и нижнего колонтитула, определить масштаб печати и ориентацию печатной страницы. Кроме того, можно добавить на печатный лист верхний и нижний колонтитулы: дату, время создания файла, имя рабочей книги, имя рабочего листа, номер страницы или произвольную текстовую информацию. При необходимости можно разрешить или запретить печать заголовков строк и столбцов электронной

Рис. 45

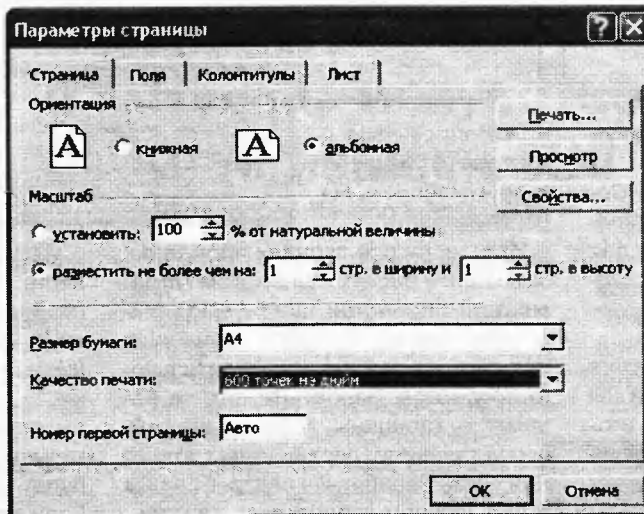


Рис. 46



таблицы, границ ячеек или примечаний к ячейкам... Согласитесь, что, например, в нашем случае результат был бы значительно лучше, если бы мы смогли разместить таблицу "Расчет заработной платы сотрудников" на одном печатном листе, желательно поперек листа, а также добавили колонтитулы и нумерацию страниц. В данном случае необходимо воспользоваться командой **Параметры страницы** меню **Файл**.

На вкладке **Страница** (рис.45) в группе **Ориентация** выберем переключатель **Альбомная**, что позволит разместить таблицу поперек печатного листа; в группе **Масштаб** установим переключатель **разместить не более чем на 1 стр. в ширину и 1 стр. в высоту**, при этом программа подберет подходящий масштаб печати. Поля со списками **Размер бумаги** и **Качество печати** служат для выбора характеристик печатного листа и качества изображения соответственно.

Теперь переключимся на вкладку **Поля** (рис.46) и установим нужные размеры в полях **верхнее, нижнее, правое, левое**, а также расстояние до колонтитулов в полях **верхнего колонтитула, нижнего колонтитула**. Флажки **горизонтально** и **вертикально** раздела **Центрировать на странице** позволят выбрать подходящее размещение всей таблицы целиком на печатном листе.

Вкладка **Колонтитулы** (рис.47) используется для создания колонтитулов, вид которых можно просто выбрать из списков **Верхний колонтитул** и **Нижний колонтитул** соответственно, либо создать самостоятельно, если воспользоваться кнопками **Создать верхний колонтитул** и **Создать нижний колонтитул**. Итак, немножко творчества. Щелчком по кнопке **Создать верхний колонтитул** и увидим перед собой окно диалогов рис.48. Выберем место размещения колонтитула. Предположим, что слева печатного листа требуется разместить имя файла, справа — дату и время создания файла. Для этого воспользуемся кнопками окна диалогов (табл.18). Щел-

Рис. 47

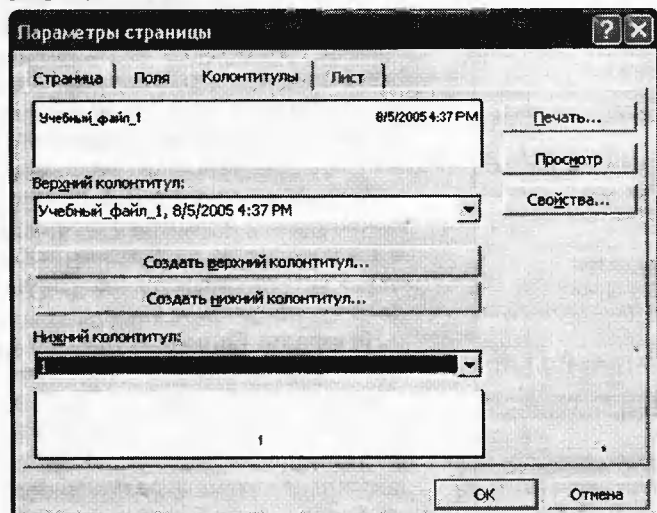
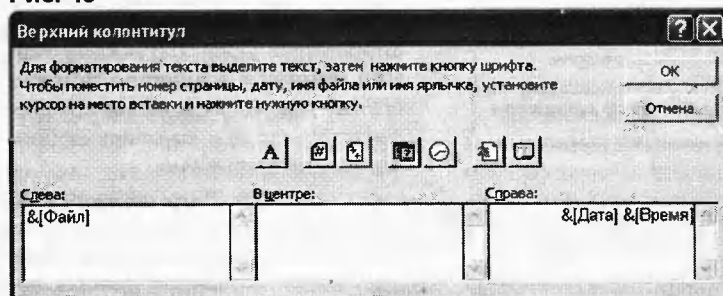


Рис. 48



чок по кнопке **OK** завершит создание колонтитула.

Аналогично, щелкнув по кнопке **Создать нижний колонтитул**, откроем соответствующее окно диалога для создания нижнего колонтитула, состоящего из номера страницы, расположенного в центре.

На вкладке **Лист** (рис.49) задаются дополнительные параметры, такие как черно-белая или черновая печать, отображение заголовков строк и столбцов (это особенно удобно при подготовке тестовых заданий для студентов или учащихся по данной теме), распечатка разделительных линий ячеек (сетки). Настройка осуществляется с помощью соответствующих флажков в группе **Печать**. Здесь же можно указать порядок вывода печатных страниц: **вниз, затем вправо** или **вправо, затем вниз**. В группе **Печатать на каждой странице** можно указать те строки или столбцы, которые будут использоваться в качестве заголовков на каждом листе при печати вашей электронной таблицы, а в поле **Выводить на печать диапазон** — ту область ячеек, которая предназначена для печати, хотя проще распечатку избранного диапазона ячеек делать другим способом, который будет описан ниже. Здесь же для выбора нужного диапазона поставьте курсор

в указанное поле и выделите мышкой нужную область на рабочем листе.

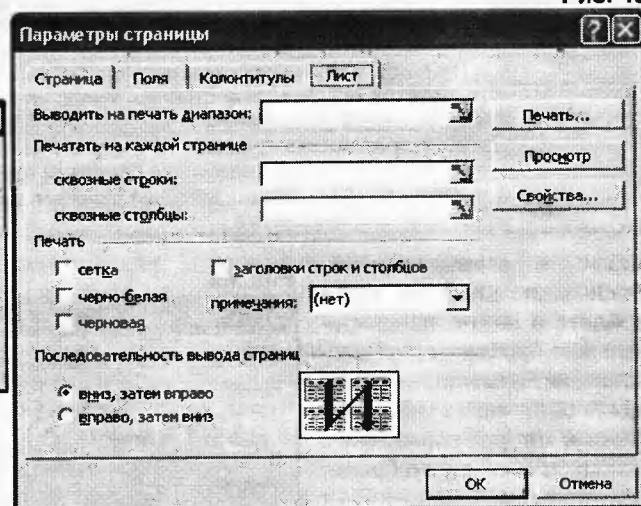
Итак, если все вкладки пройдены, завершите работу с диалогом **Параметры страницы**, щелкнув по кнопке **OK**, и снова оцените полученные результаты в режиме предварительного просмотра. Запомните, что те параметры страницы, которые вы сейчас установили, справедливы только для листа Заработная плата. Если вы в дальнейшем собираетесь печатать не только текущий рабочий лист, а всю рабочую книгу, то позаботьтесь о том, чтобы для всех листов были предварительно выбраны подходящие параметры страницы.

Если же вы устанавливаете параметры страницы для листа диаграммы (например, листа Нагрузка), то некоторые параметры, доступные для листа с электронной таблицей, здесь будут недоступны. К примеру, на вкладке **Страница** окна диалога **Параметры страницы** нельзя выбрать масштаб печати, на вкладке **Поля** нельзя задать вертикальное или горизонтальное центрирование, а вкладка **Лист** вовсе отсутствует и заменена вкладкой **Диаграмма**. Все остальные параметры задаются идентично.

Табл. 18. Инструменты окон диалога Верхний колонтитул и Нижний колонтитул

Вид кнопки	Действие
	Изменение параметров шрифта для колонтитула
	Вставка в колонтитул номеров страниц
	Вставка в колонтитул полного числа страниц
	Вставка в колонтитул текущей даты
	Вставка в колонтитул текущего времени
	Вставка в колонтитул имени текущего файла
	Вставка в колонтитул имени текущего рабочего листа

Рис. 49



И еще один момент. Если на рабочем листе, для которого вы устанавливаете параметры страницы, содержится и таблица, и диаграмма, то обратите внимание на то, что выделено. Если диаграмма, то те параметры страницы, которые вы определите, будут справедливы только для диаграммы, и в дальнейшем будет печататься только диаграмма. Если же курсор предварительно установить на любую ячейку рабочего листа, то параметры будут справедливы для всего рабочего листа.

Печать

А теперь можно переходить к самому процессу печати. Процедура распечатки зависит от того, что вы собираетесь печатать — всю рабочую книгу, текущий рабочий лист или избранный диапазон ячеек. Часть рабочего листа, которая должна быть выведена на печать, называется областью печати. По умолчанию область печати совпадает с заполненной частью рабочего листа. Если требуется распечатать не весь рабочий лист, а некоторые его фрагменты, то область печати задается вручную.

Наиболее простым представляется случай распечатки текущего рабочего листа. В данном случае достаточно щелкнуть по кнопке  на панели инструментов **Стандартная** или выполнить команду **Печать** меню **Файл** и в окне диалога **Печать** установить переключатели **все** в группе **Печатать** и **выделенные листы** в группе **Вывести на печать** (рис.50). Попутно можно указать число копий и нужно ли распечатанные страницы разбирать по копиям (флажок **Разобрать по копиям**). Аналогично производится распечатка нескольких рабочих листов, только требуется предварительно выделить их ярлычки в рабочей книге.

Для печати всей рабочей книги можно выделить все ее рабочие листы и действовать по предыдущему сценарию, либо выполнить команду **Печать** меню **Файл** и установить переключатель **всю книгу** в группе **Вывести на печать** (рис.50). В данном случае будут печататься только непустые рабочие листы.

Иногда требуется распечатать не весь

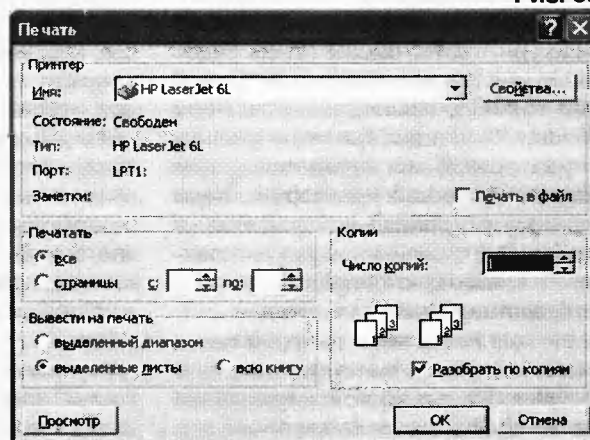
рабочий лист, а только некоторые его фрагменты. Один из способов мы рассматривали выше, когда обсуждали процедуру установки параметров страницы. Но можно поступить проще. Например, выделить нужный фрагмент, выполнить команду **Печать** меню **Файл** и установить переключатель **выделенный диапазон** в группе **Вывести на печать** (рис.50). Распечатается выделенный фрагмент, причем с теми параметрами страницы (полями, колонтитулами, ориентацией и т.д.), которые вы установили для всего рабочего листа.

Если же электронная таблица создается для многократного использования, и часто требуется печатать один и тот же фрагмент данной таблицы, то удобнее фиксировать область печати. Для этого, выделив нужный фрагмент, вы-

полните команду **Область печати/Задать** меню **Файл**. При этом область печати устанавливается жестко, просматривается в режиме предварительного просмотра, и, при необходимости распечатки другой области, эту необходимо отменить, используя команду **Область печати/Убрать** меню **Файл**.

(Окончание следует)

Рис. 50



МОДДИНГ. Наука, искусство или изыски XXI века?

(Окончание. Начало в №10/2005)

ST_HATABICH.

E-mail: hatabich@tut.by

А теперь перейдем непосредственно к моддингу.

Что должен иметь настоящий моддер?

Прежде всего это здравый ум, горячее сердце, инструмент, и самое главное — "прямые" руки.

К чему может привести Моддинг?

Прежде всего — к потере гарантии на железо, если руки не "кривые" — то к созданию своего рода шедевра железного искусства. А любое искусство требует материальных вложений. В случае "кривых" рук все действия приведут к уродованию железа (автор статьи и редакция не несет ответственности за "кривые" руки и неправильно понятые инструкции). Не исключены затраты времени и сил.

Если вы не испугались и подходите под описание моддера, то вперед!!!

Чаще всего моддингу подвергается корпус. Но не стоит покупать новенький корпус, т.к. не всегда может получится хороший результат. Лучше купить баксов за десять какой-нибудь антиквариат и извращаться над ним. Для моддинга больше всего подходят full-tower'ы, т.к. там больше простран-

ства для разгула вашего воображения.

Шаг 1

Прорезка отверстий для дополнительных вентиляторов. Обычно устанавливают вентиляторы диаметром 80, 92 или 120 мм. Причем они устанавливаются таким образом, чтобы объем вдуваемого в корпус воздуха примерно равнялся объему высасываемого. Если воздуха вдувается больше, это не страшно, хотя и нежелательно. А вот если больше будет вытягиваться, или вообще все вентиляторы стоят на выдув, то очень быстро в CD-ROM и флопах будет огромное количество пыли. Традиционная схема размещения кулеров — blowhole (на выдув). Для этого отверстия прорезают на задней стенке и в верхней плоскости корпуса. При использовании схемы suckhole (на вдув) отверстия проделывают в передней панели корпуса. Боковые стенки используются для обеих схем, хотя на них чаще ставят вентиляторы на вдув для подвода холодного воздуха к кулеру процессора и AGP/PCI-картам.

Самый "извращенный" способ установки кулера — в дно корпуса. Внизу, конечно, воздух самый холодный, но и пыли там немерено, так что в этом случае обязательно должен быть установлен пылевой фильтр.



Шаг 2

Прорезка окна, для того чтобы любой человек мог полюбоваться требухой вашего железного монстра. Контуры окна прорезаются дремелем (специальным девайсом) или электролобзиком. Возможны и другие варианты. Есть извращенцы, которые резали свои кейсы ножницами по металлу.

Я не удивлюсь, если кто-нибудь использует бензопилу или молоток и зубило :)). А вот с самым окном будет посложнее. Можно использовать готовый вариант, так называемые Plexi Window Kit. Существуют их варианты с гравировкой или отверстиями для кулеров. Купив такой наборчик, надо только дырку правильной формы пропилить. Все крепления идут в комплекте. Т.е. с помощью kit-набора не трудно сделать красивое окошечко, но вопрос — где взять такой набор. У нас я их еще не видел, ни на фирмах, ни в онлайне. Проще обойтись подручными материалами. Можно купить в магазине "Сделай сам" плексиглас и использовать его. Не советую использовать для этих целей стекло :)). Окна можно резать не только в боковой стенке корпуса, но и в других местах — сверху, сзади и т.д. Форма окна также может быть различной. Т.о. берем смело в руки лобзики, и вперед :)).



Шаг 3

Лос-Анджелес внутри? Это не мечта — это суровая реальность моддерского мира. А чем засветить? Ну, тут вариантов множество! Конечно, не стоит вешать внутрь корпуса обычную лампочку накаливания — есть другие, более совершенные методы светового декора. Можно покрасить определенные элементы и девайсы флуоресцентным маркером, краской или лаком. Туда же поставить ультрафиолетовую лампу. Зрелище просто супер! Можно с помощью светодиодных платок подсветить молекулы. Светят очень ярко. Можно использовать неоновые



шнуры. Они очень гибкие, и поэтому ими можно подсвечивать буквально все. Красиво смотрится клавиатура с таким шнурком внутри. Для подсветки кулеров можно использовать решетки с электролюминесцентной пленкой. Формы таких решеток могут быть разными, от вычурных загогулин до силуэтов обнаженных девушек. Ну и, наверное, самый простой способ — это всякого рода светодиоды. Купить их можно в любом магазине радиотоваров.

Шаг 4

Улучшаем аэродинамику "внутренностей". Основное здесь — мод шлейфов. Можно самому скрутить или разрезать шлейфы. Но лучше купить готовые, уже скрученные, экранированные и покрытые специальной краской IDE-кабели. Смотрятся они очень стильно.

Шаг 5

Покраска корпуса. Можно покрасить самому. Лучше красить в несколько слоев, предварительно обезжирив поверхность. Конечно, наилучшим вариантом будет удаление старой краски и нанесение грунтовки. Есть вариант покрасить case в автомастерской — это достаточно дешевый и простой способ. Ну а если вы любитель высокого искусства, то можно обратиться к знакомому художнику или граффитчику. Немного воображения, и из простого серенького ящика получаются настоящие шедевры.

Шаг 6

Для любителей оверклокинга предлагается установка датчиков температуры. Довольно полезная штука при максимальной нагрузке системы. Навороченные модели могут содержать более 9 датчиков, вольтметр и еще кучу прочих приамбасов.

Для контроля скорости вращения кулеров используют 4-канальный реобас, устанавливаемый в 3,5-дюймовый отсек корпуса компьютера. Он имеет несколько подсвеченных ручек для плавного регулирования напряжения, подающегося на подключенные к нему вентиляторы, в пределах от 7 до 12 В. Соответственно, каждая ручка регулирует напряжение подключенного к ней вентилятора отдельно и независимо от остальных. Подсветка заставит любой корпус, украшенный этим реобасом, выглядеть хардкорно, при этом реобас будет не только круто смотреться, но и позволит уменьшить шум, издаваемый вентиляторами, за счет уменьшения скорости их вращения в те периоды, когда от них не требуется работа на максимуме.

Шаг 7

Шумоизоляция. Кроме fanbus'a, существует еще один вариант уменьшения шума — оклейка корпуса изнутри шумоизолирующим материалом. В продаже имеются специальные шумоизолирующие маты японской фирмы AKASA, наши умельцы используют пробку или автомобильную шумоизоляцию. Где купить пробку, я не знаю, а вот про автомобильную вы сможете узнать в компаниях, занимающихся автозвуком. Оклеивать надо максимальную поверхность корпуса (изнутри, естественно), в идеале все, что не занято девайсами и кулерами. На сайте Moddin.net есть подборка статей, посвященных всем способам уменьшения шума компа.

Шаг 8

Для меня приятной мелочью являются удобные болты, а если они еще и причудливой формы, то это вообще рульно. Пример — болтик с черепом.



Придумать можно все что угодно. Штуки ради в системный блок можно встроить и ведро для мусора, и сканер, и открывашку для пива. Можно обклеить его шерстью. Я думаю, что у каждого моддера есть своя моддерская мечта, которую он жаждет воплотить в реальность.

Так что дерзайте и создавайте свои шедевры хардковского искусства!!!





Acronis TrueImage 8.0

(Окончание. Начало в №10/2005)

Создание образа диска

Основной функцией True Image, естественно, является создание образа диска, поэтому на примере именно этой операции мы более подробно познакомимся с общими принципами функционирования программы. Для того чтобы создать образ раздела или всего жесткого диска, дважды щелкнем мышью на значке "Создать образ" в главном окне программы, и нас встретит приветственное окно Мастера создания образа (рис.2).

Рис. 2

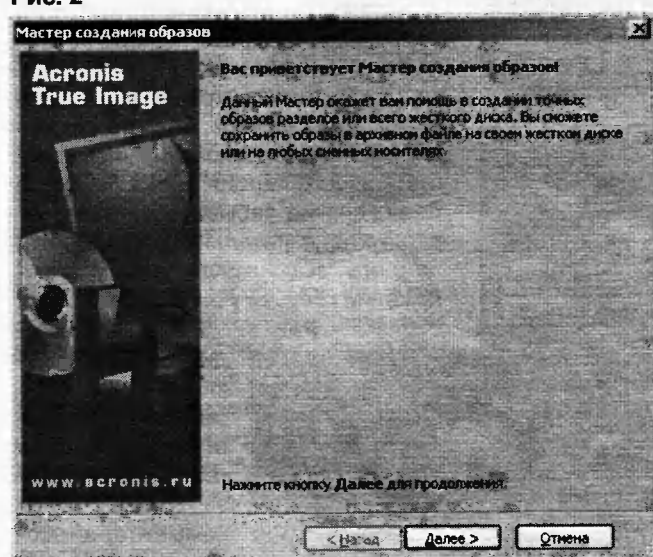
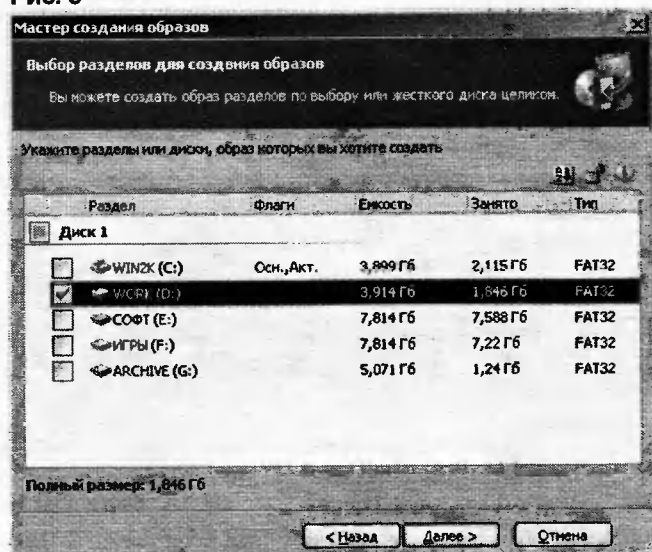


Рис. 3



Далее в соответствующем окне (рис.3) необходимо выбрать один или несколько разделов (если надо — можно выбрать и все разделы диска целиком, выделив галочкой метку диска), образ которых нам нужно создать.

Вполне логичным будет следующий шаг, на котором мы определим имя нового образа, а также место, где он будет располагаться (рис.4). При этом программа предоставляет нам максимальную свободу выбора — создаваемый образ будет находиться либо в Зоне безопасности Acronis (в этом случае образ будет создан в специально отведенном и ни для кого больше недоступном месте жесткого диска), либо на любом из имеющихся разделов жесткого диска (естественно, за исключением тех, образ

ших свою актуальность стареньких Iomega ZIP и JAZZ, по-настоящему массовых CD-R/RW, идущих им на смену DVD-R/RW, и вплоть до сугубо нишевых магнитооптических дисков. С точки зрения максимальной безопасности хранения записанного образа, использование сменных носителей будет наиболее целесообразным выбором.

Если в указанном месте файл образа уже существует, то программа предложит вам или изменить имя файла, или перезаписать его, то есть стереть существующий файл и создать на его месте новый, потому как дописать образы в уже существующий файл невозможно. Но если нельзя, но очень хочет

Рис. 4

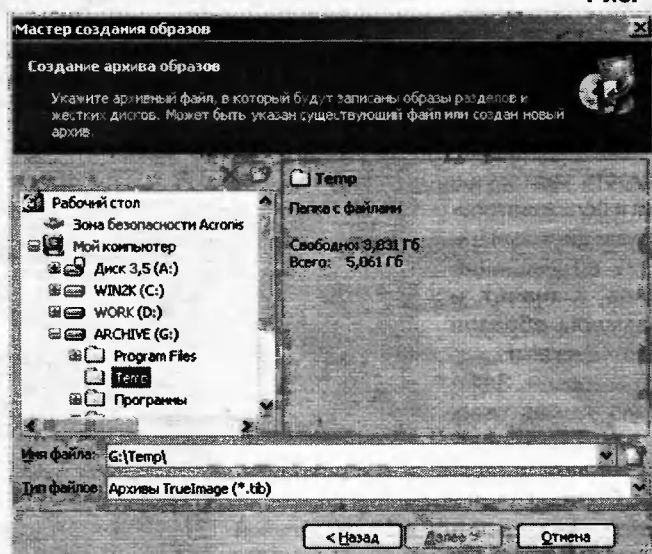
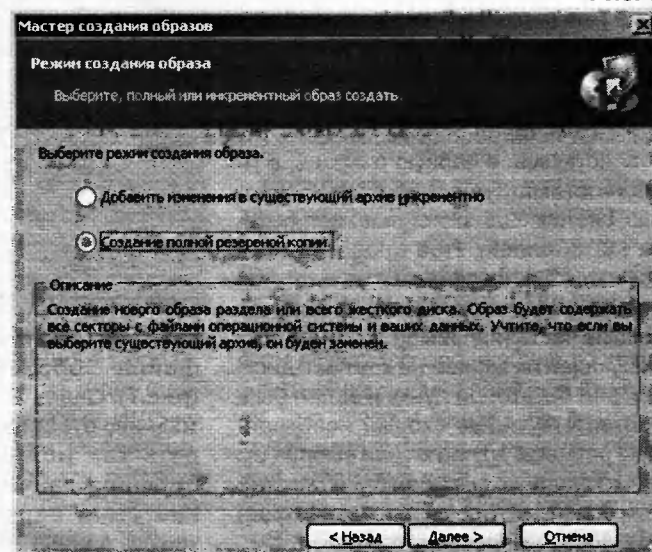


Рис. 5



которых мы будем сохранять, так как при восстановлении файл образа будет просто уничтожен), либо на сетевом диске. Кроме того, будут доступны самые разнообразные типы накопителей со сменными носителями — от теряю-

ся, тогда, как всем известно, можно. Поэтому на следующем шаге программа предложит нам создать инкрементный образ диска (рис.5), если, конечно, полное копирование этого раздела ранее уже производилось.

При этом будет осуществляться копирование только тех секторов диска, которые изменились после создания полного образа или после создания предыдущего инкрементного образа. Таким образом, создание инкрементного образа позволит существенно сократить время резервного копирования данных. При этом не стоит забывать, что при создании инкрементного образа True Image осуществляет **посекторное** сравнение резервируемого раздела и существующего образа, поэтому после каждой дефрагментации раздела старый образ потеряет свою актуальность, и лучше будет сделать новый полный образ.

Несмотря на то что объем современных винчестеров существенно возрос, их заполняемость системными и пользовательскими данными ничуть не уменьшилась, а значит, и размеры образов таких дисков изрядно возросли. Зачастую размер таких архивов исчисляется сотнями мегабайт, а то и гигабайтами. Поэтому этап разбиения образа на части уже не кажется малоактуальной операцией (рис.6).

На выбор пользователю предлагается два варианта такого разбиения. Если выбрать пункт "Автоматически", то программа в ходе создания образа сама примет решение о необходимости делить файл образа на части, руководствуясь, в первую очередь, возможностями существующих файловых систем. Однако сильно полагаться на интеллект True Image лично я все-таки бы поостерегся, и в случае необходимости разбиения файла образа на части (например, в случае последующей их записи на компакт-диск) выбрал бы второй пункт и задал бы в соответствующем окошке необходимые размеры вручную (или выбрал бы подходящий из выпадающего списка). Тем более, что процесс создания файлов образа непосредственно на дисках CD-R/RW обычно занимает неоправданно много времени. Поэтому гораздо удобнее будет сначала создать образ на жестком диске и лишь после этого переписать его на компакт-диск. В связи с этим хочу еще раз

отметить, что в своей восьмой версии True Image, создавая образ системного раздела, теперь уже не включает в него содержимое файлов подкачки (win386.swp для Windows 9x и pagefile.sys для Windows NT/2000/XP) и гибернации (hiberfil.sys), хотя сами файлы, видимо, для сохранения идентичности структуры разделов в образе присутствуют, что существенно уменьшает его итоговый размер и увеличивает скорость его создания.

Следующим шагом будет задание

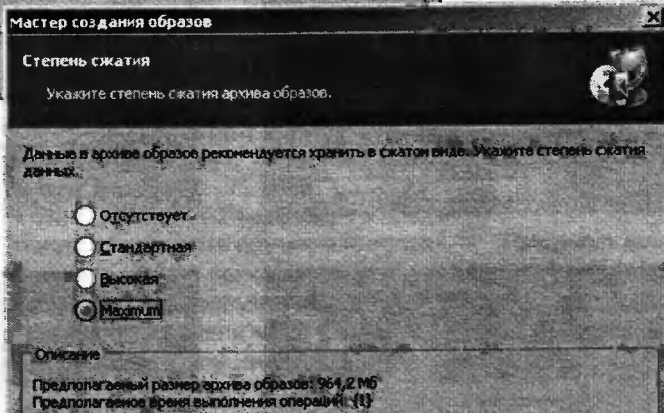
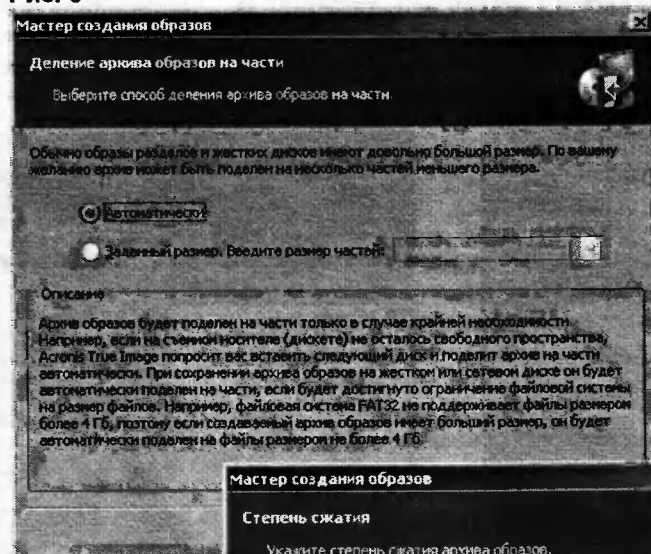
нечно, уменьшится, и весьма заметно, но при этом увеличивается и время, которое затратит True Image на его создание. И наоборот — создание образа вообще без сжатия позволит при работе с образом в дальнейшем практически без задержки получать доступ к файлам, хранящимся в образе, но размер образа не будет отличаться от размера резервируемого раздела. Так что пользователю самому придется выбирать разумный компромисс между этими крайностями, ориентируясь на предполагаемые величины размера образа и времени его создания, приведенные в секции "Описание".

В очередном окне нам будет предложено защитить создаваемый образ паролем (рис.8). Лично я пароль на сохраняемые образы никогда не ставил и вам не советую — расчет большей безопасности хранения образа я сильно сомневаюсь, зато не раз наблюдал просто феноменальное свойство любого пароля очень быстро и безвозвратно забываться, ставя чересчур подозрительного пользователя с его "высокозащищенным" в положение старухи у разбитого корыта.

Рис. 7

И самая последняя опция, требующая вашего активного вмешательства — описание создаваемого образа (рис.9). Этот комментарий может включать любую информацию о компьютере и его пользователе, жестком диске, данных раздела, времени

Рис. 6



степени сжатия файла образа (рис.7). Очень заманчиво будет задать как можно большее значение этого параметра, однако не следует забывать, что эта "палка" — о двух концах.

При максимальном уровне сжатия размер файла, ко-

Рис. 8

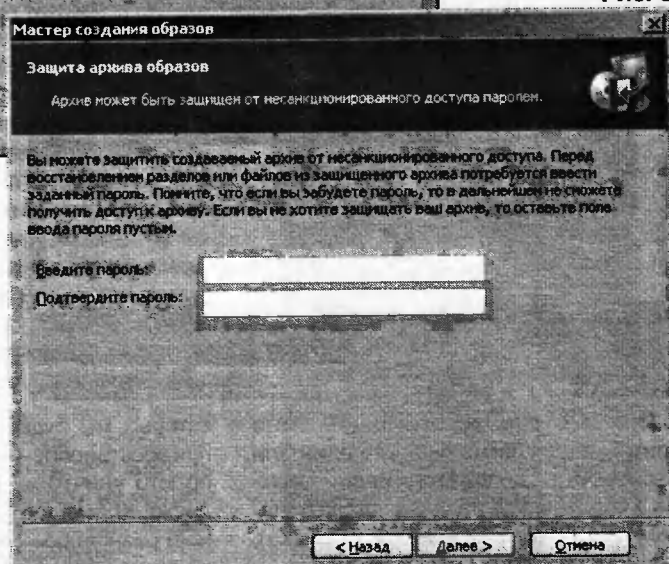
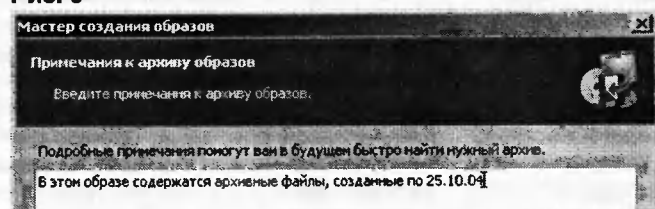


Рис. 9



создания образа, любых особенностях и обстоятельствах его создания. Как раз этой функцией я бы пренебрегать не советовал, особенно в случае наличия нескольких образов, усугубляемом не слишком частым обращением к ним в течение достаточно долгого времени. Человеческая память, знаете ли, штука очень ненадежная...

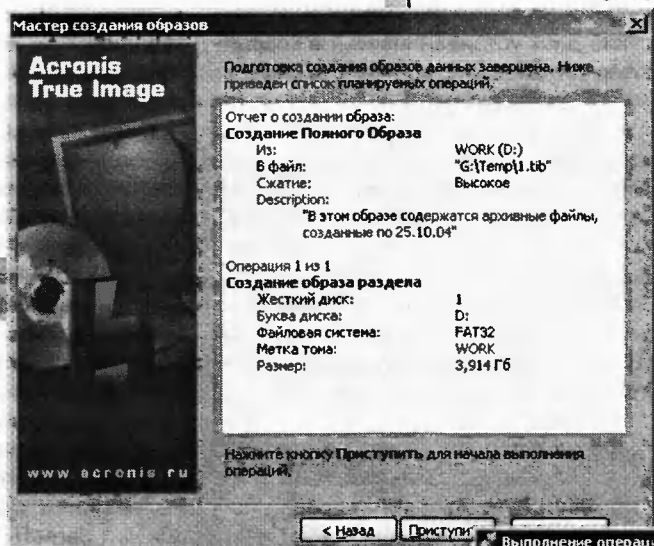
На следующем экране перед нашими глазами предстает сводный перечень всех настроек, созданных в процессе работы Мастера, а также краткое описание этих операций (рис.10). Благодаря этой информации мы можем в любой раз удостовериться в том, что все действия были произведены верно. Если же будет обнаружена какая-либо ошибка, то ее еще можно исправить, вернувшись к нужному экрану Мастера, нажимая кнопку "Назад".

И лишь после того как вы окончательно убедитесь в правильности всех настроек, можно нажать кнопку "Продолжить" на последнем экране Мастера, и True Image начнет свой труд, создавая новый образ. Весь процесс выполнения операции будет наглядно отражен в открывшемся окне (рис.11).

При необходимости, создание образа можно прервать, нажав кнопку "Отмена". Также, нажав кнопку "Скрыть", можно перевести эту операцию, занимающую подчас достаточно долгое время, в фоновый режим. При этом создание образа будет продолжаться, но пользователь получает возможность приступить к выполнению какой-нибудь другой задачи. По завершении создания образа в фоновом режиме программа автоматически завершит свою работу.

Детально ознакомившись с основными принципами работы программы True Image на примере операции создания образа дис-

Рис. 10



ка, вряд ли стоит особо подробно расписывать работу других Мастеров — практически все они похожи как близнецы (да и может ли быть иначе?). Поэтому в заключительной части обзора мы лишь вкратце остановимся на остальных модулях программы, попутно отметив наиболее характерные их особенности.

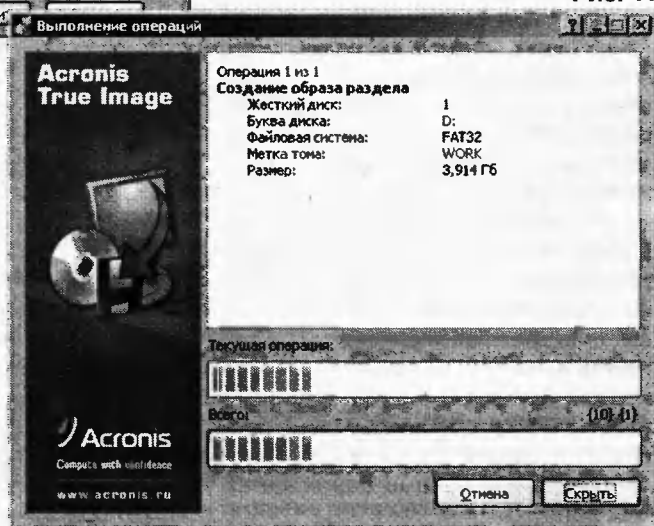
Проверка образа. Итак, образ раздела успешно создан, но как убедиться, насколько качественно True Image выполнил свою работу? Иногда, при резервировании наиболее важных данных, не будет лишним убедиться в целостности полученного образа, иначе восстановление из этого файла будет просто невозможно. Именно для этого и предназначен "Мастер проверки образов", запустить который можно при помощи ссылки "Проверка образа" в разделе "Сервис" главного окна программы.

Восстановление раздела из образа. В штатных условиях рекомендуется восстанавливать образ раздела непосредственно из Windows, и только в аварийных ситуациях следует воспользоваться комплектом спасательных

дискет (или компакт-диск). В последнем случае владельцам компьютеров, располагающих только одним приводом компакт-дисков, не стоит особо волноваться — после загрузки программа целиком размещается в оперативной памяти компьютера и к загрузочному диску уже больше не обращается. Поэтому вы можете использовать освободившийся накопитель по своему усмотрению.

Еще одним важным моментом является то, что True Image должен получить монопольный доступ к восстанавливаемому разделу, то есть никакая другая программа не должна во время восстановления работать с этим разделом. И вообще, восстанавливая дисковый раздел из образа, следует быть крайне внимательным, так как при этом осуществляется посекторная запись на него информации из образа, из чего следует практически стопроцентная вероятность уничтожения всех имеющихся в этом разделе данных.

Рис. 11



Подключение и отключение образа. Однако полное восстановление образа раздела требуется далеко не всегда — гораздо чаще бывает нужен лишь один конкретный файл (реже — несколько), содержащийся в образе. В этом случае гораздо удобнее воспользоваться достаточно полезной функцией True Image и подключить разделы, хранящиеся в файле образа, как обычный логический диск. При этом в Проводнике Windows и Моём компьютере появится виртуальный диск, обозначенный собственной буквой, при обращении к которому будет открыт доступ к любому из файлов образа, который легко можно будет скопировать, но и только — подключенный виртуальный диск будет доступен исключительно для чтения. В новой версии программы по-

явилось очень полезное (и, честно говоря, давно ожидаемое) новшество — автоматическая проверка целостности таких файлов.

По завершении копирования файлов из подключенного диска, его следует отключить, воспользовавшись соответствующим Мастером в главном окне программы. Но даже если вы забудете сделать это, ничего страшного не произойдет — после перезагрузки компьютера виртуальный диск исчезнет сам.

Клонирование диска. В процессе клонирования диска вся информация с одного диска переписывается на другой посекторно, что обеспечивает идентичность созданной копии и оригинала. Новый рабочий диск, полученный в результате клонирования, имеет полностью работоспособные операционную систему и предустановленное программное обеспечение. Таким образом, клонирование диска избавляет пользователя от рутинной и достаточно длительной процедуры установки и настройки системы и пользовательских приложений на новом жестком диске.

Клонирование диска можно осуществить в автоматическом или в ручном режиме. Понятно, что в автоматическом режиме программа может только скопировать структуру исходного диска на новый — другого результата ожидать от нее было бы странно. Чтобы результат был иным, программа должна задать ряд дополнительных вопросов о параметрах клонирования и получить ваши ответы на них, что и происходит при выборе ручного режима работы. В этом случае можно выбрать способ переноса разделов и данных:

- разделы и данные переносятся "как есть";
- место на новом диске пропорционально распределяется между переносимыми разделами старого диска;
- место на новом диске распределяется между разделами вручную.

Кроме того, можно также определить судьбу старого диска:

- сохранить структуру разделов и все имеющиеся данные на старом диске без изменений;
- уничтожить всю информацию на старом жестком диске. При этом он будет полностью очищен и перезаписан нулями, что вполне обезопасит его от любых попыток получить доступ к хранившимся на нем данным;
- создать на старом жестком диске новую структуру разделов (при этом все данные на нем будут утеряны).

По завершении процесса клониро-

вания, как в ручном, так и в автоматическом режиме, необходимо перезагрузить компьютер.

Установка нового диска

Помимо замены старого жесткого диска на новый, чаще всего большего объема, в практике сплошь и рядом встречаются случаи, когда новый жесткий диск устанавливается в компьютер не вместо старого, а вместе с ним. В этом случае старый винчестер (с установленной на него системой) является системным, а гораздо более емкому новичку отводится роль файлового архива. Но, сами понимаете, любой "свежекупленный" жесткий диск сразу работать не может, сначала его надо разметить, отформатировать — и лишь только после этого он готов к использованию. Входящие в Windows специальные утилиты для такой работы годятся мало — они безнадежно устарели. Однако в составе True Image имеется инструмент, значительно облегчающий процесс разметки и форматирования нового жесткого диска. С помощью Мастера установки новых дисков можно выбрать тип файловой системы (FAT16, FAT32, NTFS, Linux Ext2, Linux Ext3, Linux Swap, Linux ReiserFS), типы создаваемых разделов, их имена, размеры и местоположение на диске.

Планировщик

Автоматизировать процесс создания образов призван Планировщик. Основная задача этого инструмента — запуск определенных операций по расписанию. Один раз создав запланированное задание (или даже несколько независимых), пользователь может больше не беспокоиться о, например, создании резервных образов рабочих дисков. Особенно полезно это будет при создании инкрементных образов, как это и делается при резервном копировании. Впрочем, планировщик окажется полезным и при решении более привычных задач, тем более, что он поддерживает достаточно гибкую настройку параметров нового задания, которая включает-

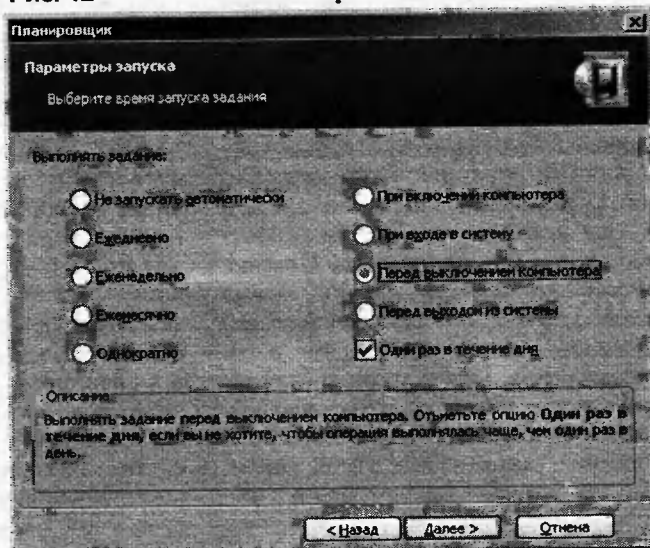
ся в определении резервируемого раздела, имени и месторасположения его будущего образа (по аналогии с ранее описанными мастерами) и, главное, в настройке периодичности запуска планируемой операции, производимой в соответствующем окне мастера "Назначить задание" (рис.12).

Здесь мы выбираем время запуска задания:

- **Ежедневно** — задание выполняется ежедневно в указанное на следующем шаге мастера время;
- **Еженедельно** — задание выполняется еженедельно или один раз в несколько недель в указанные время и дни недели;
- **Ежемесячно** — задание выполняется ежемесячно в указанное время;
- **Однократно** — задание выполняется один раз в указанное время;
- **При включении компьютера** — задание выполняется при каждой загрузке операционной системы;
- **При входе в систему** — задание выполняется каждый раз, когда происходит регистрация нового пользователя в системе;
- **Перед выключением компьютера** — задание выполняется перед каждым выключением компьютера (или его перезагрузкой);
- **Перед выходом из системы** — задание выполняется каждый раз, когда пользователь выходит из системы;
- **Не запускать автоматически** — задание не выполняется. Вы можете выбрать этот пункт, когда хотите приостановить на некоторое время работу созданного ранее задания, но не хотите удалять его полностью;

Каждое новое задание индицируется специальным значком в основном окне True Image.

Рис. 12



Создание загрузочного диска

Естественной функцией программ, осуществляющих резервное копирование и восстановление информации, является возможность их запуска и полноценной работы с аварийных носителей — дискет или компакт-дисков. Не менее естественно и включение в их состав специального инструментария для создания таких дисков.

В True Image имеется "Мастер создания загрузочного диска", запустить который можно, нажав на ссылку "Создание загрузочного диска" в разделе "Сервис" главного окна программы. При создании комплекта аварийных загрузочных дискет вам будет предложен выбор между полной версией загрузчика (способного работать со всеми без исключения устройствами, поддерживаемыми программой), и его облегченной версией, отличающейся отсутствием в ней драйверов устройств USB, PC card (PCMCIA) и SCSI (рис.13). В отличие от прошлых версий, True Image 8.0 требует для этих целей 6 дискет для полного варианта и 3 — для облегченного, что, согласитесь, слишком много (Norton Ghost, к примеру, до сих пор умудряется обходиться только одной).

Поэтому наиболее целесообразным вариантом будет выбрать создание загрузочного компакт-диска, тем более, что записывающие приводы CD — R/RW давно уже стали фактическим стандартом в компьютерной индустрии.

После создания аварийного диска (или дисков) с них можно будет загрузиться и, работая в привычном графическом интерфейсе True Image, выполнить любые стандартные операции по восстановлению дисков из образа или по их созданию.

Просмотр журнала

Все события, происходящие во время работы True Image, протоколируются в журнале работы. В любой момент, заглянув в него, можно уточнить, какую задачу в данный момент времени выполняет программа, и узнать, насколько успешно были выполнены предыдущие задания. Пример журнала показан на рис.14.

Рис. 13

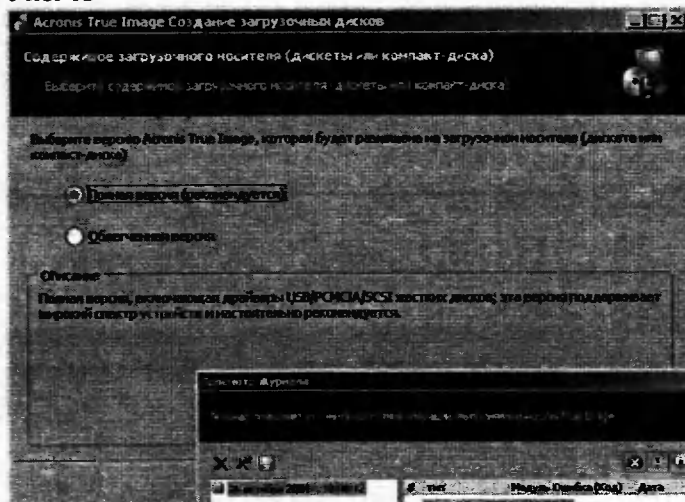
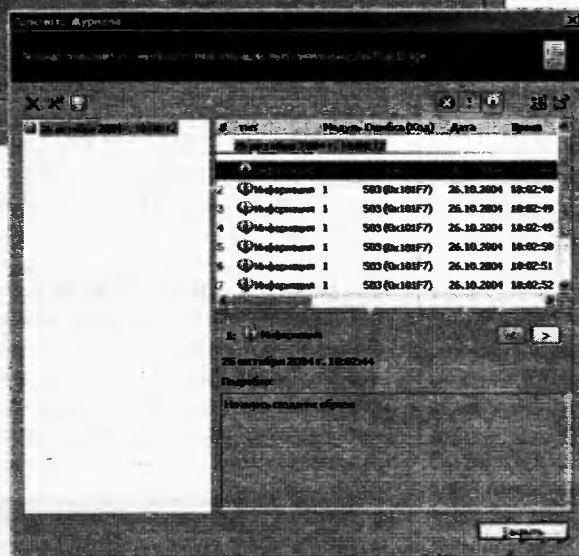


Рис. 14



Подведем итоги

Если попытаться дать программе Acronis True Image 8.0 однозначную оценку, то ее сегодня можно смело признать оптимальным выбором для подавляющего большинства индивидуальных пользователей. Она предостав-

ляет в распоряжение пользователя массу простых, и вместе с тем, чрезвычайно полезных инструментов, которые позволяют создавать резервные копии данных и управлять ими, а в случае аварии без особых усилий восстановить данные из имеющейся резервной копии. Хорошо продуманный интерфейс и подробная документация на русском языке (правда, на момент написания

статьи в дистрибутив программы входила документация для предыдущей, 7-й версии, а руководство пользователя для 8-й было представлено только для ее англоязычного варианта) и достаточно простые мастера, обеспечивающие выполнение всех функций, позволяют с легкостью пользоваться True Image малоподготовленным пользователям.

Но и достаточно "продвинутые" компьютерные асы найдут в новой программе немало для себя полезного. К примеру, достаточно эффективная автоматизация основных операций, выполняемых программой, наряду с созданием инкрементных образов и возможностью подключения образов как логических дисков, делает True Image полноценным инструментом ре-

зервного копирования, а возможность загрузки ее консоли (основанной, кстати, на Линуксе) еще до загрузки ОС позволяет восстанавливать операционную систему даже при самых сложных авариях, когда загрузка операционной системы невозможна.

С.РЮМИК,
г.Чернигов.

Знание бывает двух видов. Мы сами знаем предмет или же знаем, где найти о нем сведения.
Б. Франклин

Судя по афоризму известного просветителя и государственного деятеля, Франклин, сам того не ведая, 250 лет назад предсказал появление Интернета. Как видно, простые и понятные философские умозаключения естественным образом определяют дальнейший ход истории.

Знать, где в Интернете находится информация, помогают браузеры. От радно, что наряду со скандально известным Microsoft Internet Explorer (IE), все большую популярность набирает open-source браузер Mozilla Firefox 1.0.6 (<http://www.mozilla.org/products/firefox/>, 5 Мб). Быстрота навигации, возможность под-

ключения плагинов, макросов, надстроек (<http://addons.update.mozilla.org/>), а главное, freeware-статус — вот достоинства "огненной лисицы". Кстати, на компьютере могут мирно сосуществовать и Firefox, и IE.

Дальнейшим развитием Firefox является браузер Deer Park Alpha 1, или

Интернет-калейдоскоп, freeware #24

Firefox 1.1 (<http://www.mozilla.org/projects/deerpark/whatsnew.html>). Но не следует торопиться его устанавливать, это альфа-версия для тестеров и разработчиков дополнений, не свободная от ошибок. Практический вывод — нельзя удалять с диска старую версию программы, пока не полностью опробована новая.

Стикеры на экране монитора

Если их не имеешь, то и вопросов никому не задаешь. Однако, попробовав хоть раз "наклеить" виртуальную липучку-стикер на рабочий стол Windows, удивляешься: "И как я раньше без них обходился?"

Программа ATnotes (автор — Thomas Ascher, <http://atnotes.free.fr/dl/atnsetup.exe>, 708 Кб) позволяет создавать наклейки разного размера, цвета, содержания и даже разной даты активации. Основное их назначение — это напоминание

Рис. 1



о предстоящих событиях, встречах, а также передача текущей информации от одного пользователя компьютера другому (с первой смены на вторую).

По умолчанию программа находится в трее и активизируется нажатием правой кнопки мыши (рис.1). Редактируются стикеры прямым вводом текста.

Рисунки по координатам и координаты по рисункам

Две прямо противоположные задачи решаются двумя разными программами одного автора О.Записных: РИСУЕМ ПО КООРДИНАТАМ (<http://www.mtu-net.ru/avangard/kdrw.zip>, 207 Кб) и АНАЛИТИК (<http://www.mtu-net.ru/avangard/analitik.zip>, 452 Кб).

Первая из них имеет на входе массивы чисел, соответствующие координатам точек, и проводит между ними отрезки прямых линий. Задача школьная, как в газете "Математика", но рисунки могут получаться весьма эстетичными (рис.2).

Вторая программа имеет на входе отсканированный график в виде bmp-файла. Далее производится его экстраполяция, вычисление координат точек и передача их массивом чисел

Рис. 2

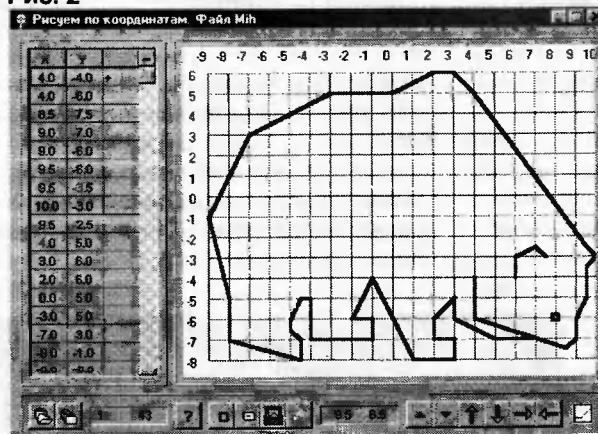


Рис. 3



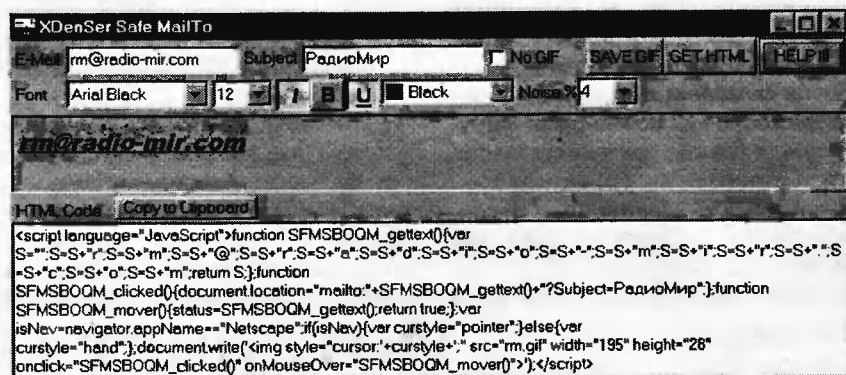
прямо в Microsoft Excel. Для повышения надежности считывания можно установить процентный порог между черными и белыми пикселями. На рис.3 для примера изображена оцифровка реальной электрокардиограммы.

Как изобрести велосипед?

Патентная система появилась более 3 веков назад, и в ее основе лежала привилегия автора на свое изобретение. Но не все технические идеи признаются изобретениями.

Попов и Маркони, Котельников и Найквист. Даже в журнале "Радиомир. Ваш компьютер" можно найти тому подтверждение. В частности, графические способы защиты e-mail от спамеров [1] нашли отражение в программе XDenSer Safe Mailto (автор — Denys S. Khanzhyev, <http://xdenser.chat.ru/safemailto.zip>, 296 Кб). Из нововведений — в gif-рисунки добавляются помехи в виде точек (рис.4), чтобы затруднить распознавание в программах типа FineReader. Кроме

Рис. 4



Например, в Англии отсеивается половина, а в Японии — треть заявок. Тем не менее, именно японцы в 50-х годах прошлого века удачно скупили законсервированные и считавшиеся бесперспективными патенты США и других промышленно развитых стран. В результате они сумели усовершенствовать технологию и поставить на поток первые полупро-

того, автоматически генерируется html-код ссылки на e-mail.

Ключевые слова для поиска: Firefox, Deer Park, липкие наклейки, Аналитик, Safe Mailto.

Литература

1. Lawyer. Защита от спама. — Радиомир. Ваш компьютер, 2005, №7, С.28.



Локальная сеть у вас дома

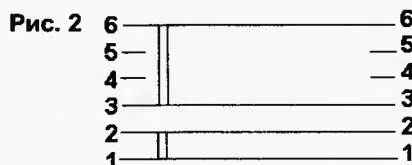
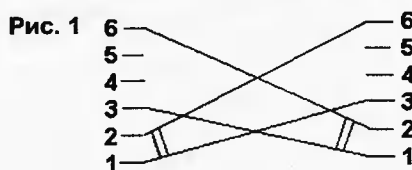
Сейчас уже многие имеют дома по два компьютера, один из них, обычно более мощный, на базе Celeron или Pentium 4, другой — менее мощный, на основе 486 процессора или одного из первых Pentium. Многие используют DOSовские программы, так как они их вполне устраивают, или используется оборудование, для которого нет подходящих драйверов в современных операционных системах или они работают некорректно. При желании нетрудно соединить эти два компьютера в полноценную локальную сеть и использовать все ее преимущества. Для этого потребуются две сетевые карты и кусок кабеля типа "витая пара".

В принципе, два компьютера можно соединить между собой без хаба с помощью так называемого "перекрестного" кабеля, другое название — "реверсивный" кабель, кросс-кабель. Если вы используете хаб, то компьютеры соединяются с ним отрезками "прямого" кабеля. В случае, если ваш более мощный компьютер имеет встроенную сетевую карту, то понадобится всего одна сетевая карта.

На рынке предлагается много сетевых карт — от самых дешевых, стоимостью около ста российских рублей, до весьма дорогих. Возьмем, например, RTL8139. Приглядевшись повнимательней, можно увидеть, что в разъеме RJ-45 на этой карте установлено всего 4 контакта вместо положенных 8, либо контакты имеются, но они не разведены на печатной плате. Эта карта предназначена для работы в полудуплексном режиме и будет нормально работать только в паре с хабом. Для нашего же случая она совершенно не подходит. Нам нужна карта, работающая в полном дуплексном режиме (*full duplex*). Обычно в технической характеристике карты указано — *support half and full duplex operation* или *twice the throughput with full duplex signaling*. Например, подойдут сетевые карты D-Link DFE-530TX или DFE-538TX. Сетевой чип, установленный в материнскую плату, обычно поддерживает полный дуплекс.

Теперь примемся за кабель. На рис. 1 и 2 приведены разводки проводов витых пар по контактам коннекторов RJ-45 для "прямого" и "перекрестного" кабелей. Двумя короткими параллельными штрихами указано то, что эти провода скручены, т.е. это одна из четырех витых пар, входящих в кабель.

Контакты 4, 5 одного коннектора соединяются витой парой с контактами 4, 5 другого коннектора (аналогично контактам 1, 2 для прямого кабеля). Контакты 7, 8 одного коннектора соединяются витой парой с контактами 7, 8 другого коннектора. Эти



соединения на рисунках не показаны.

Не пытайтесь опрессовывать провода в коннекторе с помощью подручных средств, вряд ли вам удастся обеспечить надежные контакты проводов в разъеме, и сеть будет часто отказывать. Лучше использовать специальный инструмент для обжима кабеля, но он довольно дорог и для одного раза его приобретать не имеет смысла. Можно одолжить его на время у знакомого компьютерщика. Если у вас такового нет, обратитесь в любую компьютерную фирму, и вам изготовят такой кабель за небольшую плату. Только обязательно проверьте, чтобы был изготовлен именно "перекрестный" кабель, т.к. вам могут продать заготовленный заранее кусок "прямого" кабеля.

После установки сетевой карты ее нужно настроить. Windows XP автоматически определяет и настраивает, например, сетевую карту DFE-538TX как карту RTL-8139. Такую настройку можно оставить без изменений, если вы будете работать через хаб. Для нашего случая (соединение двух компьютеров "перекрестным" кабелем) это не подходит, для обеспечения полного дуплекса нужно настроить сетевую карту именно как DFE-538TX. Для этого идем в **Панель управления**. Переключите отображение панели управления к классическому виду (как в Windows 98). Щелкаем по значку **Система**, выбираем вкладку **Оборудование**, жмем на кнопку **Диспетчер устройств**, в открывшемся окне щелкнем по крестик **Сетевые платы**, правой кнопкой мыши щелкнем по RTL-8139, и выберем пункт **Обновить драйвер**. Запускается **Мастер обновления оборудования** и спрашивает разрешения на подключение к узлу Windows Update. Отметьте **Нет, не в этот раз** и жмите кнопку **Далее**. Чтобы не повторяться, сразу замечу, что в окне после выбора одного из пунктов нужно нажимать кнопку **Далее** для перехода к следующему окну. В следующем окне выберите **Установка из указанного места**. В новом окне отметьте **Выполнить поиск наиболее подходящего драйвера** и галочкой

Б.ШИЛЬНИКОВ,
Читинская обл., п.Дарасун.
E-mail: gnzgo@mail.gin.su

отметьте **Включить следующее место поиска**. Кнопкой **Обзор** открывается обзор папок. Выберите нужную папку с драйвером на вашем компакт-диске или винчестере. Если имеется несколько папок с названиями операционных систем, выберите папку с названием вашей системы, например WinXP. Жмите **ОК**. В окне для списка сетевых карт появится строчка D-Link DFE-538TX PCI Adapter, пометьте его и нажимите **Далее**. Происходит копирование драйверов и системных файлов. Затем появляется сообщение о завершении установки сетевой карты D-Link DFE-538TX. Нажмите кнопку **Готово**. Для проверки снова идем в **Панель управления**. Щелкнем по крестик **Сетевые платы**, должна быть установлена сетевая плата D-Link DFE-538TX PCI Adapter.

В компьютере под управлением Windows 98 при загрузке появляется окно с сообщением, что найдено новое оборудование D-Link DFE-538TX PCI Adapter и предложение вставить диск с драйвером к этому устройству. Нажмем **ОК**. В появившемся окне укажем с помощью кнопки **Обзор** каталог с драйвером для Windows 98. В окне появляется название нужного файла и становится доступной кнопка **ОК**. Нажмем ее. Выполняется установка программного обеспечения для этой сетевой карты. После этого выводится вопрос: "Произвести перезагрузку сейчас?". Нажмем **Да**. После перезагрузки идем в **Панель управления** → **Система** → **Устройства**. Щелкнем **Сетевые платы**. Должна быть установлена плата D-Link DFE-538TX 10/100 Adapter.

В случае с встроенной сетевой картой драйвер находится на компакт-диске, прилагаемом к материнской плате. После автозапуска этого диска в меню обычно имеется пункт "Install LAN driver" или подобный. Выбрав этот пункт и пройдя последовательно этапы установки, вы в результате настроите сетевую карту. Аналогично, для проверки инсталляции зайдём в **Панель управления**, щелкнем значок **Сетевые карты** и посмотрим, установлен ли сетевой адаптер. Должна быть строчка наподобие такой: NVIDIA nForce MCP Networking Adapter.

После настройки сетевых плат можно перейти к настройке сети. Для Windows 98 это можно сделать следующим образом. После настройки сетевой карты у вас на Рабочем столе должен появиться значок **Сетевое окружение**. Щелкнув по нему правой кнопкой мыши, выберем в контекстном меню пункт **Свойства**. Затем выберем вкладку **Идентификация**. В поле **Имя компьютера** введем, например, Com1. В поле **Рабочая группа** — например, Home.



Имя рабочей группы для обоих компьютеров должно быть одинаковым, а имена компьютеров различаться, т.е. для другого компьютера можно ввести Com2 и Home. Лучше на английском, все-таки это "родной" язык для Windows. Перейдем на вкладку **Конфигурация**. Нажмем кнопку **Доступ к файлам и принтерам**. В появившемся окне отметим галочкой строчку **Файлы этого компьютера можно сделать общими**. Если к этому компьютеру непосредственно подключен принтер, отметим галочкой также и строчку **Принтеры этого компьютера можно сделать общими**. Затем нажмем **ОК**. На этой же вкладке выберем строчку **ТСР/П** и нажмем **Свойства**. Выберем вкладку **IP-адрес** в окне **Свойства ТСР/П**. Установим переключатель на строчку **Указать IP-адрес явным образом**. Зададим IP-адрес — 192.168.0.5, маску подсети — 255.255.255.0. Для второго компьютера соответственно — 192.168.0.10 и 255.255.255.0. Вместо цифр 5 и 10 можно указать любое число в диапазоне от 1 до 254, но IP-адреса компьютеров должны отличаться хотя бы на единицу. При выходе из окон нажимаем **ОК** для фиксации изменений. На предложение перезагрузиться ответим **Да**. После перезагрузки сеть должна заработать на обоих компьютерах. Щелкнем по **Сетевому окружению**, в окне должны быть значки обоих компьютеров. Щелкнем по значку другого компьютера, но в открывшемся окне вместо папок и файлов пусто — это потому, что мы не открыли доступ к файлам и папкам этого компьютера. Откроем на нем **Мой компьютер**, щелкнем правой кнопкой мыши по значку диска **C** и выберем пункт **Доступ**. Установим переключатель на **Общий ресурс** и тип доступа **Полный**. Нажмем **ОК**. У значка диска **C** появляется изображение руки. После этого можно открывать и просматривать файлы на обоих компьютерах.

Для печати документов на сетевом принтере его нужно установить. Идем в **Панель управления** → **Принтеры** → **Установка принтера**. Установим переключатель на **Сетевой принтер**. В другом окне с помощью кнопки **Обзор** выберем компьютер, к которому подключен принтер, и щелкнем по его значку. Появится значок принтера. Отметим его и нажмем **ОК**. По запросу программы укажем диск и папку с драйверами к этому принтеру. После завершения установки вы сможете посылать документы на печать и с другого компьютера.

Теперь о настройке сети в Windows XP. Идем в **Мой компьютер** → **Сетевое окружение** → **Установить домашнюю или малую сеть**. Второй компьютер должен быть включен. Откроется окно **Мастер настройки сети**. В окне **Метод подключения** выберем **Другое**. В окне **Другие способы подключения к Интернету** выберем пункт **Этот компьютер принад-**

лежит к сети, не имеющей подключения к Интернету. В новом окне задайте имя компьютера Com2 и описание (произвольный текст). В следующем окне задается имя для вашей сети (имя рабочей группы). В окне ввода введите Home вместо предлагаемого по умолчанию MSHOME. В следующем окне оставьте включенным пункт **Включить общий доступ к файлам и принтерам**. Затем происходит копирование файлов и выдается предложение о создании диска настройки сети. Если на втором компьютере установлена Windows 98, и вы уже настроили на нем сеть, то можно не создавать такой диск. Далее нажмем кнопку **Готово**. Снова идем в **Мой компьютер** → **Сетевое окружение** → **Отобразить сетевые подключения**. Щелкнем правой кнопкой по значку **Подключение по локальной сети** и выберем **Свойства**. На вкладке **Общие** выберем компоненту **Протокол Интернета (ТСР/П)** и нажмем **Свойства**. На вкладке **Общие** окна **Свойства** установим переключатель на пункт **Использовать следующий IP-адрес**. Наберем 192.168.0.10, маску — 255.255.255.0 и нажмем **ОК**. Затем нажмем **Закрыть** для фиксации настроек. Перезагружаем компьютер. Сеть должна заработать.

Даже если настройки сети правильные, второй компьютер появляется в сети не сразу, а через какое-то время (обычно несколько минут), когда система пропишет сетевое окружение в своих "схрихалях". Можно открыть **Мой компьютер**, щелкнуть по **Сетевому окружению**, и в окне появятся доступные папки на всех сетевых компьютерах (Windows XP). Если этого нет, щелкните по полю правой кнопкой мыши и выберите пункт **Обновить**. Если и это не помогает, перезагрузите компьютер и снова попробуйте. Также проверьте имя компьютера, имя рабочей группы, IP-адрес и маску подсети. Можно проверить работу сети с помощью команды **ping IP-адрес (Пуск → Выполнить)**. На экран выводится сообщения о переданных пакетах и скорости передачи. Либо попробуйте найти компьютер **Пуск → Найти...** → **Компьютер** и введите имя компьютера (Windows 98), или **Пуск → Поиск → Компьютер в сети** и введите имя компьютера (Windows XP).

Еще несколько замечаний по настройке сетевых карт. Иногда, несмотря на все ваши старания, сетевая карта никак не хочет устанавливаться. Идем в **Панель управления** → **Система** и смотрим установленные устройства, видим **Ethernet Controller** или название вашей сетевой карты с большим желтым знаком вопроса. Пометим ее и нажмем кнопку **Удалить**. Перезагружаем компьютер. Windows радостно сообщает, что найдено новое оборудование. Постарайтесь на этот раз настроить ее уже правильно. Этот метод применим и для других уст-

ройств — звуковой карты, модема и т.п.

Возможен и такой случай, что установленная в PCI-разъем сетевая карта определяется неправильно, например, как специфичный VGA-контроллер, и настроить ее как сетевую карту вам не удастся, Windows не позволит. Тогда нужно переставить ее в другой PCI-разъем и снова попытаться настроить ее.

Бывает, при установке какого-либо устройства в открывшемся окне почему-то отсутствует кнопка **Обзор**, и вы не можете выбрать каталог с нужным драйвером. Приходится по памяти указывать путь к нужному файлу в окне ввода. Поэтому полезно одновременно записать этот путь, например, на титульном листке компакт-диска.

При установке драйверов на какое-либо устройство Windows 98 требует указать расположение файла с нужным драйвером. Он находится на компакт-диске, прилагаемом к этому устройству. Вы пробуете указать папку с этим файлом с помощью кнопки **Обзор**, но оказывается, что у вас в системе нет CD-ROM'a! Остался один винчестер. И вы попадаете в тупиковую ситуацию. Здесь иногда помогает простая перезагрузка. Выньте компакт-диск из дисковода и перезагрузите ПК. Срочно копируем драйверы на винчестер и повторяем установку.

Чтобы этого не случилось, полезно загодя записать установочные файлы Windows и каталоги с драйверами на винчестер. Сделать это можно разными способами. Например, подключить винчестер как SLAVE к другому компьютеру и скопировать нужные файлы. Если вы раньше работали с MS DOS, можете установить сначала MS DOS, драйвер CD-ROM для DOS и Norton или Volkov Commander. И копируете с помощью NC или VC нужные файлы на винчестер.

Если один из компьютеров имеет постоянное подключение к Интернет, стоит хорошо подумать, давать ли полный доступ к файлам этого компьютера. Если вы подключаетесь к Интернету эпизодически по обычной телефонной линии и в основном смотрите WEB-сайты, то это не так страшно. Опасность резко возрастает, если вы имеете свой электронный почтовый ящик, так как многие вирусы распространяются с помощью электронной почты. Обычно они находятся в электронных письмах во вложенных файлах. Никогда не открывайте вложенные файлы неизвестного происхождения.

В случае, когда оба компьютера не подключены к Интернет, можно без больших опасений обеспечить полный доступ к файлам и принтерам на обоих компьютерах. Это создаст большие удобства в работе.

Литература

1. Руководство по настройке сетевой карты D-Link DFE-538TX PCI Adapter.



Чудо XX века: реальность и перспективы

А.КАШКАРОВ,
г.С.-Петербург.

(Продолжение. Начало в №10/2005)

Transmission Control Protocol — это протокол, тесно связанный с IP, который используется в аналогичных целях, но на более высоком уровне — транспортном уровне эталонной модели ISO OSI. Часто эти протоколы, по причине их тесной связи, вместе именуют TCP/IP. Термин TCP/IP обычно означает все, что связано с протоколами TCP и IP. Он охватывает целое семейство протоколов, прикладные программы и саму сеть. В состав семейства входят протоколы TCP, UDP, ICMP, telnet, FTP и многие другие. TCP/IP — это технология межсетевого взаимодействия, технология internet. То есть сеть, которая использует технологию internet, называется internet.

Протокол TCP занимается проблемой пересылки больших объемов информации, основываясь на возможностях протокола IP. Как это делается? Как можно переслать книгу по почте, если та принимает только письма и ничего более? Очень просто — разделить ее на страницы и отправить каждую страницу отдельным конвертом. Получатель, руководствуясь номерами страниц, легко сможет книгу восстановить. Этим же простым и естественным методом пользуется и TCP.

TCP делит информацию, которую надо переслать, на несколько частей. Нумерует каждую часть, чтобы позже восстановить порядок. Чтобы пересылать эту нумерацию вместе с данными, он обкладывает каждый кусочек информации своей обложкой — конвертом, который содержит соответствующую информацию. Это и есть TCP-конверт. Получившийся TCP-пакет помещается в отдельный IP-конверт, и получается IP-пакет, с которым сеть уже умеет обращаться.

Получатель (TCP-модуль, процесс) при получении распаковывает IP-конверты и видит TCP-конверты, распаковывает и их и помещает данные в последовательности частей в соответствующее место. Если чего-то недостает, он требует переслать этот кусочек снова. В конце концов, информация собирается в нужном порядке и полностью восстанавливается за доли секунды. Вот теперь этот массив пересылается к пользователю.

Это слегка упрощенный взгляд на TCP. В реальности пакеты не только теряются, но и могут искажаться при передаче из-за наличия помех на линиях связи. TCP решает и эту проблему. Для этого он пользуется системой кодов, исправляющих ошибки. Существует целая наука о таких кодировках. Простейшим примером может служить код с добавлением к каж-

дому пакету контрольной суммы (а к каждому байту — бита проверки на четность). При помещении в TCP-конверт вычисляется контрольная сумма, которая записывается в TCP-заголовок. Если при приеме заново вычисленная сумма не совпадает с той, что указана на конверте, значит, есть ошибка — где-то в пути имели место искажения, так что надо переслать этот пакет снова, что и делается.

Модуль TCP разбивает поток байтов на пакеты, не сохраняя при этом границы между записями. Если один прикладной процесс делает три записи в порт, то совсем не обязательно, что другой прикладной процесс на другом конце виртуального канала получит из своего порта именно три записи, причем именно таких (по разбиению), какие были переданы с другого конца. Вся информация будет получена исправно и с сохранением порядка передачи, но она может быть разбита по-другому, и на иное количество частей. Не существует зависимости между числом и размером записываемых сообщений с одной стороны и числом и размером считываемых сообщений с другой стороны. TCP требует, чтобы все отправленные данные были подтверждены принявшей их стороной. Он использует ожидания (таймауты) и повторные передачи для обеспечения надежной доставки. Отправителю разрешается передавать некоторое количество данных, не дожидаясь подтверждения приема ранее отправленных данных. Таким образом, между отправленными и подтвержденными данными существует окно уже отправленных, но еще не подтвержденных данных. Количество байтов, которое можно передавать без подтверждения, называется размером окна. Как правило, размер окна устанавливается в стартовых файлах сетевого программного обеспечения. Данные могут одновременно передаваться в обоих направлениях. Подтверждения для данных, идущих в одном направлении, могут передаваться вместе с данными, идущими в противоположном направлении. Приемники на обеих сторонах виртуального канала выполняют управление потоком передаваемых данных, для того чтобы не допускать переполнения буферов. Протокол TCP обеспечивает гарантированную доставку с установлением логического соединения в виде байтовых потоков. Он освобождает прикладные процессы от необходимости использовать ожидания и повторные пе-

редачи для обеспечения надежности. Наиболее типичными прикладными процессами, использующими TCP, являются ftp и telnet. Кроме того, TCP использует система X-Windows (стандартный многооконный графический интерфейс с пользователем) и "г-команды".

Возможности TCP даются не бесплатно, а реализация TCP требует большой производительности процессора и большой пропускной способности сети (не каждый "древний" ПК выдержит sie). Когда прикладной процесс начинает использовать TCP, то начинают напрямую общаться модуль TCP на машине пользователя и модуль на машине сервера. Эти два оконечных модуля TCP поддерживают информацию о состоянии соединения — виртуального канала. Виртуальный (дуплексный) канал потребляет ресурсы обоих оконечных модулей TCP. Один прикладной процесс пишет данные в TCP-порт, откуда они модулями соответствующих уровней по цепочке передаются по сети и выдаются в TCP-порт на другом конце канала, а другой прикладной процесс читает их из своего TCP-порта. Что входит на одном конце канала, выйдет с другого. Хотя в действительности никакая прямая линия отправителю и получателю в безраздельное владение не выделяется (другие пользователи могут использовать те же узлы и каналы связи в сети в промежутках между пакетами этих), но упрощенно, извне это именно так и выглядит.

Установка TCP-виртуального канала связи требует больших расходов на инициирование и поддержание соединения и приводит к задержкам передачи. Если вся эта суеда — излишество, лучше бы обойтись без нее. Если все данные, предназначенные для пересылки, умещаются в одном пакете, и если пользователя не особенно беспокоит надежность доставки, то можно обойтись без TCP.

Имеется другой стандартный протокол транспортного уровня, который не отягощен такими накладными расходами. Этот протокол называется UDP (User Datagram Protocol) — протокол пользовательских дейтаграмм. Он используется вместо TCP. Здесь данные помещаются не в TCP, а в UDP-конверт, который также помещается в IP-конверт. Этот протокол реализует дейтаграммный способ передачи данных.

Дейтаграмма — это пакет, передаваемый через сеть независимо от других пакетов без установления логического соединения и подтверждения приема. Дейтаграмма — самостоятельный пакет, который содержит всю необходимую для ее передачи информацию. Передача происходит безо всякой подготовки. Дейтаграммы не содержат средств обнару-



жения и исправления ошибок, поэтому при передаче данных с их помощью следует принимать меры по обеспечению надежности пересылки информации. Методы организации надежности могут быть самыми разными, обычно же используется метод подтверждения приема посылкой эхоотклика при получении каждого пакета с дейтаграммой.

Протокол UDP проще TCP, поскольку он не заботится о возможной пропаже данных, пакетов и сохранении правильного порядка данных. UDP используется для клиентов, которые посылают только короткие сообщения и могут просто заново послать сообщение, если отклик подтверждения не придет достаточно быстро. Предполагаем, что пользователь пишет программу, которая просматривает базу данных с телефонными номерами где-нибудь в удаленном месте сети. Совершенно незачем устанавливать TCP-связь, чтобы передать, например, двадцать символов в каждом направлении. Пользователь "укладывает" имя в UDP-пакет, запаковывает в IP-пакет и посылает. На другом конце прикладная программа получит пакет, прочтёт имя, посмотрит телефонный номер, положит его в другой UDP-пакет и отправит обратно. Что произойдет, если пакет по пути потеряется? Программа действует так — если она ждет ответа слишком долго, и становится ясно, что пакет затерялся, она повторяет запрос, посылает еще раз то же послание. Так обеспечивается надежность передачи при использовании протокола UDP.

В отличие от TCP, данные, отправляемые прикладным процессом через модуль UDP, достигают места назначения как единое целое. Например, если процесс-отправитель производит три записи в UDP-порт, то процесс-получатель должен будет сделать три чтения. Размер каждого записанного сообщения будет совпадать с размером соответствующего прочитанного. Протокол UDP сохраняет границы сообщений, определяемые прикладным процессом. Он не объединяет несколько сообщений в одно целое и не делит одно сообщение на части.

Альтернатива TCP—UDP позволяет пользователю рационально использовать предоставленные ресурсы, исходя из своих возможностей и потребностей.

Если нужна надежная доставка, то лучше использовать TCP.

Если нужна доставка дейтаграмм — UDP.

Если нужна эффективная доставка по длинному и ненадежному каналу передачи данных, то лучше использовать TCP.

Если нужна эффективность на быстрых сетях с короткими соединениями,

лучше всего будет UDP. Если потребности не попадают ни в одну из этих категорий, то выбор транспортного протокола неясен. Прикладные программы устраняют некоторые недостатки выбранного протокола.

Подключение к Интернету

В первую очередь необходимы модем и телефонная линия. Теперь необходимо выбрать провайдера — посредника, который предоставит доступ в Интернет. Нужно установить и настроить для работы программное обеспечение, как вариант — русскую версию программного комплекса Microsoft Internet Explorer версии 6.0 или более поздней (или другую почтовую программу). Кроме того, полезно снабдить ПК несколькими маленькими дополнительными программами, которые помогут облегчить работу с Интернетом.

Провайдеры появились в тот момент, когда родилась Интернет. Сеть родилась в тот момент, когда появились первые провайдеры. Извечный спор о курице и яйце, о первичном и вторичном...

Первоначально сеть Интернет (которая тогда называлась еще Arpanet) состояла преимущественно из постоянно подключенных компьютеров, каждый из которых обладал своим фиксированным адресом, а позднее — домашним именем.

Позднее родилась идея предоставлять доступ к Сети по телефонной линии с помощью сеансового подключения. Например, вы связываетесь по телефону с компьютером — постоянным "гражданином" Сети, подключаетесь к нему и таким образом сами становитесь частью Интернет. Появилось множество организаций, которые предоставляли доступ всем желающим. Не бесплатно, конечно... Так появились первые провайдеры.

Стать провайдером может любой, затратив деньги на мощный сервер, покупку множества телефонных входных линий для клиентов, и самое главное — на выделенный канал связи.

Этот канал — главное, что отличает провайдера от конечных пользователей. Стоит задуматься, с каким трудом отечественные телефонные линии пропускают мощный поток Интернет-информации. И одному-то пропускной способности канала не хватает... А если несколькими?

Поэтому передачу информации в Интернет провайдеры осуществляют через специальные высокоскоростные каналы связи, например, через волоконно-оптические кабели или спутниковую связь. Каналы позволяют одновременно работать в Интернет сотням и тысячам пользователей, которые не ощущают при

этом дискомфорта. В определенный момент емкости канала перестает хватать, тогда его либо модернизируют, делают более емким, либо связь катастрофически ухудшается...

Пользователю архиважно — с какой скоростью "работает" канал между его собственным компьютером и компьютером провайдера. Ведь каким бы быстрым и мощным ни был канал связи между провайдером и Сетью, хлипкость последнего участка в пределах пользователя — АТС может свести на нет все его достоинства. Именно различие в типе подключения и пропускной способности канала связи между пользователем и провайдером определяет вид доступа к Интернет, а заодно и его стоимость. Конечно, не в каждом городе России пользователь может позволить себе роскошь выбирать одну из нескольких схем. Но Интернет все активнее проникает в нашу жизнь, и самые новые виды коммуникации, еще пару лет назад доступные только в Москве и Петербурге, потихоньку "обживают" и провинцию.

Итак, какие же виды доступа в Интернет значатся в "меню" современных провайдеров?

Виды доступа условно делятся на две большие группы.

- сеансовое подключение — в этом режиме работы пользователь не подключен к Сети постоянно, соединяясь с ней через посредство телефонной линии на короткое время. Оплата взимается за каждый час работы в Сети. Данные в Сеть передаются в аналоговом виде.

- постоянное подключение — ПК подключен к быстрому каналу для доступа в Интернет, при этом данные передаются в Сеть в цифровом виде. Оплата взимается только за трафик — объем принятых компьютером данных.

Эти два вида отличаются не только временем пребывания пользователя в Сети, но и скоростью работы. При постоянном доступе ПК получает полноценную "прописку" в Интернет и собственный цифровой IP-адрес, по которому к конкретному компьютеру (точнее — к открытой для доступа части жесткого диска) может обратиться любой пользователь Интернет. Это оправдано, если ПК "по совместительству" работает в качестве сервера Сети и содержит собственный сайт пользователя или архив файлов.

В случае сеансового доступа, IP-адрес присваивается компьютеру только на время работы, выбираясь случайным образом из бесчисленного множества свободных адресов. Потому и называется он динамическим IP-адресом.

(Окончание следует)



ЛИСТАЯ СТРАНИЦЫ

Жесткие диски — потенциально самые ненадежные устройства компьютера. Поэтому **защита винчестеров и записанных на них данных** — это одна из главных задач, стоящих перед производителями. Технологии защиты данных — тема статьи *В.Цыганкова и Р.Никитина* "Спаси и сохрани" (**CHIP, 7/2005, С.64...66**).

Современные жесткие диски — очень сложные электронно-механические устройства. Уже одно наличие множества движущихся частей автоматически увеличивает риск аппаратных сбоев даже при аккуратном использовании. Если же винчестер используется в качестве сменного носителя, требования к его аппаратной части многократно возрастают, и необходимы специальные технологии, оберегающие и сам жесткий диск, и записанные на нем данные от повреждения.

Наиболее опасными для HDD считаются короткие по времени, но сильные по амплитуде удары. Последствия от такого воздействия на накопитель можно условно разделить на три группы:

- шлепок головок;
- проскальзывание и смещение дисков в пакете;
- появление люфта в подшипниках.

Первым и основным решением вопроса существенного снижения вероятности выхода из строя HDD является осторожность при обращении, как в момент работы, так и при транспортировке. Другие методы повышения безотказности — использование фирменных технологий защиты поверхности дисков внутри винчестера.

Фирма Quantum в своих изделиях использовала технологию SPS (Shock Protection System). В ее основе — конструктивные улучшения накопителя, призванные максимально увеличить поглощение энергии удара. Для этого была увеличена упругость держателя головки, вследствие чего при ударе головка либо вообще не перемещалась, либо ее сдвиг был весьма незначителен. В дальнейшем была разработана технология SPS II, отличительной особенностью которой было то, что система защищена еще и от ошибочной записи в момент удара, т.е., если в момент удара головка отклонилась от своей траектории, то процесс записи блокируется, а данные кэшируются.

GFP (G-Force Protection) — технология защиты от компании Seagate. Здесь для уменьшения эффекта шлепка головки о диск была уменьшена масса головки и ее держателя, а также увеличен зазор парения головки. Кроме того, классическая проводочная подвеска головки была заменена специальной гибкой системой Flex on Suspension с оптимальным управлением положения головки над пластиной. Кроме того, некоторое время диски Seagate (в частности, модель Barracuda U6) поставлялись с прорезиненной оболочкой, которая предохраняла винчестер от случайных ударов при транспортировке, а также защищала его электронную часть от статического электричества.

Фирменная технология компании Maxtor — ShockBlok (SB) — является дальнейшим развитием технологий Quantum. Чтобы уменьшить последствия от удара, во-первых, были уменьшены размер и масса головок.

Во-вторых, был механически укреплен поверхностный слой зоны парковки головок. Кроме того, была пересмотрена конструкция узла крепления головки — теперь она всегда опускается на поверхность пластины плашмя, что значительно снижает вероятность повреждения поверхности диска.

Одна из известных технологий компании Samsung — Shock Skin Bumper (SSB) — представляет собой аналог автомобильного бампера: система увеличивает продолжительность воздействия при одновременном уменьшении амплитуды нагрузок.

Компания Hitachi в своих винчестерах для мобильных компьютеров наряду с парковкой головок вдали от диска на специальном скате использует технологию Active Protection System. Основная часть системы — специальный микрочип, вмонтированный в материнскую плату и постоянно отслеживающий ускорение корпуса ноутбука (соответственно, и винчестера). Если ускорение превышает допустимые нормы, например, во время удара, то держатель головки по команде микрочипа автоматически переводит головку в зону парковки, благодаря этому сохраняется записанная информация и предотвращается возможное разрушение поверхности диска.

Однако, как бы не старались производители жестких дисков, полностью исключить возможность повреждения диска при неаккуратном обращении не получится. В полной мере защитить накопители от механических воздействий можно, только если использовать принципиально иные конструкции, не имеющие подающих элементов — например, блоки flash-памяти.

дительно. У Intel номера процессоров делятся на категории по трехзначным номерам. Чем больше первая цифра, тем выше класс кристалла.

Процессоры Intel Celeron D

Модель	Тактовая частота, ГГц	Soket	FSB, МГц	Объем кеша, Кб	Техпроцесс, мкм
345J	3,06	775	533	256	0,09
345	3,06	478	533	256	0,09
340J	2,93	775	533	256	0,09
340	2,93	478	533	256	0,09
335J	2,80	775	533	256	0,09
335	2,80	478	533	256	0,09
330J	2,66	775	533	256	0,09
330	2,66	478	533	256	0,09
325J	2,53	775	533	256	0,09
325	2,53	478	533	256	0,09
320	2,40	478	533	256	0,09

А.Попов в материале "Что в имени тебе моем?" (**CHIP, 7/2005, С.46...47**) приаодит расшифровку маркировок современных процессоров ф. Intel и AMD.

Два ведущих производителя процессоров для персональных компьютеров используют в названиях своих продуктов числовой код. Каждая из фирм считает, что именно ее система наиболее удачна.

Основное отличие маркировок процессоров Intel и AMD заключается в подходе к присвоению числового индекса той или иной модели. Для AMD этот показатель привязан к вычислительной мощности кристалла — чем выше число в названии процессора (performans rating), тем выше произво-

Внутри каждой категории представлены различные модели процессоров, отличающиеся, например, объемом кеша и тактовой частотой.



Процессоры Intel Pentium 4 с поддержкой технологии Hyper-Threading

Модель	Тактовая частота, ГГц	Socket	FSB, МГц	Объем кеша, Кб	Техпроцесс, мкм	EM64T	XD
660	3,6	775	800	2048	0,09	+	-
650	3,4	775	800	2048	0,09	+	-
640	3,2	775	800	2048	0,09	+	-
630	3,0	775	800	2048	0,09	+	-
570	3,8	775	800	1024	0,09	-	-
560	3,6	775	800	1024	0,09	-	-
550	3,4	775	800	1024	0,09	-	-
540	3,2	775	800	1024	0,09	-	-
530	3,0	775	800	1024	0,09	-	-
520	2,8	775	800	1024	0,09	-	-

Процессоры Intel Pentium Extreme Edition (двухъядерные)

Модель	Тактовая частота, ГГц	Socket	FSB, МГц	Объем кеша, Кб	Техпроцесс, мкм	Hyper-Threading	EM64T	XD
840	3,2	775	800	1024+1024	0,09	+	+	+

Процессоры Intel Pentium 4 Extreme Edition с поддержкой технологии Hyper-Threading (не имеют номера модели)

Тактовая частота, ГГц	Socket	FSB, МГц	Объем кеша, Кб	Техпроцесс, мкм	EM64T	XD
3,73	775	1066	2048	0,09	+	-
3,46	775	1066	512	0,13	-	-
3,40	775	800	512	0,13	-	-
3,40	478	800	512	0,13	-	-
3,20	478	800	2048	0,13	-	-

Процессоры Intel Pentium D (двухъядерные)

Модель	Тактовая частота, ГГц	Socket	FSB, МГц	Объем кеша, Кб	Техпроцесс, мкм	Hyper-Threading	EM64T	XD
840	3,2	775	800	1024+1024	0,09	-	+	+
830	3,0	775	800	1024+1024	0,09	-	+	+
820	2,8	775	800	1024+1024	0,09	-	+	+

Процессоры AMD Athlon 64

Маркировка	Модель	Тактовая частота, МГц	Socket	Напряжение питания, В	Потребляемая мощность, Вт	Объем кеша, Кб	Техпроцесс	Степпинг
ADA2800AEP4AP	2800+	1800	754	1,50	89	512	0,13	C0
ADA2800AEP4AR	2800+	1800	754	1,50	89	512	0,13	CG
ADA2800AEP4AX	2800+	1800	754	1,50	89	512	0,13	CG
ADA3000DEP4AW	3000+	1800	939	1,50	89	512	0,13	CG
ADA3000DK4BI	3000+	1800	939	1,40	67	512	0,13	CG
ADA3000DAA4BP	3000+	1800	939	Варьируется	67	512	0,09	E3
ADA3000AEP4AP	3000+	2000	754	1,50	89	512	0,13	C0
ADA3000AEP4AR	3000+	2000	754	1,50	89	512	0,13	CG
ADA3000AEP4AX	3000+	2000	754	1,50	89	512	0,13	CG
ADA3200DEP4AW	3200+	2000	939	1,50	89	512	0,13	CG
ADA3200DK4BI	3200+	2000	939	1,40	67	512	0,09	D0
ADA3200DAA4BP	3200+	2000	939	Варьируется	67	512	0,09	E3
ADA3200AEP5AP	3200+	2000	754	1,50	89	1024	0,13	C0
ADA3200AEP5AR	3200+	2000	754	1,50	89	1024	0,13	CG
ADA3200AEP4AR	3200+	2200	754	1,50	89	512	0,13	CG
ADA3200AEP4AX	3200+	2200	754	1,50	89	512	0,13	CG
ADA3400AEP5AP	3400+	2200	754	1,50	89	1024	0,13	C0
ADA3400AEP5AR	3400+	2200	754	1,50	89	1024	0,13	CG
ADA3400AEP4AR	3400+	2400	754	1,50	89	512	0,13	CG
ADA3400AEP4AX	3400+	2400	754	1,50	89	512	0,13	CG
ADA3500AEP4AS	3500+	2200	939	1,50	89	512	0,13	CG
ADA3500AEP4AW	3500+	2200	939	1,50	89	512	0,13	CG
ADA3500DK4BI	3500+	2200	939	1,40	67	512	0,09	CG
ADA3500DAA4BP	3500+	2200	939	Варьируется	67	512	0,09	E3
ADA3700DAA5BN	3700+	2200	939	Варьируется		1024	0,09	E4
ADA3700AEP5AR	3700+	2400	754	1,50	89	1024	0,13	CG
ADA3800DEP4AW	3800+	2400	939	1,50	89	512	0,13	CG
ADA3800DAA4BP	3800+	2400	939	Варьируется	89	512	0,09	E3
ADA4000AEP5AS	4000+	2400	939	1,50	89	1024	0,13	CG



Процессоры AMD Athlon 64FX

Маркировка	Модель	Тактовая частота, МГц	Soket	Напряжение питания, В	Потребляемая мощность, Вт	Объем кеша, Кб	Тех-процесс	Степпинг
ADAFX55DEI5AS	FX-55	2600	939	1,50	104	1024	0,13	CG
ADAFX53CEP5AT	FX-53	2400	940	1,50	89	1024	0,13	CG
ADAFX53DEP5AS	FX-53	2400	939	1,50	89	1024	0,13	CG
ADAFX51CEP5AK	FX-51	2200	940	1,50	89	1024	0,13	C0
ADAFX51CEP5AT	FX-51	2200	940	1,40	67	1024	0,13	CG

Процессоры AMD Athlon 64 X2 (двухъядерные)

Маркировка	Модель	Тактовая частота, МГц	Soket	Напряжение питания, В	Потребляемая мощность, Вт	Объем кеша, Кб	Техпроцесс	Степпинг
ADA4800DAA6CD	4800+	2600	939	1,35-1,40	110	1024+1024	0,13	E6
ADA4600DAA5BV	4600+	2400	940	1,35-1,40	110	512+512	0,13	E4
ADA4400DAA6CD	4400+	2400	939	1,35-1,40	110	1024+1024	0,13	E6
ADA4200DAA5BV	4200+	2200	940	1,35-1,40	110	512+512	0,13	E4

Процессоры AMD Sempron

Маркировка	Модель	Тактовая частота, МГц	Soket	Напряжение питания, В	Потребляемая мощность, Вт	Объем кеша, Кб	Техпроцесс	Степпинг
SDC2200DUT3D	2200+	1500	A	1,60	62	256	0,13	A
SDA2200DUT3D	2200+	1500	A	1,60	62	256	0,13	B
SDA2300DUT3D	2300+	1583	A	1,60	62	256	0,13	B
SDC2400DUT3D	2400+	1667	A	1,60	62	256	0,13	A
SDA2400DUT3D	2400+	1667	A	1,60	62	256	0,13	B
SDA2500AЮ2BO	2500+	1400	754	1,40	62	256	0,09	E3
SDA2500DUT3D	2500+	1750	A	1,60	62	256	0,13	A
SDA2600AЮ2BA	2600+	1600	754	1,40	62	128	0,09	D0
SDA2600AЮ2BO	2600+	1600	754	1,40	62	128	0,09	E3
SDA2600DUT3D	2600+	1833	A	1,60	62	256	0,13	B
SDA2800AЮ3BA	2800+	1600	754	1,40	62	256	0,09	D0
SDA2800AЮ3BO	2800+	1600	754	1,40	62	256	0,09	E3
SDC2800DUT3D	2800+	2000	A	1,60	62	256	0,13	A
SDA2800DUT3D	2400+	1667	A	1,60	62	256	0,13	B
SDA3000AIP2AX	3000+	1800	754	1,40	62	128	0,13	CG
SDA3000AЮ2BA	3000+	1800	754	1,40	62	128	0,09	D0
SDA3000AЮ2BO	3000+	1800	754	1,40	62	128	0,09	E3
SDA3000DUT4D	3000+	2000	A	1,60	62	512	0,13	A
SDA3100AIP3AX	3100+	1800	754	1,40	62	256	0,13	CG
SDA3100AЮ3BA	3100+	1800	754	1,40	62	256	0,09	D0
SDA3100AЮ3BO	3100+	1800	754	1,40	62	256	0,09	E3
SDA3300AЮ2BA	3300+	2000	754	1,40	62	128	0,09	D0
SDA3300AЮ2BO	3300+	2000	754	1,40	62	128	0,09	E3

Высокая скорость и поистине гигантские объемы современных винчестеров, казалось бы, должны были снять потребность в программах-дефрагментаторах. Однако этого не случилось. А.Евдокимов в статье "Время собирать фрагменты" (Подводная лодка, 7/2005, С.88...91) протестировал современные программы дефрагментации жестких дисков.

В современных компьютерах трудно добиться существенного прироста производительности за счет дефрагментации HDD. Но при регулярном проведении этой операции существенно снижается риск внезапного

обвала системы, а главное — безвозвратной утраты важной информации. Ведь реанимировать разбросанные по всему винчестеру файлы гораздо сложнее, чем компактно собранные в одном месте. Чем больше создается, удаляется, перемещается файлов на диске, тем выше вероятность системного сбоя, и уровень фрагментации в этом случае может достичь критической величины.

В процессе дефрагментации размещенная на жестком диске информация раскладывается по полочкам. Хороший дефрагментатор должен не только собрать разбросанные данные, но и по возможности упорядо-

чить их расположение на диске для ускорения доступа к ним. Также желательно, чтобы программа этого типа умела работать со swar-файлами, зоной MFT в NTFS-разделах, а также с важнейшими системными библиотеками, т.е. могла запускать процесс оптимизации в момент перезапуска Windows. При выборе программы-дефрагментатора важными свойствами являются скорость работы, объем требуемого для работы дискового пространства, наличие встроенного планировщика.

В таблице приведены результаты (оценка — по пятибалльной системе) тестирования семи дефрагментаторов.



Программа	Скорость анализа и дефрагментации	Автоматическая работа по расписанию	Дефрагментация системных и swap-файлов при перезагрузке Windows	Дополнительные возможности оптимизации данных и свободного пространства на диске	Суммарная оценка
Norton Speed Disk	4	4	-	4	12
Disk Defragmentation Wizard	3	-	5	3	11
MindSoft Defrag Lite 3.0	5	-	-	3	8
O&O Defrag 8.0 Prof. Edition	4	5	-	5	14
PerfectDisk 7.0	5	5	4	4	18
VoptXP 7.22	4	4	4	5	17
mst Defrag 1.5	4	-	-	3	7



Обзор игровых гарнитур сделал Р.Еремин в статье "Полное погружение" (Игромания, №7/2005, С.124...127).

В процессе игровых баталий (особенно сетевых) колонки не всегда позволяют четко определить положение противника, а доступ к микрофону весьма затруднен. Кроме того, надо щадить слух и нервы домочадцев. Истинному игроману не обой-

тись без игровой гарнитуры (наушников с микрофоном). Какие же требования предъявляются к мультимедийной гарнитуре? Первостепенное значение имеет высокое качество звучания. Второй важный момент — эргономика. Профессиональные гарнитуры мало приспособлены для игр, они предназначены совсем для других целей. Также при выборе кон-

кретной модели нужно обращать внимание на устройство оголовья и возможность его регулировки. Помимо игр, мультимедийная гарнитура хорошо подойдет для общения через Интернет (Skype), прослушивания музыки и т.д.

В таблице представлены сравнительные характеристики современных устройств данного класса.

Модель	Genius HS-02N	Genius HS-04S	Genius HS-04V	Creative HS300	Logitech Internet Chat Headset	Logitech stereo USB Headset 250	KOSS CS-80	KOSS CS-100	Plantronics DSP-500	Sennheiser PC130	Sennheiser PC140	Sennheiser PC155 USB
Частотный диапазон наушников	20 Гц – 20 кГц	20 Гц – 20 кГц	20 Гц – 20 кГц	20 Гц – 15 кГц	20 Гц – 20 кГц	20 Гц – 20 кГц	15 Гц – 25 кГц	30 Гц – 16 кГц	20 Гц – 20 кГц	30 Гц – 18 кГц	40 Гц – 20 кГц	18 Гц – 22 кГц
Частотный диапазон микрофона	20 Гц – 20 кГц	50 Гц – 20 кГц	50 Гц – 20 кГц	380 Гц – 15 кГц	100 Гц – 10 кГц	100 Гц – 16 кГц	-	-	100 Гц – 10 кГц	80 Гц – 15 кГц	80 Гц – 15 кГц	80 Гц – 15 кГц
Сопротивление, Ом	32	32	32	32	32	-	60	32	-	32	32	32
Регулятор громкости	Есть	Есть	Есть	Есть	Есть	Нет	Нет	Нет	Есть	Есть	Есть	Есть
Тип подключения	miniJack	miniJack	miniJack	miniJack	miniJack	USB	miniJack	miniJack	USB	miniJack	miniJack	USB или miniJack
Цена, USD	8	12	27	17	35	42	24	20	105	46	57	160
Достоинства	Низкая цена, отсоединяемая трубка-микрофон, складное оголовье	Низкая цена	Вибрация, удобная и качественная конструкция	Хорошее звучание и качественный микрофон, легкая конструкция, оправданная цена	Сменные цветные панели	Хорошее звучание и запись, независимость от звуковой карты	Отличное соотношение цена/качество, хорошее звучание, легкая и удобная конструкция	Отличное соотношение цена/качество, хорошее звучание, легкая и удобная конструкция	Использование звукового процессора, независимость от звуковой карты	Чистый звук, легкая и качественная конструкция	Чистый звук, легкая и качественная конструкция	Удачная конструкция, отличное звучание и запись, универсальность подключения
Недостатки	Невысокое качество записи и воспроизведения, слабая конструкция	Слабая конструкция, искажения на низких и высоких частотах, крупная подушка микрофона	Искажения на низких и высоких частотах, крупная подушка микрофона	Неуверенные верхние частоты	Искажения на низких и высоких частотах, завышенная цена, учитывая качество звука	Отсутствие регулятора громкости, неприятный материал обшивки амбушюр	Отсутствие регулятора громкости	Отсутствие регулятора громкости	Высокая цена	Относительно высокая цена	Относительно высокая цена	Высокая цена



Borland C++ Builder 6

для начинающих

О.ВАЛЬПА,
E-mail: sandh@narod.ru

(Продолжение. Начало в №№7-12/2004, 1-5,8-10/2005)

Эта статья завершает цикл "Borland C++ Builder 6 для начинающих" рассмотрением примеров создания двух рабочих программ.

Примеры программ

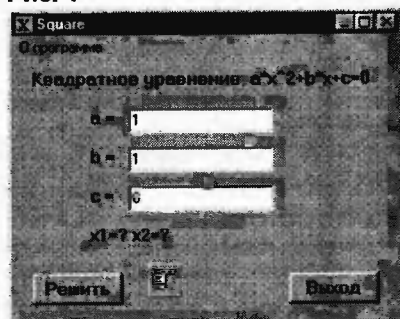
Итак, мы рассмотрели все необходимые инструменты, составляющие и способы программирования в среде разработки программ Borland C++ Builder 6. Теперь читатели вооружены достаточными знаниями, для того чтобы начать составление и разработку собственных программ. Но для закрепления изученного материала на практике я хочу предложить вниманию читателей еще несколько программ, которые помогут начинающим программистам обрести уверенность в своих силах.

Программа Square

Предлагаемая программа предназначена для вычисления корней квадратного уравнения и может быть полезна учащимся школ. На ее примере можно будет создать множество других программ, облегчающих вычисления различных уравнений и функций.

Сформируем интерфейс программы. Создайте новое приложение и поместите на его форму пять компонентов Label, три компонента Edit, две кнопки Button и компонент MainMenu. Задайте для формы Form1 свойство Name с именем "FormMain" и свойство Caption с названием "Square". Замените свойства Name компонентов Label1...Label5 на "LabelFormul", "Labela", "Labelb", "Labelc" и "Labelx1x2" соответственно, и задайте для них следующие свойства Caption: "Квадратное уравнение $a \cdot x^2 + b \cdot x + c = 0$ ", "a=", "b=", "c=" и "x1=? x2=?". Замените свойства Name компонентов Button1 и Button2 на "ButtonExe" и "ButtonOut" соответственно, и задайте для них следующие свойства Caption: "Решить" и "Выход". Измените значение свойства Name компонентов Edit1...Edit3 на "Edita", "Editb" и "Editc" соответственно, а свойства Text этих компонентов заполните значениями 1, 1 и 0. Дважды щелкните левой кнопкой мыши по компоненту MainMenu1 и в открывшемся окне FormMain задайте свойство Caption для первого пункта меню в виде названия "О программе". Закройте окно FormMain и расположите компоненты на форме в соответствии с рис.1.

Рис. 1



обработчик события OnClick следующие строки программы:

```
double a,b,c,x1,x2;
try
{
    a=Edita->Text.ToDouble();
    b=Editb->Text.ToDouble();
    c=Editc->Text.ToDouble();
}
```

```
catch(EConvertErrorE)
{
    ShowMessage("Неправильно введен коэффициент\rfПовторите ввод");
    Labelx1x2->Caption="x1=? x2=?"; // Убрать предыдущий результат
    return;
}
int d=b*b+4*a*c;
if (a==0){
    if (b==0){
        Labelx1x2->Caption="Неправильно заданы коэффициенты a и b";
        return;
    };
    x1=-c/b;
    Labelx1x2->Caption="Корень один.\rx="+String(x1)+";";
    return;
};
if (d<0){
    Labelx1x2->Caption="Детерминант меньше нуля. Нет корней!";
    return;
};
if (d==0){
    x1=(-b/(2*a));
    Labelx1x2->Caption="Детерминант равен нулю.
        Корень один.\rx="+String(x1)+";";
    return;
};
x1=(-b+sqrt(d))/(2*a);
x2=(-b-sqrt(d))/(2*a);
Labelx1x2->Caption="x1="+String(x1)+"; x2="+String(x2)+";";
```

Эти программные строки и весь проект программы под названием Square выложены в приложении к статье на сайте журнала. В них описываются переменные, которые участвуют в вычислениях корней квадратного уравнения, и включены операторы и функции для вычисления корней. Поскольку при вычислении могут встретиться запрещенные ситуации, наподобие деления на ноль и пр., в программу включены операторы проверки подобных ситуаций с выводом соответствующих сообщений пользователю программы. В программе эти сообщения присутствуют в виде строк текста, заключенных в кавычки. Если этого не делать, то программа будет выдавать системные предупреждения на английском языке или "зависать". Хорошим стилем программирования считается предупреждение всех ошибочных ситуаций и недопущение туликового выполнения программы, приводящего к ее "зависанию". В противном случае программа будет раздражать пользователя, и он от нее откажется.

Так как в тексте программы присутствуют непростые арифметические операции, например вычисление квадратного корня sqrt(), необходимо подключить к программе математическую библиотеку. Для этого в модуле программы после строки #include "USquare.h" впишите строку подключения библиотеки с математическими функциями:

```
#include <math.h>.
```

Теперь сформируйте обработчик события OnClick для кнопки "Выход" и вставьте в него программную строку для закрытия приложения:

```
Close();
```

Приложение уже можно запустить, и оно будет работать. Но в нем не будет работать главное меню. Создадим поддержку для этого меню.

Сформируйте обработчик события OnClick для первого пункта меню N1 компонента MainMenu1 путем двойного щелчка левой кнопкой мыши по самому компоненту и вставьте в него строку для модального вызова формы FormAbout с описанием программы:

```
FormAbout->ShowModal();
```



Модальный вызов не позволяет программе переключиться на другое окно, до тех пор пока текущее окно не будет закрыто. В отличие от него, немодальное окно вызывается функцией Show() и не налагает такого условия.

Поскольку формы с описанием программы нет, ее необходимо создать. Для этого выполните из главного меню команду File→New→Form. Для вновь созданной формы задайте свойство Name с именем "FormAbout" и свойство Caption с названием "О программе". Теперь поместите на эту форму четыре компонента Label, один компонент Image и одну кнопку BitBtn из палитры Additional, а также компонент MonthCalendar из палитры Win32. Для компонентов Label1...Label4 задайте свойство Caption с названиями: "Программа решения квадратных уравнений", "Версия 1.0", "Послать E-mail" и "Поиск в интернете".

Для компонентов Label3 и Label4 через свойство Font установите шрифт — синий подчеркнутый полужирный курсив. Свойство Cursor для этих компонентов поменяйте на crHandPoint. Это делается для того, чтобы выделить надписи данных компонентов и изменить при наведении на них вид курсора мыши.

Создайте обработчик события OnClick для компонента Label3 и вставьте в него программную строку вызова почтовой программы:

```
ShellExecute(Handle, "open", "mailto:sandh@narod.ru?x-
priority=normal&subject=ТЕМА_ПИСЬМА", NULL, NULL, SW_SHOWNORMAL);
```

Здесь в программной строке можно изменить адрес получателя письма и заполнить тему письма собственным текстом.

Далее создайте обработчик события OnClick для компонента Label4 и вставьте в него программную строку вызова интернет-обозревателя:

```
ShellExecute(Handle, "open", "http://www.ya.ru", 0, 0, SW_SHOW);
```

Естественно, что адрес страницы http://www.ya.ru можно заменить на любой другой подходящий реально существующий адрес.

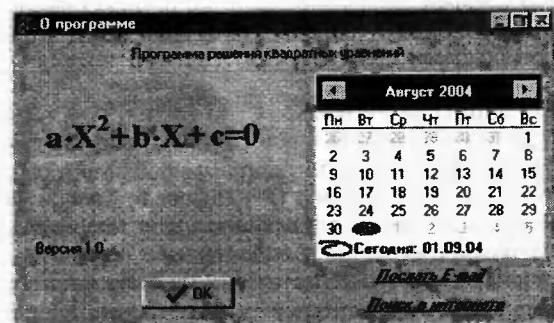
Для кнопки BitBtn1 установите свойство Kind как bkOk. Создайте обработчик события OnClick для этой кнопки и вставьте в него программную строку выхода:

```
Close();
```

Наконец, скопируйте в каталог с программой предваритель-

но созданный файл с рисунком или фотографией в формате bmp или jpg и загрузите его через свойство Picture компонента Image. Для того чтобы рисунок совпадал с размерами компонента Image, установите его свойство AutoSize в значение true. Расположите все компоненты на форме в соответствии с рис. 2 и сохраните модуль программы в файле с именем about.cpp.

Рис. 2



Поскольку мы добавили еще одну форму в наше приложение, необходимо включить в основной модуль Usquare.cpp имя файла описания новой формы с названием about.h. Для этого впишите в модуле Usquare.cpp после строки #include <math.h> программную строку:

```
#include "About.h".
```

Теперь остается лишь заменить значок-иконку программы с помощью команды главного меню Project→Options→Application на созданный во встроенном редакторе Image Edit файл иконки с расширением ico и откомпилировать программу.

Программа готова. Запустите ее на выполнение и проверьте в работе. В этой программе мы использовали новый компонент MonthCalendar, который позволяет без труда внедрить в программу красочный календарь с богатыми функциями. Кроме того, мы научились запускать из программы почтовую службу отправки электронных писем и входить в Интернет из работающей программы. Подобные приемы могут вами применяться во вновь создаваемых программах.

(Окончание следует)

Поиск и сортировки: сортировка обменом

А.КОРБИТ, Т.КРИВОНОСОВА,

г.Минск, БГУИР, ФИТиУ, каф.ВмиП.

E-mail: a_korbit@mail.ru

Метод 1. Стандартный обмен (сортировка обменом, метод "пузырька")

Пусть необходимо отсортировать массив по убыванию. Рассмотрим еще один метод сортировки, алгоритм которого можно представить в следующем виде:

1. Сравниваем первые два элемента. Если первый меньше второго, то меняем их местами.

2. Сравниваем второй и третий, третий и четвертый, ..., предпоследний и последний элементы, при необходимости меняя их местами. В конечном итоге наименьший элемент окажется на последнем месте.

3. Снова просматриваем массив с его начала, уменьшив на единицу количество просматриваемых элементов (т.е. ранее упорядоченную часть массива больше не рассматриваем).

4. Весь массив будет отсортирован после просмотра, в котором приняли участие только первый и второй элементы.

Если рассматривать массив не как горизонтальную, а как вертикальную структуру, элементы представить как пузырьки в емкости с водой, при этом масса каждого пузырька соответствует значению элемента этого массива, в этом случае при каждом просмотре один пузырек как бы поднимается до уровня, соответствующего его массе. Такой метод широко известен под названием "пузырьковая сортировка".

Пример 1.

Пусть есть целочисленный массив из N элементов.

Обозначим:

- $L=N-k+1$ — длина рабочей части массива;
- k — номер просмотра;
- i — номер проверяемой пары;
- $N-k$ — номер последней пары.

Как и раньше, вертикальной чертой будем отделять отсортированные элементы массива.

Дано: $a = 5\ 4\ 8\ 2\ 9$; $N=5$. Сортируем по возрастанию.

Первый просмотр: рассматривается весь массив.

($k=1$, $L=5$)

$i=1$ 5 4 8 2 9

 > меняем

$i=2$ 4 5 8 2 9

 < не меняем

$i=3$ 4 5 8 2 9

 > меняем

$i=4$ 4 5 2 8 9

 < не меняем

9 стоит на своем месте.

Второй просмотр: рассматриваем часть массива с первого до четвертого элемента.

```
(k=2, L=4)
i=1   4 5 2 8 | 9
      < не меняем
i=2   4 5 2 8 | 9
      > меняем
i=3   4 2 5 8 | 9
      < не меняем
```

8 стоит на своем месте.

Третий просмотр: рассматриваемая часть массива содержит три первых элемента.

```
(k=3, L=3)
i=1   4 2 5 | 8 9
      > меняем
i=2   2 4 5 | 8 9
      < не меняем
```

5 стоит на своем месте.

Четвертый просмотр: рассматриваем последнюю пару.

```
(k=4, L=2)
i=1   2 4 | 5 8 9
      < не меняем
```

4 стоит на своем месте.

Для самого маленького элемента (2) остается только одно место — первое.

А так будет выглядеть сортировка по убыванию массива, отсортированного выбором в [1]:

Пример 2.

$a = \{0, 1, 9, 2, 4, 3, 6, 5\}$.

- 1) 1, 9, 2, 4, 3, 6, 5, | 0
- 2) 9, 2, 4, 3, 6, 5, | 1, 0
- 3) 9, 4, 3, 6, 5, | 2, 1, 0
- 4) 9, 4, 6, 5, | 3, 2, 1, 0
- 5) 9, 6, 5, | 4, 3, 2, 1, 0
- 6) 9, 6, | 5, 4, 3, 2, 1, 0
- 7) 9, | 6, 5, 4, 3, 2, 1, 0

Число сравнений в данном алгоритме равно $(N^2 - N)/2$.

Алгоритм метода имеет вид:

```
Начало цикла (выполнять для i от 1 до N-1)
  Начало цикла (выполнять для j от 1 до N-1)
    если  $a[j] > a[j+1]$ 
      тогда
         $x = a[j]$ ;
         $a[j] = a[j+1]$ ;
         $a[j+1] = x$ ;
      все
    конец цикла
  конец цикла
```

В последнем примере последние два прохода не влияют на порядок расположения элементов, из-за того что они уже отсортированы. Оптимизируем алгоритм — можем запомнить, были или нет перестановки в процессе некоторого прохода. И если перестановок не было, то работу можно заканчивать.

Можно сделать еще одно улучшение, если запомнить не только сам факт, что обмена не было, но и положение (индекс $i1$) последнего обмена. Все пары соседних элементов после этого индекса уже отсортированы (следуют в требуемом порядке). Поэтому просмотр можно завершить.

Как и раньше, введем логическую переменную-флажок $flag$ для контроля, был обмен или нет, и переменную $i1$ для запоминания индекса последнего обмена. Еще нужна одна переменная R — это будет граница, на которой заканчиваем просмотр

```
 $flag = \text{"истина"};$ 
 $i1 = N-1$ ;
Начало цикла: выполнять пока  $flag = \text{"истина"};$ 
   $flag = \text{"ложь"};$ 
   $R = i1$ ;
  Начало цикла (выполнять для j от 1 до R)
    если  $a[j] > a[j+1]$ 
```

```
тогда
   $x = a[j]$ ;
   $a[j] = a[j+1]$ ;
   $a[j+1] = x$ ;
   $flag = \text{"истина"};$ 
   $i1 = j$ ;
```

все

конец цикла

конец цикла

Один из вариантов программы:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
  int *a, i, j, n, r;
  cout << "\n n-> "; cin >> n;
  a = new int[n];
  cout << "\n Input array\n";
  for (i=0; i<n; i++)
  {
    printf("\n a[%d] = ", i+1);
    cin >> a[i];
  }
  for (i=0; i<n; i++)
  for (j=0; j<(n-1)-i; j++)
    if (a[j]<a[j+1])
    {
      r=a[j];
      a[j]=a[j+1];
      a[j+1]=r;
    }
  cout << "\n Sort Array \n";
  for (i=0; i<n; i++)
    cout << " " << a[i];
  cout<<endl;
  delete [] a;
  getch();
}
```

Метод 2. Шейкерная сортировка

Если внимательно проанализировать "пузырьковую сортировку", можно заметить некоторую своеобразную асимметрию. Самый легкий пузырек занимает свое место за один проход, а самый тяжелый будет просачиваться на свое место на один шаг при каждом просмотре. Иными словами, всплывает пузырек сразу — за один проход, а тонет очень медленно — за один проход на одну позицию. Это наводит на мысль чередовать направление последовательных просмотров. Такая сортировка называется "шейкерной". Рассмотрим ее работу на том же массиве $A = \{0, 1, 9, 2, 4, 3, 6, 5\}$ (L — левая граница просмотра, R — правая).

- | | |
|---------------------------|----------------|
| 1) 1, 9, 2, 4, 3, 6, 5, 0 | ($L=1, R=7$) |
| 2) 9, 1, 6, 2, 4, 3, 5, 0 | ($L=2, R=7$) |
| 3) 9, 6, 2, 4, 3, 5, 1, 0 | ($L=2, R=6$) |
| 4) 9, 6, 5, 2, 4, 3, 1, 0 | ($L=3, R=6$) |
| 5) 9, 6, 5, 4, 3, 2, 1, 0 | ($L=3, R=5$) |
| 6) 9, 6, 5, 4, 3, 2, 1, 0 | ($L=4, R=5$) |

Алгоритм данной сортировки предлагаем написать самостоятельно.

Все эти улучшения сокращают количество операций сравнения для частных случаев, однако при неблагоприятной начальной расстановке элементов массива (подумайте, какой) приходится проделать все $(N^2 - N)/2$ операций сравнения.

Метод 3. Челночная сортировка

Челночная сортировка, называемая также "сортировкой просеиванием" или "линейной вставкой с обменом", является наиболее эффективной из всех рассмотренных выше методов и отличается от сортировки обменом тем, что не сохраняет фиксированную последовательность сравнений.

Алгоритм челночной сортировки действует точно так же,

как стандартный обмен, до тех пор пока не возникает необходимость перестановки элементов исходного массива. Сравниваемый меньший элемент поднимается, насколько это возможно, вверх. Этот элемент сравнивается в обратном порядке со своими предшественниками, по направлению к первой позиции. Если он меньше предшественника, то выполняется обмен, и начинается очередное сравнение. Когда элемент, передвигаемый вверх, встречает меньший элемент, этот процесс прекращается, и нисходящее сравнение возобновляется с той позиции, в которой выполнялся первый обмен. Сортировка заканчивается, когда нисходящее сравнение выходит на границу массива.

Процесс челночной сортировки можно проследить на следующем примере:

Исходный массив: 2 7 9 5 4.
 Сортируем по возрастанию.
 Нисходящие сравнения: 2 7; 7 9; 9 5;
 После перестановки: 2 7 5 9 4.
 Восходящие сравнения и обмен: 7 5 → 5 7; 2 5 (встретил меньший элемент) — конец восходящего сравнения;
 получен массив 2 5 7 9 4.
 Нисходящие сравнения: 9 4.
 После перестановки: 2 5 7 4 9.
 Восходящие сравнения и обмен: 7 4 → 4 7; 5 4 → 4 5; 2 4 — конец восходящего сравнения; получен массив 2 4 5 7 9.

Метод 4. Сортировка выбором наилучшего (наименьшего, наибольшего) элемента

Идея метода проста — просматривается последовательно весь массив, и *i*-й элемент сравнивается со всеми, следующими за ним. Найдя наилучший, переставляем места его и *i*-й. После перебора всех элементов массив оказывается отсортирован.

Внешний цикл изменяется от *i*=0 до *i* < (*n*-1), внутренний цикл изменяется от *j*=*i* до *j* < *n*.

Программа сортировки целых чисел по убыванию может иметь вид:

```
void main()
{
    int *a, r, i, j, n, max, i_max;
    cout << "\n Size : "; cin >> n;
    a = new int[n];
    for ( i=0; i<n; i++)
        cin >> a[i];
    for ( i=0; i<(n-1); i++)
    {
        max=a[i]; i_max=i;
        for ( j=i+1; j<n; j++)
            if( max<a[j] )
            {
                max=a[j]; i_max=j;
            }
        r=a[i]; a[i]=max; a[i_max]=r;
    }
    for(i=0;i<n;i++)
        printf("%d",a[i]);
    delete [ a;
    getch();
}
```

Литература

1. А. Корбит, Т. Кривоносова. Поиск и сортировки: сортировка выбором. — Радиомир. Ваш компьютер, 2005, №8, С.36.
2. А. Корбит, Т. Кривоносова. Поиск и сортировки: сортировка вставками (включениями). — Радиомир. Ваш компьютер, 2005, №9, С.32.
3. В. Котов и др. Методы алгоритмизации. — Мн.: Народная асвета, 1996.
4. А. Крячков и др. Программирование на С и С++. Практикум. — М.: Горячая линия-Телеком, 2000.



Р.ФАСИХОВ,

г.Казань.

E-mail: fasihov@mail.ru

Написание плагинов для перехватчика API-функций

Сам себя не похвалишь — никто не похвалит.

В этой статье я хочу показать, как пишутся плагины для моего перехватчика API-функций (API-шпиона) **kerberos 1.01**. Исполняемый файл **kerberos 1.01** можно взять на www.wasm.ru (в разделе "Утилиты").

Без поддержки плагинов создание нового API-шпиона имело бы мало смысла, т.к. в этом случае можно было бы использовать любой иной. С другой стороны, собственные плагины дают возможность безграничного контроля над приложением.

Общие аспекты использования плагинов:

- одновременно может работать только один плагин;
- файл плагина всегда называется **ke_plug.dll**;
- плагин считается установленным, если лежит в директории с **ke_core.dll**.

Код **kerberos 1.01** написан исключительно на ассемблере (**masm32**) как на самом лучшем языке программирования, поэтому и плагин тоже пишется на этом языке. В каждом плагине должна присутствовать функция инициализации **plug_setup**. Она возвращает в регистр **eax** значение 0, если рапорт шпиона необходим, иначе рапорт создан не будет. Это нужно, если вы собираетесь использовать **kerberos** как загрузчик программ.

Также в каждом плагине должна быть функция **plugin** — это основной код плагина. Функции **plugin** и **plugin_setup** должны являться экспортируемыми, иначе ничего не будет работать (самое лучшее — воспользоваться готовым примером из папки **ke_plug**).

Давайте рассмотрим шаблон, который подойдет для всех плагинов:

```
ke_plug.asm
.....
;Возвращаем значение рапорт нужен/не нужен

plug_setup proc
mov eax, REPORT_OFF
ret
plug_setup endp

;Вызывается каждый раз, при перехвате любой функции,
;которая есть в базе

plugin proc
pop ret_address ;Снимаем со стека адрес возврата
pop func_addr ;Адрес начала перехваченной функции
pop func_par ;Количество параметров функции
pop func_nam ;Указатель на имя функции
pop func_return_address ;Адрес возврата в вызывающую программу
pop func_idx ;Индекс функции в базе
pushfd
pushad
mov eax, func_addr

;Ваш код должен быть здесь

.if eax == FUNC_0

;Выполняем действия, которые хотим
```



```

;.....
;Восстанавливаем регистры и вызываем функцию
;Можно не вызывать, но надо будет сбалансировать стек
;от параметров функции

```

```

popad
popfd
call func_addr

```

```

;Если у нас другая функция, для которой установлен плагин

```

```

.elseif eax == FUNC_1
;.....
;Никакая функция не подошла - просто вызываем
;перехваченную функцию

```

```

.else
popad
popfd
call func_addr
.endif

```

```

;Возвращаемся туда, откуда пришли

```

```

push ret_address
ret
plugin endp

```

Нет ничего сложного. Надеюсь, что непонятные моменты прояснятся при разборе конкретного примера.

Рассмотрим, как можно сделать граббер музыки с записью в wav-файл. Я тестировал его на демке UberNuss, если будете пробовать на ней, не забудьте вытащить из нее Win32.exe. Размер демки — 4 Кб, она сделана как com-файл, который распаковывается и запускает сам PE-файл.

Программа, из которой будет вытаскиваться музыка, должна пользоваться для вывода звука следующими функциями из winmm.dll:

waveOutOpen, waveOutWrite.

Первым делом модифицируем файл ke_spy.txt, чтобы в нем были эти самые функции. В противном случае не будет перехвата этих функций и, соответственно, не будет вызова нашего кода из плагина. Если нужны только две функции, то нет необходимости перехватывать остальные — так надежнее. Сам ke_spy.txt для нашего плагина будет выглядеть так:

```

[winmm.dll]
waveOutOpen ,6
waveOutWrite ,3.

```

Теперь перейдем к самому плагину. Я приведу только его ключевую часть, полностью код смотрите в исходниках.

Главное, что делает плагин:

- открывает файл для вывода в него данных;
- пишет первую часть заголовка (по шаблону);
- записывает вторую часть заголовка (берется из параметров функции **waveOutOpen**;
- дописывает заголовок wav-файла.
- выводит буфер, переданный в функцию **waveOutWrite**, в выходной файл;
- после окончания своей работы закрывает выходной файл.

Описания функции имеются в MSDN, описания самого формата wav-файла можно легко найти в Сети.

Остается заметить, что в плагине не заполнены два поля DWORD по смещениям 04h и 024h. Они должны быть соответственно равны: Размер Файла — 8 и Размер Файла — 44. Хотя можно послушать и так, я пробовал в WinAmp5.0.

```

;Имя выходного файла и заголовок wav-файла

```

```

ofile db "m.wav",0

```

```

wavhead 0 db "RIFF"
db 0,0,0,0
db "WAVE"
db "fmt "
dd 10h
wavhead 1 db "data"
db 0,0,0,0
install_plugin proc
invoke LoadLibrary,offset sz_winmm
mov hwinmm, eax
invoke GetProcAddress, eax, offset sz_waveOutOpen
mov waveOutOpen_addr, eax
invoke GetProcAddress, hwinmm,offset sz_waveOutWrite
mov waveOutWrite_addr, eax
invoke CreateFile,offset
ofile,GENERIC_WRITE,NULL,NULL,CREATE_ALWAYS,FILE_ATTRIBUTE_NORMAL,NULL
mov hwavfile, eax
ret
install_plugin endp

```

```

plugin proc
pop ret_address
pop func_addr
pop func_par
pop func_nam
pop func_return_address
pop func_idx
pushfd
pushad
mov eax, func_addr

```

```

;Если адрес перехваченной функции совпал, то действуем

```

```

.if eax == waveOutOpen_addr

```

```

;Пишем в файл заголовок wav-файла

```

```

invoke WriteFile,hwavfile,offset wavhead_0,20,ADDR bwri,NULL

```

```

;Пишем в файл структуру, переданную функции waveOutOpen
;(должна быть в заголовке)

```

```

mov eax, [esp+2Ch] ;LPWAVEFORMATEX
invoke WriteFile,hwavfile, eax,10h,ADDR bwri,NULL

```

```

;Дописываем в файл заголовок

```

```

invoke WriteFile,hwavfile, offset wavhead_1,8,ADDR bwri,NULL
popad
popfd
call func_addr

```

```

;Если мы перехватили waveOutWrite, то выводим ее звуковой буфер
;в наш файл, а уже потом играем

```

```

.elseif eax == waveOutWrite_addr
mov eax, [esp+28h] ;LPWAVEHDR
mov ecx, [eax] ;lpData
mov edx, [eax+4] ;dwBuffer
invoke WriteFile,hwavfile, ecx, edx,ADDR bwri,NULL
popad
popfd
call func_addr ; Вызвали waveOutWrite
.else
popad
popfd
call func_addr ; Ни одна не совпала - просто вызываем ее
.endif
push ret_address ; Возвращаемся
ret
plugin endp

```

Напоследок хочу сказать, что при помощи плагинов можно делать что-нибудь более интересное, например, цеплять шпиона к создающимся процессам. Для этого нужно осуществить перехват CreateProcess и написать для нее обработчик, который внедрит ke_core.dll при помощи CreateRemoteThread.

На сайте журнала в приложении к статье выложены исходные тексты проекта.



В.КАРЦЕВ,
г.Минск, БГУИР.

Управление потоком сообщений в приложениях, использующих библиотеку MFC

(Продолжение. Начало в №10/2005)

Следует сказать, что функция `PreTranslateMessage` вызывается до того как сообщение будет обработано `Windows` функциями `TranslateMessage` и `DispatchMessage`. Это и позволяет вовремя отфильтровать (проверив содержимое `*pMsg`) требуемое сообщение и изменить его параметры.

Напомню, что функция `TranslateMessage` получает на основе клавиатурных сообщений символьные сообщения. Такие сообщения попадают в очередь сообщений потока, откуда они могут быть впоследствии извлечены посредством функций `PeekMessage` и `GetMessage`. Например, сочетание клавиатурных сообщений `WM_KEYDOWN` и `WM_KEYUP` функция `TranslateMessage` преобразует в символьное сообщение `WM_CHAR` или `WM_DEADCHAR`. Другими словами, в функцию `PreTranslateMessage` сообщения `WM_KEYDOWN` и `WM_KEYUP` попадут раньше сообщений `WM_CHAR` и `WM_DEADCHAR`. Это одна из причин, по которой в примере обрабатываются комбинации сообщений `WM_KEYDOWN` и `WM_KEYUP`. Вторая причина состоит в том, что сообщение `WM_CHAR` говорит лишь о факте нажатия клавиши на клавиатуре, а не о моментах времени нажатия и отпускания этой клавиши пользователем.

Функция `DispatchMessage` отвечает за "доставку" сообщений оконной процедуре.

После модификации содержимого метода `PreTranslateMessage` можно создать функции-члены класса `CControlMessageDlg`, отвечающие за обработку нажатия соответствующих клавиш на клавиатуре и кнопок в окне приложения. Их текст в файле `ControlMessageDlg.cpp` имеет вид:

```
void CControlMessageDlg::OnButtonplus()
{
    // TODO: Add your control notification handler code here
    if (Calc)
    {
        m_strResult.Format("%f", m_dFirst+m_dSecond);
        m_cstrStaticResult="сложения";
        ResCalc=true;
        UpdateData(false);
    }
}

void CControlMessageDlg::OnButtonminus()
{
    // TODO: Add your control notification handler code here
    if (Calc)
    {
        m_strResult.Format("%f", m_dFirst-m_dSecond);
        m_cstrStaticResult="вычитания";
        ResCalc=true;
        UpdateData(false);
    }
}

void CControlMessageDlg::OnButtonmultiplication()
{
    // TODO: Add your control notification handler code here
    if (Calc)
    {
        m_strResult.Format("%f", m_dFirst*m_dSecond);
        m_cstrStaticResult="умножения";
        ResCalc=true;
        UpdateData(false);
    }
}
```

```
void CControlMessageDlg::OnButtondivision()
{
    // TODO: Add your control notification handler code here
    CString str;
    if (Calc)
    {
        if (m_dSecond)
        {
            m_strResult.Format("%f", m_dFirst/m_dSecond);
            m_cstrStaticResult="деления";
        }
        else
        {
            (GetDlgItem(IDC_EDITFIRSTOPERAND)->GetWindowText(str);
            m_cstrStaticResult="»»;
            if (atof((LPCTSTR)str)<0) m_strResult="-INF";
            else
            if (atof((LPCTSTR)str)>0) m_strResult="»+INF";
            else m_strResult="NaN";
        }
        ResCalc=true;
        UpdateData(false);
    }
}
```

Эти функции имеют как различия, так и много общего. Например, все они заканчиваются вызовом функции `UpdateData`. Это позволяет обеспечить идентичность "содержимого" элементов управления в окне диалога и значений соответствующих им переменных.

Переменные `m_dFirst` и `m_dSecond` в программе имеют тип `double`. В них хранятся значения первого и второго операндов соответственно. Переменные `m_strResult` и `m_cstrStaticResult` являются объектами класса `CString` и хранят соответственно содержимое окна редактора `IDC_EDITRESULT` и элемента управления `IDC_STATICRESULT`. Напомню, что у элемента управления `Group Box`, расположенного в нижней части диалогового окна, в свойстве `Caption` указан текст "Результат". В начале работы приложения элемент управления `IDC_STATICRESULT` содержит пустую строку. При выполнении какого-либо арифметического действия за счет `IDC_STATICRESULT` можно рядом со словом "Результат" вывести одно из четырех слов: "сложения", "вычитания", "умножения", "деления". В связи с этим в каждом из приведенных обработчиков выполняется модификация содержимого переменной `m_cstrStaticResult`. В случае, если предпринята попытка выполнить операцию деления на ноль (обработчик `OnButtondivision`), в переменную `m_cstrStaticResult` заносится пустая строка.

Может возникнуть вопрос: "А почему для модификации содержимого `IDC_EDITRESULT` используется переменная `m_strResult` типа `CString`, а не переменная типа `double`?" Ведь `m_dFirst` и `m_dSecond` имеют именно такой тип. Дело в том, что выполняемое действие может быть делением, а второй операнд может оказаться в этот момент нулем. Тогда в `IDC_EDITRESULT` следует отобразить величину `+INF` (в случае положительного первого операнда), либо `-INF` (в случае отрицательного первого операнда), либо `NAN` (если будет попытка вычислить `0/0`). Указанные величины `+INF`, `-INF` и `NAN` широко используются в програм-



мировании для обозначения соответственно бесконечности (infinity в переводе с английского означает бесконечность) и нечислового характера результата (NaN — not a number). Часто используют две величины NaN: +NaN и -NaN. Мы в программе не будем учитывать знак перед NaN. Чтобы обеспечить преобразование числового результата типа double в переменную типа CString, в обработчиках использован метод Format класса CString.

В приведенных функциях применяется член класса CControlMessageDlg — элемент данных ResCalc. Элемент данных ResCalc принимает значение true, если только что была нажата одна из кнопок, позволяющих выполнить вычисления, и false, если произошла корректировка значения хотя бы одного из операндов. Для обработки событий, связанных с действиями пользователя по корректировке значений первого или второго операндов, необходим еще один обработчик. Именно один, так как при корректировке содержимого IDC_EDITFIRSTOPERAND и IDC_EDITSECONDOPERAND выполняются фактически одинаковые действия. Для того чтобы и при корректировке первого, и при корректировке второго операндов вызывался один и тот же обработчик (назовем его EditChange), то есть и извещение о корректировке значения первого операнда, и извещение о корректировке второго операнда направлялись в один и тот же обработчик, выполним следующие действия.

В той части карты сообщений, которая расположена в файле CControlMessageDlg.h, перед макросом DECLARE_MESSAGE_MAP(), но после строки //}}AFX_MSG, вставим такую строку:

```
afx_msg void EditChange();
```

В файле CControlMessageDlg.cpp, в карте сообщений, после строки //}}AFX_MSG_MAP, но перед макросом END_MESSAGE_MAP(), добавим следующий макрос

```
ON_CONTROL_RANGE(EN_CHANGE, IDC_EDITFIRSTOPERAND,
IDC_EDITSECONDOPERAND, EditChange).
```

Тогда можно в файл CControlMessageDlg.cpp добавить следующую функцию

```
void CControlMessageDlg::EditChange()
{
    CString str1, str2;
    GetDlgItem(IDC_EDITFIRSTOPERAND)->GetWindowText(str1);
    GetDlgItem(IDC_EDITSECONDOPERAND)->GetWindowText(str2);
    if (((str1!="0") && (atoi((LPCTSTR)str1)==0))) || ((str2!="0")
    && (atoi((LPCTSTR)str2)==0))) Calc=false;
    else {Calc=true; UpdateData(true);}
    if (ResCalc)
    {
        m_strResult.Empty();
        m_cstrStaticResult="";
        ResCalc=false;
        UpdateData(false);
        GetDlgItem(IDC_EDITFIRSTOPERAND)->SetWindowText(str1);
        GetDlgItem(IDC_EDITSECONDOPERAND)->SetWindowText(str2);
        m_EditFirst.SetSel(str1.GetLength(), str1.GetLength());
        m_EditSecond.SetSel(str2.GetLength(), str2.GetLength());
    }
}
```

Команды и извещения элементов управления

Известно, что командные сообщения поступают от таких объектов как меню, акселераторы и кнопки панелей инструментов. Их в нашем приложении нет. Но дело в том, что при нажатии кнопки типа Button формируется сообщение BN_CLICKED, воспринимаемое системой и обрабатываемое как команда. Кроме того, сообщение WM_COMMAND фор-

мируется в случае диалогового приложения при нажатии клавиш <Enter> и <Esc>. В случае клавиши <Enter> в параметре wParam командного сообщения может быть указан либо код IDOK, либо идентификатор элемента управления, являющегося по умолчанию так называемой нажимающейся кнопкой (push button). В случае клавиши <Esc> в wParam будет указан идентификатор IDCANCEL.

Мы рассмотрим фильтрацию и обработку командных сообщений и извещений элементов управления с использованием функции OnCmdMsg. Эта виртуальная функция применяется для маршрутизации и диспетчеризации командных сообщений. Функция OnCmdMsg определена в классе CCmdTarget, являющемся базовым для классов, которые должны обеспечивать возможность обработки сообщений. Наследником класса CCmdTarget является класс CDialog (он наследник класса CWnd, а CWnd — класс, производный от CCmdTarget), и следовательно, он также имеет возможность вызывать виртуальную функцию OnCmdMsg. Эта функция может быть переопределена. Ее прототип имеет вид (он описан в файле AFXWIN.h):

```
virtual BOOL OnCmdMsg(UINT nID, int nCode, void* pExtra,
AFX_CMDHANDLERINFO* pHandlerInfo);
```

Функция содержит четыре параметра:

- идентификатор команды nID;
- код команды извещения nCode;
- родовой указатель pExtra, значение которого зависит от значения второго параметра (nCode);
- указатель (обычно этот параметр должен быть NULL) на структуру AFX_CMDHANDLERINFO. Тип AFX_CMDHANDLERINFO так определен в файле AFXPRIV.h:

```
struct AFX_CMDHANDLERINFO
{
    CCmdTarget* pTarget;
    void (AFX_MSG_CALL CCmdTarget::*pmf)(void);
};
```

Если pHandlerInfo не равен NULL, функция OnCmdMsg вместо выполнения диспетчеризации команды заполняет необходимой информацией поля структуры, на которую указывает pHandlerInfo.

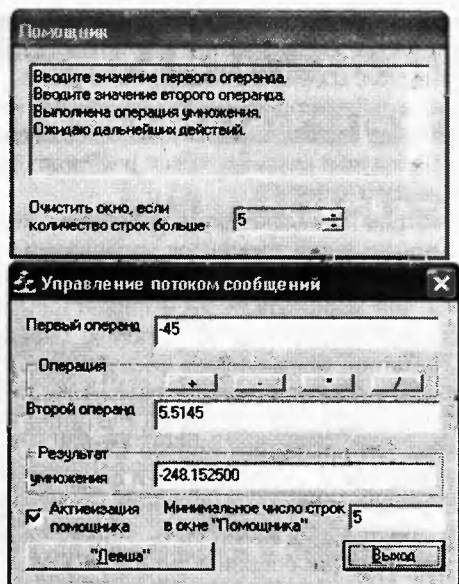
Во время выполнения программы соответствующим образом переопределенную функцию OnCmdMsg можно использовать для фильтрации потока сообщений и "распределения" сообщений в соответствующие объекты. Параметр nCode может принимать ряд значений, среди которых величины, задаваемые символическими константами CN_COMMAND, CN_EVENT, CN_UPDATE_COMMAND_UI и т.д. Обработка сообщений, не обработанных внутри переопределенной функции OnCmdMsg, выполняется за счет вызова соответствующего метода OnCmdMsg базового или корневого класса (root class). Функция OnCmdMsg возвращает ноль, если сообщение не было обработано.

Чтобы продемонстрировать использование OnCmdMsg, внесем изменения в наше приложение:

1. Добавим в окно приложения кнопку (рис.2), позволяющую задавать один из двух возможных режимов работы мыши:

- работает пользователь, для которого наиболее удобно работать левой рукой (режим "Левша");
- работает пользователь, для которого наиболее удобно работать правой рукой (режим "Правша") (подобное изменение в приложение мы внесем лишь для демонстрации использования процессов фильтрации и корректировки сообщений, так как обеспечить необходимую configura-

Рис. 2



цию кнопок мыши можно и за счет настроек Windows). На кнопке будем выводить надпись, из которой можно узнать, к какому режиму работы мыши можно перейти, нажав на эту кнопку.

2. Создадим еще одно окно диалога с идентификатором `IDD_DIALOGERROR`. Его предназначение — комментировать действия пользователя, выполненные им в главном окне приложения, и сообщать об ошибочных действиях пользователя. Таким действием может быть, например, попытка деления на ноль. В этом случае дополнительное окно «поможет» пользователю узнать о его ошибке. Поскольку это окно производит такую «неоценимую помощь», назовем его «Помощник». Это диалоговое окно сделаем немодальным. В него поместим окно многострочного редактора `IDC_EDITERROR`, в которое будем выводить сообщения о работе приложения. Также в окно «Помощника» поместим элемент управления типа `Static Text` с идентификатором `IDC_STATICERROR`. Он будет использоваться для отображения текста «Очистить окно, если количество строк больше». И наконец, разместим в окне «Помощника» окно однострочного редактора `IDC_EDITLIMITSTRING` и элемент управления типа `Spin` с идентификатором `IDC_SPINLIMITSTRING`. Элемент управления `IDC_EDITLIMITSTRING` будет выступать по отношению к `IDC_SPINLIMITSTRING` в качестве окна сопровождения. Элементы управления `IDC_EDITLIMITSTRING` и `IDC_SPINLIMITSTRING` будем использовать для регулирования допустимого количества строк в `IDC_EDITERROR`. Окно «Помощника» изображено на рис.2 вместе с основным окном приложения.

3. В основное окно приложения добавим флажок `IDC_CHECKDIALOGERROR`, у которого в свойство `Caption` введем текст «Активизация помощника». Установка этого флажка будет приводить к созданию и выводу на экран окна «Помощника», сброс флажка — к разрушению окна «Помощника». Кроме того, в основное окно приложения добавим элемент управления типа `Static Text`, содержащий в свойстве `Caption` текст «Минимальное число строк в окне «Помощника», и однострочный редактор типа `Edit Box`, с помощью которого при активном окне «Помощника» будем задавать минимально возможное количество строк, которое может быть отображено в `IDC_EDITERROR`. Идентификатором этого окна редактора будет `IDC_EDITQUANTITYSTRING`. Изменение значения, хранимого в `IDC_EDITQUANTITYSTRING`, при работающем «Помощнике», должно приводить к изменению нижней грани-

цы диапазона значений, которые можно задавать с помощью `IDC_EDITLIMITSTRING` и `IDC_SPINLIMITSTRING`.

Для функционирования окна «Помощника» потребуется создать класс `CDialogError`, являющийся потомком класса `CDialog`. Таким образом, в проект нужно добавить файлы `DialogError.h` и `DialogError.cpp`.

Функцию `OnCmdMsg` переопределим и в классе `CControlMessageDlg`, и в классе `CDialogError`. Текст функции `CControlMessageDlg::OnCmdMsg` находящийся в файле `ControlMessageDlg.cpp` имеет вид:

```
BOOL CControlMessageDlg::OnCmdMsg(UINT nID, int nCode,
void* pExtra, AFX_CMDHANDLERINFO* pHandlerInfo)
{
    // TODO: Add your specialized code here and/or call the base class
    if ((bHelpError) && (nID!=IDOK))
    {
        if ((nID==IDC_BUTTONDIVISION) && (!m_dSecond))
            DlgErr.iCodeError=1;
        else
            if ((nID==IDC_BUTTONPLUS) || (nID==IDC_BUTTONMINUS) ||
                (nID==IDC_BUTTONMULTIPLICATION) ||
                (nID==IDC_BUTTONDIVISION))
                DlgErr.iCodeError=0;
            else if (nID!=IDC_BUTTONMOUSE) DlgErr.iCodeError=-1;
            else
            {
                if (typeMouseButton==Left)
                    DlgErr.iCodeError=2;
                else
                    DlgErr.iCodeError=3;
            }
        if (DlgErr.OnCmdMsg(nID, nCode, pExtra, pHandlerInfo)) return true;
    }
    return CDialog::OnCmdMsg(nID, nCode, pExtra, pHandlerInfo);
}
```

Текст функции `CDialogError::OnCmdMsg`, находящейся в файле `DialogError.cpp`, имеет вид:

```
BOOL CDialogError::OnCmdMsg(UINT nID, int nCode, void* pExtra,
AFX_CMDHANDLERINFO* pHandlerInfo)
{
    // TODO: Add your specialized code here and/or call the
    base class
    if ((nID==IDC_BUTTONPLUS) || (nID==IDC_BUTTONMINUS) ||
        (nID==IDC_BUTTONMULTIPLICATION) ||
        (nID==IDC_BUTTONDIVISION) ||
        (nID==IDC_BUTTONMOUSE))
    {
        UpdateData(true);
        if (iQuantityString+1>m_iEditLimitString)
        {
            m_cstrEditError="Ожидаю дальнейших действий.\xd\xa";
            iQuantityString=1;
            UpdateData(false);
        }
        switch (iCodeError)
        {
        case 1: {m_cstrEditError.Delete(m_cstrEditError.GetLength()-
            sizeof("Ожидаю дальнейших действий.")-1,
            sizeof("Ожидаю дальнейших действий.")+1);
            UpdateData(false);
            m_cstrEditError+="Делить на ноль нельзя!\xd\xa";
            m_cstrEditError+="Ожидаю дальнейших действий.\xd\xa";
            iCodeError=-1;
            iQuantityString+=1;
            UpdateData(false);
            return true; break;
        }
        case 0:
        case 2:
        case 3: {
            m_cstrEditError.Delete(m_cstrEditError.GetLength()-
                sizeof("Ожидаю дальнейших действий.")-1,
                sizeof("Ожидаю дальнейших действий.")+1);
            UpdateData(false);
            switch (nID)
            {
            case IDC_BUTTONPLUS: m_cstrEditError+="Выполнена
                операция сложения.";
                break;
            case IDC_BUTTONMINUS: m_cstrEditError+="Выполнена
                операция вычитания.";
```



Программа для поиска и замены

(Окончание. Начало в №№8-10/2005)

FAQ

В процессе работы с программой у вас могут возникать различные вопросы. Да что там, иногда вопросы возникают даже у меня, хоть я и сам написал эту программу :-). В этом разделе я попытаюсь ответить на некоторые из них.

Вопрос. Я точно знаю, что обрабатываемый файл содержит искомые последовательности, но программа почему-то ничего в нем не заменяет (никакого сообщения об ошибке также не появляется). Почему?

Ответ. Вот некоторые возможные причины:

- программа запущена в режиме "read only";
- неверно указано имя ini-файла или обрабатываемого файла (если файл с указанным именем существует, то программа действительно может не выдать никакого сообщения об ошибке);

- может быть, вы сами в отладочных целях закоментировали в ini-файле какие-то пары последовательностей, а потом забыли раскомментировать;

- возможно, при работе программы какая-то ошибка все-таки была, но сообщения о ней вы не увидели. Например, если программа была запущена из bat-файла без дальнейшей проверки кода ее завершения, то при выполнении следующих команд bat-файла текст сообщения об ошибке может уйти за край экрана.

Посмотрите также следующие вопросы и ответы на них.

Вопрос. Мне нужно заменить в файле некоторые последовательности, содержащие русские буквы (эти последовательности заданы в ini-файле в виде строк), но программа почему-то их не заменяет (в то же время другие последовательности заменяются успешно). Почему?

Ответ. Скорее всего, не совпадают кодировки ini-файла и обрабатываемого файла. Кстати, замечу, что если вы указываете "cp" вместо имени ini-файла и затем вводите его содержимое с клавиатуры, то оно будет в альтернативной кодировке DOS (CP866).

Вопрос. Почему-то не происходит замена последовательностей, содержащих русские буквы, если я указываю, что при их поиске не должен учитываться регистр букв. Почему? Я уже проверил, что кодировки ini-файла и обрабатываемого файла совпадают.

Ответ. Скорее всего, в первой строке ini-файла не указана его кодировка, или указана неверная кодировка. Напоминаю — не указывать кодировку можно, только если это альтернативная кодировка DOS (CP866).

Вопрос. Мне нужно заменить в файле все последова-

тельности "х???", где "???" — любые байты, на "у???", где "???" — те же байты, что соответствовали "???" в заменяемой последовательности. Я создал следующий ini-файл:

```
'х???'
```

```
'у???'
```

Но программа не выполняет замену. Почему?

Ответ. Как я уже упоминал в разделе "Формат ini-файла", символы "?" в строке не являются подстановочными символами "?". Правильный ini-файл должен быть таким:

```
'х',???
```

```
'у',???
```

Вопрос. Мне нужно заменить "News" на "Новости", а "News Archive" — на "Архив новостей". Я создал для этого такой ini-файл:

```
'News'
```

```
'Новости'
```

```
'News Archive'
```

```
'Архив новостей'
```

Но при его использовании не происходит замена "News Archive" на "Архив новостей", как если бы соответствующая пара последовательностей вообще не была описана в ini-файле. Почему?

Ответ. Потому что перебор искомых последовательностей происходит именно в порядке их перечисления в ini-файле, пока не будет обнаружено совпадение. Поэтому, как только программа встречает в обрабатываемом файле "News", сразу происходит замена на "Новости", а до проверки совпадения со второй искомой последовательностью дело так и не доходит. Правильный ini-файл должен быть таким:

```
'News Archive'
```

```
'Архив новостей'
```

```
'News'
```

```
'Новости'
```

Вопрос. Мне надо заменить в текстовом файле последовательность, в которой есть несколько пробелов подряд, но программа не выполняет эту замену. Почему?

Ответ. Скорее всего, причина в следующем. В текстовом файле последовательность пробелов не всегда бывает представлена именно пробелами (ASCII-код 32). Для этого также может использоваться табуляция (ASCII-код — 9) или какая-либо комбинация пробелов и табуляций (это зависит от количества пробелов и от их расположения относительно начала строки файла). А так как программа всегда рассматривает обрабатываемый файл как двоичный и не выполняет

```
break;
case DC_BUTTONMULTIPLICATION:m_cstrEditError+="Выполнена
                                операция умножения.";
                                break;
case IDC_BUTTONDIVISION:m_cstrEditError+="Выполнена
                                операция деления.";
                                break;
case IDC_BUTTONMOUSE:
{m_cstrEditError+="Переход к режиму";
 if (iCodeError==3)
m_cstrEditError+="\"Левша\"";
 else
m_cstrEditError+="\"Правша\"";
 m_cstrEditError+=" выполнен.";
```

```
}
}
//\xd\xa - переход на новую строку
m_cstrEditError+="\xd\xaОжидаю дальнейших действий.\xd\xa";
iQuantityString+=1;
iCodeError=-1;
UpdateData(false);
}
}
return CDialog::OnCmdMsg(nID, nCode, pExtra,
pHandlerInfo);
}
```

(Окончание следует)



замену табуляций на эквивалентное число пробелов (в отличие от чтения ini-файла, когда такое преобразование выполняется), то искомая последовательность в таком случае не будет найдена и заменена.

Возможен такой выход из положения — преобразовать файл, заменив в нем все табуляции на эквивалентное число пробелов (это можно сделать, скажем, с помощью Dos Navigator, используя встроенный редактор), а потом уже выполнять замену.

Вопрос. У меня есть текстовый файл в кодировке UTF-8 (где, как известно, символ может быть представлен несколькими байтами), и в этом файле мне надо заменить одни строки на другие. Можно ли это сделать с помощью описанной программы?

Ответ. Да, можно (ini-файл, конечно, тоже должен быть в кодировке UTF-8). Но при этом нельзя использовать поиск без учета регистра букв и использовать для обозначения произвольного символа текста подстановочный символ "?" (ведь он соответствует только одному байту, а символ текста может занимать несколько байтов).

Интересно, что при создании программы возможность обработки текстов в кодировке UTF-8 вообще не предусматривалась. Эта возможность имеется лишь за счет того, что кодировка UTF-8 разработана очень удачно: во-первых, первые 128 Unicode-символов (соответствующие стандарту ASCII) в ней кодируются точно так же, как в кодировке ASCII, то есть байтами 0 – 127, а во-вторых, значение первого байта любого символа заведомо не может встретиться во втором, третьем и т.д. байтах любого многобайтового символа.

Вопрос. А нет ли какой-нибудь похожей программы, но с поддержкой регулярных выражений? Это надо мне для обработки текстовых файлов.

Ответ. Попробуйте воспользоваться программой "sed". Существуют ее версии для различных операционных систем. Информацию об этой программе можно найти, например, на сайте <http://sed.sourceforge.net>. Лично я пользуюсь расширенной версией "sed" под названием "super-sed"

("ssed"); скачать ее исполняемый файл (для Windows), а также исходные тексты и документацию можно на странице <http://sed.sourceforge.net/grabbag/ssed/>.

Кстати, сразу укажу на принципиальное различие между моей программой и "sed" — у меня все замены выполняются за один проход, а в "sed" при выполнении каждой команды, обрабатывающей текущий фрагмент текста, происходит новый проход по этому фрагменту.

Вопрос. А если мне надо производить поиск и замену с использованием регулярных выражений не только в текстовых файлах, но и в двоичных?

Ответ. Тогда попробуйте использовать Perl (www.perl.com). Ниже для примера приведена Perl-программа, заменяющая в двоичном файле "test.bin" байты 0x00 – 0x1F на байт 0x2E и выводящая количество выполненных замен.

```
#!/usr/bin/perl
```

```
my $c; # Содержимое обрабатываемого файла.
my $n; # Количество выполненных замен.
```

```
open (F,'<','test.bin') || die ("Can't open file! $!");
binmode (F);
sysread (F,$c,(-s F)) || die ("Read error! $!");
close (F) || die ("Close error! $!");
```

```
$n=($c=~s/[\x00-\x1F]/\x2E/g); # Непосредственно замена.
```

```
open (F,'>','test.bin') || die ("Can't open file! $!");
binmode (F);
syswrite (F,$c) || die ("Write error! $!");
close (F) || die ("Close error! $!");
```

```
print ("Выполнено замен: ", $n+0);
```

Литература

1. И.Рощин. Добавление в HTML-документы информации об их кодировке. — Радиомир. Ваш компьютер, 2003, №10, С.13.
2. Р.Галеев/ИИ-TECH. Сारा о кодировках. — Радиомир. Ваш компьютер, 2004, №№1-4.

А.КОРБИТ,

г.Минск, БГУИР, ФИТиУ, каф. ВмиГ
E-mail: a_korbit@mail.ru

Утечка памяти

Утечка памяти является одной из наиболее распространенных и трудно обнаруживаемых ошибок. Небольшие утечки памяти не проявляют себя до тех пор, пока приложение не будет запущено на компьютере с небольшим объемом оперативной памяти или параллельно с большим количеством других одновременно выполняемых проектов.

Так как обычно программисты стараются работать на ПК с большим объемом оперативной памяти и при проверке функционирования своего приложения не запускают на выполнение другие приложения, то такие ошибки, как правило, могут остаться незамеченными. Последствия утечек памяти могут быть достаточно серьезными и привести даже к краху операционной системы с необходимостью ее переустановки.

Практически у утечки памяти причина одна — где-то в приложении под некоторый объект выделяется динамическая область оперативной памяти, которая не освобождается после использования данного объекта. Эта причина имеет несколько конкретных проявлений, на которые программисту следует обратить внимание:

1. Указатель на объект создан оператором new и не освобожден оператором delete.
2. Указатель на объект создан оператором new[], а освобожден оператором delete, хотя нужно использовать оператор delete [].
3. Переменной, хранящей указатель на некоторый объект, присвоено новое значение, а объект не освобожден.
4. При броске исключения область памяти, выделенная динамически для переменных данной функции, не освобождена.

Ниже приведен листинг класса, в котором содержатся эти ошибки, и код программы, которая из-за них создает утечку памяти:

```
#include <iostream.h>
#include <conio.h>
class X
{
    int *m; // указатель на начало динамического массива
public:
    X(); // декларация конструктора
};

X::X() // код конструктора
{
    m=new int[5]; // захватили память для пяти элементов
    for(int i=0; i<5; i++) // проинициализировали
```



```

        *(m+i)=i;
        cout<<"array created"<<endl; // сообщили
    };
    X f1(void)
    {
        X a; // создали первый массив
        X b; // создали второй массив
        b=a; // скопировали содержимое
        return b; // функция возвращает второй массив
    };

```

```

void main(void)
{
    X a=f1(); // создали объект a и вызвали функцию f1()
}

```

Запускаем эту программу на исполнение и получаем:

```

array created
array created
Press any key to continue

```

На самом деле работает это так. При запуске функции `main()` вызывается функция `f1()`, в которой создаются два объекта класса `X`. В каждом из них переменной `m` присваивается значение указателя на некоторую область оперативной памяти, отведенную для хранения пяти целых чисел. Так как в данном классе отсутствует перегруженный оператор присваивания, то операция присваивания одного объекта данного класса другому трактуется как побитовое копирование значений соответствующих переменных одного объекта переменным другого объекта. При этом переменные `m` обоих объектов будут указывать на одну и ту же область памяти, зарезервированную для объекта `a`. Указатель на другую область памяти будет потерян.

Оператор `return` вызывает конструктор, который по умолчанию выполняет операцию побитового копирования переменных исходного объекта класса в создаваемый объект. Этот же конструктор используется и при создании объекта `a` в функции `main()`.

Включение в данный класс деструктора, уничтожающего указатель на целочисленный массив, приведет к появлению явных ошибок, так как при завершении работы функции `f1()` будут вызваны два деструктора. Они попытаются освободить одну и ту же область оперативной памяти. Хотя это и не приведет к остановке программы, но при инициализации объекта `a` в функции `main()` будет использован некорректный указатель на захватываемую область памяти.

Для устранения этих ошибок класс `X` и сам код программы нужно переделать следующим образом:

```

#include <iostream.h>
#include <conio.h>
class X
{
    int *m;
public:
    X();
    X(X &); // декларация конструктора копирования
    ~X(); // декларация деструктора
    void operator=(X &); // функция оператор перегрузки
                        // операции присваивания
};

X::X()
{
    m=new int[5];
    for(int i=0; i<5; i++)
        *(m+i)=i;
    cout<<"array created"<<endl;
}

```

```

};
X::X(X &r) // код конструктора копирования
{
    m=new int[5]; // захватили нужный объем памяти
    for(int i=0; i<5; i++) // скопировали первый
                        // объект во вновь созданный
        m[i]=r.m[i];
    cout<<"array copywrited"<<endl;
};
void X::operator=(X &x) // код функции оператора
                        // перегрузки операции присваивания
{
    if (this-&x) return;
    delete m;
    m=x.m;
    x.m=NULL;
};
X::~X() // декларация деструктора
{
    if (m) delete m;
    cout<<"array destroyed"<<endl;
}
X f1(void)
{
    X a;
    X b;
    b=a;
    return b;
};

```

```

void main(void)
{
    X a=f1();
    getch();
}

```

Запускаем откорректированную программу на исполнение и получаем:

```

array created
array created
array copywrited
array destroyed
array destroyed

```

Особенности приведенного программного кода.

1. В деструкторе класса `X` использована операция `delete`, хотя по теории необходимо использовать операцию `delete[]`. Различия в этих операциях при уничтожении массивов трудно уловимы. А в некоторых случаях использование `delete[]` приводит к возникновению ошибок. Правильный выбор нужной операции определяется "методом научного тыка".

2. Проверка в деструкторе значения указателя `m` перед его уничтожением является простой формальностью, так как освобождение нулевого указателя не приводит ни к чему плохому. Т.е. использование данной операции является своего рода "правилом хорошего тона".

3. Проверка значения указателя перед его уничтожением может потребоваться тогда, когда этот указатель содержит адрес объекта класса, который имеет непустой деструктор, который, в свою очередь, может быть создан, а может и не быть создан в процессе работы с исходным классом.

4. Вызов деструктора для объекта класса, который не был создан, может привести к непредсказуемым последствиям. Поэтому в конструкторе пользовательского класса необходимо присвоить нулевое значение всем указателям на содержащиеся в нем объекты классов. И в деструкторе нужно проверять значение указателя на объект перед его уничтожением.



5. Если в процессе работы с пользовательским классом уничтожается указатель на один из содержащихся в нем объектов, то данному указателю следует присвоить нулевое значение.

6. В конструкторе копирования класса X создается указатель на новую область оперативной памяти, в который и копируется содержимое массива исходного объекта класса. Это позволяет впоследствии использовать оба объекта класса.

7. Для разнообразия оператор присваивания переопределен иначе. После его применения нельзя использовать исходный объект класса. Данный метод обычно используется совместно с методом, который вызывается в конструкторе копирования.

8. Условный оператор, расположенный в начале функции переопределения операции присваивания, имеет очень большое значение, особенно при используемом методе реализации данного оператора. Он обеспечивает возможность корректной обработки операции присваивания объекта самому себе.

Для предотвращения возможности возникновения утечек памяти следует при программировании следовать некоторым правилам.

1. Не нужно использовать динамическую память, если создаваемый объект используется только в данной функции.
2. Если использование динамической памяти неизбежно, нужно:

- создать класс для работы с этой памятью;
- в класс нужно включить копирующий конструктор, деструктор, освобождающий всю используемую данным классом память, и перегрузить операцию присваивания для основных типов данных, которые будут использоваться с данным классом.

3. Если функция, захватывающая память, возвращает значение, позволяющее получить доступ к данной памяти, то это значение не должно быть ссылкой, поскольку освободить память по ссылке невозможно.

4. Никогда не изменяйте значения указателя на **захваченную область памяти**, пока эта область не будет освобождена.

5. Не используйте операцию приращения для указателя, полученного при помощи оператора new

Литература

1. Н. Секунов. Самоучитель Visual C++. — СПб.: BHV — Санкт-Петербург, 2000.

Низкоуровневый доступ к диску в Win 2000/XP

С. РУСАКОВ,

г. Тюмень, ТГУ.

E-mail: sergey_rus@bk.ru

Иногда необходимо получить доступ к жесткому диску на физическом уровне, т.е. независимо от файловой системы считывать и записывать данные непосредственно на диск или дискету. Такой задачей может быть создание "ключевой дискеты", защищенной от копирования, или скрытая передача данных. В данной статье речь пойдет как раз о скрытии данных и о способе низкоуровневого доступа к дискам на Visual Basic.

Сначала разберем методику скрытия информации в Win 2000/XP.

Как известно, каждая дорожка по окружности делится на секторы. Программы записывают в любой сектор информацию объемом не более 512 байтов. Несколько смежных секторов диска объединяются в группу, называемую кластером. Кластер — это минимальная "порция" дискового пространства, которая может быть выделена файлу. Один кластер для дискет 3,5 дюйма обычно содержит один сектор. Если файл не помещается в кластер, то ему выделяется еще один, и т.д. То есть, даже если размер файла — 1 байт, то под него будет отведено 512 байтов. А что же остальные байты кластера? Они просто не используются! Это легко проверить, создав на дискете файл размером один байт и посмотрев его свойства — там указано, что на диске он занимает 512 байтов. Если вы удалите файл, а на его место запишете новый, то перезапишется только часть кластера, а остальные данные, которые там были, останутся нетронутыми. Этим фактом легко воспользоваться для сокрытия своей информации.

Рассмотрим случай, когда мы хотим записать информацию в файл на дискете (имя файла должно быть меньше или равно 8 байтов). Итак, задачи, которые предстоит выполнить для записи информации:

- найти структуру данных, описывающую файл (имя, расширение, тип, размер, положение и т.п.);
- считать из нее размер файла и номер кластера, содержащего начало файла или каталога;
- проследить в таблице размещения файлов (FAT) цепочку кластеров и определить последний из них;
- непосредственно записать данные на свободное место

то в кластере.

Это сложный и "правильный" способ.

Второй способ проще:

- записать в файл информацию встроенными средствами VB;

- найти структуру данных, описывающую файл (имя, расширение, тип, размер, положение и т.п.);

- искусственно "уменьшить" размер, записав соответствующее значение в соответствующее поле в данной структуре.

Мы рассмотрим второй способ. Для простоты будем считать, что записываемая нами информация помещается в один остаток. Если это не так, то придется немного изменить код для получения записанной информации. Для чтения информации придется выполнить все те же шаги, кроме последнего. Вместо него мы будем считывать информацию... Итак, по порядку.

Чтобы найти нужную нам структуру данных, необходимо ознакомиться с логической структурой диска.

Первый сектор на самой внешней дорожке (загрузочный сектор) содержит важную информацию о логической структуре диска. Начиная с байта с порядковым номером 12, в загрузочном секторе размещается так называемый блок параметров BIOS (BPB), содержащий информацию о длине и расположении остальных логических частей диска и некоторые другие важные параметры.

В табл.1 приведена структура BPB.

Для любых ОС, совместимых с MS-DOS, количество байтов в секторе всегда равно 512.

За загрузочным сектором располагается таблица размещения файлов. Она начинается в секторе ressecs и содержит информацию о принадлежности каждого кластера тому или иному файлу. Обычно на диске создается несколько копий FAT (обычно две), содержимое которых идентично. Дублирование информации в нескольких копиях FAT повышает надежность ее хранения. Количество копий FAT и размер каждой копии (в секторах) задаются полями Fatcnt и Fatsize блока BPB. Сразу за последней копией FAT на диске располага-



Табл. 1

Смещение	Описание (имя переменной в программе)	Размер в байтах
11	Количество байтов в секторе (Sectsize)	2
13	Количество секторов в кластере (Clustsize)	1
14	Количество секторов перед FAT (Ressecs)	2
16	Количество копий FAT (Fatcnt)	1
17	Количество элементов корневого каталога (Rootsize)	2
19	Количество секторов на диске (Tot_sect)	2
21	Дескриптор носителя	1
22	Количество секторов в FAT	2

ется корневой каталог. Содержимое каталога описывает все зарегистрированные в нем подкаталоги и файлы. Каждый элемент корневого каталога и всех подкаталогов представлен 32-байтной структурой данных, представленной в табл. 2.

Для нас представляют ценность имя файла, его расширение и размер. Следует сказать, что если имя файла меньше 8 байтов, то недостающие справа байты заполняются пробелами.

Мы будем уменьшать размер файла. Итак, приступим.

Сначала создадим модуль с именем Module1, в котором объявим процедуры и функции для работы с дискетой напрямую.

Option Explicit

```
Private DriveToProcess As String
Public BytesPerSector As Integer
Private SectorsPerCluster As Byte
```

```
Public Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory"
(Destination As Any, Source As Any, ByVal Length As Long)
```

```
Private Const GENERIC_READ = &H80000000
Private Const GENERIC_WRITE = &H40000000
Private Const FILE_SHARE_READ = &H1
Private Const FILE_SHARE_WRITE = &H2
Private Const OPEN_EXISTING = 3
Private Const INVALID_HANDLE_VALUE = -1&
```

```
Private Declare Function CreateFile Lib "kernel32" Alias
"CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess
As Long, ByVal dwShareMode As Long, ByVal lpSecurityAttributes As
Long, ByVal dwCreationDisposition As Long, ByVal
dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long
Private Declare Function CloseHandle Lib "kernel32" (ByVal hObject
As Long) As Long
Private Declare Function SetFilePointer Lib "kernel32" (ByVal
hFile As Long, ByVal lDistanceToMove As Long, lpDistanceToMoveHigh
As Long, ByVal dwMoveMethod As Long) As Long
Private Declare Function ReadFile Lib "kernel32" (ByVal hFile As
Long, lpBuffer As Any, ByVal nNumberOfBytesToRead As Long,
lpNumberOfBytesRead As Long, ByVal lpOverlapped As Long) As Long
Private Declare Function WriteFile Lib "kernel32" (ByVal hFile As
Long, lpBuffer As Any, ByVal nNumberOfBytesToWrite As Long,
lpNumberOfBytesWritten As Long, ByVal lpOverlapped As Long) As Long
Private Declare Function FlushFileBuffers Lib "kernel32" (ByVal
hFile As Long) As Long
Private Declare Function LockFile Lib "kernel32" (ByVal hFile As
Long, ByVal dwFileOffsetLow As Long, ByVal dwFileOffsetHigh As
Long, ByVal nNumberOfBytesToLockLow As Long, ByVal
```

```
nNumberOfBytesToLockHigh As Long) As Long
Private Declare Function UnlockFile Lib "kernel32" (ByVal hFile As
Long, ByVal dwFileOffsetLow As Long, ByVal dwFileOffsetHigh As
Long, ByVal nNumberOfBytesToUnlockLow As Long, ByVal
nNumberOfBytesToUnlockHigh As Long) As Long
Private Const FILE_BEGIN = 0
```

```
Public b1 As Byte, b2 As Byte, b3 As Byte, b4 As Byte
Public Function DirectReadDriveNT(ByVal sDrive As String, ByVal
iStartSec As Long, ByVal iOffset As Long, ByVal cBytes As Long) As
Variant
```

```
Dim hDevice As Long
Dim abBuff() As Byte
Dim abResult() As Byte
Dim nSectors As Long
Dim nRead As Long
nSectors = Int((iOffset + cBytes - 1) / BytesPerSector) + 1
hDevice = CreateFile("\\.\\" & UCase(Left(sDrive, 1)) & ":",
GENERIC_READ Or GENERIC_WRITE, FILE_SHARE_READ Or
FILE_SHARE_WRITE, 0, OPEN_EXISTING, 0, 0)
If hDevice = INVALID_HANDLE_VALUE Then Exit Function
Call SetFilePointer(hDevice, iStartSec * BytesPerSector, 0,
FILE_BEGIN)
ReDim abResult(cBytes - 1)
ReDim abBuff(nSectors * BytesPerSector - 1)
Call ReadFile(hDevice, abBuff(0), UBound(abBuff) + 1, nRead, 0)
CloseHandle hDevice
CopyMemory abResult(0), abBuff(iOffset), cBytes
DirectReadDriveNT = abResult
End Function
```

```
Public Function DirectWriteDriveNT(ByVal sDrive As String, ByVal
iStartSec As Long, ByVal iOffset As Long, ByVal sWrite As String)
As Boolean
```

```
Dim hDevice As Long
Dim abBuff() As Byte
Dim ab() As Byte
Dim nRead As Long
Dim nSectors As Long

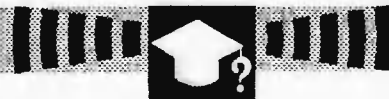
nSectors = Int((iOffset + Len(sWrite) - 1) / BytesPerSector) + 1
hDevice = CreateFile("\\.\\" & UCase(Left(sDrive, 1)) & ":",
GENERIC_READ Or GENERIC_WRITE, FILE_SHARE_READ Or
FILE_SHARE_WRITE, 0, OPEN_EXISTING, 0, 0)
If hDevice = INVALID_HANDLE_VALUE Then Exit Function
abBuff = DirectReadDriveNT(sDrive, iStartSec, 0, nSectors *
BytesPerSector)
ab = StrConv(sWrite, vbFromUnicode)
CopyMemory abBuff(iOffset), ab(0), Len(sWrite)
Call SetFilePointer(hDevice, iStartSec * BytesPerSector, 0,
FILE_BEGIN)
Call LockFile(hDevice, FWord(iStartSec * BytesPerSector),
SWord(iStartSec * BytesPerSector), FWord(nSectors *
BytesPerSector), SWord(nSectors * BytesPerSector))
DirectWriteDriveNT = WriteFile(hDevice, abBuff(0),
UBound(abBuff) + 1, nRead, 0)
Call FlushFileBuffers(hDevice)
Call UnlockFile(hDevice, FWord(iStartSec * BytesPerSector),
SWord(iStartSec * BytesPerSector), FWord(nSectors *
BytesPerSector), SWord(nSectors * BytesPerSector))
CloseHandle hDevice
End Function
```

```
Public Function SWord(ByVal dw As Long) As Integer
CopyMemory SWord, ByVal VarPtr(dw) + 2, 2
End Function
```

```
Public Function FWord(ByVal dw As Long) As Integer
CopyMemory FWord, dw, 2
End Function
```

```
Public Function Size_To_Byte(dw As Long)
CopyMemory b1, ByVal VarPtr(dw), 2
CopyMemory b2, ByVal VarPtr(dw) + 1, 2
CopyMemory b3, ByVal VarPtr(dw) + 2, 2
CopyMemory b4, ByVal VarPtr(dw) + 3, 2
End Function
```

Основные функции — DirectReadDriveNT и DirectWriteDriveNT. Параметры каждой функции — буква диска, сектор, смещение. Для чтения еще указывается размер считываемых данных, а для записи — строка, которую будем записывать. Функция Fword



возвращает первое слово числа `dw`, а `SWord` — второе. Функция `Size_To_Byte` разбивает число для записи на четыре байта. Теперь можно приступить непосредственно к кодировке — будем дописывать информацию в файл `1.txt`. Потом вы можете изменить его имя. Создаем форму, на нее ставим кнопку `Command1`. В код формы пишем такие строки:

```
Dim ressecs As Long
Dim fatsize As Long
Dim fatcnt As Byte
Dim sectsize As Long
Dim tot_sect As Long
Dim FAT() As Byte
Dim MainDan() As Byte
Dim clustsize As Long
Dim rootsize As Long
Dim SectNu As Long
Dim DataStart As Long
Dim DATA() As Byte
Dim i As Long
```

```
Public Sub Get_Main_Data()
DATA() = DirectReadDriveNT("A:", 0, 11, 2)
Call CopyMemory(sectsize, DATA(0), 2)
```

```
DATA() = DirectReadDriveNT("A:", 0, 16, 1)
Call CopyMemory(fatcnt, DATA(0), 1)
```

```
DATA() = DirectReadDriveNT("A:", 0, 19, 2)
Call CopyMemory(tot_sect, DATA(0), 2)
```

```
DATA() = DirectReadDriveNT("A:", 0, 17, 2)
Call CopyMemory(rootsize, DATA(0), 2)
```

```
DATA() = DirectReadDriveNT("A:", 0, 13, 1)
Call CopyMemory(clustsize, DATA(0), 1)
```

```
DATA() = DirectReadDriveNT("A:", 0, 14, 2)
Call CopyMemory(ressecs, DATA(0), 2)
```

```
DATA() = DirectReadDriveNT("A:", 0, 22, 2)
Call CopyMemory(fatsize, DATA(0), 2)
End Sub
```

```
Public Sub Find_Get_File_Description()
Dim find As Boolean
```

```
i = 0
find = False
Do While find = False
i = i + 1
MainDan() = DirectReadDriveNT("A:", ressecs + fatsize *
fatcnt, 32 * i, 32)
If MainDan(0) = 49 And MainDan(1) = 32 Then find = True
Loop
End Sub
```

Первая из этих процедур читает необходимую для дальнейшей работы информацию из загрузочного сектора. Вторая — ищет файл `1.txt` на диске в разделе описаний (49 —

ASCII-код "1", 32 — пробела). Осталось только добавить код для кнопки:

```
Private Sub Command1_Click()
Dim fsizeold As Long
Dim stroka As String
```

```
BytesPerSector = 512
fsizeold = FileLen("A:\1.txt")
stroka = InputBox("Введите секретную строку")
stroka = stroka & Space((fsizeold \ 512 + 1) * 512 - Len(stroka))
```

```
Open "A:\1.txt" For Append As #1
Print #1, stroka
Close #1
```

```
Get_Main_Data
Find_Get_File_Description
Size_To_Byte (fsizeold)
Call DirectWriteDriveNT("A:", ressecs + fatsize * fatcnt, 32 *
i + 28, Chr(b1) & Chr(b2) & Chr(b3) & Chr(b4))
End Sub
```

```
Private Sub Command2_Click()
Dim fsizeold As Long
```

```
fsizeold = FileLen("A:\1.txt")
BytesPerSector = 512
Get_Main_Data
Find_Get_File_Description
Size_To_Byte ((fsizeold \ 512) * 512 + 512)
Call DirectWriteDriveNT("A:", ressecs + fatsize * fatcnt, 32 *
i + 28, Chr(b1) & Chr(b2) & Chr(b3) & Chr(b4))
End Sub
```

Все! Теперь наш файл полностью занимает кластер. Если вы заранее знаете, сколько информации будете в него писать, то можно изменить в значении `Size_To_Byte` размер на тот, который вам нужен, а затем спокойно программным образом считать данные. Если же нет, то можно это посмотреть вручную (после окончания данных идут одни пробелы).

Один из плюсов этого метода — если вы не будете изменять файл, в который записали информацию, то даже если полностью заполнить дискету какими-либо данными, ваша секретная строка не изменится.

Внимание! Если скопировать файл с дискеты на жесткий диск, а затем снова скопировать на дискету, то информация потеряется! И еще надо упомянуть о важном моменте — для работы вам необходимы права администратора системы!

В качестве домашнего задания предлагаю вам попробовать изменить программу так, чтобы она работала не только с файлом `"A:\1.txt"`, а вообще с любым заданным файлом. И чтобы можно было произвести чтение и запись любой по размеру строки.

Литература

1. В.В.Фаронов. TurboPascal 7.0. Практика программирования. Учебное пособие. — Нолидж, 1998.

Алгоритм решения японских кроссвордов

На страницах журнала тема решения японских кроссвордов уже затрагивалась, но рассказать о самих кроссвордах, видимо, все же надо.

Кроссворд сразу ассоциируется с чем-то, что надо разгадывать. В Европе японские кроссворды пришли в виде зашифрованных рисунков. Сегодня такого типа кроссворды называются японскими лишь по способу

представления зашифрованной картинки и методам их разгадывания. В этих головоломках используются не слова, как в классических европейских кроссвордах, а ряды чисел, посредством которых и ведется раскодирование образа в кроссворде.

Изображение в японском кроссворде выглядит как прямоугольник, разлинованный на клеточки, которые мо-

гут быть либо закрашенными, либо нет (в таких ячейках, как правило, ставят знак "X"). Слева или справа (кому как нравится) от поля кроссворда просят ставить циферки, но для расшифровки картинки этого никак не достаточно, поэтому подписываются не только строчки, но и столбцы (это также делается либо сверху над столбцом, либо снизу).

Е БОНДАРЕНКО,

г.Петропавловск, Казахстан.

E-mail: Ewg_sha@mail.ru



Вам, наверное, интересно, что это за загадочные числа везде написаны. Во-первых, циферки, как уже было упомянуто, кодируют изображение, но каждое конкретное число отвечает лишь за один блок (последовательность закрашенных клеточек без пропусков между ними), показывая его размер. Например, если вы видите число пять в одной из строк, то это значит, что надо закрасить пять подряд идущих ячеек в этой строке. Во-вторых, порядок следования чисел определяет последовательность размещения блоков в строке или столбце: снизу вверх для столбцов и слева направо для строк. т.е., если в столбце стоит число 3, а под ним 2, то это значит, что в этом столбце нужно закрасить (где-то) сверху 3 клеточки, а ниже под ними 2 клеточки (первый столбец на рис.1). То же относится и к строкам.

Между закрашенными клеточками обязательно должна присутствовать как минимум одна незакрашенная клетка. На рис.1 вы можете увидеть пример решенного японского кроссворда. Догадаться, что изображено?

Вот так, закрашивая и зачеркивая ячейки, можно получить очень реалистичные картинки. Но если бы, кроме перечисленных правил, не было других ограничений, многие кроссворды были бы практически неразрешимы, во всяком случае, для человека. Поэтому к этим головоломкам предъявляются еще требования:

- головоломка должна иметь единственное логическое решение (метод предположений в решении или его варианты недопустимы);
- в информационных полях не может быть нулей;
- сетка головоломки может иметь по горизонтали и вертикали только кратное пяти число клеток (например: 5, ..., 15, 20, 25, ...);
- изображение не должно быть симметричным и должно заключать в себе легко читаемый образ;
- цифры не могут быть равны количеству клеток в принадлежащих им линиях сетки (например, если в линии всего 20 клеток, недопустимо иметь в задании тоже "20", или, как варианты, "7" "42", "10" "9" и т.д.).

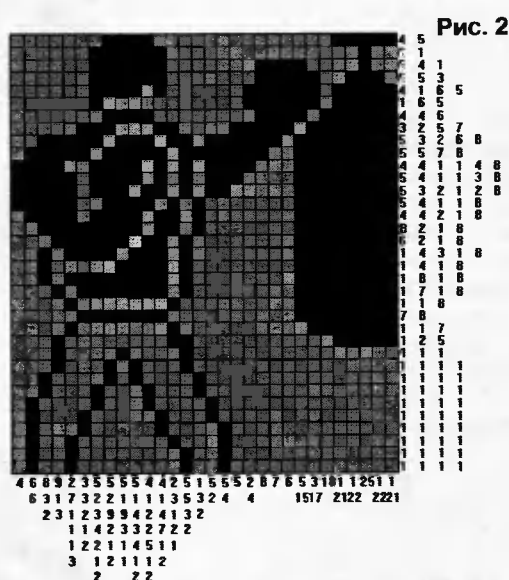
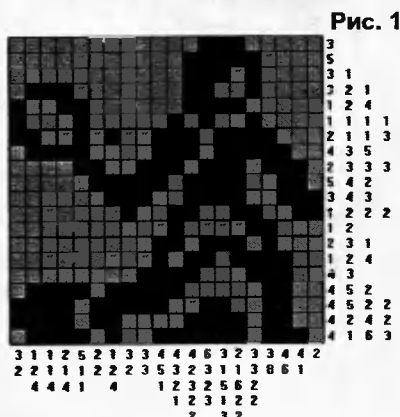
Несмотря на все эти требования к кроссвордам, решить их, надо сказать, не так просто, как может показаться на первый взгляд, но нет ничего невозможного, и за пару дней почти любой кроссворд можно решить.

Но ведь на самом-то деле я хотел рассказать не об этом.

Разобравшись в доступных алгоритмах решения, я понял простую вещь — они не предназначены для решения

неоднозначных кроссвордов, т.к. дают отличные результаты лишь в случае, если кроссворд действительно "японский" (имеет одно решение). В против-

можного варианта истинному решению. Для примера, вторая строка (рис.7) из кроссворда, представленного на рис.1, будет иметь в массиве ин-



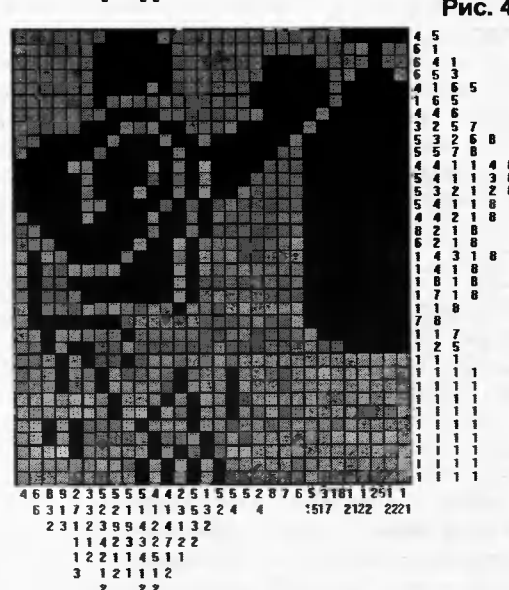
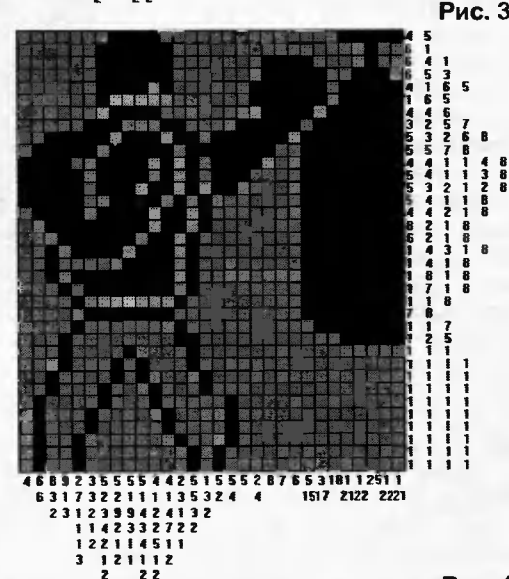
ном случае приходится заниматься гаданием, и мало того, еще и запоминанием каждого решения и сравнением его с новым (решения могут совпадать). Потому создал свой алгоритм. Думаю, что он может претендовать на звание "универсального" алгоритма.

Для начала, вот один пример "интересного" кроссворда. Сравните рис.2...6 и найдите **много** отличий! Этот кроссворд встретился мне на http://sw710.narod.ru/razn/yar_sw.htm, где также затрагивается вопрос многовариантности в головоломках. Я "дорешался" до 112 миллионов вариантов решений (это серьезно). Но что интересно, так как применялся алгоритм последовательного перебора, то можно сделать вывод, что решений существует как минимум в два раза больше.

Ниже описаны основные моменты алгоритма, применяемого при поиске решений. Если вам встретятся непонятные выражения, их расшифровка приводится в самом конце статьи.

В процессе поиска используются два способа хранения поля кроссворда (области изображения).

Первый — двумерный логический массив (наиболее очевидный способ представления), в котором первый индекс соответствует строкам, а второй — столбцам кроссворда. Значение истины, записанное в ячейку кроссворда, показывает, что соответствующая индексам ячейки клетка кроссворда является в текущий момент времени закрашенной. Использование этого массива обусловлено тем, что при таком способе хранения данных легко устанавливать соответствие воз-



декс 1 (начиная с нуля). Тогда элементы массива

$M1[1,0]=M1[1,1]=M1[1,2]=M1[1,3]=M1[1,4]=...$
 $=M1[1,10]=M1[1,16]=M1[1,17]=M1[1,18]=M1[1,19]=0$,
 в то время как

$M1[1,11]=M1[1,12]=M1[1,13]=M1[1,14]=M1[1,15]=1$.

Второй массив определяется в трех координатах, где первый индекс указывает на строку кроссворда, второй — на порядковый номер блока в строке, а третий принимает значение типа хранимой информации. По индексу 0 записывается информация о положении первой ячейки блока в строке, а по индексу 1 — длина блока. Организация второго массива позволяет “передвигать” блоки строки без постоянного пересчета всех точек блока. Изменению подвергаются лишь значения массива, имеющие в качестве третьего индекса единицу. Аналогично предыдущему примеру, массив M2 для второй строки кроссворда будет иметь следующий вид:

$M1[1,0,0]=11$ и $M1[1,0,1]=5$.

Как можно заметить, этот вариант несколько короче предыдущего.

Суть предлагаемого метода заключается в том, что от исходного кроссворда на каждом шаге отрезается лишь шапка (верхняя часть), и осуществляется ее эвристический анализ на возможность существования решения при текущем состоянии. В случае, если текущее состояние признается не имеющим будущего (нет решений при любой перетасовке оставшейся части кроссворда), производится перемещение последнего блока последней строки шапки (до достижения блоком конфликтного положения, т.е. дальнейшее движение невозможно по какой-либо причине).

По этой же схеме сдвигается блок, предшествующий текущему, при невозможности произвести движение в строке (блоки возвращаются в начальное положение). Если шапка содержит лишь две строки, осуществляется движение первой из строк, а после перебора всех состояний в обеих строках, алгоритм завершает свою работу. В противном случае (строка больше двух) реализуется движение предыдущей строки (если нельзя двигать предыдущую строку, рекуррентно двигаются предшествующие ей), и производится новый срез шапки, состоящей из меньшего числа строк (наиболее оптимальным считается уменьшений на 1).

Когда состояние шапки признается многообещающим (имеющим решения), захватывается шапка большего размера, и алгоритм повторяется по той же схеме. При достижении последней строки кроссворда активизируется функция полного перебора блоков

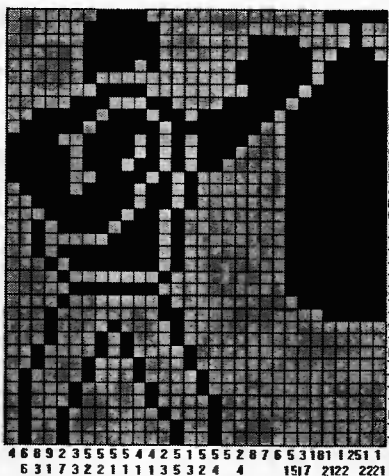


Рис. 5

- размер блока больше требуемого, и последняя ячейка в блоке занимает последнюю строку шапки.

Число критериев можно сократить до двух, сформулировав их следующим способом:

- размер последнего блока больше соответствующего ограничения, при условии, что ограничение не ноль (т.е. существует);
- ограничения нет, а размер блока не нулевой.

И последнее. Хочется отметить, что результаты сравнения описанного выше подхода с другими показывают, что по скорости поиска, в случае, когда решение единственное, он практически не отстает от аналогов, а при наличии нескольких вариантов — значительно опережает их (выигрыш зависит от структуры исходного кроссворда и составляет от 200 до 4000% по отношению к другим подходам).

Вот и все. Если у кого-нибудь возникнут вопросы или предложения, буду рад их выслушать и поделиться новыми идеями. Писать прошу по адресу Ewg_sha@mail.ru

Используемые термины

Ограничения — показывают размер и положение предполагаемых областей (блоков) ограничений в каждой из строк и столбцов. Задаются для любого кроссворда.

Блок (область) ограничений — группа закрашенных ячеек.

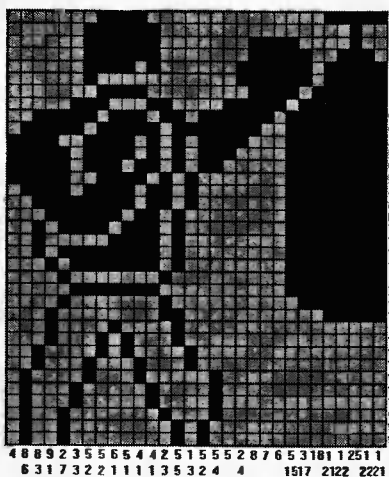


Рис. 6

последней строки. Если при этом находится решение, оно сохраняется, а счетчик количества решений увеличивается на единицу. На рис.8 показан срез, состоящий из трех строк.

Остается сущая мелочь — проверить наличие возможности решений при текущем размещении закрашенных ячеек. Сделать это и в самом деле очень просто. В первую очередь подсчитываем количество идущих подряд закрашенных ячеек (размер блока) в столбце и сравниваем его с длиной блока, которая должна быть по заданию. Для столбца можно сформулировать следующие принципы, показывающие невозможность решения для определенной закрашки шапки:

- в столбце закрашенных блоков больше, чем это предусмотрено по информационным полям;
- размер блока отличен от указанного в ограничениях на столбец, и последняя закрашенная ячейка блока не последняя в столбце шапки;

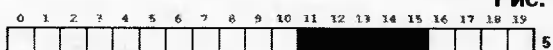


Рис. 7

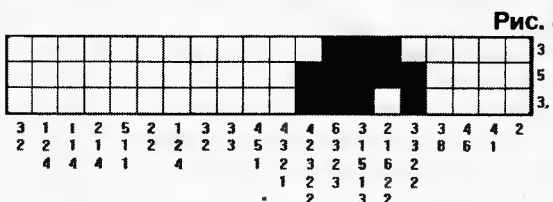


Рис. 8

“Движение блока” — перемещение блока с текущей позиции в следующую.

“Движение строки” — перемещение первого с конца блока, который может быть перемещен. При этом, если перемещается не последний блок, то все следующие за ним (расположенные справа) “возвращаются”, т.е. стремятся занять положение как можно левее (но при этом соблюдая исходную последовательность расположения блоков по строке).

“Возврат в исходное положение” — при невозможности “движения строки”, все ее блоки перемещаются в начало строки последовательно друг за другом, с интервалом между блоками, равным единице.



Простое восстановление поврежденной BIOS компьютера

В. ФЕДОРОВ,

г. Липецк.

E-mail: connectiv@yandex.ru

Как известно, Base Input Output System (BIOS) материнских плат компьютеров является основной программой, позволяющей операционным системам (ОС) работать с аппаратными ресурсами материнских плат (МВ — MotherBoard). BIOS является как бы мостиком между ОС и различными устройствами, которые на разных МВ хотя и выполняют одинаковую функцию, но порой являются совершенно несовместимыми устройствами. Таким образом, BIOS как бы сглаживает противоречия между аппаратными ресурсами различных МВ и позволяет ОС корректно использовать их. BIOS как бы состоит из множества программ, выполняющих основные функции: чтение с гибких и жестких дисков, инициализацию видеоадаптеров, прием информации с клавиатуры и т.п. BIOS располагается в специальной микросхеме ПЗУ (ROM).

МВ, поддерживающие процессоры до 486, использовали ПЗУ с ультрафиолетовым стиранием серии 27xxx. Данные МВ настолько устарели, что рассматривать их нет смысла. "Последние" 486-е компьютеры и МВ, поддерживающие современные процессоры (вплоть до Pentium 4), используют для хранения BIOS ПЗУ с электрическим стиранием (FLASH ROM). Но, как известно, все выходит из строя, и в том числе ПЗУ, содержащее BIOS. Симптомы могут быть разными: от "пропадания" некоторых функций в целом работающего компьютера до полной его неработоспособности (темный экран и отсутствие признаков "жизни") в целом. Конечно, данные неисправности могут являться результатом выхода из строя других компонентов МВ, однако в 15% случаев причиной отказа аппаратной части "материнок" является выход из строя ПЗУ.

Для замены ПЗУ BIOS необходима "чистая", или непрошитая, ПЗУ того же типа, что и заменяемая, и программатор FLASH ПЗУ (конечно, подойдет аналогичная микросхема ПЗУ с другой МВ, содержащее которой нужно предварительно стереть программатором). Микросхему ПЗУ устанавливают в панельку программатора и, если нужно, предварительно стирают. Далее загружают в программатор "прошивку" BIOS'а для вашей МВ (они доступны в Интернете на сайтах производителей МВ). После

программирования микросхему ПЗУ устанавливают в МВ и проверяют ее работу. Если она "оживла", то можно утверждать, что "слетела" BIOS, в противном случае — на МВ вышли из строя другие компоненты.

Существует еще один способ, который я называю "зверским". Для него в качестве программатора используется другой, заведомо работоспособный компьютер. Суть способа основывается на том, что при старте компьютер переписывает содержимое BIOS из более медленной FLASH ПЗУ в более быструю оперативную память (RAM). После этого FLASH ПЗУ в работе не участвует и "затеняется". Предварительно на компьютер инсталлируют специальную программу-программатор и переносят записываемую "прошивку". Затем включают компьютер и после загрузки ОС аккуратно вынимают микросхему ПЗУ из панельки (если ПЗУ находится не на панельке, а намертво припаяна к плате, использовать данный компьютер в качестве программатора не представляется возможным). Далее аккуратно устанавливают программируемую ПЗУ на место снятой и запускают программу-программатор. Она должна обнаружить ПЗУ (если ПЗУ не обнаружена, то можно быть уверенным, что она неисправна, либо неправильно установлена). При необходимости стирают "новую" ПЗУ и загружают "прошивку", которой программируют микросхему BIOS. При успешном программировании будет выдано сообщение об успешном завершении процесса. В противном случае микросхема является неисправной. Выключают компьютер и вынимают запрограммированную ПЗУ, установив на ее место штатную. Если после этого компьютер, использованный в качестве программатора, заработал, то примите поздравления — вы его не "спалили". Если еще заработала и восстанавливаемая вами МВ, то вы — счастливчик. Конечно, я категорически не рекомендую этот способ и не несу ответственности за последствия, которые могут возникнуть в результате его применения. Отмечу лишь, что если вы рискнете использовать его, вынимайте ПЗУ сначала с той стороны, где на нее подается напряжение питания (выводы с большей нумерацией). Программируемую ПЗУ устанавливайте сначала той частью, где находится общий провод

питания. Также нужно помнить, что ПЗУ может быть в корпусах двух типов: PDIP и PLCC. Если нужно запрограммировать, к примеру, ПЗУ в корпусе PLCC, то и компьютер для программирования также нужно брать с подобным ПЗУ.

Облегчить процесс программирования "слетевших" BIOS позволяет технология Twin-BIOS. Суть ее заключается в том, что на МВ устанавливают две идентичные ПЗУ с одинаковыми "прошивками" (т.н. BIOS1 и BIOS2). Если рабочая BIOS выйдет из строя или будет заражена вирусом, другая будет готова заменить ее. Чтобы определить, поддерживает ли ваша МВ функцию Twin-BIOS, посмотрите на нее, и если увидите две ПЗУ, то можно быть уверенным, что данная функция имеется.

Далее на примере МВ 60JY-2 фирмы CHANTECH покажем возможности функции Twin-BIOS. Для активизации соответствующей ПЗУ необходимо войти в программу SETUP BIOS, и в опции Advance BIOS Features в строке Boot BIOS сделать выбор между BIOS1 (по умолчанию) и BIOS2.

Для апдейта или перепрограммирования из BIOS1 нужно включить компьютер и в момент прохождения POST нажать клавиши ALT и F2. Появятся приглашения для апдейта BIOS1 (нажать F1) и BIOS2 (нажать F2). Для продолжения POST без апдейта BIOS нужно нажать Esc. При этом защита FLASH от записи (Flash BIOS Protection) в опции Advance BIOS Features программы SETUP BIOS должна быть выключена. Если будет нажата F1, апдейт BIOS1 произведется с флоппи-дисковода. Апдейт BIOS2 (F2) будет произведен из BIOS1 (Enter) или с флоппи-дисковода (Esc).

Если в SETUP установлена загрузка с BIOS2, нажатие на ALT и F2 в момент старта компьютера также приведет к появлению приглашения для апдейта BIOS1 (нажать F1) и BIOS2 (нажать F2). Если будет нажата F1, апдейт BIOS1 произведется из BIOS2 (Enter) или с флоппи-дисковода (Esc). Апдейт BIOS2 (F2) будет произведен с флоппи-дисковода.

Во всяком случае, следует помнить, что выключать компьютер во избежание его повреждения в момент программирования FLASH ПЗУ категорически запрещается.



ВОЗВРАЩАЯСЬ К НАПЕЧАТАННОМУ
(РМ. ВК, №6/2005, С.37)

А.АНИСОВИЧ,

г. Минск.

E-mail: radar1@yandex.ru

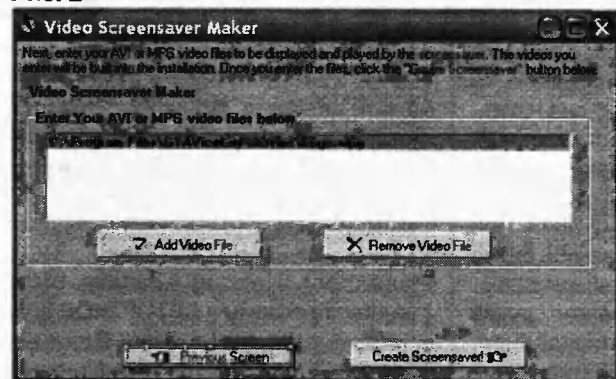
Как преобразить свой компьютер

Рис. 1

В прошлой статье мы коснулись лишь верхов настройки Windows XP на свой вкус. Было рассказано о внесении в список фоновых рисунков рабочего стола своих рисунков, рисовании иконок и курсоров, настройке указателей и изменении звуковой схемы. В этой статье речь пойдет о программах и утилитах для "приукрашивания" Windows XP. Специально для ее написания мною был приобретен компакт-диск под названием "Тюнинг Windows XP". На этом диске имеются такие программы по интересующей нас теме как "Acme Photo ScreenSaver Maker", "Super Screensaver Builder", "Video Screensaver Maker", "Style XP". Кроме того, на данном диске оказалось внушительное количество утилит по настройке, оптимизации и работе с Windows XP, тем оформления для ОС, скринсейверов, иконок, программ для визуального оформления рабочего стола. В вышеперечисленный список попали самые лучшие, на мой



Рис. 2



взгляд, программы. Пройдемся по каждой из них.

"Acme Photo ScreenSaver Maker" (shareware, www.acme-photo-screensaver-maker.com) — утилита для создания скринсейверов с музыкальным оформлением из вручную выбираемых изображений (рис.1). Создание скринсейвера состоит из 8 шагов: добавления изображений, музыки, выбора фона, создания статических надписей, движущихся надписей, выбора эффектов, добавления информации о "художестве" и ... установки ограничений (shareware) на использование созданного скринсейвера (!). С помощью кнопки на форме можно просмотреть созданное "чудо" и построить настоя-

щий установщик, который можно продать за деньги :-). Спешу предупредить об одной вещи — в созданном скринсейвере будут меняться (прокручиваться) только первых 5 изображений, независимо от того, сколько вы внесли их в список — 10 или 200. Но всем известно, что ничего совершенного на свете не бывает. Да и этот "минус" компенсируется огромным количеством эффектов, заложенных в самой программе.

"Super Screensaver Builder" (shareware, www.supersaversoft.com) — программа, также предназначенная для создания хранителей экрана с музыкальным оформлением из изображений (рис.2). Данная программа име-

ет сходство с предыдущей и особого описания не требует.

"Video Screensaver Maker" (trial) — программа для создания хранителей экрана из видеофайлов (рис.3). Из-за того что программа триальная, придется при каждом запуске делать такую неприятную операцию как ввод в текстовое поле полного названия программы для запуска. После этого появляется окно, приведенное на рис.4, в котором нужно выбрать типе файлов — AVI или MPEG. После нажатия кнопки будет создан скринсейвер из видеофайла.

"Style XP" (shareware, www.tgtsoft.com) — это, пожалуй, главная программа, ради которой и была написана эта ста-

Рис. 3

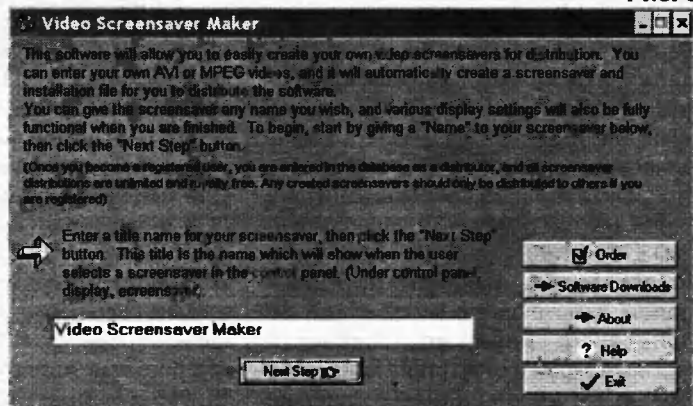
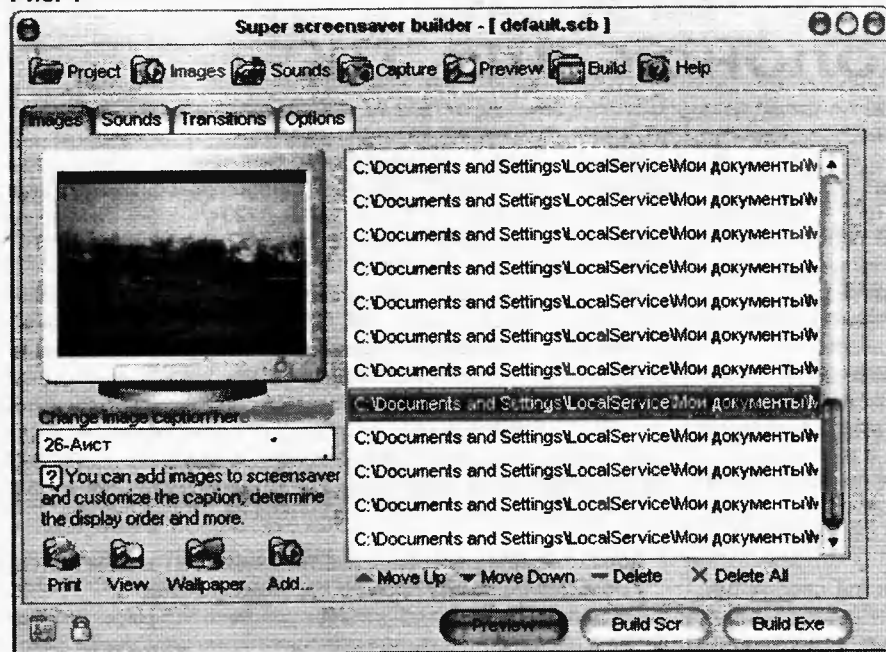


Рис. 4



- загрузчики — здесь можно заменить стандартный загрузочный рисунок на новый, более стильный (рис.8);

- прозрачность — для установки прозрачности различных меню, панели задач;

- автосмена — для автоматической смены тем оформления.

Кроме того, программа позволяет скачивать свеже созданные темы оформления, значки, загрузчики, экраны приветствия прямо с сайта разработчика.

В случае возникновения у вас каких-либо вопросов, пишите мне на E-mail: radar1@yandex.ru.

Рис. 6



Рис. 5

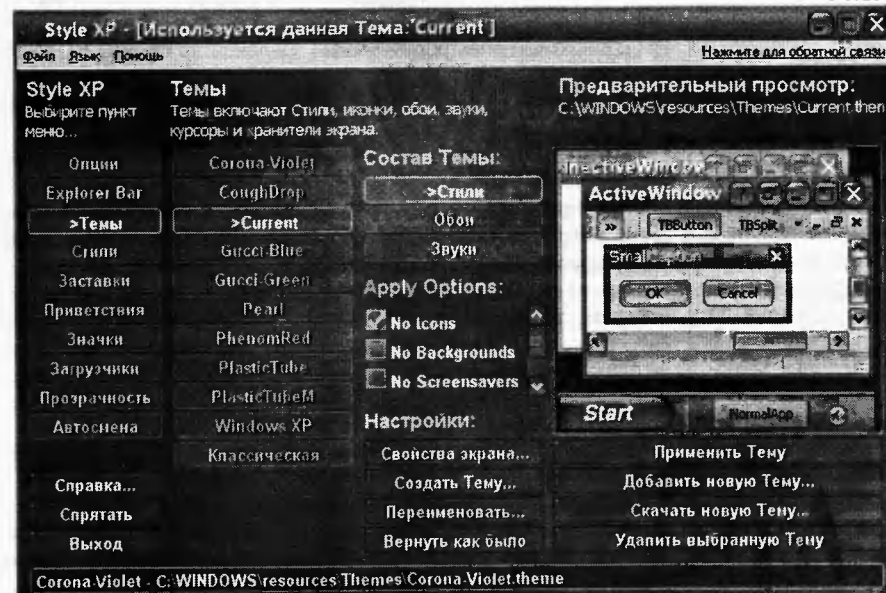


Рис. 7



Рис. 8



тъя. Она предназначена для визуального оформления Windows XP (рис.5). "Style XP" состоит из 10 основных пунктов меню:

- опции — настройка самой программы;
- Explorer Bar — для настройки Explorer;
- темы оформления для ОС;
- стили;
- заставки — для рабочего стола Windows;
- приветствия — здесь можно заменить привычный всем экран приветствия (рис.6, 7);
- значки — для обновления кеша значков;



Я.ОЧАКОВСКИЙ,
с.Петропавловское Алтайского кр.

Системные переменные

Наверное, ни один уважающий себя программист не ограничится только теми средствами, которые предлагает стандартный Спектр-Бейсик. Поэтому в этой статье будет сказано несколько слов о других возможностях, предоставляемых операционной системой компьютера. В первую очередь речь пойдет о так называемых системных переменных.

Байты памяти с 23552 до 23733 предназначены для специального использования. В них и располагаются системные переменные. Не надо путать их имена с именами переменных в программе. Компьютер не распознает ссылки к этим переменным из Бейсик-программы по их именам. Имена используются только для мнемонического обозначения этих переменных при их описании.

Все системные переменные изменяются путем записывания в ячейки памяти с помощью оператора POKE числа из определенного диапазона. Записываемые числа должны быть целыми и в пределах от 0 до 255.

Однако, экспериментировать с системными переменными не советуют. Если вам неизвестен смысл той или иной ячейки памяти, лучше оставить ее в покое и не испытывать фортуны, потому что при изменении системных переменных мы вторгаемся в святая святых компьютера. При грубом и неумелом обращении и вмешательстве в операционную систему предугадать результат бывает совершенно невозможно. Если вы имеете жесткий диск, то делать (т.е. экспериментировать с неизвестными переменными) этого вообще не стоит, т.к. результат может быть автоматически сохранен на носителе, и, возможно, вы уже никогда не сможете вернуть жесткий диск к нормальной работе, что впоследствии вынудит вас установить новый носитель.

Наверное, я вас запугал. Не бой-

тесь, просто грамотное использование данной области памяти позволит обогатить программу на Бейсике совершенно потрясающими эффектами и, вместе с тем, сохранить ее объем без потери возможностей или увеличить ее быстроедействие.

Для наглядности привожу таблицу, содержащую в себе имена соответствующих системных переменных, их функций, адреса и влияние, оказываемое на систему.

Информация, содержащаяся в первом столбце таблицы, имеет следующее обозначение:

- X — переменная не должна изменяться, так как это может нарушить работу системы;

- N — изменение переменной не приводит к длительному эффекту;

- цифра — число байтов в переменной (для двухбайтовых переменных, младший байт загружается первым).

Рассмотрим несколько конкретных примеров использования системных переменных.

Ряд ячеек в области системных переменных имеет непосредственное отношение к аппаратуре.

Все прекрасно знают, что при наборе или редактировании строк программы, если понадобится ввести несколько одинаковых символов (или удалить часть текста), то вовсе не обязательно бешено колотить по одной и той же клавише, а достаточно нажать ее и подождать примерно полсекунды — нужный символ сам начнет размножаться (или удаляться) достаточно быстро.

Часто при написании программ (особенно игровых) возникает необходимость в изменении времени задержки перед повторением или частоты самих повторений.

Первая проблема легче решается с помощью системной переменной REPDEL, расположенной по адресу 23561. Изменение времени задержки

производится оператором POKE 23561, N, где N — любое число от 0 до 255. Но не переборщите, иначе проще будет не дожидаться повторений, а упорно тыкать в клавиатуру (напомню, начальное значение — 35).

Частота повторений регулируется системной переменной 23562 (REPPER). Она изменяется так же, как и REPDEL, в диапазоне чисел от 0 до 255. Начальное значение — 5.

В длинных и продолжительных циклах, особенно если нельзя останавливать программу оператором PAUSE0, бывает очень трудно зафиксировать нажатие какой-либо клавиши, и возникает ситуация, когда жми не жми — все впустую. Это связано с тем, что компьютеру в таком случае необходимо выполнить огромное количество разных операций, а когда очередь доходит до IF INKEY\$, оказывается, что клавиша уже давно опущена. Бороться с подобными ситуациями поможет системная переменная LASTK, занимающая ячейку 23560. В этой ячейке хранится код последней нажатой клавиши, причем, после опускания клавиши, находится он там до тех пор, пока вы не нажмете другую или не выключите компьютер. Таким образом, чтобы определить, какая из клавиш была нажата, необходимо узнать содержимое ячейки 23560, а затем перевести его в символ. Читаются значения ячеек памяти функцией PEEK addr, где addr — адрес от 0 до 65535. Переводит числовые значения в символы функция CHR\$ cod, где cod — числовое значение соответствующего символа. Поэтому строки типа: LET K\$=INKEY\$ в описываемых случаях можно заменить на LET K\$= CHR\$ PEEK 23560.

Не забывайте только о своевременном сбросе переменной LAST K. Это необходимо делать, вписав в программу в необходимом месте оператор POKE 23560,0.

Обозначение	Адрес	Имя	Описание
N8	23552	KSTATE	Используется при чтении с клавиатуры
N1	235560	LAST K	Запоминается последняя нажатая клавиша
1	23561	REPDEL	Отрезок времени (в 50-х долях секунды), в течение которого клавиша должна быть зафиксирована в нажатом состоянии. Начальное значение — 35, но может быть изменено
1	23562	REPPER	Задержка (в 50-х долях секунды) между последовательными опросами клавиш. Начальное значение — 5
N2	23563	DEF ADD	Адрес аргументов функций пользователя, если они используются, иначе — 0
N1	23565	K DATA	Второй байт управления цветом с клавиатуры
N2	23566	TV DATA	Байты цвета AT, TAB управления монитором
X38	23568	STRMS	Адреса подключения каналов



2	23606	CHARS	Адрес символического набора — 256. Обычно этот набор находится в ПЗУ, но может быть размещен в ОЗУ с указанием в CHARS адреса размещения
1	23608	RASP	Продолжительность звукового сигнала, сообщающего о неправильных действиях с вашей стороны или о других ошибках
1	23609	PIP	Длительность задержки, устраняющей дребезг клавиатуры
1	23610	ERR NR	Код сообщения — 1. Начальное значение — 255 (для "-1"), т.е. PEEK 23610=255
X1	23611	FLAGS	Управляющие флажки Бейсика
X1	23612	TV FLAG	Флажок монитора (телевизора)
X2	23613	ERR SP	Адрес в аппаратном стеке, используемый как адрес возврата при ошибке
N1	23617	MODE	Режим. Спецификация [K], [L], [C], [E] или [G] курсора
2	23618	NEW PPC	Номер строки, на которую должен быть сделан переход
1	23623	SUB PPC	Порядковый номер выполняющегося оператора в строке
1	23624	BORDER	Цвет рамки экрана. Содержит атрибуты
2	23625	E PPC	Количество текущих строк
X2	23627	VARS	Адреса переменных (не путать с системными)
N2	23629	DEST	Адрес переменной в задании
X2	23631	CHANS	Адрес канала данных
X2	23633	CURCHL	Адрес данных для ввода-вывода
X2	23639	PROG	Адрес Бейсик-программы
X2	23637	NXTLIN	Адрес следующей строки в программе
X2	23639	DATADD	Адрес терминатора последнего символа в DATA
X2	23641	ELINE	Адрес выведенной команды
2	23643	K CUR	Адрес курсора
X2	23645	CH ADD	Адрес последнего интерпретируемого символа в операторах PEEK, NEW, LINE или POKE
2	23647		Адрес символа, следующего за маркером [?]
X2	23649	WORK SP	Адрес временной рабочей области
X2	23651	STK BOT	Адрес "дна" программируемого стека
X2	23653	STK END	Адрес начала резервной области памяти
N1	23655	BREG	B — регистр калькулятора
N2	23656	MEM	Адрес области, используемой как память калькулятора (обычно MEMBOT, но не всегда)
1	23658	FLAGS2	Старшие флажки
X1	23659	DF SZ	Число строк (включая и одну нижнюю) в нижней части экрана
2	23660		Количество верхних строк программы в автоматическом листинге
2	23662	OLD	Номер строки, на которую указывает оператор Continue
1	23664	OSPPC	Номер оператора в строке, на которую указывает оператор Continue
N1	23665	FLAGX	Переменные флажки
N2	23666	STR LEN	Величина расстояния между строками
N2	23668	T ADDR	Адрес следующего символа в синтаксической таблице
2	23670	SEED	Начальное значение для RND. Возможно изменение функцией RANDOMIZE
3	23672	FRAMES	Счетчик кадров (приращивание через каждые 20 мс)
2	23675	UDG	Адрес первого, определяемого пользователем, символа
1	23677	COORDS	X — координата точки графопостроителя, Y — координата точки графопостроителя
1	23678		
1	23679	P POSN	33 — позиционное число для позиционирования принтера
1	23680	PR CC	Младший байт адреса позиции для оператора LPRINT при печати
1	23681		Не используется
2	23682	ECHO E	33-позиционное и 24-строковое числа (в нижней половине) конца входного буфера
2	23684	DF CC	Адрес PRINT — позиции в области экрана
2	23686	DF CCL	Подобно DF CC для нижней части экрана
X1 X1	23688	S POSN	33-позиционное число для PRINT-позиции, 23-строковое число для PRINT-позиции
	23689		
X2	23690	S POSNL	Подобно POSN для нижней части экрана
1	23692	SCR CT	Счетчик сверток — всегда на 1 больше числа сверток, которые должны быть проведены перед остановом со сверткой. Если вы установите число, больше чем 1 (скажем, 255), то экран будет сворачиваться без запроса подтверждения
1	23693	ATTR P	Сплошные цвета
1	23694	MASK P	Используется для высвечивания 8-битных цветов, установленный в "1", показывает биты атрибутов, которые берутся не из ATTR P, а из того, что указано на экране
N1	13695	ATTR T	Временный указатель цветов
N1	23696	MASK T	Временный MASK P
1	23697	P FLAG	Старшие флажки
N30	23698	MEMBOT	Область памяти калькулятора. Используется для записи чисел, которые не могут быть размещены в программируемом стеке калькулятора
2	23728		Не используется
2	23730	RAMTOP	Адрес последнего байта области Бейсик-программы
2	23732	P-RAMT	Адрес последнего байта физического ОЗУ

(Окончание следует)



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

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



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

Продам: техническую документацию на Р-399А; авиационную дисконную антенну, работающую и в диапазоне 2 м; фильтры на 215, 465 и 500 кГц, а также на на 34,785 МГц; кольца ферритовые диаметром 20 — 45 мм; измерительные головки (50 — 300 мкА); реле РЭС, РПА, РПВ, РЭВ, В1В; конденсаторы КСО, К-15, переменной емкости (в т.ч. вакуумные); резисторы безындуктивные ТВО, УНУ, МОУ (25 — 150 Ом); частотомер ЧЗ-54 (0 — 150 МГц).
630098, г.Новосибирск,
ул.Приморская, 23 — 132. В.Г.Ермола.
Тел. 8383-345-77-30.

Куплю 2 ИМС CD4536BE (или аналог).
220040, г.Минск, ул.Некрасова, 33 — 51.
Тел. 8-029-557-01-44.

Юный радиолюбитель (11 лет) с благодарностью примет в дар компьютер или комплектующие к нему.
195067, г.С.-Петербург,
пр.Маршала Блюхера, 56 — 64. А.Феоктистов.
Тел. (812) 543-98-39.

Куплю: тиратроны ТГ-3-0,11; строботроны ИСШ-5, ИСШ-15 или обменяю на техническую документацию на осциллографы С1-83, С1-99 и магнитофоны Исык-Куль МПК-101-1, Союз 110, Союз 111.
650501, г.Кемерово, ул.Полесская, 2 — 1.
В.Д.Ладурко.
Тел. 8-904-377-20-80.

Продам: трансивер IC-746PRO, радиостанцию "Маяк" на диапазон 2 м.
397600, Воронежская обл., г.Калач,
ул. Энгельса, 51. А.В.Гревецев.

Продам радиоприемник "Калина" (цифровая шкала, ЭМФ по 2-й и 3-й ПЧ).
Куплю радиолампы 6Ж53П, 6С58П, 6С59П или обменяю на ЭМФ (215 и 500 кГц).
169300, Республика Коми, г.Ухта, а/я 36.
А.Н.Данилов.

Продам: радиоприемник "Катран" Р-399А (1 — 30 МГц), трансивер "Волна" (160 — 10 м), радиостанции Р-123М и "Фермер" (27 МГц, 10 каналов), усилитель мощности Р-140 (блок на ГУ-43Б), радиолампы ГУ-19, Г-811, EF-86, ECC-82, ECC-83 и др.
170004, г.Тверь, ул.Речная, 12. В.Л.Кабурневе.
Тел. 42-86-16.

Продам: X-1-19, C-6-1, C-1-19, C-1-64, 4-3-38, Г-3-35, ГСС-6, ГСС-7, Г-3-8А, Г-4-102А, БП 2 (1 — 30 В, 5 А), ВК-7-9, Р-399 с панорамным блоком, Р-405 с УМ, радиостанции "ПАЛЬМА" и "БАКЛАН" (144 — 146 МГц), удлинитель радиотелефонный VOYAGER GT-598 DX (35 Вт), коллекцию бытовой радиоаппаратуры (более 150 наименований), мачты телескоп 10 м, 7 м с изолятором (ГР) и 4 колена по 1 м, колесо мачты с двумя редукторами (высотой 2,5 м), антенну коллинеарную ARS-2-B (144 МГц), редуктор для поворота антенн (размеры 50x25 см), редуктор на базе ПР-4 (самодельный, с сельсинами), УМ Р-140 с УСС (в стойке на ГУ-43Б, с резервными лампами), УМ Р-134 (транзисторный, 200 Вт), УМ SYNCRON-300 (транзисторный, 300 Вт), антенный тюнер MFJ-949C (до 300 Вт), антенные нагрузки 100 Вт/75 Ом и 1000 Вт/75 Ом, трансивер IC-F4-4 (440 МГц, 4 Вт, канальный, с чехлом), ключ телеграфный с памятью, настольный станок для намотки трансформаторов, настольный станок для намотки трансформаторов со счетчиком.
Всеволод Всеволодович.
Тел. (327-2) 91-70-76.
E-mail: un7gj@topmail.lz

Куплю документацию на модем GVC 2400 V.42bis и схему подключения.
E-mail: northman@yandex.ru

Ищу транзистор МТБ1306 или его аналог (используется в фильтре питания материнской платы на VIA KT333 под Socket-A).
Игорь.
E-mail: seval@atnet.ru

Продам автомобильную радиостанцию Dragon SY-550 (141 — 149,95 МГц, 10 ячеек памяти, мощность 25 Вт).
Тел. 8-905-56-1-56-46. Александр.
E-mail: alex222@inbox.ru

Куплю трансивер, недорого.
Игорь.
E-mail: konkordv@svitonline.com

Продам радиоприемники Р-250 и Р-323 в полном рабочем состоянии, с документацией. Самовывоз из С.Петербурга.
E-mail: m-20_pobeda@mail.spbnet.ru

Продам литературу по ZX SPECTRUM.
165341, Архангельская обл., пос.Вычегодский, ул.Ленина, 2 — 11. Владимир.
Ученик 11 класса из небогатой семьи с благодарностью примет в дар любой компьютер или комплектующие к нему.
671950, Бурятия, г.Закаменск, ул.Байкальская, 81. А.Бутин.
E-mail: butart_zkm@rambler.ru

Продам: радиолампы ГИ-7Б (2 шт.), радиостанцию Р-401М (60 — 70 МГц), приемник Р-311.
Тел. в Климовичи Могилевской обл.
(8-022-44-5-62-74). Иван Архипович.

Куплю кварцы на частоты 10300, 11060, 12500 и 14670 кГц в старых (больших) металлических или стеклянных корпусах. Разброс — ±1%.
655016, Республика Хакасия, г.Абакан, ул.Крылова, 88 — 139. Ю.Бабишев.
Тел. 4-33-13.

Продам остатки журналов "Радио", "Техника Молодежи", "Юный техник" по сходной цене, а также книги по радиотехнике. Для получения подробного каталога необходимо прислать оплаченный конверт с обратным адресом.
455000, Челябинская обл., Магнитогорск, ОУПС, а/я 247.00. Евгений.
Тел. 8 904 932 77 65 (в любое время до 19.00 МСК).

Куплю (в Самарско-Уральском регионе, или в Москве, или в Рязани) новый (или почти новый) радиоприемник Р-399А с передающей приставкой, документацией, ремшлангами и ЗИПом.
456014, Челябинская обл., г.Аша, ул.Озиминая, 32 — 29. В.Зеленский.
E-mail: rk9af@mail.ru

Продаю две новые 40-канальные автомобильные СВ-радиостанции (АМ, 4 Вт); механические кодеры 10-2 двух типоразмеров; взаимозаменяемые переключатели П2К; транзисторы ГТ905, КП904, КТ704, КТ904, КТ907, КТ807; сегментные импортные индикаторы, класеры для микросхем, оптический болометр 100-0-100 мкА.
141980, Московская обл., г.Дубна, ул.Мира, 9/6 — 4. И.Семенов.
Тел. (096) 21 4 54 00.

Продаю конденсаторы К75-40А (100 мкФх3000 В), К50-18 (10000 мкФх80 В), от приемника Р-311 с фотошальной; поворотное устройство R-7000 с пультом для УКВ-антенны; в корпусе от Р-311 базовый приемник UA1FA с кварцевым фильтром по 1-й ПЧ; кварцевые фильтры ФП2П-153-01-25М-40, ФП2П-153-00-25М-18, 10,7 МГц KVARC (QD001-15); электрохимические фильтры ФЭМ-500-0,6С, ФЭМ-500-6; НЧ-фильтр Д-3,4; кварцевые резонаторы на 100, 465, 1000, 5000, 5500, 6000, 10000 кГц и т.д. (до 300 шт.); реле РЭС6, РЭС8, РЭС10, РЭС15, РЭС22, РЭС49, РЭС55; транзисторы КП103, КП302, КП303, КП304, КП305, КП350, КП312, КП903, КП904, КП905, КП907, 2Т602, КТ603, КТ605, КТ606, КТ608, КТ635, КТ801, КТ805АМ, КТ829, КТ837, КТ826, КТ819, КТ904, КТ907, КТ908, КТ909, КТ911, КТ920, КТ925, ГТ402, ГТ403, ГТ404, ГТ806, П210, П702; трес (нерж.) Ж4,8 мм — 100 м.
352506, Краснодарский край, г.Лабинск, ул.Ленина, 34/1. В.Бурый, UA6AK.
Тел. (8 86169) 7-25-65.

Продаю носимую радиостанцию ALAN CT-145 (130 — 170 МГц).
396900, Воронежская обл., г.Семилуки, а/я 1.
Тел. (8 07372) 2 — 43-40 с 22.00 — 23.00. Михаил.

Продаю или меняю Г-811, ГК-71, ГУ-29, ГУ-50, ЭМФДП-500с-0,5, ЭМФ-500-0,6с, реле "холопушки" от РСБ-5.
Куплю вакуумный КТЕ (300 пФ), КТЕ от радиостанции "Микрон".
453265, Башкортостан, г.Салават-15, а/я 6.
Евгений, RA9WD.
Тел. (34763) 3-09-19, 89272327121.

Продам ноутбук NEC PC-9821 LS150 AILE.
E-mail: SEIVAN33@RAMBLER.RU

Продаю трансивер TS-850S.
Тел. в г.Черновцы 8 (037) 2-55-33-35. Николай.
E-mail: urfyex@yandex.ru

Продаем цифровые шкалы для трансиверов до 30 МГц на ПИК. Управление ПЧ+ и ПЧ- (ПЧ — любая), входной сигнал — от 100 мВ. Размеры — 4x4 см, индикатор — АЛС318 (семь разрядов). Можно использовать как частотомер до 30 МГц, точность измерения — 10 Гц.
Россия, 398007, г.Липецк, а/я 1464.
Коллективная радиостанция "РКЗГВД".
E-mail: RK3GWD@mail.ru

Школьник с благодарностью примет в дар любой компьютер и/или комплектующие к нему.
671950, Бурятия, г. Закаменск, ул.Байкальская, 81. А.Бутин.

Продаю осциллограф С1-65.
E-mail: kudm_77@mail.ru

Продаю вариометр роликовый (34 мкГн, диаметр провода — 2 мм; ВЧ-реле от Р-130М с вакуумным переключателем (2 шт.); измеритель RCL (1 Ом — 5 МОм; 10 пФ — 100 мкФ, 1 мкГн — 100 мГн).
184506, Мурманская обл., г. Мончегорск-6, а/я 19. Анатолий.
Тел. 88153659290, +79211755384 (моб.).
E-mail: ua1zhv@monch.mels.ru

Ученик 10 класса примет в дар или купит за условную цену верхнюю плату от компьютера Profi+ (плата версии 4 или старше).
357108, Ставропольский край, г.Невинномысск, Менделеева, 8 — 41. А.Шарихин.
Тел. (928) 315-21-04 (для SMS использовать только три последних цифры).

Срочно куплю ГУ-66А (5 шт.), ГУ-88А, ГУ-23А. Прошу в ответе указывать год выпуска и цену.
E-mail: infoinfo@list.ru

Продаю радиостанцию Ailco DR-330 (330 — 370 МГц, работает на частотах 336 МГц, выделенных Российской речным регистром для яхт), мощность — 35 Вт, шнур питания, инструкция (на русском языке), упаковка.
Тел. в Москве 8-916-72-27-403. Саша.
E-mail: alex2ed@inbox.ru

Ищу специалистов, которые могут указать год выпуска трансивера FT-102 S/N 2J040676.
Россия, 398037, г.Липецк, ул.Труда, 19.
В.Шугайлов, RU3GS.

Куплю принципиальную схему радиостанции "Недра-3" (желательно с описанием, можно ксерокопией).
413210, Саратовская обл., Советский р-н, пос.Степное, ул.50 лет Победы, 47 — 3.
С.Малин, UA4CPX.

Ищу руководство по эксплуатации (на русском языке) трансивера YAESU FT-2500M.
Куплю радиолампы ГМИ-11.
347800, Ростовская обл., г.Каменск-Шахтинский, ул.Ворошилова, 91. А.Косоголов, RA6MX.
Тел. 8-86365-7-06-02 (дом.), 8-9034733134 (моб.).

Продаю милливольтметры В3-13 (до 1 МГц), В3-38Б (до 5 МГц); ВЧ-вольтметр до 450 МГц (изготовлен по схеме UB5WCC); лабораторный блок питания Б5-47; ГСС-6А; КСВ-метр; трансформаторный блок питания для импортного трансивера; недорого две радиостанции "Нира" (на частоту 2290 кГц) с блоками питания от сети 220 В и 9-диапазонные КВ-усилители на 2-х ГИ-7Б и 3-х ГУ-50 (бестрансформаторный); Hi-Fi-оконечные усилители на микросхемах (2x100 и 2x200 Вт).



КУПЛЮ, ПРОДАМ, ОБМЕНЯЮ

Куплю или обменяю РЭВ-15, В1-В (или аналогичные), ВЧ-разъемы СР-50-164П, СР-50-165Ф. 429500, Чувашия, п.Кутеси, ул.Советская, 36. Н.Свинцов, RA4YM. Тел. 8(3540) 2-16-98.

Продам милливольтметр ВЗ-38, частотомер ЧЗ-45. Куплю трансформаторы ТАН73, ТАН132, радиолампы 6ПЗС, 5Ц4С. А.Конов. E-mail: valerych-arz@mail.ru

Продаю или меняю спутниковый тюнер "Skymaster x15" (новый, в упаковке) на переднюю панель к автомагнитоле "LG Icc6230". Владимир. E-mail: uw-7276@rambler.ru

Продаю недорого 3 шаговых электродвигателя в г.Минске, Саша. E-mail: radar1@yandex.ru

Куплю транзисторы КТ610, КТ904, КТ907; передатчик на 100 МГц (1...3 Вт). Тел. +375293257284.



Радиомир. Лучшие конструкции. Выпуск 1.

ООО "Радиомир Пресс", 2005 — 112 с.

Книга представляет собой сборник статей, опубликованных в разные годы в журнале "Радиомир" и заново отредактированных для данного издания.

В выпуске приведены схемы и описания устройств, используемые радиолюбителями в различных областях.

По поводу приобретения книг можно обращаться в редакцию по E-mail: rm@radio-mir.com, а также во все организации, имеющие журнал "Радиомир" в розничной продаже. Информация о них приведена рядом.

При приобретении книги через редакцию ее стоимость составляет:

- для жителей России — 50 руб. рублей;
- для жителей Беларуси — 3900 бел. рублей;

Правила приобретения — аналогично адресной подписке на журналы через редакцию.

Выберите себе вариант подписки!

Подписка через почтовые отделения

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.presscafe.ru);
- для жителей Беларуси: 48996 (489962 — для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" и через киоски Мингорсоюзпечати.
- для жителей Казахстана: 25785 — подписка по каталогу АРЗИ "Российская Пресса".

Радиомир. КВ и УКВ

- для жителей России и стран СНГ (кроме Беларуси): 48924 — подписка по каталогу Агентства "Роспечать" (71545 — годовая);
- для жителей Беларуси: 00137 (001372 — для организаций) — подписка по каталогу РО "Белпочта" "Газеты и журналы Республики Беларусь" (раздел "Издания РФ, распространяемые по прямым договорам") и через киоски Мингорсоюзпечати.
- для жителей Казахстана: 46502 — подписка по каталогу АРЗИ "Российская Пресса".

Радиомир. Лучшие конструкции

- для жителей России и стран СНГ (кроме Беларуси): 20020 — подписка по каталогу Агентства "Роспечать";
- 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.

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

Внимание! Адресная подписка через редакцию. Подписаться на имеющиеся в наличии отдельные номера журналов, а также на любой период, начиная со следующего после оплаты месяца, можно через редакцию. Для этого нужно оплатить необходимую сумму через Сбербанк России или оформить почтовый перевод на наш расчетный счет. Текущие цены приведены в таблице. В цену включена доставка журналов в отдельном конверте по адресу подписчика, указанному в графе "Назначение платежа" при оплате через Сбербанк России или в графе "Для письма" при оплате почтовым переводом. Наложным платежом журналы не высылаются.

Наши платежные реквизиты:

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

- для жителей Беларуси —
получатель: УП "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) 249-41-47.

Год	Можно заказать следующие номера журналов			Расценки на 1 экз. (с учетом пересылки)		
	Радиомир	Радиомир. Ваш компьютер	Радиомир. КВ и УКВ	По России и СНГ (рос. руб.)	По Беларуси (бел. руб.)	По Украине (гривны)
2002	2 — 12	1 — 12	1 — 3, 5 — 12	30	1500	10
2003	1 — 12	1 — 12	1, 5 — 8, 10 — 12	35	1800	10
2004	1 — 12	1 — 6, 8 — 10, 12	1 — 12	40	2100	10
2005	1 — 12	1 — 12	1 — 12	45	3000	11
2006	1 — 12	-	1 — 12	49	3000	12

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.Гильярдовского, 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.Москва, ул.Земляной вал, д.34.

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

В ООО "Альянс-Книга КПК": 115533, г.Москва, ул.Нагатинская, д.6, эт.1, оф.ВII (ст. метро "Нагатинская"), тел. (095) 258-91-94, 258-91-95, 258-91-96, 111-05-98, 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.Независимости, 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").

и не подпадающие в США и других странах

КОМПЬЮТЕРЫ **СИТИПРИНТ** ДЛЯ ДОМА И ОФИСА



Безграничные возможности

Откройте для своей семьи новые способы обучения, общения, развлечений – приобретите компьютер от компании “СитиПринт” на базе процессора **Intel® Pentium® 4** с технологией HT.



СитиПринт
КОМПЬЮТЕРЫ И ОРГТЕХНИКА

220006, г. Минск, ул. Полевая 28-7А.
Тел./факс (017) 2850134, 2218728.
www.citiprint.by

Подписка – 2006!

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.presscafe.ru);

— для жителей Беларуси: **00137**, (**001372** — для
организаций) — подписка по каталогу РО "Белпочта"
"Газеты и журналы Республики Беларусь" и через киоски
Мингорсоюзпечати.

*В мире оживших звуков
Рядом с телефоном
Танцуем от питания
Автоматика всегда поможет
Сам себе лекарь
Вокруг автомобиля
Первым делом технология
Видеотехника
Измерения
Не только новичку
Личная радиосвязь
Справочный материал
Радиолюбительская ярмарка
Компьютер "вдоль и поперек"*

радиомир

КВ, 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.

*Новости
QUA
Дипломы
DX-info
Соревнования
Эфирная робинзонада
Компьютер на радиостанции
УКВ
Усилители
Трансиверы
Антенны
Доска объявлений*



— для жителей России и стран
СНГ (кроме Беларуси): **20020** —
подписка по каталогу Агентства
"Роспечать";

— 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.

Лучшие конструкции

Радиомир

Выпуск 1:

*Автоматика
Звукотехника
Автомобильная электроника
Телефония
Медицина
Видеотехника
Источники питания*

Внимание! Адресная подписка через редакцию!

Подписаться на имеющиеся в наличии отдельные номера, а также на любой период, начиная со следующего после оплаты месяца, можно через редакцию. Информация на <http://radio-mir.com> или по e-mail: rm@radio-mir.com

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

В Украине журналы "Радиомир", "Радиомир. КВ и УКВ" можно приобрести в фирме "Торм", тел. (044) 287-47-64, 287-72-73.