

Windows 3.0:
в защищенном режиме:
достигая невозможного
(недокументированные
функции Windows)

“Тихое” решение ценой
в 10 000 000 000 рублей
(принятие основ
законодательства
об авторском праве)

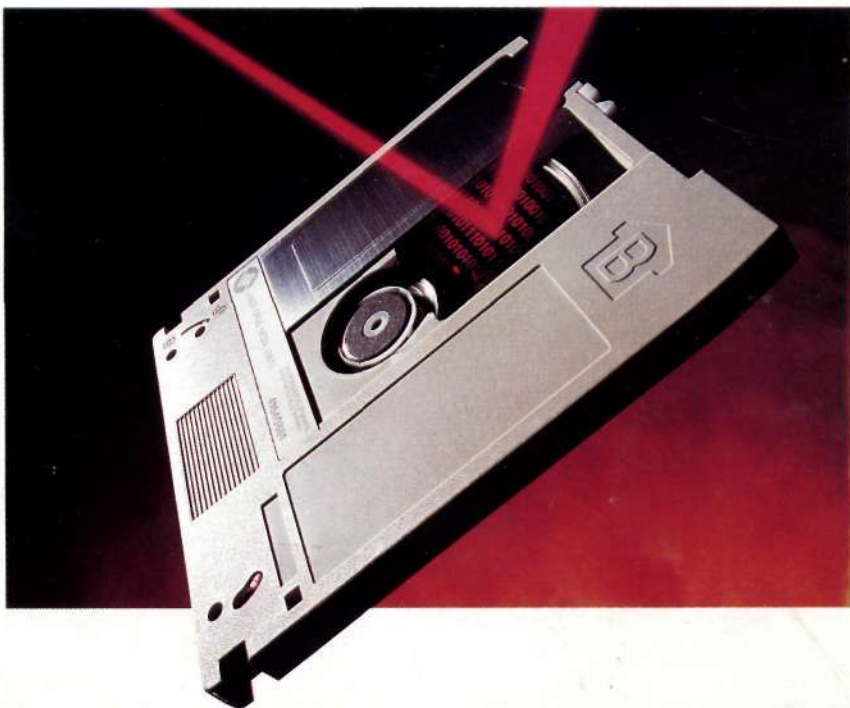
15 реляционных
баз данных:
простота доступа,
мощные программные
средства

Знакомство
с системой
BIOS

Магнитооптические диски: сбылись ли наши мечты о перезаписываемых оптических дисках?

ЧИТАЙТЕ В СЛЕДУЮЩИХ НОМЕРАХ

- Локальные сети
(специальный обзор)
- Недорогие
интегрированные
программные пакеты
- Как избежать порчи данных
на жестком диске
- Введение в практику
объектно-ориентированного
программирования – Си++





TAOS[®]
INTERNATIONAL, LTD.

НАИЛУЧШИЕ КОМПЬЮТЕРЫ АМЕРИКИ

ТОРГОВЫЕ ПРЕДСТАВИТЕЛЬСТВА

В Москве
MARUBENI CORPORATION
Московское представительство,
Секция МАС 4
СССР, Москва 123610
Краснопресненская наб., 12
Центр Международной Торговли
комната 1908 (19 этаж)
Москва, СССР
Тел. Москва 253-1821/3
Телекс. 413391, 413146 Marbn Su
Факс. Москва 230-2731

В Хабаровске
MARUBENI CORPORATION
Хабаровское отделение
СССР, 680065
Хабаровск, Амурский Б-р, 2
Гостиница „Интурист“
комната 1007
Тел. Хабаровск 39-91-12

ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

МЕЖДУНАРОДНЫЙ
КОМПЬЮТЕРНЫЙ КЛУБ (МКК)
СССР, Москва 101813
Проезд Серова 4.
Общество „Знание“
Тел. Москва 921-09-02
Телекс. 411630
Факс. Москва 921-09-02

Genius

Что поможет вам удержаться на самом гребне стремительного вала новейших издательских систем?

Конечно, это устройства Genius!

GS-C105 Color Scanner

Этот сканер распознает до 4096 цветов и способен считывать цветные, полутоновые и штриховые изображения, а также текст.



GS-B105G Gray Scale Scanner

Не обязательно быть профессионалом, чтобы с помощью этого сканера на 256 полутонов создавать высококачественную графику.

iPhoto Software

Это высокоэффективная программа, предназначенная для системы Windows 3.0. Она позволяет считывать, преобразовывать и улучшать цветные и полутоновые изображения.

CAT OCR Software

Интеллектуальная программа оптического распознавания символов. Ее можно "обучить" точному считыванию документов.



Официальный дилер

UNITECH

СОВМЕСТНОЕ
СОВЕТСКО-ГЕРМАН-
СКОЕ ПРЕДПРИЯТИЕ
"УНИТЕХ"

119899, Москва,
НИВЦ ИГУ
Тел. (095) 939-50-69
939-16-19 939-23-44
Факс (095) 938-21-36

Dr. Genius
Simply Better



KYE

KUPYING Enterprise Co., Ltd.
492, Chung-Hsin Rd., Sec.5,
San-Chung, Taipei Hsien, Taiwan
TEL:(02)995-6645
FAX:(02)995-6649

СОБЫТИЯ

- 7** Второй Международный компьютерный форум в Москве.
До встречи через год!
Илья Липкин
- 8** Для Apple — представитель в СССР, для советских пользователей — новая кодовая таблица
Рубен Герр
- 9** "Тихое" решение ценой в 10 000 000 000 рублей
Леонид Малков

КОМПАНИИ



- 35** Малоизвестные страницы из жизни Epson.
Юрий Кузьмин

АППАРАТНЫЕ СРЕДСТВА

- 11** Магнитооптические диски: Сбылись ли наши мечты о перезаписываемых оптических дисках?
М. Дэвид Стоун



КАК СТАЛО ИЗВЕСТНО

- 37** Уготована ли OS/2 в 1992 году роль "убийцы" системы QEMM?
Гас Вендитто
- 38** 15 программ-бестселлеров в июне

НОВИНКА! ДАВАЙТЕ ПОПРОБУЕМ

- 29** Компьютер Toshiba 3200SXC с цветным ЖК-экраном.
Патрик Хонен
- 30** Карманные модемы — от максимума возможного до минимума необходимого
- 31** Увеличьте поле зрения. *Билл Хауард*



- 32** Фирма Motorola создает беспроводную ЛВС.
Стив Райни
- 33** LANster фирмы Hayes - простая сетевая ОС, практически не уступающая системе LANtastic. *Рик Эйр, Рон Андерсон*

ПОЛИГЛОТ

- 41** Windows 3.0: хорошая память и отличная работоспособность.
Эндрю Шулман
- 55** Windows 3.0 в защищенном режиме — достигая невозможного.
Эндрю Шулман

НОВОСТИ

- 54** **117**

КОРОЛИ, КАПУСТА И ... КОМПЬЮТЕРЫ

- 69** Программирование с человеческим лицом. *Рубен Герр*
- 71** Новая ОС короля. *Билл Макроун*
- 73** Вести из мира Macintosh. *Джим Симур*
- 75** Куда подевались любители? *Джон Дворак*
- 76** Первое место присуждается ... *Уильям Ф. Закман*



БАЗЫ ДАННЫХ

- 77** 15 реляционных баз данных: Простота доступа, мощные программные средства. *Дэвид Кэлман*
- 93** Алло! СУБД слушает вас. *Юрий Филипчук*

ПРОГРАММИСТУ-ПРОФЕССИОНАЛУ

- 102** Постигая интерфейс прикладных программ оболочки Windows. *Рей Дункан*
- 106** Транспортировка конструкций

МАЛЕНЬКИЕ ХИТРОСТИ

- 108** О вреде языкознания. *Рубен Герр*

ИНСТРУМЕНТАРИЙ

- 109** Zortech DOS 386 C++: простой и мощный компилятор. *Кааре Кристиан*
- 110** Пакет Object Toolkit облегчает объектно-ориентированное программирование. *Рик Эйр*

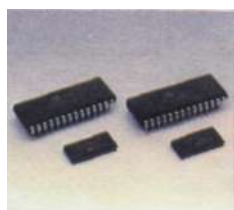
НЕ "СЛОВОМ" ЕДИНЫМ

- 111** Изменение начальных параметров печати в редакторе Word 5.5
- 112** Повышение полиграфического качества текста при работе с индексами
- 113** А WYSIWYG ли это?



НАСТАВНИК

Переназначение и конвейерная обработка. *Джефф Просис*
Знакомство с системой BIOS. *Джефф Просис*



ПОЛЬЗОВАТЕЛЬ ПОЛЬЗОВАТЕЛЮ

- 114** Переходы между каталогами
- 114** Запоминание исходного пути
- 116** Настройка AUTOEXEC. BAT
- 116** Рекурсивные командные файлы

22 августа, четверг, после дождя...

Белая асфальтовая крошка на дороге, девушка, испуганно глядящая вслед танковой колонне. Пока шла могучая техника, мороженое в руке растаяло и накапало на юбку. И сзади тихо-тихо: "Допрыгались". Наверное, немногим в это августовское утро удалось сохранять полное спокойствие и равнодушно наблюдать за тем, как в город стягиваются войска, по телевидению "крутят" симфоническую музыку (чем же классики-то виноваты?), а дикторы замогильными голосами обещают процветание, дисциплину и социалистический выбор. Кто-то, прихватив пару пачек сигарет, рванул к "Белому дому", кто-то с вытаращенными глазами начал рыскать по квартире в поисках забытого партбилета, а кто-то, сразу почуяв открывающиеся ослепительные перспективы, засел строчить первый за долгие годы вынужденного молчания донос.

В редакции PC Magazine/USSR утром 19-го под "Лебединое озеро" проходил стихийный обмен мнениями. Всем было абсолютно ясно, что при новом порядке нашему журналу не жить: ни один западный бизнесмен, будучи в здравом уме, не вложит в Советском Союзе ни цента. Настроение было просто отвратительное. Кто-то сказал, что, если народу удастся выбраться и из этого дерьма, то в России наконец-то можно будет нормально жить. Вскользь оброненная фраза вызвала только грустные улыбки — трудно было поверить, что озлобленный на всех и вся народ встанет на сторону малочисленных демократов против могучего, многомиллионного "авангарда рабочего класса".

К счастью, шесть лет перестройки не прошли даром. Полуголодная свобода показала себя лучшим агитатором, чем полусытое рабство. В эти дни демократами были все, кто выступил против диктатуры, а таких оказалось большинство. Много лет назад пропетые Булатом Окуджавой слова "Возьмемся за руки, друзья, чтоб не пропасть поодиночке" стали пророческими.

Наверное, России надо было пройти и через это испытание, чтобы поверить в себя, в свои силы, в своих людей, вновь обрести чувство собственного достоинства, без которого невозможно рассчитывать на уважение со стороны других стран и народов. Наконец-то появилась уверенность, что робкие, полужаженные демократические преобразования перерастут в подлинную свободу — свободу слова, печати, совести, предпринимательства..., российский рынок станет вожденным для крупнейших западных компаний, а отечественные специалисты будут котироваться на уровне выпускников Гарварда.



Фото Дмитрия Кучеренко



ФИРМА CPV ПРЕДСТАВЛЯЕТ: КОМПЛЕКТНОЕ СЕМЕЙСТВО СИСТЕМЫ ПЕРЕДАЧИ ДАННЫХ



Акустический модем
A+M 2400 Alpha с
принадлежностями.
Телефонный
адаптер ZZP:
№ А 200402X

- ▶ Акустические стыкующие устройства
- ▶ Акустические модемы
- ▶ Настольные модемы
- ▶ Лаптоп-модемы
- ▶ Встраиваемые платы-модемы для персональных компьютеров
- ▶ Устройства дистанционного включения персональных
- ▶ Устройства и программное обеспечение для передачи данных
- ▶ и, и, и...

CPV DATENSYSTEME GMBH
GUTENBERGRING 1-5 · 2000 NORDERSTEDT
TEL. 040/528 10 57 · FAX 040/528 25 95



Приятно себя чувствовать немножко героями, но всему хорошему приходит конец, переворотам тоже. Вместе со всеми устоял и наш журнал. Постепенно в разговорах между сотрудниками редакции политический лексикон уступил место таким привычным, греющим душу "Windows, DOS, сети, оболочки...".

Итак, во втором номере вы прочтете большую статью о магнитооптических накопителях. Быть может, в будущем они придут на смену "Винчестерам", из которых мы уже начинаем вырастать. Еще один обзор посвящен реляционным базам данных: какая лучше, какая хуже, какая для кого... Мы надеемся, что программисты найдут в этом номере кое-что и для себя: где еще опубликуют описание недокументированных функций для среды Windows, которые можно было бы вызывать из программ на Си? Как всегда, есть здесь информация и для начинающих, и для деловых людей, и для хакеров, но пересказывать содержание всего журнала в редакционной статье — дело неблагодарное.

Редакция



Я уверен, что августовские события в Москве откроют дорогу для широкого внедрения западных технологий и окажут положительное влияние на процесс компьютеризации в России.

А. Канн,
президент СК Ферлаг Интернешнл

PC Magazine/USSR is published under license from Ziff Communications Company, New York, New York. Editorial items appearing in PC Magazine/USSR that were originally published in the U.S. edition of PC Magazine are the copyright property of Ziff Communications Company, which reserves all rights. Copyright © 1991 Ziff Communications Company.

The following are trademarks of Ziff Communications Company: PC Professionale, PC, PC Magazine, PC Labs, PC MagNet, PC Magazine Award for Technical Excellence and PC Magazine Editor's Choice Award.

ИЗДАТЕЛЬ

SKVerlagGmbH
 Генеральный менеджер
И.Шляйгер
I.Schleiger
 SK Verlag GmbH
 Lulticher Str. 14, D-5000
 Cologne 1
 Tel. +(49)-221-516063
 Fax +(49)-221-529725

**ПРЕДСТАВИТЕЛЬСТВО
 В МОСКВЕ
 "СК ФЕРЛАГ ИНТЕРНЕТНЛ"**

Президент
А.Канн
 Генеральный директор
В.И.Чвертко

117607 Москва ул. Раменки, 12
 Тел.(095)931-00-03

РЕДАКЦИЯ

Главный редактор
Ю.Л.Кузьмин
 Зам. главного редактора
Р.Г.Герр
 Научные редакторы
И.Л.Шипкин
Ю.Н.Плахтий
 Художественно-технический редактор
С.Б.Мысова
 Корректор
Т.А.Шестернева
 Художник
Е.Л.Цыпин
 Оператор
А.Б.Кузовлева
 Рекламное бюро
В.Л.Демидов
 103009 Москва ул. Герцена, 5/7,
 изд-во МГУ, редакция журнала
 PC Magazine/USSR
 Тел. (095)229-52-63

Журнал сверстан в учебном центре
 СП "Интермикро".

Формат 60 x 90/8
 Гарнитура тайме.
 Печать офсетная. Усл.печ.л. **14,0**.
 Уч.-изд. л. 13,45.
 Тираж **100 000** экз. Изд. №2.
 Цена 4 р. 80 к.

LIST OF ADVERTIZERS

В настоящем выпуске помещена
 реклама следующих фирм

ZEOS International, Ltd.	2-я обл.
Кип Ying Enterprise	1
CPV Datensysteme GmbH	4
"Вертер"	6
СП "Параграф"	28
СП "Интермикро"	31, 72
"Рубикон"	34
Nantucket	36
"Ленэкспо"	39
IQ Electronics	40
СП "Унитех"	44
Softscribe, Ltd	70
Summit Systems	91
Hantarex S.U.	101
Wang Service	118
Comtek Expositions, Inc.	3-я обл.
Novell	4-я обл.

Отпечатано в типографии изд-ва
 "Калининградская правда".
 236000 Калининград, ул. Карла
 Маркса, 18
 ©1991 СК Ферлаг Интернешнл.

Полное или частичное
 воспроизведение или размножение
 каким бы то ни было способом
 материалов, опубликованных в
 настоящем издании, допускается
 только с письменного разрешения
 издательства "СК Ферлаг
 Интернешнл"

Уважаемые читатели!

С 1992 г. открыта подписка на журнал PC Magazine/USSR на русском языке. В течение года будет выпущено 8 номеров.

Стоимость подписки составляет:

29 руб. 40 коп. на полгода (номера с 1 по 4),

58 руб. 80 коп. на год (номера с 1 по 8).

Для оформления подписки достаточно перечислить ее стоимость фирме ВИКОМ на р/с 161803 КСБ "Стройкредит" в МГУ Госбанка СССР (на счет 345034), МФО 201791 и выслать в адрес фирмы ВИКОМ квитанцию почтового перевода или копию платежного поручения и заявку по следующему образцу:

фамилия, имя, отчество (или название организации)
адрес (с почтовым индексом)
количество экз. _____ срок подписки (год, полгода) _____
КОПИЯ КВИТАНЦИИ (ПЛАТЕЖНОГО ПОРУЧЕНИЯ) ПРИЛАГАЕТСЯ

ВНИМАНИЕ! Подписка ограничена!
 Прием заявок до 1 февраля 1992 г.

Фирма **ВИКОМ** с правом организации подписки и рекламы.
 127254, Москва, ул. Гончарова 6, МТЭ, ВИКОМ.
 тел.: 218-30-49



ВЕРТОП

**РАЗРАБАТЫВАЕТ,
ПРОИЗВОДИТ
НА СОВЕТСКОЙ И ЗАПАДНОЙ ЭЛЕМЕНТНОЙ БАЗЕ
ПРОДАЕТ И
УСТАНОВЛИВАЕТ «ПОД КЛЮЧ»**

**СЕТИ
МОДЕМЫ
СРЕДСТВА ТЕЛЕКОММУНИКАЦИИ**

1. Связь IBM PC XT/AT с ЕС ЭВМ

Через стойку ЕС-7920

- эмуляция терминала 7927;
- перекачка файлов со скоростью до 24 Кбайт/с;
- виртуальный винчестер на дисках ЕС;
- Связь между программами PC и ЕС;
- связь с ADABUS (NATURAL CONNECTION), SQL/DS;
- работа в локальных сетях PC-LAN, NETWARE;
- интерфейс пользователя типа NORTON COMMANDER

Цена 3100 руб.
Срок гарантии 3 года

Через стойку ЕС-7970:

- эмуляция терминала 7927;
- перекачка файлов;
- связь между программами PC и ЕС

Цена 2800 руб.

Подключение ЕС 1840 через 7920, 7970

- Возможности, как у PC

Цена 2600 руб.

2. Подсоединение ЕС ЭВМ (ЕС 1045-1068) к локальной сети типа ETHERNET под управлением NOVELL NETWARE через канал ЕС ЭВМ

- скорость до 300 Кбайт/с;
- эмуляция терминала, передача файлов, данных, электронная почта

Цена от 15 до 20 тыс. руб.

3. Локальные сети

типа ETHERNET
(собственного производства)

- Все ориентированы на сетевую среду
- NOVELL NETWARE 2.15 и более поздние версии
- 1 Мбит/с, кабель РК 75, до 600 м — 2000 руб.;
- 10 Мбит/с, кабель РК 75, до 900 м — 4100 руб.;
- 10 Мбит/с, кабель РК 50/ПК 75, до 800 м, полностью совместим с ETHERNET NE1000/NE2000 — 4200 руб.;
- 10 Мбит/с, волоконно-оптический кабель типа NE1000/NE2000 — от 10 до 15 тыс. руб.

4. Установка сетевых операционных систем NETWARE, TCP/IP, DECNET, 3+COM, BANYAN VINES, IBM PC LAN

- волоконно-оптические преобразователи;
- трансиверы, рипитеры, разветвители;
- серверы на базе AT 386/486;
- станции бесперебойного питания;
- соединители, волоконно-оптические и другие кабели;
- модемы AT-Hayes, 2400 — 5500 руб.
2400, MNP5 — 8000 руб.;
- факс-платы, факс-модемы;
- многопортовые RS-232, ИРПС, автономные коммутаторы RS-232;
- платы расширения:
ЦАП/АЦП, КОП, CENTRONICS, КАМАК, BSC, ИРПС, RS-232, выносные шины

ВПЕРВЫЕ В СССР!

РАЗРАБОТЧИКАМ РЭА, СНАБЖЕНЦАМ:

1. База данных "Советские микросхемы и компоненты"

- Все о 6000 микросхем, 15 000 конденсаторов, 8000 резисторов
- Более 30 Мбайт информации — размеры, электрические параметры, типы корпусов, адреса изготовителей, цены по прейскуранту 1991 г.
- База данных для IBM PC или ЕС ЭВМ

Цена 1100 руб.

2. Библиотеки P-CAD 4.50

- Более 1000 элементов

Цена 950 руб.

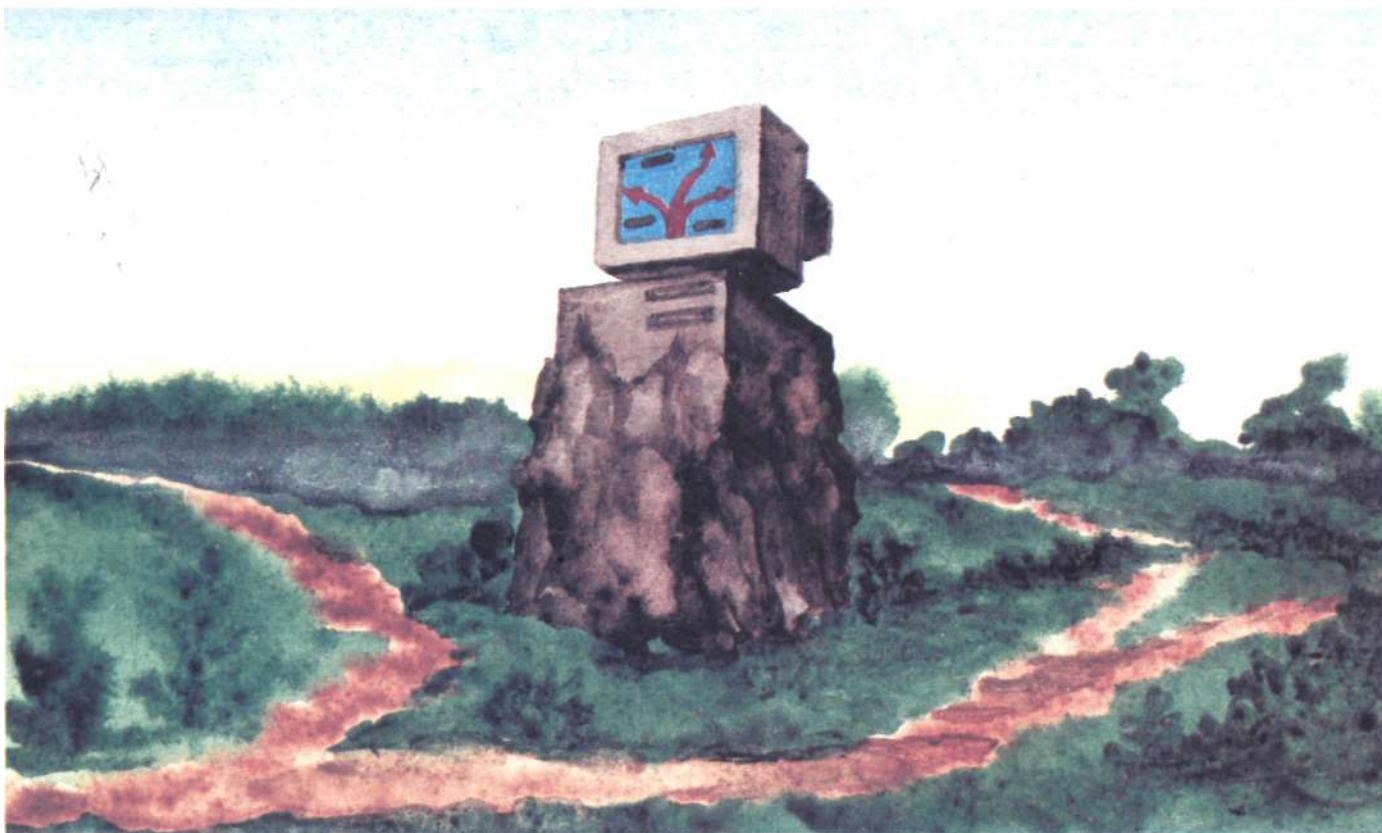
Поставка

по договору, гарантирующему сопровождение
и обновление базы

Цены даны без стоимости дискет (или магнитных лент)
и почтовых расходов.

Заявки принимаются только в письменном виде по адресу:

117049 Москва,
а.я. 58, "Мегабит".
Тел.: (095)297-55-82; (095)125-00-06
Факс (095)125-87-59



Второй Международный компьютерный форум в Москве. До встречи через год!

Илья Липкин

Люди тянутся друг к другу, и это не знамение времени — так было всегда. Как полюса магнита, слабые тянутся к сильным, глупцы к мудрецам, мужчины к женщинам (и наоборот). Людей объединяют общие интересы, профессия, язык, любовь к искусству, наконец, просто непреодолимое желание общения с интересным собеседником.

Существует по меньшей мере несколько версий того, что же в действительности следует считать венцом творения. Например, один из героев братьев Стругацких не без оснований полагает, что это отнюдь не ЧЕЛОВЕК — ЦАРЬ ПРИРОДЫ, а рюмка коньяку с ломтиком лимона. С другой стороны, если задуматься, нормальный интеллигент в одиночку не будет пить даже "Камю" под марокканские цитрусы — нужна компания. Следовательно, и человек как личность, и классный "дринк" являют собой, по сути, только переходную ступень от оди-

ногого унылого неандертальца к хорошей тусовке.

Не теряя времени на этимологию слова "тусовка", примем его в качестве термина, определяющего временное, динамически уравновешенное сообщество людей, половина из которых находится в состоянии перманентного общения, а остальные, не понимая ни слова, делают умные лица, чтобы не выйти из состояния сопричастности.

Разные исторические эпохи были богаты самыми разнообразными тусовками. Вспомним ассамблеи петровских времен, маевки начала века или подпольные "сейшены" джазовых фанов. Все эти, на первый взгляд, так непохожие друг на друга события объединяет одно — ни с чем не сравнимое удовольствие посвященных, общающихся на языке, абсолютно непонятном для окружающих.

Кстати, синонимами термина "тусовка" кроме стародавних "ассамблея" и "маевка" можно считать та-

кие слова, как "съезд", симпозиум, "сессия", "комитет" и в последние годы ставшее очень модным слово "форум".

Так вот, вся эта довольно-таки длинная вводная часть была просто необходима для того, чтобы читатель как можно острее ощутил атмосферу Второго Международного компьютерного форума (МКФ), проходившего с 17 по 20 июня 1991 года в Москве под эгидой Международного компьютерного клуба (МКК).

Очевидно, из-за того, что программисты и электронщики любят собираться в стаи, всякого рода массовые компьютерные мероприятия проводятся у нас в стране чуть ли не раз в две недели. Этак немудрено и затеряться, и все-таки Второй МКФ стал заметным событием в мире информатики.

Хлопотное, но приятное бремя спонсорских забот легло на плечи четырех фирм из СССР, США и Германии, а первую скрипку здесь сыграла американская компания ZEOS International, широко известная во всем мире своими надежными, быстрыми и, что немаловажно, недороги-

Международный компьютерный клуб (МКК) был создан в декабре 1988 г. в Москве советскими и иностранными организациями с целью объединения интересов советских и зарубежных разработчиков и пользователей современных информационных технологий.

ми вычислительными системами. Лидирующее положение и великолепная продукция фирмы неоднократно отмечались на страницах самых престижных компьютерных изданий. На московском форуме присутствовал Грег Херрик, президент ZEOS International.

Второй спонсор, фирма Nantucket, хорошо знакома советским пользователям благодаря своим популярным пакетам серии Clirreg. Отечественные программисты должны быть особенно благодарны руководству фирмы, поскольку изделия с маркой Nantucket одними из первых стали продаваться за рубли. На МКФ фирму представляла Вирджиния Клаф, директор по оперативному управлению в СССР.

Внес свой вклад в осуществление проекта проведения второго компьютерного форума Главный научно-исследовательский вычислительный центр Лечебно-оздоровительного объединения при Совете Министров СССР во главе с директором Р.А.Эльчиным.

И наконец, в самом конце списка спонсоров скромно пристроился красный от смущения фирменный знак новорожденного PC Magazine/USSR. На форуме присутствовало руководство журнала и вся его редакция (рекомендуем внимательно посмотреть с. 4).

О своем участии в МКФ заявило более 35 зарубежных фирм, среди которых были такие киты, как Apple Computer, Borland International, BYTE, Computerland, Cray Research, Digital Research, Intel, Seagate, Unix International и др.

Небольшой выставочный зал Центра международной торговли, что на Красной Пресне, удивительно контрастировал с душной тридцатиградус-

ной Москвой не столько своим прохладным и чистым воздухом, сколько царившей здесь непринужденной и доброжелательной обстановкой. Языкового барьера не существовало. Вообще, компьютерный форум выгодно отличается от многих других выставок в первую очередь тем, что в зале практически отсутствуют зевачки, а профессионалы из любых двух стран всегда найдут общий язык. Переходя на серьезный тон, можно сказать, что здесь было немало действительно важных и интересных событий.

Начнем с главного. В рамках форума состоялась Всемирная телеконференция, объединившая примерно 60 компьютерных клубов из США, Канады, Австралии, Англии, Франции, ФРГ, СССР и Японии. Экспоненты и посетители смогли стать участниками телекоммуникационного "круглого стола" и обменяться мнениями со своими коллегами во всем мире в режиме реального времени. Технически это стало возможным благодаря использованию системы GlobalNET Ассоциации пользователей персональных компьютеров (APCUG), членом которой, как известно, является и Международный компьютерный клуб.

На МКФ под руководством и при непосредственном участии ведущих разработчиков из фирмы Borland International проводился Первый летний семинар Ассоциации пользователей продукции Borland International в СССР (БорАГ). Были также подведены итоги конкурса "Борланд Контест", и тут уж дело не обошлось без МКК, СП "Интерквадро" и СП "Новинтех" — учредителей БорАГ. СП

"ПараГраф" представило на суд специалистов свой программный комплекс "Русское слово", предназначенный для работы с русским текстом на компьютере. Презентация фирмы Seagate проходила под девизом "Инновационные технологии в области изготовления жестких дисков". Именно здесь автор данной статьи понял, что слова "Seagate" и "надежность" — почти синонимы.

Во время форума на сцене уютного конференц-зала ЦМТ выступило более двух десятков докладчиков из стран Западной Европы, США и СССР. Темы выступлений охватывали весь спектр современной индустрии информатики — от анализа состояния компьютерного рынка до проблем правовой охраны программ в СССР. Если учесть, что одной из основных целей Второго МКФ было "вовлечение советских специалистов в международное разделение труда, создание наиболее благоприятных условий для реализации их интеллектуального потенциала", становится очевидной особая важность правовой охраны интеллектуальной собственности.

Кстати, конференц-зал ЦМТ стал местом проведения еще одного очень милого мероприятия — презентации PC Magazine/USSR. Природная скромность не позволяет мне в деталях описать это действие (между прочим, частично показанное по ЦТ), но само присутствие на церемонии первых лиц компьютерного мира говорит о многом.

Думается, что Второй МКФ послужил благородному делу сближения компьютерных специалистов из разных стран, помог им обменяться идеями и опытом и еще на один шаг приблизил "наших" к сегодняшнему уровню мировой информатики.

Для Apple - представитель в СССР, для советских пользователей — новая кодовая таблица

Рубен Герр

11 июля 1991 г. фирма Apple Computer Inc. (США), СП Интермикро (СССР — Австрия) и финансовая группа Prosystem (Австрия) подписали соглашение, в рамках которого Интермикро будет выступать в качестве ШС фирмы Apple в Советском Союзе. Это означает, что отныне продажу и обслуживание всех изделий Apple, в том числе всемирно известных компьютеров Macintosh, будет организовывать Интермикро (непосредственно или через своих партнеров).

Фирмы Apple и Интермикро весьма серьезно подошли к задаче вывода компьютеров Macintosh на советский рынок: будут орга-

низованы центры обучения и обслуживания, все наиболее важные программы переведены на русский язык. Но, увы, — для кириллицы был принят свой, новый вариант кодировки.

Остается только сожалеть, что в свое время резиновый призыв Польшаева о переводе делопроизводства на латинский алфавит не нашел должного отклика. Пока существует вычислительная техника, нас преследуют проблемы, связанные с необходимостью заставить компьютер "говорить по-русски" (равно, впрочем, как и по-украински, белорусски и т.д.). Были у нас

коды КОИ-7 и КОИ-8, были у нас ДКОИ-7 и 8, были у нас "основная", "болгарская" и "альтернативная" кодировки. И вот, не успела "альтернативная" кодировка утвердиться в качестве фактического стандарта, как появились особая кодировка для Windows и еще одна для компьютеров Macintosh. Нельзя сказать, что кодировка Macintosh не похожа ни на что: прописные буквы расположены так же как в альтернативной кодировке, а вот строчные — так же как в кодировке Windows. Очень "удобно", не правда ли?

Представители Интермикро утверждают, что предназ-

наченные для СССР компьютеры Macintosh будут обязательно снабжены и программами для преобразования шрифтов, но для авторов русифицированных версий программ, которые хотели бы встроить в свои пакеты средства экспорта файлов, мы на следующей странице помещаем таблицу кодировки Macintosh. Эта кодировка, кстати, прошла утверждение по крайней мере в трех отделениях разных уровней фирмы Apple и вызвала бурные дискуссии в советских организациях, уже работающих на компьютерах Macintosh.

СОБЫТИЯ

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0																
1		☘	✓	◆	🍏											
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	А	В	С	Д	Е	Ф	Г	Н	И	Ј	К	Л	М	Н	О
5	Р	Q	R	S	T	U	U	Ш	Х	У	Z	[\]	^	_
6	,	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	р	q	r	s	t	u	u	ш	х	у	z	{		}	~	
8	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
9	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A	†	°	¢	£	§	•	¶	®	©	™	™	™	™	™	™	™
B	∞	±	≤	≥	і	ц	д	Ј	Є	є	і	і	љ	љ	њ	њ
C	Ј	S	т	✓	f	≠	Δ	«	»	...	™	ћ	ќ	ќ	s	
D	—	—	“	”	‘	’	÷		Ў	ў	Ц	ц	№	Ё	ё	я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	

- ASCII в нестандартной позиции
- символы кириллицы

"ТИХОЕ" РЕШЕНИЕ ЦЕНОЙ В 10 000 000 000 РУБЛЕЙ

Леонид Малков

В СССР даже потрясения порой оказываются какими-то малозаметными, и потому не особенно потрясают воображение.

31 мая 1991 г. в Москве произошло историческое событие, которого кто-то ждал, кто-то опасался, а большинство о нем и не ведало. Тем не менее ему, очевидно, суждено сыграть определенную роль в развитии советской индустрии программирования. В этот день было подписано Постановление Верховного Совета СССР "О введении в действие Основ гражданского законодательства Союза ССР и республик"*, где, в частности, недвусмысленно говорилось о правовой охране программного обеспечения ЭВМ.

По самой грубой оценке, на рядом персональном компьютере установлено программного обеспечения тысяч этак на 10 рублей, и практически все оно — суть, несанкционированные копии, изготовление которых непосредственно затрагивается новым нормативным актом.

Откуда взялась эта цифра? Давайте прикинем. Причем будем исходить именно из рублевых цен, так как большинство программных продуктов в СССР можно вполне легально и в русифицированном виде купить за рубли.

Начнем с утилиты Norton Commander. Во всем Союзе вряд ли найдется компьютер, где ее нет. Фирма Symantec, поглотившая компанию Peter Norton и соответственно

получившая права на ее изделия, любезно продаст Commander в СССР за рубли — всего за 1300. Видите — 13% ориентировочной суммы у нас уже есть. Приступим теперь к поиску остальных 87%.

На мировом рынке доминируют три основных класса прикладных программ: это редакторы текстов, базы данных и электронные таблицы. В СССР наибольшая доля несанкционированных копий приходится именно на эти изделия. Не секрет, что далеко не все программы доступны за рубли, поэтому можно считать цены на пакеты одного класса приблизительно одинаковыми.

РЕДАКТОРЫ ТЕКСТОВ:

Здесь самый нижний уровень представлен редактором Лексикон. Если на вашем компьютере есть эта программа, считайте, что вы не доплатили СП "Параграф" от 500 до 1000 руб. в зависимости от наличия (или отсутствия) в вашей версии встроенного блока контроля орфографии. Если же вы пользуетесь текстовым про-

цессором MS-WORD, ежегодные тиражи которого во всем мире составляют сотни тысяч экземпляров, то тот же "ПараГраф", продающий эту программу по лицензии Microsoft в составе пакета Русское Слово, не достигался уже почти 3000 руб. Кстати, продаваемая отдельно программа проверки орфографии, например Орфо, стоит более 1000 руб.

СУБД:

Пакет Paradox фирмы Borland, уверенно обошедший всем известную систему dBase по количеству проданных копий, можно приобрести в СССР примерно за 10 тыс. руб. СУБД Clipper стоит сегодня около 15 тыс. руб.

ЭЛЕКТРОННЫЕ ТАБЛИЦЫ:

Цена пакета Lotus 1-2-3 — 5000 руб. Его основной конкурент — QuattroPro фирмы Borland — в русифицированном виде стоит приблизительно столько же (американская версия продается дешевле).

КОМПИЛЯТОРЫ С ЯЗЫКОВ ПРОГРАММИРОВАНИЯ:

Turbo C++ 1.0x Professional предлагают за 4200 руб., а на Turbo Pascal 6.0 Professional установлена цена 5775 руб.

ИНТЕГРИРОВАННЫЕ ПАКЕТЫ:

Пакет Мастер (СП "ПараГраф") стоит в зависимости от версии до 5000 руб.

За модную нынче оболочку Windows вам придется заплатить никак не менее 4000 руб.

Из приведенных выше цен видно, что пара фирменных пакетов, установленных на одитиг-персональном компьютере, влетает в "копеечку" — что-то около 10 тыс. руб. Можно без преувеличения сказать, что на деле эта цифра должна быть куда выше.

На Западе стоимость программ, приходящихся на один компьютер, сопоставима с ценой самого "железа". Поскольку в Союзе IBM-совместимые машины в среднем стоят от 30 до 50 тыс. руб., 10 тыс. руб. за прикладные пакеты, согласитесь, — цена умеренная.

Фактически импортное программное обеспечение в СССР в сравнении с курсом доллара продается относительно дешево. Если сейчас обменный курс балансирует между 40 и 50 рублями за доллар, то аналогичный курс в пересчете на программное обеспечение составляет около 10 — 15 рублей за доллар. Скажем, в США цена пакета Paradox 3.5 (последняя версия) по каталогу — 795

долл., а в СССР он стоит 9900 руб. (коэффициент: $9900/795=12,45$).

Причина такой ценовой политики заключается, во-первых, в том, что и эти-то цифры кажутся несоразмерно высокими в сравнении с зарплатами сотрудников, работающих на компьютерах, а, во-вторых (и это главное), в колоссальном распространении несанкционированного копирования программ.

Несмотря на то что законодательные решения о правовой охране программ во многих странах были приняты относительно недавно (в США в 1980 г., во Франции в 1985 г.), нигде отсутствие соответствующих правовых норм не было связано с таким гигантским по масштабам вольным копированием, как в СССР. По сути, во всех цивилизованных странах закон закреплял реально существовавшие отношения между производителями и пользователями, в то время как в Союзе речь может идти только о полном переосмыслении этих отношений.

Исключительность ситуации усугубляется еще и тем, что советские пользователи видят чуть ли не особый шарм именно в вольном тиражировании фирменного программного обеспечения. Нельзя сказать, что на Западе компьютерные воры перевелись совсем, но, если смотреть на них сквозь призму общественного мнения, то они вне закона. Позитивное отношение в СССР к несанкционированному копированию можно как-то объяснить, в частности, былым добрым отношением интеллигенции к самиздату. Не секрет, что многие книги десятилетиями не издавались официально и нелегальное распространение выглядело гражданской добродетелью (кстати, эта деятельность зачастую сурово каралась). Кажется, нечто подобное происходит и сейчас, но только в отношении программ для ЭВМ.

До недавнего времени (до появления в СССР различных форм собственности) серьезно наладить независимое производство конкурентоспособных отечественных программ было практически невозможно, да и западные программные продукты за рубли не продавались. Ситуация в корне изменилась несколько месяцев назад: зимой начались поставки Lotus 1-2-3, в апреле — редактора Microsoft Word в составе пакета Русское Слово. Решение фирмы Borland о продаже всех своих изделий за рубли было принято в июле, а компания Sumantec начала поставки за рубли в конце лета 1991 г. Вместе с тем, по-

требность в профессиональном программном обеспечении была насущной с самого момента появления в СССР первых IBM-совместимых компьютеров. В результате, отношение программистов к пиратскому распространению программ можно охарактеризовать следующим образом: если государственные экономические структуры препятствуют нормальному ходу дел, мы сами решим свои проблемы. Подобная реакция имела не экономический, а скорее идеологический подтекст: как аукнется — так и откликнется. Ныне такое мировоззрение постепенно превращается в анахронизм. В частности, при теперешних темпах экономизации отношений несанкционированное копирование является скорее не формой идеологического протеста, а вариантом черного рынка (получение прибыли лицами, не имеющими прав на продукт, или уклонение от законного приобретения легкодоступного товара).

Важная составляющая борьбы за легальный рынок программ в СССР — деидеологизация, демифологизация, деидеализация свободного копирования программ.

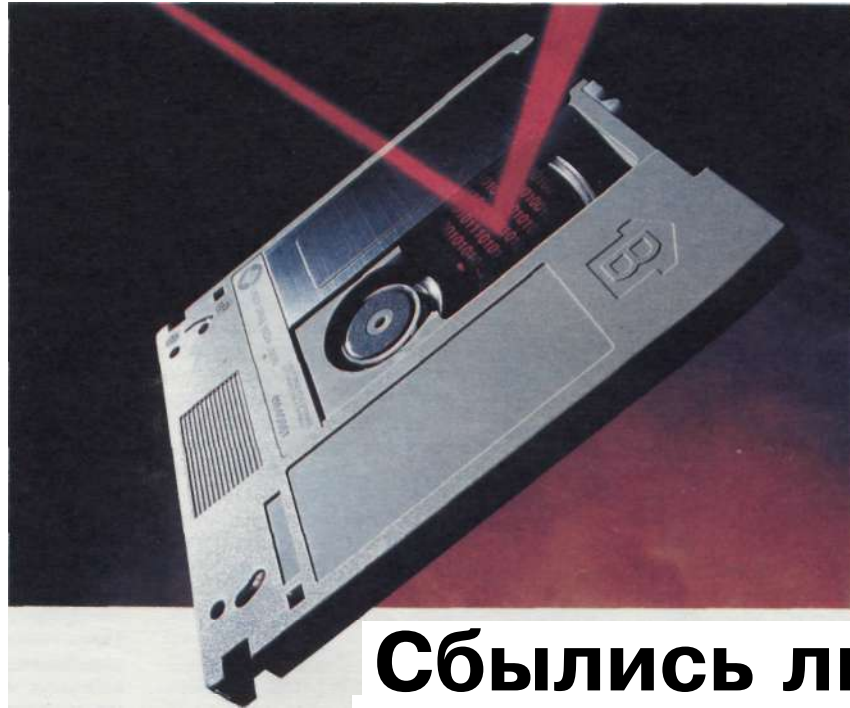
Но вернемся к принятому постановлению. Можно не сомневаться, что подробный анализ и дискуссии по поводу его содержания будут регулярными в профессиональной компьютерной периодике. Отметим лишь, что четвертый раздел "Основ" посвящен авторскому праву. К примеру, статья 134 прямо указывает, что программы для ЭВМ являются объектом авторского права. Пикантная деталь: в списке объектов программам предшествуют произведения садово-паркового искусства и картографии, а сразу после программ упоминаются произведения хореографии и пантомимы. Очевидно, с точки зрения законодателей, что программы, что пантомима, что парковые ансамбли — все едино.

Итак, подведем итог: 10 тыс. руб. умножаем на 1 млн. персональных компьютеров (при самой грубой оценке парка ПК в СССР). В результате получается внушительная сумма — 10 000 000 000 руб. Если вам кажется, что здесь слишком много нулей, вспомните о больших и мини-ЭВМ, которые вовсе не принимались в расчет.

Словом, если в отношении программного обеспечения "Основы" трудно назвать совершенным документом, то возможные последствия их принятия выглядят достаточно впечатляюще.

" Полный текст "Основ" был напечатан в "Ведомостях Съезда Народных депутатов СССР и Верховного Совета СССР", №261 (26 июня) 1991 г.

Магнитооптические диски:



Сбылись ли наши мечты о перезаписываемых оптических дисках?

Они быстрее гибких, но медленнее жестких дисков. Они дороже жестких дисков, но дешевле 8-мм стримеров. Они надежны и легко заменяемы. Они могут быть многократно перезаписаны. Но остановится ли на магнитооптических накопителях ваш окончательный выбор?

М. Дэвид Стоун

С тех пор, как в 1972 году первые лазерные видеосистемы продемонстрировали чудесные возможности оптических дисков с однократной записью, мир томился ожиданием появления подобных устройств, но обладающих возможностью повторной записи. Несмотря на то, что фирмы-изготовители давно объявили о существовании необходимой технологии, рынок бытовой электроники и по сей день кормит своих потребителей обещаниями скорого появления перезаписываемых (или стираемых) компакт-дисков и соответствующей аппаратуры. Несколько иное положение сложилось на рынке персональных компьютеров. Здесь фирмы-изготовители уже разрешились множеством разнообразных периферийных устройств, способных осчастливить

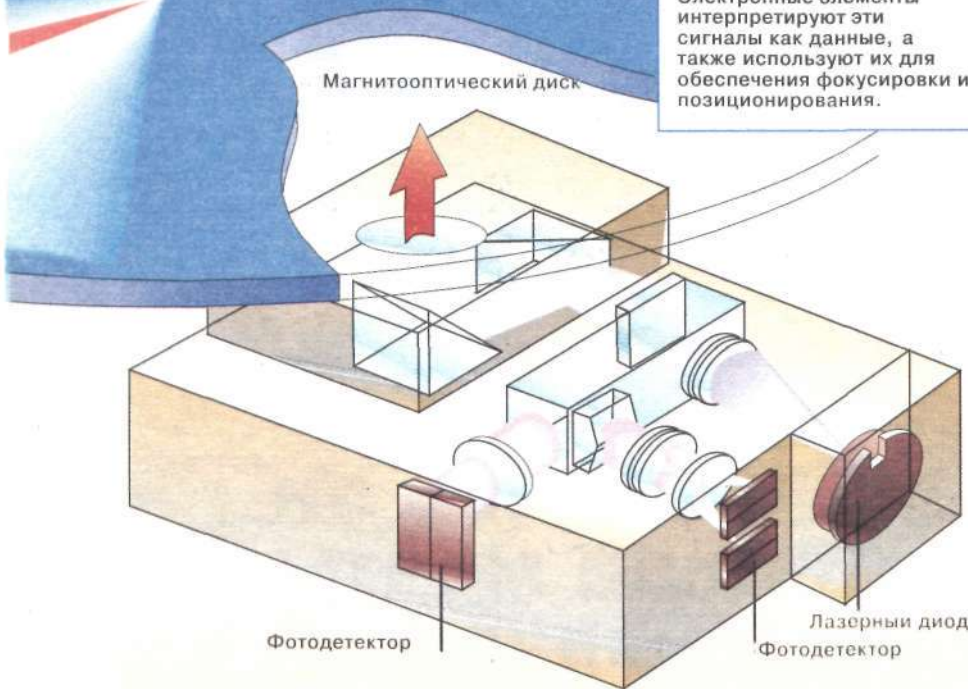
КАК РАБОТАЕТ МАГНИТООПТИЧЕСКИЙ НАКОПИТЕЛЬ



As the name suggests, magneto-optical drives use both magnetism and light to read from and write to a disk. Writing to disk actually takes two passes: an erase pass and a write pass. Reading the disk takes only one pass and is therefore faster than writing.

ОПТИЧЕСКАЯ ГОЛОВКА

Блок чтения/записи ("головка") в магнитооптических накопителях представляет собой систему линз, зеркал и призм, обеспечивающих передачу лазерного луча в точку чтения/записи и обратно. Источником света является лазерный диод. Свет в электрические сигналы преобразуют фотодетекторы. Электронные элементы интерпретируют эти сигналы как данные, а также используют их для обеспечения фокусировки и позиционирования.



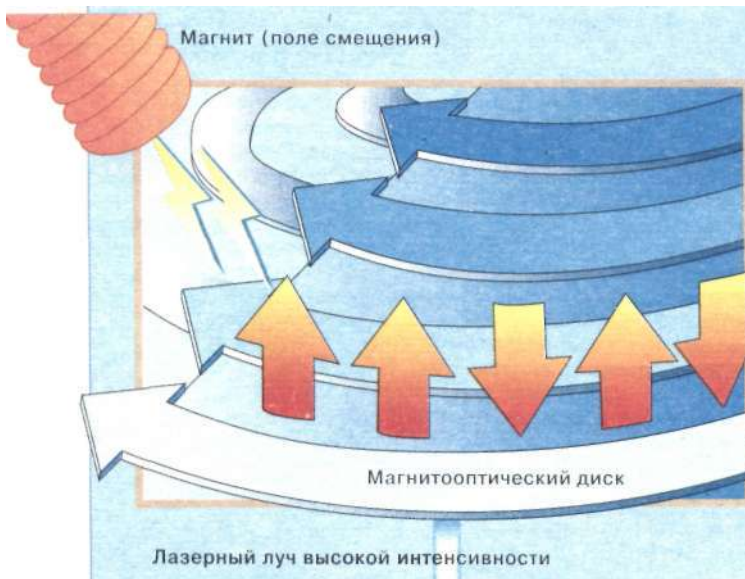
пользователей чудесными возможностями стираемых оптических дисков.

Самые жизнеспособные оптические приборы, обладающие возможностью перезаписи и доступные уже сегодня, — это магнитооптические (МО) диски, наследующие достижения магнитной и оптической технологий записи. Появившиеся в 1988 году МО-диски соединили в себе портативность гибкого диска и накопителя Vernoulli Vox, великолепную среднюю скорость доступа жесткого диска, надежность КДПЗУ и емкость, сравнимую с DAT.

Но прежде, чем покупать эти сказочные оптические накопители, учтите, что у них есть слабые стороны. Они сравнительно дороги, они пока еще не могут сравниться по вместимости с жесткими дисками большого объема, и, кроме того, они медленнее современных жестких дисков (см. врезку "От гибкого диска к магнитооптическому диску: на чем остановить свой выбор").

В данном обзоре представлено 17 внешних магнитооптических накопителей. Это законченные самостоятельные устройства, рассчитанные на конечного пользователя, в комплект которых входят контроллер SCSI, МО-кассета, дисковод и программное обеспечение. Цены их колеблются от 3 до 7 тыс. долл. Сравните это со стоимостью 300-Мбайт жесткого диска, цена которого не превышает 1,5 — 2 тыс. долл. Причем 300 Мбайт (вместимость одной стороны МО-кассеты) — далеко не предел для жестких дисков, ведь известно, что сегодня их максимальная емкость может достигать 1,2 Гбайт, тогда как современные магнитооптические диски позволяют записать не более 500 Мбайт информации на каждой стороне. Кроме того, жесткий диск быстрее, чем

ЧТЕНИЕ И ЗАПИСЬ В МАГНИТООПТИЧЕСКОМ НАКОПИТЕЛЕ



Магнитооптические дисководы записывают информацию на диск с помощью магнитного поля, называемого полем смещения, и луча лазера/Рабочий слой диска сохраняет данные в виде участков намагниченности разной полярности (доменов). Вертикальные стрелки указывают направление намагниченности доменов. Когда определенная точка на диске нагрета лазерным лучом достаточно высокой интенсивности, ее "сопротивляемость" изменению полярности резко падает. Поле смещения может теперь изменить намагниченность прогретых участков для записи информации. В процессе чтения используется лазерный луч, мощность которого недостаточна для критического нагрева.

МО-накопитель. Тестируемые МО-накопители выполняли в лаборатории PC Magazine DOS⁶recT скорости доступа, в среднем, примерно в четыре раза медленнее, чем жесткий диск компьютера Compaq Deskpro 386/25e, на котором, кстати, и проводились измерения (см. врезку "Магнитооптические диски: быстрее, чем гибкие, и медленнее, чем жесткие диски"). Поэтому не выбрасывайте пока свой старый испытанный винчестер — в качестве основного накопителя он все еще вне конкуренции.

ВТОРИЧНЫЕ НАКОПИТЕЛИ

Для того чтобы обеспечить, скажем, хранение файлов больших размеров, вам не обойтись без МО-дисков, которые могут оказаться поистине идеальными накопителями. Не найдете вы более надежного средства и для резервного копирования жесткого диска, хотя это весьма дорогое удовольствие.

Сочетание большой емкости, сменяемости и возможности перезаписи делает МО-накопитель превосходным средством решения задач, требующих значительных объемов дискового пространства (например, графических задач). Напомним читателям, что PC Magazine почти на 100% создается с помощью компьютеров. А одна только обложка журнала может занимать до 40 Мбайт дисковой памяти. Мы должны порой в течение нескольких месяцев "иметь под рукой" графический файл такого размера.

Существует немало прикладных задач, требующих больших объемов дискового пространства и словно созданных для МО-накопителей, — это обработка изображений в САПР, запись почтовых звуковых сообщений, а также сообщений, передаваемых посредством факсимильной связи. Что же касается комплексного пред-

ставления информации, или, как его называют профессионалы, "мультимедиа", то МО-накопители пока еще не обладают достаточно высокой скоростью доступа, чтобы обеспечить необходимую реактивность системы. В среднем, тестируемые МО-накопители работали на компьютере Deskpro, всего в три раза быстрее, чем НГМД.

Само собой напрашивается применение МО-накопителей для резервного копирования жесткого диска. По своим функциональным характеристикам МО-накопители значительно превосходят стримеры, поскольку являются устройствами с произвольным доступом. Если ваш жесткий диск вышел из строя и его нужно восстановить с помощью резервной копии на стримере, то, прежде чем воспользоваться спасенными данными, вам придется ждать установки нового жесткого диска. В случае же, когда резервная копия жесткого диска хранится на МО-диске, вы можете спокойно продолжать работать, хотя и с несколько меньшей производительностью.

Сменяемость, необходимая при работе с информацией, требующей секретности, делает МО-накопители особенно привлекательными — вы можете спокойно запереть свои данные на ночь в сейф, нимало не заботясь об охране компьютера. Преимущество МО-дисков перед гибкими дисками становится особенно ощутимым, если вы представите себе процесс перевозки с работы домой и обратно пары сотен мегабайт прикладных программ и данных на обычных пятидюймовых дискетах...

Наконец, последнее преимущество МО-накопителей состоит в том, что

сама технология обеспечивает защиту от физического разрушения информации, поскольку в процессе работы головка чтения/записи не соприкасается с поверхностью диска (как это происходит у винчестеров или гибких дисков).

ЛАЗЕРНЫЙ ЛУЧ И МАГНИТНОЕ ПОЛЕ

Как правило, способы записи и чтения информации являются либо чисто магнитными (жесткие диски, гибкие диски, устройства записи на магнитную ленту), либо чисто оптическими (например, WORM-накопители). Магнитооптические накопители, как это следует из названия, объединили в себе и магнитную, и оптическую составляющие. Накопитель записывает данные (представленные в виде колебаний магнитного поля) на диск при помощи магнитного поля и луча лазера. Считывание информации с диска производится посредством одного только лазера. При этом электроника накопителя анализирует эффект влияния на отраженный лазерный луч структур, сформированных на диске в процессе записи.

В целом работа магнитооптического диска подобна работе любого другого магнитного устройства. Поверхность диска покрыта специальным магнетиком, а магнитное поле, называемое полем смещения, используется для изменения намагниченности определенных участков диска. Однако, в отличие от чисто магнитных материалов, записывающий слой устроен так, что при нормальной температуре интенсивность поля смещения оказывается недостаточной, чтобы как-то изменить намагниченность диска. Вот тут-то лазерный луч и вступает в игру. (См.



АППАРАТНЫЕ СРЕДСТВА

диаграмму "Чтение и запись на магнитооптическом накопителе".)

Эффект сопротивления ферромагнетика воздействию внешнего магнитного поля носит название коэрцитивной силы, причем, чем она выше, тем меньше вероятность случайного перемангничивания материала побочными магнитными полями. В этом-то и кроется одно из преимуществ МО-дисков перед традиционными магнитными накопителями. Положите постоянный магнит на гибкий диск, и вы почти наверняка уничтожите всю записанную на нем информацию. Положите тот же магнит на магнитооптический диск — и ничего страшного, скорее всего, не случится.

Безусловно, высокая коэрцитивность усложняет задачу осуществления записи и стирания, но оказывается, вы можете управлять этой характеристикой вещества, просто изменяя его температуру, — коэрцитивная сила ферромагнетика уменьшается с увеличением температуры. При сравнительно низких температурах этот эффект проявляется незначительно, однако все магнитные материалы имеют критическую температуру, называемую температурой Кюри, при которой коэрцитивная сила резко падает до нуля.










УВЕЛИЧЕНИЕ ТЕМПЕРАТУРЫ

Для магнитооптических дисков температура Кюри составляет приблизительно 150 градусов Цельсия (или 300 градусов по Фаренгейту). Это значительно выше той температуры, при которой диск может эксплуатироваться или храниться. Однако эта величина достаточно низка для того, чтобы лазерный луч мог быстро нагреть магнитный слой до критической точки. Когда некоторая микрообласть на поверхности диска нагрета, она может быть подвергнута воздействию магнитного поля смещения, которое изменит ее намагниченность. После этого рабочая поверхность быстро ос-



ОТ ГИБКОГО ДИСКА К МАГНитоОПТИЧЕСКОМУ:

Никто еще не изобрел накопитель, который удовлетворил бы всех пользователей персональных компьютеров. Поэтому ваш выбор должен основываться на таких характеристиках, как емкость, цена и производительность. Мы провели тестирование наиболее распространенных средств хранения данных — от гибких дисков до магнитооптических устройств — по следующим функциональным показателям: - работа в качестве первичного накопителя (для хранения программ и данных, время доступа к которым должно быть минимальным), - работа в качестве вторичного накопителя (для хранения программ и данных, которые должны быть под рукой, но время доступа к ним не столь критично), - резервное копирование (для обеспечения сохранности и создания архива). Оценки (отлично, хорошо, приемлемо или слабо) основываются на исследовании потребностей основного контингента пользователей. Правда, не исключено, что у вас есть свои специфические требования, которые могут превратить в целом неважный накопитель в устройство, решающее все ваши проблемы. Например, сменяемость оптических дисков может быть для вас более важным критерием, чем производительность (вспомните о секретности данных!), а жесткие требования к надежности (устойчивости к разрушению информации) могут оказаться решающим аргументом в их пользу. Для каждой из рассматриваемых технологий мы поместили информацию о емкости накопителя и его цене, а также об основных фирмах-изготовителях устройств такого типа. Таблица составлена в обратном хронологическом порядке.

	Дата выпуска	Первичная память	Вторичная память	Резервное хранение
 Магнитооптические диски и фазопеременные диски	1988	Слабо	Отлично	Отлично
 Магнитная лента на 4-мм кассетах	1988	Неприменительно	Неприменительно	Отлично
 Магнитная лента на 8-мм кассетах со спиральным считыванием	1987	Неприменительно	Неприменительно	Отлично
 Диски с однократной записью	1985	Слабо	Хорошо	Слабо
 Магнитная лента на мини-кассетах 6,35 мм DC-2000	1984	Неприменительно	Неприменительно	Отлично
 Сменные кассетные диски (Bernoulli)	1983	Хорошо	Хорошо	Хорошо
 Жесткие диски типа "Винчестер"	1974	Отлично	Неприменительно	Хорошо
 Магнитная лента на мини-кассетах 6,35 мм DC-6000	1972	Неприменительно	Неприменительно	Отлично
 Гибкие диски	1971	Слабо	Неприменительно	Слабо

АППАРАТНЫЕ СРЕДСТВА

тывает и восстанавливает свою высокую коэрцитивность.

Чтобы записать какую-либо информацию, во время вращения диска электроника МО-накопителя включает и выключает лазер. Когда определенные участки поверхности диска оказываются в зоне действия поля смещения, лазер выборочно нагревает их, в результате чего изменяется их намагниченность.

В то время как лазер включается и выключается, а магнитное поле остается постоянным, на поверхности диска за один цикл вращения могут образоваться зоны намагниченности одной и той же полярности. Для получения стандартного двоичного кода

необходимо нанести на диск микрообласти двух противоположных полярностей, но устройство не в состоянии сделать это в течение одного цикла, поскольку поле смещения не может изменяться достаточно быстро.

Чтобы справиться с этой проблемой, современные МО-накопители используют двухпроходную схему записи. Во время первого прохода накопитель стирает информацию с того участка диска, на который он в дальнейшем будет производить запись. В процессе стирания лазер постоянно включен, а поле смещения изменяет

полярность на противоположную той, которая будет использована в режиме записи. Попросту говоря, цикл стирания сбрасывает все данные на участке записи в нули, а цикл записи вставляет в нужных местах единицы. (В скором времени на рынке должны появиться устройства, базирующиеся на новых оптических технологиях, которые могут осуществлять запись на диск за один проход. Для получения дополнительной информации о таких технических приемах, как изменение фазового состояния, использование окрашенных полимеров, модуляция магнитного поля и модуляция мощности светового луча см. врезку "В поисках самого быстро-

на чем остановить свой выбор

Основной производитель устройств	Цена дисковой подсистемы (тыс. долл.)	Основной производитель носителей информации	Максимальная емкость диска или ленты	Цена носителя информации (долл.)	Цена в расчете на мегабайт (долл.)
Canon, Hitachi, Matsushita, Maxoptix, MOST, Pioneer, Ricoh, Sharp, Sony	2,7-6,0	Asahi Chemical, Canon, Dalcet Chemical, Fuji Photo, Hitachi, Hoechst, Kasei, Kuraray, Kyocera, Maxell, Mitsubishi, Mitsui, NEC, Philips/Dupont Optical, Pioneer, Ricoh, Seiko Epson, Sharp, Sony, Sumitomo Chemical, TDK, 3M, Tosoh, Verbatim	1 Гбайт	130-250	0,13-0,25
ArDat, Giga Trend, Hewlett-Packard, Sony, WangDAT, Wangtek	2,5-6,0	BASF, DIC Digital, Fuji Photo, JVC, Maxell, Sony, TDK	2,5 Гбайт	30-45	0,01-0,02
Exabyte	7,0-8,0	BASF, DIC Digital, Fuji Photo, Maxell, Sony, TDK	5 Гбайт	40	0,01
Hitachi, ISI, LMSI, Mitsubishi, Optimem, Panasonic, Pioneer, Ricoh	2,5-4,0	Maxell, Philips, /Dupont Optical, Pioneer, Plasmon, TDK	1 Гбайт	100-200	0,10-0,20
Archive, Colorado Memory Systems, Irwin, Mountain, Wangtek	0,4-1,4	Carlisle, Sony, 3M	150 Мбайт	30-40	0,20-0,27
Iomega	1,119-2,499	Iomega	44 Мбайт	90-140	3,18-4,50
Conner, DEC, Fujitsu, Hewlett-Packard, Hitachi, IBM, Maxtor, Micropolis, Miniscribe, NEC, Quantum, Seagate, Western Digital	0,2-9,0	-	1,2 Гбайт	-	7,50-10,00
Archive, Wangtek	1,0-4,0	Carlisle, Gigatek, Sony, 3M	525 Мбайт	35-75	0,07-0,14
Chinon, Matsushita, Mitsubishi, Sony, TEAC, Toshiba	0,06-0,10	CAO, Fuji Photo, Maxell, 3M, Verbatim	1,44 Мбайт	1-2	0,69-1,39

го накопителя: перспективные оптические технологии".)

Читая данные с диска, МО-накопитель использует эффект влияния магнитного поля на отраженный световой луч. По своей природе лазер представляет когерентный источник света. Это значит, что свет, излучаемый лазером, имеет строго определенную длину волны и является поляризованным: вектор электрической напряженности электромагнитной волны колеблется в одной плоскости.

Когда поляризованный свет отражается от намагниченной металлической поверхности, плоскость его поляризации поворачивается — явление, известное как эффект Керра. Направление поворота зависит от полярности намагниченной поверхности. Для МО-дисков угол поворота не превышает одного градуса, но этого уже достаточно для того, чтобы прочитать данные с диска.

ДРУГИЕ ОПТИЧЕСКИЕ СИСТЕМЫ

Создание МО-накопителей явилось кульминационной точкой развития оптической технологии записи информации, заявившей о себе в 1972 году появлением первых лазерных видеодисков. Хотя разработка подобных систем преследовала цели, далекие от хранения компьютерных данных, технология производства видеодисков основывается на тех же принципах записи и чтения информации с помощью лазера. После того как были найдены технические решения для создания лазерных видеоси-

стем, было относительно просто перейти к изготовлению компакт-дисков и КДПЗУ.

В основе КДПЗУ и собственно компакт-дисков лежит одна и та же технология. Данные на КДПЗУ хранятся в виде последовательности углублений на поверхности диска. Углубления влияют на отражающую способность поверхности. Накопитель на КДПЗУ считывает данные, анализируя изменения в отраженном лазерном луче.

Подобная технология предназначена для массового тиражирования дисков с однажды записанной на них информацией. Процесс производства включает создание мастер-диска и штамповку копий (очень похоже на процесс изготовления грампластинок). Такая технология пригодна для производства электронных энциклопедий и справочников, однако она не позволяет дописывать на диск новую информацию либо изменять уже существующую.

Первые WORM-накопители (write once, read many — однократная запись, многократное чтение) появились в США в 1985 году. Так же как в КДПЗУ, запись на WORM-диски могла быть произведена только один раз. Однако здесь запись информации мог осуществить уже сам пользователь, применяя WORM-диски в качестве архива. Существуют несколько различных способов, используемых в WORM-накопителях

для изменения физического состояния поверхности диска. Однако все они направлены только на изменение отражающей способности поверхности. Как и у КДПЗУ, накопитель считывает данные, анализируя состояние отраженного луча.

Будущее WORM-технологии связано с созданием так называемых многофункциональных накопителей, объединяющих свойства WORM и перезаписывающих оптических накопителей (в зависимости от используемого типа дисков) в одну дисковую подсистему. Многофункциональные накопители уже начинают появляться на рынке и стоят лишь на пару сотен долларов дороже, чем их соперники, также способные осуществлять перезапись.

КРИТЕРИИ ВКЛЮЧЕНИЯ В ОБЗОР

Для того чтобы быть включенным в этот обзор, магнитооптический накопитель должен представлять собой законченную дисковую подсистему, состоящую из дисководов, контроллера, программного обеспечения, руководства по применению и некоторых других компонентов, необходимых конечному продукту. Каждый из рассматриваемых МО-накопителей должен был соответствовать стандартам ISO (Международная организация стандартов) и ANSI (Американский национальный институт стандартов) для 133-мм 600-Мбайт оптических накопителей с возможностью перезаписи. Помимо этого, каждый из накопителей должен поставляться пользователю, и, само собой, подсистема должна была вовремя и в рабочем состоянии прибыть в лабораторию PC Magazine. (В настоящее время не существует стандартов для 89-мм магнитооптических накопителей, и к моменту составления обзора еще не было промышленных образцов аппаратуры этого класса.)

Некоторые системы, не вошедшие в обзор, возможно, появятся в продаже к моменту его публикации. Сюда входят накопители фирм Advanced Digital Information, IPS Technology, Mitsubishi, Panasonic, Sharp, Sumo Systems, TASS International и Tristar Technology. Компания Hewlett-Packard предлагает через дистрибьюторов свой аппарат с одним из интерфейсов на выбор. Механизм дисководов был разработан фирмой Sony по спецификации HP.

Выбирая лучший из 17 рассматриваемых здесь накопителей, важно определить характеристики и производительность каждой из составных частей подсистемы: механизма дисководов с SCSI-контроллером, программного обеспечения и МО-кассеты.

ХАРАКТЕРИСТИКИ МЕХАНИЗМОВ

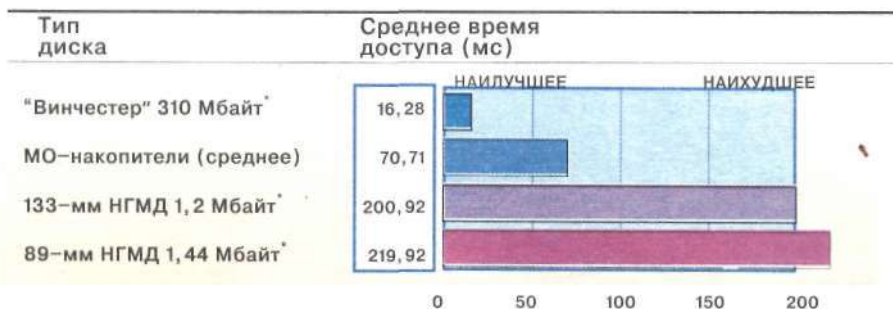
Компании, аппаратура которых представлена в обзоре, использовали в

МАГНИТООПТИЧЕСКИЕ ДИСКИ:

быстрее, чем гибкие, медленнее, чем жесткие



В качестве первичного накопителя МО-подсистемы оказываются для большинства задач слишком медленными. В среднем рассмотренные устройства продемонстрировали вчетверо большее время доступа к диску, чем 310-Мбайт жесткий диск компьютера Compaq Deskpro 386/25e. МО-подсистемы по своим характеристикам ближе к требованиям, предъявляемым к вторичным накопителям: по всем тестам их быстродействие оказалось примерно втрое выше, чем у НГМД.



Результаты тестов приводятся для устройств установленных в лаборатории PC Magazine на компьютере Compaq Deskpro 386/25e

American Classic.

PC MAGAZINE

FIRST LOOKS
IBM's Model 90: PS/2s Go the Next Round with 486 Power And XGA Graphics

AFTER HOURS
PCs Go Home Again: 5 Machines to Help You Work, Learn, Or Just Relax

SOFTWARE
Now You C-> It, Now You Don't: PC Labs Tests 9 Menuing Systems

COMPUTERS
More 386 Power: PC Labs Tests 26 33-MHz Systems

LEADER: 12 (20) THE INDEPENDENT GUIDE TO PERSONAL COMPUTING VOLUME 9 NUMBER 1

BEST OF 1990

• 7th Annual Awards for Technical Excellence
• The Year's Top Products: Editors' Personal Picks
• Gaffe Riot: Bill Howard Savors the Year's Memorable Moments




Jetzt auf Deutsch.

PC PROFESSIONELL

ARTIKEL
Harte Konkurrenz für Intel: AMDs 386/33-Prozessor ab sofort lieferbar

SOFTWARE
Damit Sie nie wieder Daten verlieren: Backup-Programme im Vergleich

CONNECTIVE
LAN Manager 2.0 gegen Netware 3.1: Welches System ist besser?

GRUNDLAGEN
Programmierung des Protected mode unter MS-DOS 4.01 und Windows 3.0

DES MAGAZIN FÜR PROFESSIONELLE PC ANWENDER • JAHRESLANG • 4. QUARTAL

386 SX-20: Der neue Standard

IM TEST: **44**
386SX-20



GUI-Spreadsheets: Halten sie, was sie versprechen?
Alle Programme im Vergleich

И, наконец, на русском ...

PC MAGAZINE

КОРОТКО, КАЧЕСТВО И ...
Семь светлых треков в век компьютеризации (типичные ошибки пользователей при организации работы на ПК)

КОМПЬЮТЕРЫ
Персональный компьютер сегодня (какие модели IBM/PC и в какой конфигурации наиболее перспективны в ближайшем будущем)

ПРОГРАММЫ
Персональная информационная система на вашем столе

МАЛЕНЬКИЕ ХИТРОСТИ
Настройка макроконструкций, меню и шаблонов в программе Word for Windows

№ 1. Июль 1991. PC Magazine USSR Edition. ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР СССР/91

Последнее слово еще не сказано: процессор 486 в компьютерах PS/2



- 24-разрядный цветной однопроходный сканер
- ObjectScript открывает окна в Бейсике и Си
- Версия Paradox 3.5 — усовершенствованное управление памятью и высочайшая эффективность
- Комтек'91 — продолжение следует...

своих разработках лишь три типа механизмов дисководов, трех разных фирм. Одиннадцать подсистем таких компаний, как Applied Programming Technologies, FWB, MicroNet, N/Hance, OCEAN, Optima, Peripheral Land, Pinnacle Micro, Procom, Storage Dimension и Testar, построены на базе дисковода фирмы Sony. Подсистемы фирм Consan, DynaTek, Relax Technology, Specialized Systems и Xuxis, их всего пять, имеют в своей основе механизм фирмы Ricoh. И только один накопитель — ASC MO-SS — использует новый дисковод

фирмы Maxoptix (Таити). Компании Sharp, Matsushita и Mitsubishi Electric грозятся в скором времени выпустить свои варианты механизмов для MO-накопителей. Фирмы NEC и Toshiba также готовятся принять участие в борьбе за рынок.

В целом накопители, созданные на базе дисководов Ricoh, дешевле, чем их соперники с механизмом Sony. Самая дешевая подсистема из представленных в обзоре — это накопитель фирмы DynaTek, построенный

на основе механики Ricoh. Его цена чуть больше 3 тыс. долл. Стоимость самого дешевого накопителя на базе дисковода Sony, выпускаемого фирмой APT, составляет приблизительно 4 тыс. долл. Самый дорогой накопитель с механизмом Ricoh — модель Xuxis XY 600RW — стоит около 5 тыс. долл., а цена наиболее дорогой системы на дисковом Sony — Testar Laser Vault — составляет около 7 тыс. долл. Накопитель MO-SS фирмы ASC, построенный на базе механики Maxoptix, относится к дорогим системам и стоит более 5,5 тыс. долл.

В ПОИСКАХ САМОГО БЫСТРОГО НАКОПИТЕЛЯ: перспективы ближайшего будущего

Интересуетесь ли вы будущим магнитооптических накопителей? PC Magazine предлагает обзор перспективных технологий, основанных на изменении фазового состояния, использовании полимерных красителей, модуляции магнитного поля и модуляции интенсивности излучения лазера. Благодаря новым технологиям MO-накопители станут более быстрыми и надежными.

М. Дэвид Стоун

Магнитооптические накопители пока еще сильно отстают от жестких дисков по такому важному параметру, как скорость записи. В первую очередь это связано с тем, что запись производится в два прохода. Именно поэтому современные направления исследований главным образом сосредоточены на поиске однопроходной технологии записи.

Современные MO-накопители записывают данные сразу после предварительного цикла стирания, который необходим для того, чтобы на всем участке записи остаточная намагниченность рабочего слоя диска имела одну и ту же величину и полярность. Собственно цикл записи изменяет полярность определенных участков-доменов магнитного слоя на противоположную. Необходимость двух проходов обусловлена тем, что ориентация векторов намагниченности доменов записываемого слоя задается магнитным полем (называемым полем смещения), направление которого технически не может изменяться достаточно быстро. Таким образом, домены противоположных полярностей не могут быть записаны за один проход.

Направление вектора индукции поля смещения определяется либо постоянным магнитом, на переориентацию которого требуется значительный промежуток времени, либо электромагнитом, так же достаточно инерционным.

Наиболее часто обсуждаемая альтернатива двухпроходной записи — это технология, основанная на изменении фазового состояния, уже реа-

лизованная в системе фирмы Matsushita (в настоящее время продается фирмой Panasonic и другими компаниями). Существует также технология, связанная с использованием полимерных красителей, которая пока находится в стадии разработки.

Известны еще по крайней мере два метода однопроходной записи, являющиеся по сути модернизацией существующей магнитооптической технологии. Они базируются на модуляции магнитного поля и модуляции мощности излучения лазера.

ИЗМЕНЕНИЕ ФАЗОВОГО СОСТОЯНИЯ И ПОЛИМЕРНЫЕ КРАСИТЕЛИ

Технологии, основанные на изменении фазового состояния и использовании полимерных красителей, уже применялись на протяжении некоторого времени в накопителях с однократной записью (WORM). В обоих случаях лазер служит для нагрева записываемого слоя с целью изменения его физического состояния. И в перезаписываемых дисках, и в дисках с однократной записью нагрев поверхности лазерным лучом изменяет ее отражающую способность.

Под изменением фазового состояния подразумевается способность вещества переходить из кристаллического состояния в аморфное. Достаточно осветить некоторую точку на поверхности диска лучом лазера определенной мощности, как вещество в этой точке перейдет в аморфное состояние. Говоря точнее, лазерный луч нагревает

его вещество до температуры плавления, после чего вещество остывает, переходя в аморфное состояние.

Технология, основанная на использовании полимерных красителей, допускает повторную запись. В этом случае поверхность диска покрывается двумя слоями специального пластика, каждый из которых чувствителен к свету "своей" строго определенной частоты. Для записи используется частота, игнорируемая верхним слоем, но вызывающая реакцию в нижнем. В точке падения луча нижний слой разбухает и образует маленькую выпуклость, влияющую на отражающие свойства диска. Для стирания выпуклости дисковод использует другую частоту, которая влияет только на верхний слой, сглаживающий выпуклость.

Обе технологии позволяют производить запись данных на диск за один проход. Их основной недостаток заключается в том, что процессы записи и стирания влекут за собой физические изменения в записываемом слое, что ограничивает время его жизни. Например, диски для накопителя фирмы Panasonic, работающего по принципу изменения фазового состояния, рассчитаны на 100 тыс. циклов чтения/записи. Сравните это с миллионом циклов для MO-кассеты.

В накопителе Panasonic "проблема долгодетия" решается путем автоматического перераспределения данных на диске в случае обнаружения сомнительного сектора. Практическую пригодность такой схемы еще предстоит проверять и проверять, поскольку действующая модель накопителя выпущена только в декабре 1990 г. Однако специалисты фирмы Panasonic уверены, что подобный подход позволит сделать их накопители такими же надежными, как и любые другие. Компания претендует на создание диска, выдерживающего миллион циклов записи/чтения. Когда дисководы, построенные на использовании полимерных красителей, выйдут на уровень рыночных образцов, их произво-

Разумеется, механизмы дисководов отличаются друг от друга не только по стоимости, но и по своей производительности. Хотя Ricoh- и Sony-дисководы показали сравнимые результаты в тестах чтения, системы на базе дисковода Sony значительно превосходили соперников в тестах записи. Накопитель MO-SS фирмы ASC с механикой Махортix вошел в пятерку лучших по всем показателям (см. "Тестирование производительности магнитооптических накопителей").

В процессе тестирования выяснилось, что среднее время доступа, оп-

ределяемое собственно механизмом, влияет на общую производительность меньше, чем этого можно было ожидать. (Среднее время доступа равно среднему времени поиска в сумме со средним временем реакции. Другими словами, это время, необходимое накопителю для поиска нужной дорожки и перемещения головки на эту дорожку. Дисковод Ricoh, со средним временем доступа 66,7 мс, может похвастаться лучшими характеристиками, чем механизм Sony, чье время

доступа составляет 107,5 мс, а Махортix (Таити) опережает конкурентов с чемпионским временем — всего 48,6 мс. Более высокая скорость вращения у дисковода Sony (2400 об/мин) в сравнении со скоростью вращения механики Махортix (2200 об/мин) и Ricoh (1800 об/мин) обуславливает некоторое его преимущество с точки зрения общей производительности.

ИНТЕРФЕЙСЫ

Несмотря на то, что производительность системы в определенной степени зависит от типа используемого дисковода, существует множество других факторов, влияющих на характеристики накопителя (см. таблицу спецификаций). Само собой разумеется, что одна и та же электроника в сочетании с различными типами механизмов будет обеспечивать разную производительность.

Законченная подсистема включает в свой состав механизм дисковода, интерфейс, состоящий из SCSI-контроллера и программного обеспечения, и MO-кассету. Контроллер в сочетании с механикой играют здесь наиболее важную роль. Например, 16-разрядный контроллер обеспечит более высокую производительность, чем 8-разрядный. Далеко не все фирмы, продающие MO-накопители, предлагают подсистемы, содержащие полный набор комплектующих, однако, как уже упоминалось, в этой статье мы рассматриваем только законченные изделия (и, кстати, их цены). Отдельно представлен перечень продавцов, предлагающих MO-подсистемы по частям.

Таблица спецификаций содержит разнообразную информацию об интерфейсах. Здесь есть и сведения о производителе SCSI-адаптера, и тип адаптера (8- или 16-бит), и список утилит, включенных в пакет интерфейса, с указанием производителей программного обеспечения. В таблице вы встретите такие знакомые названия, как Adaptec, Corel и Western Digital.

Все подсистемы комплектуются утилитами форматирования низкого уровня, которые позволяют подготовить новую кассету к работе. Однако на том сходство между ними и заканчивается. Некоторые подсистемы позволяют создавать только одну логическую область на диске, заставляя пользователя работать с 300-Мбайт логическим диском независимо от того, устраивает это его или нет. Если вы предпочитаете разбивать большие диски на меньшие логические области, скажем, по 100 Мбайт каждая, вам, разумеется, больше подойдет подсистема, позволяющая создавать множественные логические устройства (с помощью ее собственного программного обеспечения или утилиты FDISK DOS).

дители, скорее всего воспользуются аналогичной схемой повышения надежности.

МОДУЛЯЦИЯ МАГНИТНОГО ПОЛЯ

Фирма Sony недавно продемонстрировала тонкопленочную магнитную головку, способную изменять свою полярность на противоположную всего лишь за несколько наносекунд. Практически это позволяет изменять магнитное поле (или модулировать его) синхронно с потоком поступающих данных. Если одновременно с переориентацией магнитного поля можно будет включать лазер, то запись данных на диск за один проход станет вполне осуществима.

Модуляция магнитного поля (MFM — Magnetic Field Modulation) аналогична процессу записи на жестких дисках, однако, чтобы исключить возможность механического повреждения диска, здесь магнитная головка расположена достаточно далеко от его поверхности. Фирма Sony объявила, что MFM-технология обеспечит возможность чтения информации с MO-дисков, уже используемых в настоящее время (не ясно только, будет ли возможна также и запись на них). Согласно данным Эдварда Ротшильда, давнишнего обозревателя оптических технологий из фирмы Rotchild Consultants, компании Hitachi и Fujitsu тоже работают над созданием аналогичных схем. Ротшильд предсказывает, что промышленные образцы MFM магнитооптических накопителей появятся уже в ближайшие год-два.

МОДУЛЯЦИЯ ИЗЛУЧЕНИЯ ЛАЗЕРА

Еще одна альтернативная технология известна под названием модуляция интенсивности излучения лазера. Она была предложена фирмой Nicon и в настоящее время разрабатывается компаниями Sony и IBM.

Модуляция интенсивности излучения получила свое название вот из каких соображений: в трех разных режимах работы дисковода — а это

режим чтения с низкой интенсивностью, режим записи со средней интенсивностью и режим записи с высокой интенсивностью — используются три уровня интенсивности лазерного луча.

Модуляция интенсивности требует более сложной структуры диска, чем у существующих MO-кассет, и, кроме того, наличия двух магнитов в механизме дисковода (магнит смещения, как у MO-дисководов, и инициализирующий магнит, установленный перед магнитом смещения и имеющий противоположную полярность).

В самом простом случае диск имеет два рабочих слоя — инициализирующий и записывающий. Инициализирующий слой сделан из такого материала, что инициализирующий магнит может изменять его полярность без дополнительного воздействия лазера. Допустим, полярность инициализирующего слоя соответствует двоичному нулю. Когда рассматриваемый участок поверхности диска достигает лазера и магнита смещения, получается одно из двух: Если лазер настроен на среднюю мощность, поверхность диска нагревается как раз настолько, чтобы намагнитить записывающий слой. Проще говоря, на записываемом слое будет также установлен двоичный ноль. Если лазер освещает диск лучом большой интенсивности, слой нагревается до температуры Кюри (около 150 градусов Цельсия) и намагничивается в соответствии с полярностью магнита смещения. Иными словами, этот участок будет соответствовать двоичной единице. Таким образом, при переключении мощности лазера, запись данных может происходить за один проход.

Единственная схема прямой (однопроходной) записи, реализованная на сегодня, — это технология изменения фазового состояния, используемая в накопителе фирмы Matsushita. Но при таком обилии перспективных технологий победителя называть еще рано.

С другой стороны, если вы используете DOS 3.x, ограничивающий размер логического диска до 32 Мбайт, у вас могут возникнуть проблемы, связанные с появлением слишком большого количества логических дисков. Некоторые из накопителей предлагают специальные утилиты для создания в среде DOS 3.x логических дисков большего размера, вплоть до полного объема МО-диска. Можно

также использовать Compaq DOS3.31, который позволяет создавать логические диски размером до 300 Мбайт (как в DOS 4.01), но в этом случае следует помнить, что вашей МО-подсистеме может потребоваться специальная программная поддержка.

Если вы собираетесь использовать МО-накопитель для резервирования

жесткого диска, то, возможно, вы захотите приобрести систему, имеющую специальный режим резервного копирования. Фирма Testmag, к примеру, предлагает специальный режим работы своего накопителя, называемый "виртуальным стримером". Подобные специальные режимы часто обеспечивают более быстрое резервное копирование, чем, скажем, XCOPY, поскольку они не тратят

РЕДАКТОР СОВЕТУЕТ

- HammerDisk600S
фирмы FWB
- LaserVault
фирмы Testmag

Магнитооптические накопители занимают пока еще полупустую нишу на сегодняшнем рынке. К моменту создания настоящего обзора на суд редакции были представлены 17 накопителей (между прочим, как минимум, еще 8 подсистем не были готовы вовремя). Это аппараты фирм Advanced Digital Information, IPS Technology, Mitsubishi, Panasonic, Sharp, Sumo Systems, TASS International и Tristar Technology. Причем практически все подсистемы (в состав которых должны были входить дисковод, контроллер, программное обеспечение и кассета) были настолько сырыми, что их вряд ли можно было пускать в серию. С учетом всего сказанного выше мы наметили в качестве победителей два аппарата - это LaserVault фирмы Testmag и HammerDisk600S фирмы FWB. Особого упоминания заслуживает также накопитель ROS600 фирмы DynaTek.

В настоящее время МО-накопители обладают возможностью произвольного доступа к файлам (как в жестких дисках), имеют большую емкость (в общем сравнимую с емкостью некоторых жестких дисков), и, конечно, не надо забывать об их замечательном свойстве сменяемости (вы можете вынуть диск из накопителя и унести его домой или запереть на ночь в сейф). Все эти особенности обеспечивают уникальные преимущества МО-дисков перед остальными устройствами внешней памяти. Например, в настольных издательских системах они незаменимы для хранения графических изображений. Они очень удобны в качестве вторичных накопителей (можно держать под рукой программы, время доступа к которым не слишком критично), а также для резервного копирования жестких дисков. Однако, судя по устройствам, представленным в обзоре, большинство МО-подсистем пока еще страдают откровенно слабым дизайном.

Заметим, что сегодня на рынке существуют лишь три основных производителя механизмов для оптических накопителей с возможностью перезаписи - это фирмы Sony, Ricoh и Maxoptix. Кроме того, в числе наших требований к отобранному подсистемам было их соответствие стандартам ISO и ANSI для 133-мм оптических накопителей. И, тем не менее, в большинстве случаев вам вряд ли удастся использовать один и тот же диск для работы на двух разных устройствах. Поэтому, если вам необходимы несколько МО-накопителей с гарантированной совместимостью, советуем покупать их у одного производителя. И, наконец, при выборе подходящего устройства обращайте внимание не столько на его цену и производительность, сколько на комплектность предлагаемой подсистемы и ее дополнительные возможности.

Если у вас возникла потребность в МО-накопителе и к тому же есть деньги и необходимые знания, покупайте подсистему LaserVault фирмы Testmag. Видимо, благодаря тому, что она относится к разряду наиболее дорогих (6885 долл.), она обеспечивает высокую производительность во всех режимах работы. Не менее важно и то, что она является наиболее полной, законченной, а также имеет множество дополнительных возможностей. Оснащенная специальным режимом резервного копирования и автоматически устанавливаемым программным обеспечением, LaserVault производит впечатление цельного изделия. Подсистема предлагает следующие дополнительные возможности: поддержка дисков с размером сектора 1024 байт, обмен данными между операционными системами DOS и Macintosh, а также управление несколькими операционными системами с одного диска. Если вас устраивает цена, подсистема LaserVault - достойная покупка.

Другое устройство, выделенное нами в разряд лучших, — это накопитель HammerDisk600S фирмы FWB. Превосходя подсистему LaserVault по общей

производительности, он несколько беднее оснащен необходимыми принадлежностями и не имеет таких широких возможностей. Цена накопителя составляет 6085 долл., и, безусловно, он этих денег стоит. Быстрый дисковод с хорошим набором программных утилит (отсутствует утилита, подобная LaserBack для накопителя LaserVault), HammerDisk600S не обеспечивает лишь одной возможности, доступной LaserVault. Речь идет о поддержке секторов размером 1024 байт. Если вы можете обойтись без расширенных возможностей LaserVault и хотите получить небольшой выигреш в скорости, накопитель HammerDisk600S станет для вас наилучшим выбором.

Аппарат ROS600 фирмы DynaTek заслуживает особого упоминания. Несмотря на то, что это самая дешевая подсистема из всех рассмотренных (3245 долл.), она на удивление хорошо сконструирована. Здесь не предусмотрены возможности обмена данными между разными компьютерными платформами и поддержки множественных операционных систем на одном диске (к тому же накопитель имеет среднюю производительность), но простая начальная установка в сочетании с набором превосходных утилит и способностью форматирования дисков с размером сектора 1024 байт поднимают подсистему DynaTek на уровень наиболее конкурентоспособных.

Последний накопитель, достойный упоминания, - это подсистема W6501 фирмы N/Hance, являющаяся собой яркий пример сочетания тех крайностей, которые типичны для сегодняшнего состояния МО-индустрии. W6501 обладает уникальным и очень мощным механизмом, доказавшим свое превосходство над остальными в большинстве важнейших тестов. Но из-за конструктивных недостатков подсистема полностью не прошла один из тестов. Кстати, большинство рассмотренных накопителей обнаруживают крайне неравномерные результаты при тестировании.

время на обновление таблицы размещения файлов, правда, они могут конфликтовать с возможностью произвольного доступа к файлам, и вы не сможете воспользоваться своими данными, пока не восстановите их на жестком диске.

Имейте также в виду, что некоторые подсистемы сопровождаются такими дополнительными удобствами, как например, возможность загрузки компьютера с MO-дисководом или форматирование в фоновом режиме, что позволяет производить разметку новой кассеты, не останавливая работу всей системы. Фоновое форматирование не слишком важно для подсистем, способных быстро справиться с такой задачей, но оно может быть очень полезным для накопителей с циклом форматирования 20 — 40 минут.

Еще одна представляющая интерес дополнительная возможность — это способность MO-накопителей перемещать данные между разными компьютерными платформами. Например, режим резервного копирования накопителя Testmag обеспечивает обмен данными между системами PC и Mac.

СТАНДАРТЫ И СОВМЕСТИМОСТЬ

Как это часто случается с новыми технологиями, первые MO-накопители были, большей частью, не совместимы друг с другом, несмотря на наличие двух стандартов.

Отсутствие совместимости не причинит особых неудобств, если вы собираетесь использовать все свои диски только на одном компьютере. Но сам факт, что MO-диски обладают способностью сменяемости, предполагает возможность их использования с разными накопителями. В идеале MO-накопители должны позволять вам обмениваться кассетами так же просто, как это делается с помощью флоппи-дисков, но тут требуется введение единого стандарта.

Приемлемые стандарты были предложены ISO и ANSI, но, к сожалению, они все еще не введены повсеместно. Важно то, что между ними есть одно существенное различие.

Стандарт ISO разрешает использование любой из двух несовместимых схем форматирования: CCS (continuous composite servo — непрерывное комбинированное слежение) и SS (sampled servo — шаблонное слежение). В настоящее время формат CCS значительно более популярен, чем SS, отчасти из-за того, что он первым появился на рынке. Здесь играет роль и тот факт, что CCS — единственный формат, разрешенный стандартом ANSI. Поэтому любой ANSI-совместимый накопитель будет совместим с ISO, в то время как накопитель, отвечающий стандарту ISO, может быть несовместим с ANSI.

Поскольку в настоящее время формат CCS наиболее популярен, и только CCS-накопители могут быть совместимы с обоими стандартами (ANSI и ISO), мы сфокусируем наше внимание на 133-мм MO-накопителях, использующих формат ANSI. Следует иметь в виду, что приверженцы SS-подхода утверждают, что он превосходит вариант CCS. Современные отношения между CCS и SS-форматами напоминают битву между видеоформатами VHS и Betamax, правда, сейчас еще слишком рано предсказывать победителя. Пока нужно лишь помнить о том, что CCS и SS — это два несовместимых формата, и если вам необходимо перенести данные на MO-кассете между двумя системами, приобретайте накопители одного типа.

РАЗМЕР СЕКТОРА

К сожалению, различия в схемах форматирования — не единственная проблема взаимозаменяемости. Размер сектора на диске представляет собой еще одно узкое место. Стандарты ISO и ANSI определяют два варианта форматирования MO-дисков — с размером сектора 512 байт или 1024 байт. Главное преимущество использования 1024-байт секторов состоит в их несколько большей емкости — 325 Мбайт на сторону — по сравнению с 300 мегабайтами при размере сектора 512 байт. Правда, если ваши данные организованы как множество маленьких файлов, выигранные при форматировании мегабайты могут быть потеряны. К тому же, поскольку нормальная длина сектора в DOS составляет 512 байт, для дисков с таким форматом легче писать драйверы.

Механизмы всех MO-систем, фигурирующих в данном обзоре, позволяют записывать на диск и читать с диска секторы любого размера. Однако большинство устройств этого не допускают. Из 17 MO-накопителей 12 поддерживают диски с размером сектора 512 байт. Остальные пять подсистем, созданные фирмами Advanced Storage Concepts, Applied Programming, Dynatek, N/hance и Testmag, позволяют работать с двумя форматами. Возможность работы с двумя форматами, очевидно, имеет преимущество с точки зрения совместимости. Тем не менее, диски с длиной сектора 512 байт являются как бы стандартом для всех рассматриваемых накопителей.

Интересно, что ни одна из пяти подсистем, допускающих использование двух размеров сектора, при тестировании не показала существенного выигрыша одного формата по сравнению с другим. При выполнении любого теста одни системы

работали лучше при длине сектора 1024 байт, другие — при длине сектора 512 байт. Поэтому для каждого конкретного MO-накопителя разница в быстродействии при использовании двух форматов может оказаться от пренебрежимо малой до значительной.

Даже если ограничить рамки обзора подсистем стандартами ISO и ANSI и длиной сектора 512 байт, нельзя не упомянуть о проблемах, связанных с совместимостью. В стандартах ISO и ANSI строго определен только физический, но не логический формат записи. Гарантию совместимости в какой-то мере может дать то, что для управления устройствами используются одни и те же программные средства (см. таблицу "Магнитооптические подсистемы: основные характеристики").

Пытаясь обеспечить совместимость, ANSI разрабатывает стандарт логического формата, который будет использован как для уже существующих, так и для проектируемых MO-систем. Это значит, что, покупая сегодня MO-кассету диаметром 133 мм, можно не очень беспокоиться о том, что она устареет в ближайшем будущем (хотя, вероятно, через некоторое время вам понадобится новый драйвер). Что же касается дисков диаметром 89 мм, для них стандарт физического формата еще только обсуждается.

Следующая нестандартная характеристика, которую необходимо отметить, — это функция привилегированной записи, позволяющая увеличить емкость диска в системе Maxortix до 500 Мбайт и больше (имеется в виду одна сторона!). Дело в том, что в стандартах ISO и ANSI используется техника записи с постоянной угловой скоростью (CAV). В накопителе Maxortix кроме этого варианта записи, предназначенного для дисков стандарта ISO, можно использовать технику с зонной постоянной угловой скоростью (Z-CAV), позволяющую заметно увеличить емкость внешних, более длинных дорожек, и, таким образом, более рационально использовать рабочую поверхность диска. Возможность записи большего объема информации — очевидное преимущество техники Z-CAV, хотя и дается оно ценой отказа от стандартного формата записи. Фирма Maxortix сейчас прилагает максимальные усилия, чтобы установить стандарты на запись с использованием техники Z-CAV.

ЧЕМ БОЛЬШЕ ГИГАБАЙТОВ, ТЕМ МЕНЬШЕ ДОЛЛАРОВ

Несмотря на противоречия, которыми богата всякая новая технология, будущее MO-дисков выглядит вполне безоблачным. Эдвард Ротшильд из фирмы San Francisco's Rotshild Consultants Inc. следит за развитием оп-

АППАРАТНЫЕ СРЕДСТВА

тической технологии с 1974 г. Он предполагает, что к концу этого десятилетия емкость оптических кассет увеличится в 30 раз и достигнет 10 Гбайт на одной стороне при диаметре диска 133 мм. Он также считает, что благодаря увеличению объема производства стоимость накопителей в течение следующих двух лет упадет ниже уровня 2000 долл. Цены на кассеты ожидается та же участь (сейчас они составляют от 150 до 250 долл.).

Что касается самого ближайшего будущего, то большинство обозревателей полагает, что 89-мм оптические диски станут сенсацией 1991 г. Пока ISO и ANSI еще только разрабатывают соответствующий стандарт записи, по крайней мере три фирмы — Sharp, Ocean Technologies и Pinnacle — объявили о начале производства или уже приступили к производству кассет диаметром 89 мм емкостью по 125 Мбайт на каждой стороне. Ротшильд предполагает, что уже в этом году 89-мм оптические

диски начнут выпускать компании Panasonic, Ricoh, Sony, а может быть, и кто-нибудь еще. Итак, если сейчас МО-накопители вас не очень интересуют, есть шанс, что скоро вы все равно станете обладателем такого ус-

тройства: цены неуклонно падают, а характеристики улучшаются.

Область применения МО-дисков еще довольно узка. Если вам необходим накопитель большой емкости с произвольным доступом или если ваши резервные копии должны обладать возможностью выбора произвольного файла, МО-диск — это как раз то, что нужно. Вам остается только сделать правильный выбор из 17 рассмотренных здесь систем.

ТОЛЬКО ФАКТЫ

FWB HammerDisk600S

Фирма-изготовитель: FWB Inc.

Цена: с набором кабелей и кассетой — 6085 долл.;

дополнительная кассета — 295 долл.

Требуемые ресурсы: DOS 3.3

или более поздняя версия, O3Y

640 Кбайт

Система основана на

механизме фирмы Sony.

Устройство недорогое, но

обладает хорошими

характеристиками и гарантирует

совместимость с DOS, OS/2,

Novell и Xenix

ФИРМА FWB

Система

HammerDisk 600S

Уинн Л. Рош

Благодаря простоте установки и "тесному сотрудничеству" с DOS, HammerDisk 600S фирмы FWB стал оптимальным вариантом для пользователя, не требующего от устройства высокого быстродействия. Подсисте-

МАГНИТООПТИЧЕСКИЕ ПОДСИСТЕМЫ: основные характеристики

Изделия перечислены в порядке убывания цен.

Рассматриваемые здесь 17 МО-подсистем представляют собой внешние устройства с интерфейсом SCSI. Все они совместимы со стандартом ISO/ANSI для оптических дисков диаметром 133 мм. В каждом случае мы указываем цену полного комплекта.

	DynaTek ROS600	Relax Optical 600 Plus	Consan RS600/N	APT Odessa ROS-3250EIS	SST Storage Stak II	OCEAN Tidalwave 650	PLI Peripheral Infinity Optical	Xyxis 600RW XY
Цена тестируемой конфигурации (тыс. долл.)	3,245	3,377	3,495	3,995	4,422	4,495	4,799	4,995
ФИЗИЧЕСКИЕ ХАРАКТЕРИСТИКИ								
Фирма-изготовитель механизма	Ricoh	Ricoh	Ricoh	Sony	Ricoh	Sony	Sony	Ricoh
Среднее время доступа (мс)	66,7	66,7	66,7	107,5	66,7	107,5	107,5	66,7
Габаритные размеры (см)	12,7x25,4x25,4	10,2x25,4x27,9	12,7x25,4x33	12,7x20,3x30,5	35,6x25,4x33	12,7x25,4x27,9	12,7x25,4x30,5	15,2x25,4x33
Разрядность интерфейсной платы	16-bit	8-bit	16-bit	8-bit	16-bit	8-bit	8-bit	16-bit
Размер интерфейсной платы (в долях полной длины)	1/2	1/2	1	1/2	1	1/2	1/2	1/2
Фирма-изготовитель интерфейсной платы	DynaTek	Relax Technology	Procomp xUSA	APT	Western Digital	Future Domain	PLI	Adaptec
НОСИТЕЛЬ								
Емкость (сектор 512 байт, обе стороны)	1Гбайт	570Мбайт	596Мбайт	560Мбайт	650Мбайт	564Мбайт	582Мбайт	600Мбайт
Максимальный размер раздела (сектор 512 байт)	320Мбайт	285Мбайт	327Мбайт	280Мбайт	281Мбайт	282Мбайт	281Мбайт	287Мбайт
Возможность работы с секторами длиной 1024 байта	■	□	□	■	□	□	□	□
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ								
Размер драйвера DOS	16Кбайт	13Кбайт	62Кбайт	16Кбайт	16Кбайт	7Кбайт	17Кбайт	22Кбайт
Потребность в драйвере DOS	■	■	■	■	■	■	■	■
Возможность работы с другими операционными системами	OS/2, Mac, Unix	Mac, Unix	OS/2,	Нет	OS/2, Mac, Unix	Mac, Unix	Mac,	OS/2, Mac, Unix
Утилита установки	Нет	Trantor	Procomp USA	APT	Columbia	OCEAN	PLI	Xyxis
Утилита форматирования	DynaTek	Trantor	Procomp USA	APT	Columbia	OCEAN	PLI	Xyxis
Утилита создания резервной копии	DynaTek	Нет	Нет	APT	Columbia	Нет	PLI	Нет

■ — Да □ — Нет

АППАРАТНЫЕ СРЕДСТВА

му стоимостью 6085 долл. (вторую по цене среди всех рассмотренных) нужно покупать по частям: накопитель фирмы Sony стоимостью 5495 долл. со средним временем доступа 107,5 мсек, управляющий адаптер за 295 долл. и кассета за 295 долл. вместе составляют 6085 долл. HammerDisk 600S требует минимума операций при установке, но по сравнению с другими устройствами этого класса ей немного не хватает быстроедействия.

Фирма FWB ориентируется в основном на рынок компьютеров Apple. Поэтому система HammerDisk 600S сопровождается множеством программ для Macintosh. Установка устройства на IBM PC предполагает использование управляющего контроллера IN-200 фирмы Always Technology, собранного на микросхемах Xilinx. Эту плату фирма настраивает при сборке, так что пользователю остается только вставить ее в слот и начать работать. Диски, записанные на HammerDisk 600S в DOS, читаются на Macintosh, и нао-

борот. Существуют также версии для работы в OS/2, Unix и Novell.

Все программное обеспечение, необходимое для работы подсистемы, прошито во встроенном ПЗУ. При помощи переключателей DIP на плате адаптера для нестандартных PC можно выбрать альтернативный адрес BIOS (отличный от C800M).

Дисковод размерами 12,0 5x24, 1 x24,75 см соединяется с 25-контактным разъемом на плате адаптера при помощи кабеля длиной 51 см, поставляемого фирмой FWB. В системе HammerDisk 600S есть два 50-контактных SCSI-разъема и внешние заглушки, которые поставляются вместе с системой.

После установки, которая состоит лишь в подсоединении кабеля, заглушек и в перезагрузке компьютера, на его экране возникает сообщение о том, что в вашей вычислительной системе появилось новое периферийное устройство.

Как и стандартные диски DOS, оптические кассеты нужно подготовить к работе утилитами FDISK и FORMAT. Форматирование при длине сектора 512 байт занимает сравнительно мало времени — около 10 мин. Система не поддерживает секторы длиной 1024 байт.

После форматирования Hammer Disk 600S можно использовать как обычный накопитель. С него даже можно загружаться, если при помощи опции /S скопировать системные файлы. Управляющий адаптер имеет режим работы системного контроллера флоппи-дисков.

Результаты, показанные Hammer Disk 600S' при тестировании под управлением DOS, одни из лучших. Время доступа к файлу и время доступа к диску — в первой пятерке, характеристики в тесте с XCOPY тоже вполне приличные.

Единственным недостатком HammerDisk 600S можно считать высокую цену. Но, принимая во внимание хорошие характеристики, следует

Pinnacle REO-650	Procom MEOD650/E	SD LaserStop Erasable Optical 650MB	N/Hance W6501	MicroNet SB-SMO/DOS	Optima Concorde 600MO	ASC MO-SS	FWB HammerDisk 600S	Tecmar LaserVault
5,239	5,294	5,295	5,450	5,479	5,590	5,595	6,085	6,885
Sony 107,5 15,2x20,3x33	Sony 107,5 17,8x20,3x x30,5	Sony 107,5 15,2x20,3x x38,1	Sony 107,5 12,7x20,3x x30,5	Sony 107,5 17,8x20,3x x30,5	Sony 107,5 10,1x27,9x x25,4	Maxoptix (Tahiti) 48,6 17,8x20,3x x30,5	Sony 107,5 12,7x25,4x x25,4	Sony 107,5 12,7x20,3x x30,5
16-bit	16-bit	16-bit	16-bit	16-bit	16-bit	16-bit	16-bit	16-bit
1	1	3 4	1	1 2	3 4	1 2	3 4	1
Western Digital	Procom	Adaptec	Adaptec	MicroNet	Optima	ASC	Always	Adaptec
600Мбайт 281Мбайт	588Мбайт 291Мбайт	562Мбайт 281Мбайт	594Мбайт 297Мбайт	586Мбайт 293Мбайт	564Мбайт 282Мбайт	596Мбайт 298Мбайт	574Мбайт 287Мбайт	1Гбайт 322Мбайт
□	□	□	■	□	□	■	□	■
35Кбайт □	7Кбайт ■	16Кбайт ■	130Кбайт ■	10Кбайт □	11Кбайт □	19Кбайт ■	20Кбайт □	36Кбайт ■
OS/2, Mac, Unix Columbia	OS/2, Mac, Unix Procom	OS/2, Mac, Unix SD	Unix N/Hance	OS/2, Unix MicroNet	OS/2, Mac, Unix Optima	Нет ASC	OS/2, Mac, Unix FWB	OS/2, Mac, Unix Tecmar/ Adaptec/ Tecmar/ Adaptec/ Tecmar
Columbia	Procom	SD	N/Hance	MicroNet	Optima	ASC	Нет	Adaptec/ Adaptec/ Adaptec/ Adaptec
Нет	Нет	SD	N/Hance	Нет	Нет	Нет	Нет	Нет



ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ МАГНИТООПТИЧЕСКИХ НАКОПИТЕЛЕЙ

При записи время доступа в случае накопителей с механизмами фирмы Sony меньше, чем для накопителей с механизмами фирмы Ricoh, но при чтении показатели обеих систем сходны. Накопитель ASC MO-SS фирмы Maxoptix претендовал на первенство, но по результатам наиболее важных тестов чемпионом оказалось устройство W6501 фирмы N/Hance с механизмом фирмы Sony.

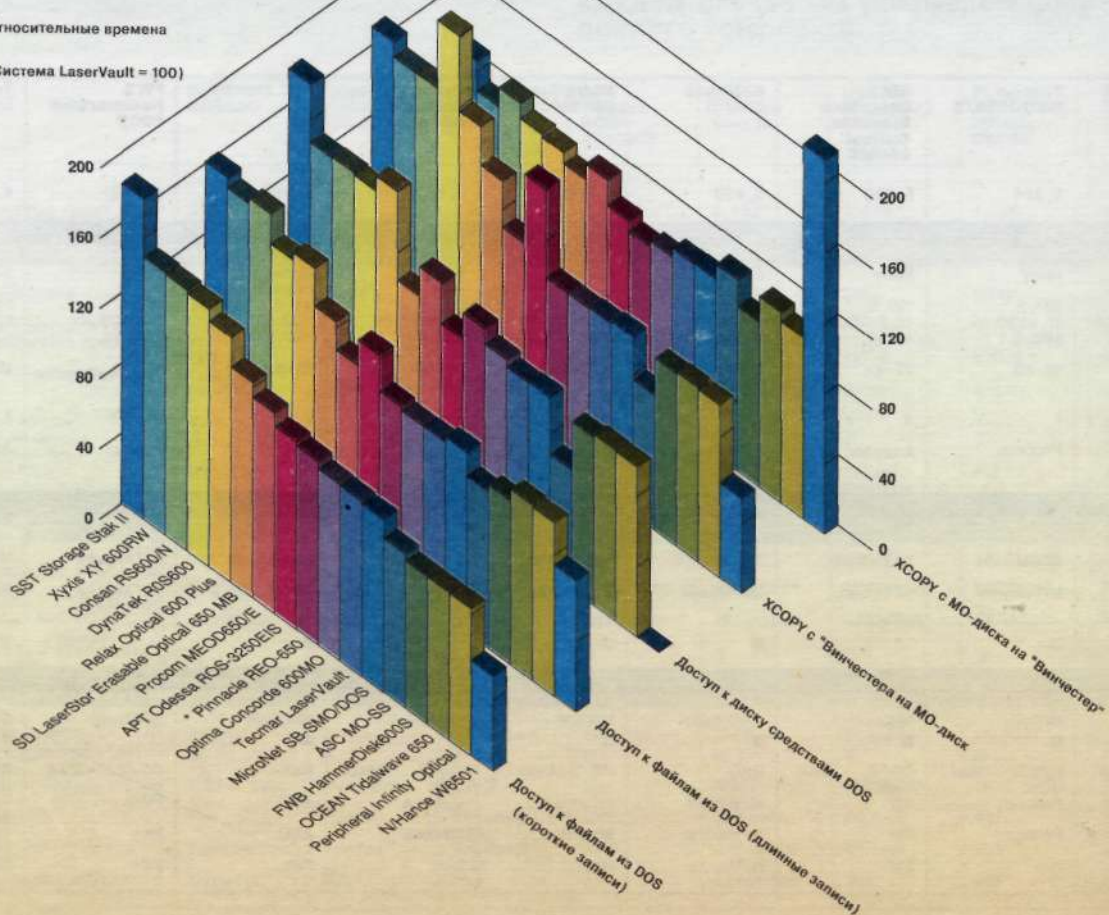
АНАЛИЗ ТЕСТОВ

При сравнении 17 MO-дисковых подсистем, рассматриваемых в настоящем обзоре, самыми важными были тесты, связанные с доступом DOS к файлам и выполнением утилиты XCOPY. Каждый тест доступа DOS к файлам — это измерение времени создания, чтения и записи для небольших записей (типа баз данных) и больших записей (типа программ). Тест с использованием XCOPY показывает, какие операции диск выполняет быстрее — запись данных на MO-диск или чтение. Немного менее важны результаты тестирования времени доступа DOS к диску, которые отражают скорость вращения и скорость перемещения головки чтения/записи.

Семнадцать накопителей, о которых идет речь, различаются в основном механизмами, которые в них применяются. Два наиболее популярных разработаны фирмами Sony и Ricoh. В четырех из пяти тестов модели, в которых использован механизм Sony, были

быстрее моделей с механизмом Ricoh. Этот результат частично объясняется более высокой скоростью вращения механизмов Sony и меньшим временем позиционирования головок. Разница между двумя механизмами отчетливо проявилась при выполнении копирования с помощью XCOPY с винчестера на MO-диск. Все модели с механизмом Ricoh работали существенно медленнее, чем модели с механизмом Sony. Повышенная скорость вращения механизма фирмы Sony и сниженное время позиционирования ускоряют процесс двухпроходной записи, который используется во всех MO-дисках. С другой стороны, при копировании с помощью XCOPY с MO-диска на винчестер результаты были весьма близки и распределены настолько равномерно, что победителя назвать не удалось. Система ASC MO-SS, которая наиболее часто оказывалась в лидирующей пятерке, уникальна, так как в ней используется механизм фирмы Maxoptix (Таити). Она заняла

Относительные времена
(Система LaserVault = 100)



первое место в тесте доступа к диску и вторые места в тестах доступа к файлам (большие записи) и копирования с помощью XCOPY с винчестера на MO-диск. Эти результаты показывают, что более высокая скорость вращения механизма Махорих ускоряет двухпроходную запись. В двух самых важных тестах наилучшие результаты показал MO-накопитель фирмы N/Hanse. В устройстве W6501 фирмы N/Hanse используется механизм фирмы Sony, а также применен драйвер, который осуществляет параллельный доступ к файлам и существенно улучшает характеристики системы, особенно при записи данных на MO-диск. Однако при чтении это преимущество теряется, так как в этом случае параллельный доступ только замедляет передачу данных (это замедление несущественно, если диск используется для архивирования). Кроме того, для модели W6501 фирмы N/Hanse не удалось измерить время доступа к диску, потому что неправильно обрабатывалось прерывание 25п.

Из-за этого использование любой программы, запрашивающей специфическую информацию о диске (например, число байтов или дорожек в секторе или число секторов), с этим устройством проблематично. Таким образом, можно сделать заключение, что, хотя механизм и является важным фактором, влияющим на быстрдействие MO-устройства, не менее важны характеристики драйвера. Необходимо отметить, что мы убрали строку SHARE из файла AUTOEXEC.BAT, когда тестировали систему REO-650 фирмы Pinnacle с помощью XCOPY. Программа SHARE позволяет старым версиям DOS работать с разделами диска более 32 Мбайт. Нам не удалось скопировать информацию с устройства Pinnacle на жесткий диск, когда программа SHARE находилась в памяти: система зависала при попытке копирования файла размером более 1 Мбайт. Звонки в фирму не прояснили ситуацию. Когда SHARE убрали, утилита XCOPY выполнялась без проблем.

сделать заключение, что система HammerDisk 600S фирмы FWB предназначена для тех, кто заплатит немного больше, чтобы снизить вероятность конфликтов и повысить быстрдействие.

ФИРМАТЕСМАР Устройство LaserVault М. Дэвид Стоун

Устройство LaserVault оправдывает ожидания, связанные с репутацией фирмы Тесмаг, чьи накопители на магнитной ленте можно назвать безупречными. LaserVault построен на основе дискового механизма фирмы Sony со средним временем доступа 107,5 мсек и оснащен аппаратным и программным обеспечением, необходимым для использования его в качестве виртуального накопителя на магнитной ленте. Это обстоятельство, а также высокие показатели при тестировании позволяют считать, что он стоит денег, которые за него просит фирма. (6885 долл. — самая высокая цена, упоминающаяся в обзоре.) Что касается тестирования, то LaserVault по всем характеристикам оказался "в середине турнирной таблицы" или чуть выше. Пакет позволяет поддерживать секторы длиной 1024 байт. После форматирования с большими размерами сектора, результаты оказались лучше в одних тестах и хуже в других. Например, при записи посредством XCOPY с винчестера на MO-диск при длине сектора 1024 байт результаты были хуже, а при выполнении обратной операции — намного лучше, чем при длине сектора 512 байт. Однако, при сравнении с остальными накопителями результаты, показанные LaserVault, оказались несколько выше, независимо от того, какой формат записи данных использовался. Многие конкуренты показали лучшие результаты в одних тестах и худшие в других, но LaserVault не таков. Его стабильные показатели и высокая надежность оправдывают высокую цену. Тестируемая система была оснащена 16-разрядной платой с интерфейсом SCSI и кабелем; фирма Тесмаг за 795 долл. предлагает набор программных и аппаратных средств для подключения накопителя к компьютерам с микроканальной архитектурой, а также набор средств для подключения к Macintosh за 295 долл. Максимальная емкость диска с длиной сектора 1024 байт — 325 Мбайт на одной стороне (что соответствует стандарту), а максимальный размер раздела — 310 Мбайт. Возможна поставка дополнительного программного обеспечения для сете-

- Результаты тестирования *доступа к файлам (малые записи)* зависят от скорости работы механики накопителя, быстрдействия контроллера и шины данных. Тестирование выполняется без программного кэширования диска. Лучшие результаты показывают программы, работающие с малыми блоками данных.
- Результаты тестирования *доступа к файлам (большие записи)* зависят от скорости работы механики накопителя, быстрдействия контроллера и шины данных. В этом тесте сводится к минимуму влияние небольшой аппаратной кэш-памяти. Тестирование выполняется без программного кэширования диска. Лучшие результаты получают при загрузке больших файлов.
- Ш Тест *доступа к диску* использует прерывание 21h DOS для прямой записи информации в 1000 случайно выбранных секторов. Этот тест должен работать с любым устройством, которое DOS распознает как диск. Он позволяет получить среднее время обращения к сектору.
- В тесте "XCOPY с Винчестера на MO-диск" измеряется время, которое требуется программе XCOPY с ключом /S для копирования файловой системы размером 115 Мбайт на чистый отформатированный MO-диск. Эта процедура может применяться для архивирования жесткого диска на MO-накопитель.
- В тесте "XCOPY с MO-диска на Винчестер" измеряется время, которое требуется программе XCOPY с ключом /S для копирования файловой системы размером 115 Мбайт с MO на чистый отформатированный жесткий диск. Эта процедура может применяться для восстановления данных MO на жесткий диск.

КАК МЫ ПРОВОДИЛИ ТЕСТИРОВАНИЕ

Лаборатория журнала PC Magazine тестировала накопители на компьютере Compaq Deskpro 386/25e с оперативной памятью объемом 4 Мбайт, жестким диском с интерфейсом ESDI емкостью 310 Мбайт (контроллер Compaq) и встроенным адаптером VGA. Тестирование проводилось под управлением операционной системы Compaq DOS 4.01. В память загружались драйверы, необходимые для работы с устройствами; переменным FILES и BUFFERS в CONFIG.SYS присваивалось значение 20. В файле AUTOEXEC.BAT устанавливались необходимые параметры PROMPT и PATH, а также запускалась программа SHARE.EXE, необходимая при использовании носителей большой емкости. Жесткий диск был разбит на два раздела по 155 Мбайт (логические устройства C и D). Для тестирования с помощью XCOPY на диске D была создана файловая система размером 115 Мбайт. Все MO-диски были отформатированы с длиной сектора 512 байт.

APT Odessa ROS-3250EIS

Фирма-изготовитель: Applied Programming Technologies
Цена: с набором кабелей и кассетой — 3995 долл.; дополнительная кассета - 225 долл.
Требуемые ресурсы: DOS 3.2 или более поздняя версия, ОЗУ 512 Кбайт

Система построена на базе механизма фирмы Sony; низкая цена, скромные характеристики и странный драйвер, не полностью совместимый с DOS.

Consan RS600/N

Фирма-изготовитель: Consan Inc.
Цена: с набором кабелей и кассетой - 3495 долл.; дополнительная кассета — 155 долл.
Требуемые ресурсы: DOS 2.0 или более поздняя версия, ОЗУ 640 Кбайт

Эта недорогая система соответствует стандарту DOS и построена на микропроцессоре Intel 80C188, но при ее установке возникают проблемы, а показатели невысокие.

Pinnacle REO-650

Фирма-изготовитель: Pinnacle Micro Inc.
Цена: с набором кабелей — 5239 долл.; дополнительная кассета — 249 долл.
Требуемые ресурсы: DOS 3.2 или более поздняя версия, ОЗУ 512 Кбайт

Система базируется на механизме фирмы Sony. Цена и показатели — средние. Единственная особенность — несовместимость с утилитой SHARE.

DynaTek ROS600

Фирма-изготовитель: DynaTek Automation Systems Inc.
Цена: с набором кабелей и кассетой — 3245 долл.; дополнительная кассета — 225 долл.
Требуемые ресурсы: DOS 2.x или более поздняя версия, ОЗУ 256 Кбайт

Накопитель построен на базе механизма фирмы Ricoh и является самым дешевым из рассматриваемых здесь. Прилагаемые утилиты с развитой системой меню делают установку очень простой. Однако при тестировании характеристики устройства оказались ниже среднего уровня.

Relax Optical 600 Plus

Фирма-изготовитель: Relax Technology Inc.
Цена: с набором кабелей - 3377 долл.; дополнительная кассета — 200 долл.
Требуемые ресурсы: DOS 3.3 или более поздняя версия, ОЗУ 512 Кбайт

Система Optical 600 Plus построена на базе механизма Ricoh. Ее достоинства — невысокая цена и простота установки — не перекрывают недостатков: низкой производительности и затрудненности адаптации к различным аппаратным конфигурациям.

Optima Concorde 600MO

Фирма-изготовитель: Optima Technology Corp.
Цена: с набором кабелей и кассетой - 5590 долл.; дополнительная кассета - 395 долл.
Требуемые ресурсы: DOS 3.2 или более поздняя версия, ОЗУ 512 Кбайт

Система построена на базе механизма фирмы Sony. Сочетает средние характеристики с хорошим обслуживанием. Цена выше средней.

MicroNet SB-SMO/DOS

Фирма-изготовитель: MicroNet Technology Inc.
Цена: с набором кабелей и кассетой — 5479 долл.; дополнительная кассета - 279 долл.
Требуемые ресурсы: DOS 2.x или более поздняя версия, ОЗУ 512 Кбайт

Дорогая система, построенная на механизме фирмы Sony, поставляется с качественным программным обеспечением и микропроцессором, оптимизирующим многозадачный режим SCSI, но результаты работы непредсказуемы.

N/Hance W6501

Фирма-изготовитель: N/Hance Systems Inc.
Цена: с набором кабелей и кассетой - 5450 долл.; дополнительная кассета - 250 долл.
Требуемые ресурсы: DOS 3.0 или более поздняя версия, ОЗУ 640 Кбайт

Система построена на базе механизма фирмы Sony и укомплектована мощным драйвером, который позволяет ей занимать первые места в самых важных тестах. К сожалению, этот драйвер в ряде случаев несовместим с DOS, а это может доставить неприятности неподготовленному пользователю.

Xuxis XY 600RW

Фирма-изготовитель: Xuxis Corp.
Цена: с набором кабелей — 4995 долл.; дополнительная кассета - 200 долл.
Требуемые ресурсы: DOS 3.3 или более поздняя версия, ОЗУ 512 Кбайт
 Фирма Xuxis подчеркивает свою специализацию — оптические накопители, но для ее системы, базирующейся на механизме фирмы Ricoh, характерны высокая цена и характеристики ниже среднего уровня

PLI Peripheral Infinity Optical

Фирма-изготовитель: Peripheral Land Inc.
Цена: с набором кабелей - 4799 долл.; дополнительная кассета - 299 долл.
Требуемые ресурсы: DOS 3.3 или более поздняя версия, ОЗУ 512 Кбайт

Система базируется на механизме фирмы Sony. При средней цене она ориентирована в основном на компьютеры Macintosh, но, если установить ее с правильными опциями, она может достаточно хорошо работать и в комплексе с PC.

Procom MEOD650/E

Фирма-изготовитель: Procom Technology Inc.
Цена: с набором кабелей - 5294 долл.; дополнительная кассета - 295 долл.
Требуемые ресурсы: DOS 2.0 или более поздняя версия, ОЗУ 640 Кбайт

Система построена на механизме фирмы Sony. Благодаря поставляемой утилите DupoBoot она может стать полезным дополнением к существующей периферии с интерфейсом SCSI. Цена и характеристики - средние.

ASC MO-SS

Фирма-изготовитель: Advanced Storage Concepts Inc.
Цена: с набором кабелей и кассетой - 5595 долл.; дополнительная кассета — 300 долл.

Требуемые ресурсы: DOS 3.3 или более поздняя версия, ОЗУ 640 Кбайт

Единственная система, в которой используется механизм фирмы Maxortix, показала отличные результаты, хотя компоновка узлов оставляет желать лучшего.

SST Storage Stak II

Фирма-изготовитель: Specialized Systems Technology Inc.
Цена: с набором кабелей — 4422 долл.; дополнительная кассета - 165 долл.
Требуемые ресурсы: DOS 3.3 или более поздняя версия, ОЗУ 512 Кбайт

Специализация фирмы SST - недорогие внешние накопители. К сожалению, эта система, построенная на механизме фирмы Ricoh, показала характеристики "в конце турнирной таблицы".

SD LaserStor Erasable Optical 650 MB

Фирма-изготовитель: Storage Dimensions Inc.
Цена: с набором кабелей - 5295 долл.; дополнительная кассета - 255 долл.
Требуемые ресурсы: DOS 3.x или более поздняя версия, ОЗУ 512 Кбайт

Характеристики не блестящие, цена — высокая, документация — невразумительная, общая оценка — низкая.

OCEAN Tidalwave 650

Фирма-изготовитель: OCEAN Microsystems Inc.
Цена: с набором кабелей и кассетой - 4495 долл.; дополнительная кассета - 249 долл.
Требуемые ресурсы: DOS 3.3 или более поздняя версия, ОЗУ 512 Кбайт

Система основана на механизме фирмы Sony. При умеренной цене она обладает весьма хорошими характеристиками. Вместе с системой поставляется небольшой, немного капризный драйвер.

Многофункциональные накопители сочетают преимущества двух подходов

Многофункциональные накопители наступают на пятки оптическим устройствам с многократной перезаписью. Многофункциональный накопитель — это устройство, которое, в зависимости от того, какой тип диска вы используете, может работать в режимах как однократной записи, так и с перезаписью. На первый взгляд, такой аппарат не заслуживает отдельного разговора, но на самом деле это сильный претендент на лидерства на будущем рынке оптических устройств. Цена многофункционального накопителя лишь не намного превышает цену системы с возможностью перезаписи, например, многофункциональный накопитель E511 фирмы Sony стоит 4895 долл., а цена накопителя E501 с возможностью перезаписи — 4650 долл., в то время, как стоимость устройства с однократной записью колеблется в пределах от 2000 долл. до 4000 долл. Очевидно, имеет смысл купить многофункциональный накопитель, даже если вы не уверены, что будете использовать оба его режима. Сейчас фирмы, производящие многофункциональные накопители, разделились на три лагеря. В первом, куда входят Hewlett-Packard и Sony, магнитооптические диски предпочитают использовать и в том и в другом режимах. Во втором лагере, который возглавляет фирма Pioneer, для однократной записи используют диполимерные диски, а для режима с перезаписью — MO-диски, поддерживающие различные форматы. В третьем, куда входит и Panasonic, для однократной записи, и для перезаписи используют диски, записанные по технологии, основанной на изменении фазового состояния рабочего слоя.

МНОГОФУНКЦИОНАЛЬНЫЕ УСТРОЙСТВА ФИРМ HEWLETT-PACKARD И SONY

Накопители фирм HP и Sony не были готовы к моменту, когда проводилось тестирование устройств для настоящего обзора. Специалисты обеих фирм утверждают, что в своих разработках они руководствуются стандартами ISO и ANSI для дисков с возможностью перезаписи, но распространяют их и на диски с однократной записью. Устройства работают как обычные MO-накопители, но в режиме однократ-

ной записи они просто "отказываются" стирать информацию. Ключевой момент здесь в нанесении на диск, предназначенный для однократной записи, специального идентифицирующего кода прямо в процессе его производства. Диск распознает код и принимает решение о том, с каким диском он работает — для однократной или многократной записи. Правда, если вы вставите диск со специальным кодом в обычный накопитель, он выдаст сообщение об ошибке и откажется читать или записывать информацию.

МО-НАКОПИТЕЛИ ФИРМЫ PIONEER

Подход фирмы Pioneer к разработке многофункциональных накопителей, кроме нее самой (компания Pioneer не смогла предоставить нам устройство для тестирования к моменту написания обзора), разделяют фирмы Laser Magnetic Storage International (устройство LMSI LD520-602 стоимостью 3700 долл.) и TenX Technology (устройство TenX OptiXchange стоимостью 5995 долл.). Эти накопители используют для однократной записи диполимерные диски, а также MO-диски с многократной записью. (Кроме того, версия LMSI может использовать диски для однократной записи, соответствующие стандарту

ANSI.) Однако применяемые здесь MO-диски не совместимы с дисками стандарта ANSI. Фирма Pioneer использует альтернативный формат ISQ, называемый sampled servo (SS), и, кстати, утверждает, что формат SS превосходит в техническом отношении CCS, так как приводит к более высокой плотности записи и позволяет существенно облегчить головки накопителя. В результате емкость диска получается больше, а быстродействие — выше. Однако в наших тестах дисководы обеих фирм — как LMSI, так и TenX — показали результаты, свойственные средним CCS-накопителям. При копировании 115 Мбайт информации с диска и на диск утилитой XCOPY оба накопителя показали быстродействие ниже, чем большинство дисководов стандарта CCS (намного больше 40 мин. при копировании информации с жесткого диска и около 18 мин. для восстановления ее на диске). Интересно, что устройство LMSI

LD520-602 с 8-разрядной платой фирмы Corel продемонстрировало более высокое быстродействие, чем устройство TenX OptiXchange с 16-разрядной платой фирмы Adaptec.

НАКОПИТЕЛИ ФИРМЫ PANASONIC

Третий подход к проблеме многофункциональных накопителей состоит в использовании носителей, на которые информация записывается по технологии, основанной на изменении фазового состояния как для дисков с однократной записью, так и для дисков с возможностью перезаписи. Подобные дисководы, в которых используются механизмы фирмы Matsushita, поставляют фирмы Panasonic и Reflection Systems. К моменту публикации обзора фирма Corel Systems также предполагает выпустить устройство, построенное на базе механизма Matsushita. Дисководы, основанные на использовании изменения фазового состояния, — первые коммерческие оптические накопители с возможностью перезаписи, в которых запись осуществляется за один проход. Действительно, в тестах копирования на оптический диск с помощью утилиты XCOPY оба устройства показали гораздо лучшие результаты, чем большинство проверенных нами MO-накопителей (около 28 минут на копирование 115 Мбайт с жесткого диска). Только системы фирм N/Hance и Advanced Storage Concepts имеют результаты того же порядка или более высокие. Короче говоря, накопители, использующие изменение фазового состояния, в среднем работают быстрее, чем MO-накопители, хотя и тут есть исключения. "Фазовые" накопители, кроме того, показали совершенно различные результаты при копировании данных на диск. Система RF-7010 фирмы Reflection продемонстрировала только предпоследний результат (17 мин. 40 сек.), тогда как устройство фирмы Panasonic показало лучшее среди всех время — 13 мин. 40 сек. Такая разница в производительности говорит, что подходить к выбору оптического накопителя необходимо исходя из всех его компонентов, включая SCSI-контроллер и программное обеспечение.

ТОЛЬКО ФАКТЫ

Тесмар LaserVault

Фирма-изготовитель: Тесмар Inc.
Цена: с набором кабелей — 6885 долл.; дополнительная кассета - 295 долл.

Требуемые ресурсы: DOS 3.1 или более поздняя версия, ОЗУ 640 Кбайт

Система построена на базе механизма фирмы Sony. Самая дорогая среди рассматриваемых, но показывает хорошие характеристики и предоставляет много возможностей пользователю

вых операционных систем NetWare 286 и NetWare 386 фирмы Novell, а также для нескольких версий Unix. Замечательная особенность Laser Vault — возможность создавать на одном диске разделы для разных операционных систем. Сейчас в число этих систем входят DOS, Macintosh и AUX (хотя подразделы AUX можно создать только в разделе для Macintosh). Допускается также обмен данными между разделами DOS и

АППАРАТНЫЕ СРЕДСТВА

Macintosh, осуществляемый через специальный обменный раздел. Доступ к разделу производится с помощью программы LaserBack, которая превращает MO-диск в виртуальный накопитель на ленте. Копирование данных с помощью утилиты LaserBack происходит гораздо быстрее, чем посредством XCOPY. И поскольку обменный раздел доступен программе LaserBack как при работе с DOS, так и при работе с Macintosh, с помощью этой программы можно копировать и восстанавливать данные в любой среде.

Установить LaserVault несложно. Нужно вставить в разъем 16-разрядную плату АНА-1540В фирмы Adaptec, соединить накопитель и плату кабелем и вставить в накопитель сетевой шнур. Для установки программного обеспечения нужно скопировать на жесткий диск все файлы, включая два драйвера, общим объемом около 35 Кбайт. Затем программа установки добавляет необхо-

димые строки вида "DEVICE=..." в ваш файл CONFIG.SYS. В программе установки имеется возможность создать на MO-кассете один раздел для DOS, один обменный раздел или разделить диск поровну между этими двумя разделами. Если вам нужно разбить диск более сложным образом, можно изначально определить необходимое количество разделов, либо потом добавить дополнительные разделы. Руководство пользователя написано хорошо и содержит полную информацию о том, как нужно устанавливать устройства с интерфейсом SCSI. Очевидно, LaserVault — одно из наиболее богатых возможностями устройств, представленных в настоящем обзоре. Хорошие временные характеристики, высокая скорость копирования при использовании программы LaserBack, возможность создавать на одном диске разделы для разных операционных систем, обмен данными между разделами, созданными в разных средах, — все это заслуживает особого внимания. Всех перечисленных удобств достаточно для оправдания высокой цены устройства.

Совместное советско-американское предприятие

Параграф

представляет лучшие продукты года
по самым низким в мире ценам

Русский Парадокс

Новый лидер баз данных - для всех программистов и пользователей

Русский Парадокс включает самую последнюю версию лучшей в мире системы управления базами данных Paradox 3.51 корпорации Borland® и программу РЗР СП ПараГраф®, обеспечивающую полную русификацию СУБД Paradox 3.51.

Русский Парадокс поставляется с комплектом документации на английском и на русском языках. Владельцам Русского Парадокса предоставляется право на льготное получение новых версий СУБД Paradox и на обслуживание в режиме "hot-line".

Русский Парадокс - это стандарт поставки СУБД Paradox 3.51 на советском рынке, утвержденный корпорацией Borland®.

Русский Парадокс - это Ваш шанс перейти одновременно со всем миром на новый стандарт баз данных с отживающего dBase .

Цена системы Русский Парадокс - 9900 рублей.

РусскоеСлово

Комплексное решение всех проблем при работе с русским текстом на персональном компьютере.

РусскоеСлово включает популярнейший в мире текстовый процессор Microsoft® Word 5.0, поставляемый по лицензионному соглашению с корпорацией Microsoft®, пакет программ и комплект шрифтов СП ПараГраф®, обеспечивающих все необходимое для работы с русским текстом в среде данного текстового процессора.

РусскоеСлово поставляется с полным комплектом документации, его владельцы могут пользоваться поддержкой в режиме "hot-line".

Цена комплекса РусскоеСлово - 2990 рублей.

Наш адрес: 103051, Москва, Петровский бульвар, 23.

Телефоны: (095)-200-25-66; 924-17-81; 200-03-90.

ФАКС: (095)-928-27-68

PARAGRAPH

Компьютер Toshiba 3200SXC с цветным ЖК-экраном

Патрик Хонен

Глядя на цветной жидкокристаллический VGA-дисплей нового портативного компьютера модели T3200SXC компании Toshiba, специалист, наверное, подумал бы: "не иначе как активная матрица на базе тонкопленочных транзисторов". Простой же человек воскликнул бы: "Вот это да!".

Действительно, цвета на экране выглядят яркими и насыщенными, изображение очень четкое. Если говорить кратко, то различия в качестве изображения между ЖК-дисплеем компании Toshiba и обычным цветным VGA-монитором отсутствуют.

Однако высокое качество отразилось на цене компьютера. Модель T3200SXC на базе 20-МГц процессора 386SX с 1-Мбайт ОЗУ, 89-мм НГМД емкостью 1,44 Мбайт и 120-Мбайт НЖМД с временем доступа 19 мс стоит 7249 долл. Это на 2850 долл. больше стоимости портативного компьютера Toshiba T3200SX, сопоставимого по всем параметрам, за исключением того, что в нем используются 16-МГц процессор 386SX и газоразрядный VGA-дисплей. Кроме того, поскольку энергопотребление цветных ЖК-дисплеев довольно велико, компания Toshiba приняла решение комплектовать новым дисплеем машины, питающиеся только от сети переменного тока.

Основным поставщиком дисплея с размером по диагонали 26 см для компании Toshiba является фирма Display Technologies — совместное предприятие компании Toshiba и японского отделения корпорации IBM. Вideosистема выполнена на базе комплекта ИС фирмы Paradise.

Если в прошлом качество цветных ЖК-дисплеев вас не удовлетворяло, то это, возможно, из-за того, что они изготавливались по тех-

нологии пассивной матрицы. В этой матрице для включения и выключения каждого ряда и столбца матрицы элементов изображения (ЭИ) используется отдельный генератор запускающих импульсов. В результате на каждый ЭИ приходится лишь незначительная часть времени, отводимая на заряд всего ряда и столбца, и область управляющих напряжений (диапазон яркостей), приходящаяся на один ЭИ, оказывается довольно малой.

В дисплее на базе активной матрицы для управления каждым ЭИ используется свой собственный транзистор; следовательно, на каждом ЭИ постоянно присутствует заряд. В результате изображение на экране оказывается существенно ярче и контрастнее и не содержит полос и теней, характерных для дисплеев с пассивной матрицей.

Еще одно достоинство нового дисплея состоит в том, что изображение на его экране хорошо видно под любым углом зрения — это немаловажно, если вы собираетесь использовать компьютер для показа демонстрационной графики небольшим группам людей. В то же время наилучшее восприятие изображения на дисплее с пассивной матрицей достигается только тогда, когда вы смотрите на экран строго под прямым углом.

Однако у дисплеев с активной матрицей есть и недостаток — гораздо более высокая трудоемкость изготовления. Для управления ЭИ красного, зеленого и синего свечения каждая активная матрица цветного VGA-дисплея должна включать в себя около одного миллиона транзисторов, расположенных в нужном порядке и с высокой точностью. Подобная технология оказывается дорогостоящей даже при изготовлении этих изделий на заводе, оснащенном самым новейшим оборудованием. В результа-

те цена на дисплеи с активной матрицей колеблется в пределах от 2 до 3,5 тыс. долл. и больше.

Подобно тому, как цветной экран нового компьютера изменил представление о современной технологии, габариты и масса модели T3200SXC заставили поинтересоваться тем, каким должен быть портативный компьютер. Новая модель имеет высоту уже 112 мм, ширину 371 мм, глубину 396 мм и весит 7,8 кг. Компания Toshiba разумно распорядилась образовавшимся за счет увеличения габаритов дополнительным пространством внутри корпуса компьютера и предусмотрела большие возможности расширения ресурсов, что позволяет использовать модель T3200SXC в качестве альтернативы настольному компьютеру.

В корпусе новой машины может быть установлено ОЗУ емкостью до 13 Мбайт. В 16-разрядное гнездо расширения полной длины могут устанавливаться платы с шиной ISA, а второе гнездо рассчитано под 8-разрядные платы половинной длины с шиной ISA или платы,

число гнезд расширения, то можно приобрести расширительное шасси производства этой же компании, имеющее пять 8-разрядных гнезд. Шасси и соответствующая интерфейсная плата стоят 950 долл.

Качество ЖК-дисплея таково, что вы, по всей вероятности, не захотите подключать к модели T3200SXC внешний монитор в случае использования этой машины в качестве настольной системы. Единственное, что может побудить приобрести внешний монитор, — это необходимость иметь более высокую разрешающую способность (разрешение встроенного ЖК-дисплея составляет 640 x 480 ЭИ). Но в этом случае вам также понадобится купить плату типа SuperVGA, поскольку встроенный в компьютер VGA-порт обеспечивает разрешение всего 640 x 480 ЭИ.

Насколько интересен для пользователей цветной экран модели T3200SXC, настольного его габариты, цена и отсутствие батарейного питания непривлекательны для тех, кому необходим портативный компьютер в традиционном понимании этого слова. Однако как средство показа демонстрационной графики при ведении деловых переговоров, а также как устройство, которое можно брать с собой на работу, машина T3200SXC имеет несомненные преимущества.



Дисплей на базе активной матрицы тонкопленочных элементов в компьютере модели T3200SXC обеспечивает такое же высокое качество воспроизведения цветов, как и VGA-мониторы настольных компьютеров.

выпускаемые компанией Toshiba (например, плату адаптера сети Ethernet стоимостью 559 долл.) Третье гнездо зарезервировано для специальной платы-модема со скоростью передачи данных 2400 бит/с (279 долл.) Если требуется большее

Цена по каталогу: конфигурация с 1-Мбайт ОЗУ и 120-Мбайт жестким диском — 7249 долл. Toshiba America Information Systems Inc., Computer Systems Division, 9740 Irvine Blvd., Irvine, CA 92718; 800-457-7777.

Карманные модемы — от максимума до минимума необходимого

Карманный вариант модема Hayes 2400

Кристофер Барр

Карманный вариант модема фирмы Hayes на 24.00 бит/с, подключаемого как внешнее устройство, не имеет такого широкого набора функций, как большинство современных модемов, однако его появление на рынке карманных модемов можно считать вполне своевременным.

Основное достоинство этого устройства — портативность (масса всего 85 г) — является и его главным недостатком: нет места для установки элементов питания. Модем получает питание из последовательного порта и телефонной линии.

Потребление мощности от батареи портативного компьютера минимально; питание из последовательного порта "используется" лишь командами AT, выдаваемыми на модем. Все другие функции модема "записываются" через гнездо RJ-11; одним из недостатков такой конфигурации является то, что нельзя пользоваться акустическим соединителем, и, кроме того, при подключении к некоторым цифровым сетям иногда невозможно установить соединение из-за отсутствия в них необходимого напряжения питания.

Далее, модем не имеет звуковой индикации установления соединения. Не имеет он и световых индикаторов, а также встроенных средств компрессии данных и контроля ошибок.

В карманном модеме Hayes 2400 все компактно. По размерам это устройство несколько меньше колоды

игральных карт, оно не требует внешнего кабеля и не имеет ни одного DIP-переключателя.

Модем конструктивно выполнен как одно целое с 25-см кабелем, на конце которого имеется 9-контактный соединитель, и телефонным шнуром длиной около 2 м с вилкой RG-11.

В модеме частично реализован набор AT-команд устройства Hayes Smartmodem 2400, что делает его, как можно догадаться, совместимым со всеми модемами фирмы Hayes. В модеме предусмотрено энергонезависимое ЗУ для хранения двух различных функциональных конфигураций и четырех (максимум) телефонных номеров.

Программное обеспечение модема — Smartcom EZ

— представляет собой частичную реализацию полнофункционального пакета Smartcom Exec. Этот "урезанный" пакет программ поддерживает лишь эмуляцию виртуального терминала VT 102, а в качестве протокола передачи файлов с исправлением ошибок — только протокол Xmodem. Кроме того, отсутствие языка сценариев делает его еще более скромным. Однако можно пользоваться любыми связными программами, которые распознают набор AT-команд модема Hayes Smartmodem 2400.

Если все, что вам требуется, — это удобство доступа к службе электронной почты, и если вы всегда будете подключены к обычному аналоговому телефонному каналу, то карманный



Карманный модем Hayes 2400 не требует батарей, но имеет лишь минимум необходимого.

модем Hayes 2400 позволит "экономить состояние" только на одних батареях.

Цена по каталогу: 179 долл.

Требуемые ресурсы: последовательный 9-контактный порт RS-232, гнездо типа RJ-11. Hayes Microcomputer Products Inc., P.O. Box 105203, Atlanta, GA 30348, 404-449-8791

Портативный модем WorldPort 9600

М. Дэвид Стоун

Когда речь идет о компьютерах и же с ними, "портативный" часто означает, что для уменьшения размеров чем-то "все-таки пришлось пожертвовать. Однако этого не скажешь о модеме WorldPort 9600 — устройстве на 9600 бит/с производства фирмы Touchbase Systems.

Это довольно-таки дорогое устройство (899 долл.) имеет размеры всего 25 x 125 x 70 мм, т. е. сравнимо с обычной аудиокассетой, только чуть-чуть толще.

Такое малогабаритное исполнение (масса немного

больше 200 г, включая встроенный звукогенератор) имеют многие модемы, выпускаемые фирмой Touchbase.

Кроме совместимости на скоростях 9600 и 4800 бит/с с рек. V.32 МККТТ, устройство также совместимо с рек. V.22bis на скорости 2400 бит/с, на скорости 1200 бит/с — как с американским стандартом Bell 212, так и с рек. V.22, принятой в Европе, а на скорости 300 бит/с — как со стандартом Bell 103, так и с рек. V.21. Оно даже совместимо с более старым европейским стандартом — рек. V.23 МККТТ.

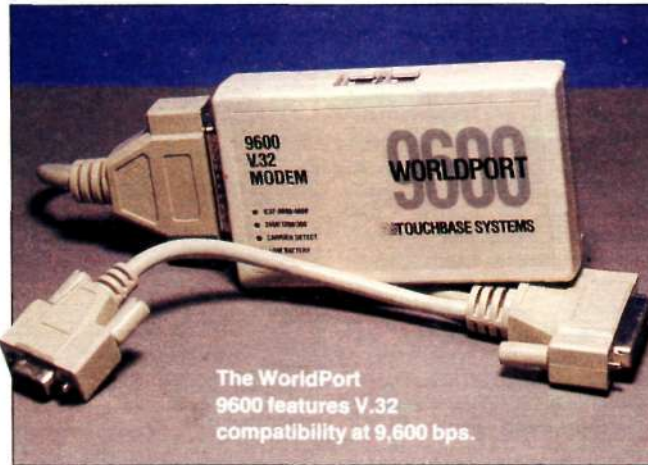
Контроль ошибок, кото-

рый можно считать своего рода полезной роскошью на скорости 2400 бит/с, на более высоких скоростях — вещь абсолютно необходимая. Поэтому и не удивительно, что в модеме WorldPort 9600 реализованы протоколы с MNP 2 по MNP 4. Проведенные испытания показали, что модем спокойно может обмениваться данными с модемами типа Hayes Ultra 96 как на скорости 9600 бит/с, так и на скорости 2400 бит/с.

Единственное, что отсутствует в модеме, — это сжатие данных. Однако вместе с модемом поставляется программа MTEZ (фирма MagicSoft), которая реализует процедуру сжатия данных MNP 5 на программном уровне. Программные процедуры сжатия несколько менее эффективны,

чем аппаратные, но все-же передача файлов осуществляется быстрее. В процессе испытаний использовался модем именно с такой программой, и было установлено, что никаких проблем при обмене файлами с модемами типа Ultra 96, когда применяется процедура MNP 5, не возникает. Здесь уместно заметить, что, хотя во многих модемах, соответствующих рек. V42bis, эта процедура реализована, она не является частью данного стандарта на сжатие данных.

Если у вас уже есть связанная программа, которой вы предпочитаете пользоваться, модем WorldPort 9600, скорее всего, с ней работать будет. Единственным исключением могут



быть программы, написанные специально для модемов типа Hayes.

Модем WorldPort 9600 имеет индикаторы скорости передачи, несущей и падения напряжения батареи ниже нормы, т. е. светодиодов больше, чем на большинстве портативных модемов. Хотя это и не полный комплект контрольных индикаторов, отсутствие некоторых из них — не столь уж высокая цена за такое портативное устройство.

Цена по каталогу: 899 долл.

Требуемые ресурсы: последовательный порт. Touchbase Systems Inc., 160 Laurel Ave., Northport, NY 11768, 800-541-0345, 516-261-0423.

Увеличьте поле зрения

Билл Хауард

В некоторых портативных компьютерах с VGA-дисплеем, например в машинах компании Toshiba, в текстовом режиме не используются области шириной примерно полтора сантиметра в верхней и нижней частях экрана. Дело в том, что в этом режиме в VGA-стандарте используется 400 строк, в то время как в графическом режиме — 480. Программа Laptop UltraVision компании Personics Corp. стоимостью 69,95 долл. увеличивает изображение на экране, позволяя эффектив-

но использовать весь экран. Установка этой программы не обязательно требуется на каждом портативном компьютере, так как многие из них имеют подобную опцию выбора растянутых и сжатых шрифтов в меню задания параметров работы. Как узнать, что ваш портативный компьютер выиграет от применения новой программы? Для этого достаточно запустить графическую программу, а затем текстоориентированную программу. Если изображение на экране сожмется по вертикали, то есть смысл приобрести про-

грамму Laptop UltraVision.

Остальная часть Laptop UltraVision аналогична версии этой программы для настольных компьютеров и работает с любым портативным компьютером, оснащенным EGA- или VGA-монитором. Имеется возможность вывода на VGA-дисплей 25, 36, 50 или 63 строк (максимум 43 строки на EGA-дисплее). В отличие от варианта UltraVision для настольного компьютера, данная версия выводит в каждой строке 80 знакомест. Имеется возможность выбора курсора в виде прямоугольника или символа

подчеркивания, вывода цветного изображения полутонами (оттенками серого) и использования различных шрифтов. Часть шрифтов имеют только светлое начертание (Old English, Broadway); другие дают более жирные, темные легко читаемые начертания.

Цена по каталогу: 69,95 долл.

Требуемые ресурсы: EGA- или VGA-дисплей, ОЗУ 16 Кбайт, DOS версии 2.0 или более поздней.

Personics Corp., 63 Great Rd., Maynard, MA 01754; 800-445-3311, 508-897-1575.

ХОТИТЕ БЫТЬ С КОМПЬЮТЕРОМ НА «ТЫ»?

Ведущие компьютерные фирмы — APPLE, AUTODESK, NOVELL, RANK XEROX — доверили СП «Интермикро» обучение по направлениям: локальные сети, настольные издательства, САПР...

Авторизованный Учебный центр СП «Интермикро» — это интенсивное интерактивное обучение в прекрасно оборудованных IBM PC/AT и Macintosh классах, прошедшие зарубежную стажировку преподаватели, фирменные методические пособия, телефонная поддержка, гостиница для приезжих.

Сертификаты учебного центра имеют международный статус.

Ждем Вас по адресу: Москва, ул. Нижняя Красносельская, д.39

Телефон: 261-16-91, 264-31-21

Фирма Motorola создает беспроводную ЛВС

Стив Райни

Компания Motorola — главный производитель средств двусторонней радиосвязи в США, используя свой богатый опыт, разработала беспроводную локальную вычислительную сеть (ЛВС) своей собственной фирменной архитектуры. Эта архитектура называется архитектурой "беспроводных внутренних сетей зданий" (Wireless Inbuilding Network — WIN), а сама сеть получила название Altair.

Даже в этой, только что появившейся области технологии у компании уже имеется несколько конкурентов. Предлагаемая система, в частности, соперничает с беспроводными сетями WaveLAN фирмы NCR (см. рубрику "First Looks" в журнале PC Magazine, May 14, 1991) и SLW фирмы Telesystems (см. статью "The Next Wave: LANs Without Wires", PC Magazine, May 29, 1990).

Система Altair работает на частоте 18 ГГц — это так называемая "крайне высокая" частота, обычно используемая в спутниковой связи. Ее важным достоинством является то, что она довольно устойчива к помехам и может работать в условиях контроля радиобмена. Что касается сети WaveLAN, то в ней используются специальные методы модуляции с расширением спектра сигналов в диапазонах более загруженных частот вблизи полосы сотовой радиотелефонной связи.

Важная отличительная особенность системы состоит в ее полной независимости от операционных систем (ОС) ЛВС. Средства сети WaveLAN фирмы NCR и модуль ARLAN-450 фирмы Telesystems размещаются внутри персонального компьютера (ПК), который должен иметь сетевой интерфейс, и требуют специальных программ-драйверов для каждой поддерживаемой ими ОС. Что же касается модулей системы

Altair, то они просто подключаются к кабельным сегментам ЛВС и осуществляют передачу пакетов в формате Ethernet. Модули совместимы со всеми ОС, имеющими средства управления сетевыми адаптерами Ethernet, в том числе с NetWare, LAN Manager, LANtastic, Unix и VMS.

Стоимость базового комплекта системы Altair составляет 7490 долл., причем в него входят все компоненты, необходимые для построения радиосистемы связи двух сегментов ЛВС типа Ethernet, в том числе тонкий кабель и терминатор. Несмотря на то что предлагаемая конфигурация дороже базового комплекта WaveLAN (2780 долл.), стоимость построения сети в случае крупных ЛВС, содержащих до 30 рабочих станций, падает до 715 долл./порт.

Развернуть сеть беспроводной связи с применением модулей Altair быстрее, чем объяснить, как это делается. Представьте себе два сегмента ЛВС Ethernet, один с 33 узлами, а другой — с семью. Одним из узлов более крупного сетевого сегмента служит модуль управления со встроенными приемопередатчиком и антенной.

Одним из узлов меньшего сегмента ЛВС является модуль пользователя. Эти модули обеспечивают передачу пакетов в формате Ethernet между кабельными сегментами этой ЛВС. С одним и тем же модулем управления могут взаимодействовать до пяти модулей пользователя, к каждому из которых может подключаться до шести компьютеров или других устройств. Модуль управления выполняет функции интеллектуального моста, читающего адрес каждого пакета и направляющего этот пакет соответствующему модулю пользователя.

Мы провели контрольно-оценочные испытания модулей по методике лаборатории тестирования ЛВС журнала PC Magazine. В результате оказалось, что средняя пропускная способ-



ность сети с сервером на базе ОС NetWare 3.1 в режиме обработки запросов ПК с микропроцессором 386SX составляет около 470 кбит/с. Это лишь немногим меньше, чем в случае сети WaveLAN; для сравнения можно отметить, что такой показатель приблизительно соответствует скорости обмена с жестким диском компьютера серии AT. В то же время по стандартному медному кабелю тот же самый ПК осуществлял бы обмен данными со средней скоростью более 2 Мбит/с. Таким образом, обе беспроводные ЛВС требуют, чтобы пользователь жертвовал пропускной способностью в пользу гибкости и легкости развертывания сети, хотя любая из этих систем обладает вполне достаточной пропускной способностью для большинства офисов.

Расстояние между модулем управления и модулями пользователя может составлять до 40 м, если путь для распространения радиоволн свободен. Если же на пути распространения радиосигналов имеются препятствия, например стена сухой кладки или конторская мебель, то это расстояние уменьшается примерно до 12 м. Однако высокочастотные сигналы, используемые в системе Altair, не проникают через обычные бетонные или кирпичные стены, тогда как в случае сети WaveLAN, которая работает на длинах волн сотовой радиотелефонной сети, сигналы проникают сквозь стены довольно легко.

Модули Altair передают информацию с уровнями излучения, существенно бо-

лее низкими, чем предельные значения, которые регламентируются стандартами США и Европы. Для установки этих модулей, вообще говоря, необходимо разрешение Федеральной комиссии по связи (ФКС), однако компания Motorola получила от ФКС лицензию, которая распространяется на всю территорию США, так что модули можно эксплуатировать по условиям этой лицензии.

Предлагаемые компанией Motorola модули Altair являются отличной альтернативой средствам построения обычных (проводных) ЛВС; по пропускной способности предлагаемая система уступает сетям на базе медного кабеля, однако она, безусловно, является достойным конкурентом беспроводной сетевой системы WaveLAN.

ТОЛЬКО ФАКТЫ

Altair — беспроводная ЛВС типа Ethernet
Motorola Inc., 3215
N. Wilke Rd., Arlington
Heights, IL
60004, 800-233-0877

Цены по каталогу:
модуль управления —
3995 долл.; модуль
пользователя — 3495
долл.

В отличие от других
беспроводных ЛВС
модули системы Altair
способны работать с
любой сетью типа
Ethernet и любой ОС
ЛВС, обеспечивая
связь между кабельными
сегментами на
расстояниях до 40 м.
Модули отличаются
исключительной
простотой установки и
обладают достаточно
высокой пропускной
способностью.

LANstep фирмы Hayes — простая сетевая ОС, практически не уступающая системе LANtastic

Новая недорогая и достаточно производительная сетевая ОС, имеющая, однако, один недостаток — необычный интерфейс с пользователем

Рик Эйр,
Рон Андерсон

Когда вы опаздываете на дружескую вечеринку, то лучше всего появиться в компании достаточно эффектно. Возможно, именно так и думала гигантская фирма по средствам связи Hayes Microcomputer Products, когда она планировала создание своей первой сетевой ОС — LANstep.

Система появляется с существенным опозданием, если учесть, что такие ОС, как LANtastic, ELS и полдюжины других, уже давно зарекомендовали себя на рынке.

Система построена на базе ОС "недосовского" типа и имеет весьма ориги-

нальный интерфейс с пользователем. Для ее работы требуется, чтобы в сетевой сервер, причем не обязательно специализированный, была загружена ее собственная операционная подсистема. Система предусматривает значительную гибкость и предоставляет возможность задавать для рабочих станций конфигурацию либо станции с LANstep, либо станции с DOS.

Рабочие станции первого типа осуществляют начальную загрузку при помощи сервера и работают в защищенном режиме, тогда как станции с DOS загружают сетевую информацию, обеспечивающую доступ к сетевым услугам, со своих соб-

ственных накопителей. Рабочие станции с DOS предусматривают разделение принтеров, файлов и программ, но не могут передавать свои локальные файлы по сети в широковещательном режиме. Для системы LANstep характерна также высокая потребность в вычислительных ресурсах. Так, например, в компьютере IBM PS/2 модели 60, имеющем конфигурацию рабочей станции с DOS, эта ОС занимает около 95 Кбайт памяти.

Если сравнивать возможности предлагаемой ОС с возможностями других ОС ЛВС такого класса, то она практически не уступает им. Подобно большинству своих конкурентов она реализует коллективное использование файлов и принтеров, причем возможно разделение прикладных программ, рассчитанных на работу как в среде DOS, так и в среде Windows. Система содержит простые для использования инструментальные средства, позволяющие вводить в сеть дополнительные прикладные программы, увеличивать число пользователей и рабочих станций.

В системе LANstep реализована также мощная программа электронной почты, позволяющая создавать списки почтовых отправок, рассылать заказные почтовые сообщения и дополнять сообщения файлами или даже целыми каталогами. Программа содержит встроенный редактор и является вполне эффективной для большинства небольших сетей. (За отдельную плату можно приобрести шлюз.)

Самой примечательной особенностью предлагаемой ОС является ее интерфейс с

ТОЛЬКО ФАКТЫ

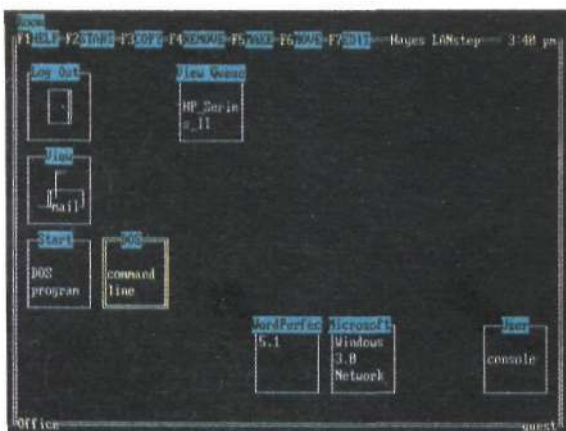
Система LANstep
Фирма Hayes
Microcomputer Products
Inc., P.O. Box 105203,
Atlanta, GA 30348;
404-441-1617.

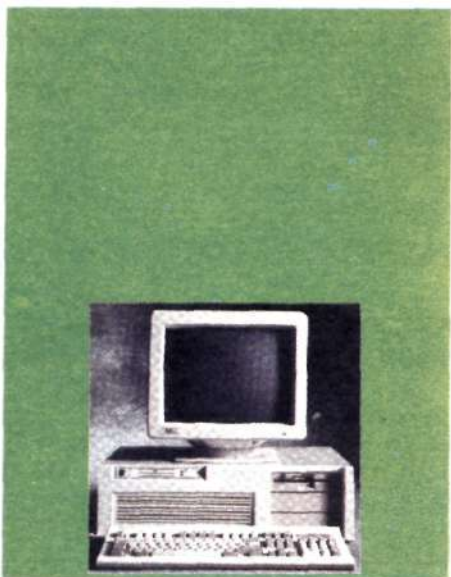
Цена по каталогу:
базовый комплект на
пять станций - 595
долл.; на каждые
дополнительные пять
станций - 395 долл.

Требуемые ресурсы:
для сетевого
программного сервера -
персональный
компьютер на базе
микропроцессора 286
или более мощного,
20-Мбайт ОЗУ,
1,5-Мбайт расширенная
память, сетевая плата;
для рабочей станции с
LANstep - персональный
компьютер с
микропроцессором 286
или более мощным,
1-Мбайт расширенная
память, сетевая плата;
для рабочей станции с
DOS - персональный
компьютер с
микропроцессором 8088
или более мощным,
сетевая плата.
LANstep - сетевая ОС,
обладающая более
широкими
возможностями по
сравнению с обычными
ОС ЛВС данного класса.
Система требует
больших аппаратных
ресурсов, а ее
интерфейс с
пользователем является
нестандартным, хотя и
простым в освоении.

пользователем. Этот интерфейс основан на модели помещения с несколькими комнатами — так называемой "экранной конторы". Комнаты этой конторы служат для выполнения определенных функций, как, например, комната DOS, или обеспечивают переход из одной программной среды в другую, как, например, комната почтового ящика. Экранная контора реализует те же самые функции, что и Program Manager в Windows, однако символные пиктограммы предлагаемой ОС делают интерфейс с пользователем, на первый взгляд, довольно примитивным.

(продолжение см. стр. 39)





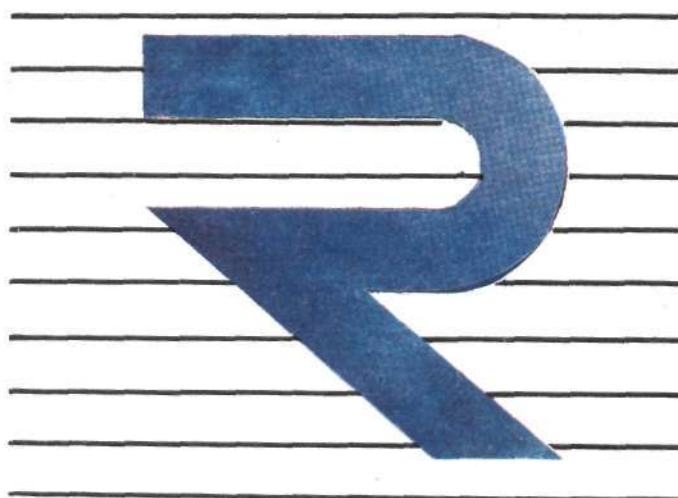
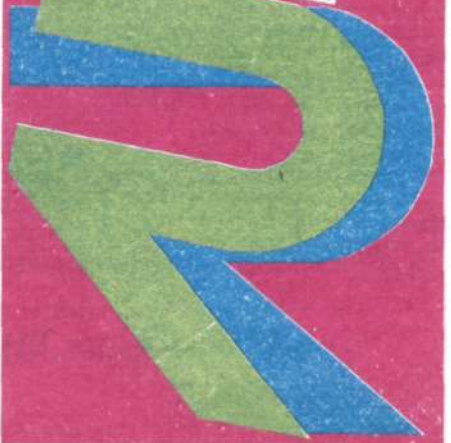
**Ленинградская
ФИРМА**



Предлагает



Обеспечивает



Р У Б И К О Н

**один из крупнейших
поставщиков
компьютеров
в Северо-Западном
регионе СССР**

**полный ряд компьютеров,
совместимых с IBM PC/AT
ксероксы, телефаксы,
телефоны**

гарантийное обслуживание

**Телефоны: (812) 105-64-34
(812) 268-47-48
(812) 268-61-98**

Факс: (812) 269-11-83

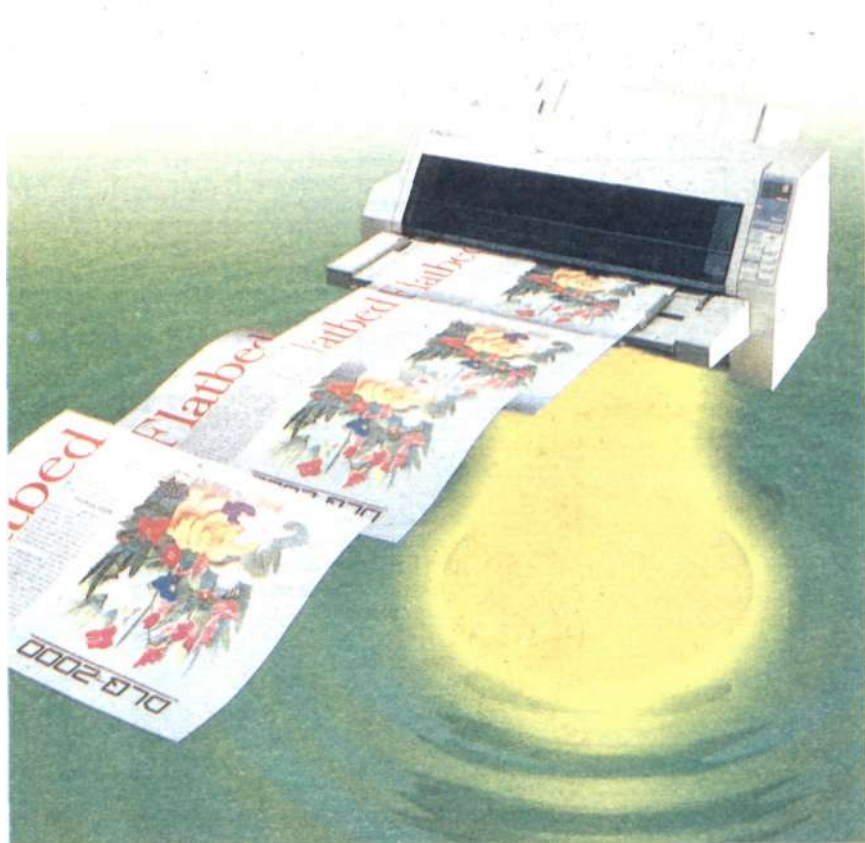
Телекс: 121194 RUB SU

Р У Б И К О Н

**это
высокое
качество
обслуживания,
надежное партнерство
и самые низкие цены!**

Малоизвестные страницы из жизни Epson

Юрий Кузьмин



На моем рабочем столе, как и у десятков тысяч пользователей персональных компьютеров, стоит матричный принтер фирмы Epson. Принтеры Epson настолько повсюду, что в сознании большинства тех, кто работает на компьютерах, название фирмы ассоциируется с жужжанием печатающего механизма и листами бумаги, на которых строка за строкой появляется "машинописный" текст.

Да, фирма Epson — крупнейший в мире изготовитель как матричных печатающих устройств, так и мини-принтеров для разнообразного специализированного оборудования. Однако, вероятно, немногие знают, что Epson — лишь половина названия фирмы. Полное же ее имя — Seiko Epson Corp., и это говорит как минимум о том, что фирма известна не

только принтерами, но и своими знаменитыми кварцевыми часами.

Тем не менее и это еще не все. Корпорация Seiko Epson, ежегодный оборот которой превышает 3 млрд. долл., имеет более 35 отделений в 22 странах (не считая целого ряда предприятий в Японии, где находится и ее штаб-квартира). Специалисты фирмы создали первые в мире жидкокристаллический индикатор для настольных калькуляторов, микросхему — синтезатор музыки, карманный цветной телевизор, блокнотный компьютер и немало других оригинальных изделий.

Сегодня номенклатура продукции Seiko Epson насчитывает десятки различных наименований. Большим спросом, в частности, пользуются ее КМОП-микросхемы, специализиро-

ванные ИС, а также БИС памяти и микропроцессорные комплекты. Будучи изготовителем всемирно известных кварцевых часов, фирма снискала также популярность как поставщик высококачественных кварцевых генераторов и модулей часов реального времени. Seiko Epson, кроме того, является крупнейшим в мире производителем высококоэрцитивных самарий-кобальтовых магнитов и магнитных материалов на основе неодима.

Как уже говорилось выше, фирма Seiko Epson — пионер в области компьютеров блокнотного типа. И сейчас в ассортименте ее изделий такие пользующиеся большим спросом устройства, как 16-разрядный портативный компьютер PX-16, карманная ЭВМ с мини-принтером и портом оп-



КОМПАНИИ

тического интерфейса, получившая название ЕНТ-7L, а также карманный компьютер ЕНТ-10 с сенсорным ЖК-экраном и 256-Кбайт памятью.

Разрабатывая свои технологические процессы, Seiko Epson создала ряд собственных конструкций компактных сборочных роботов, которые теперь широко применяются на производственных предприятиях различных компаний по всему миру.

Можно продолжать этот список и дальше, но для достаточно полного описания продукции Seiko Epson не хватило бы и целого журнала, ведь здесь еще жидкокристаллические дисплеи, карты памяти, шаговые электродвигатели, накопители на жестких -и гибких магнитных дисках, печатающие головки, микропреобразователи угол-код, магнитооптические диски и многое другое.

На открытии московского представительства Seiko Epson (Центр международной торговли на Красной Пресне, телефон 253-81-74) в мае 1991 г. его глава, г-н Т.Касаи, сказал, что деятельность фирмы в Советском Союзе первоначально будет концентрироваться на сбыте печатающих устройств различных типов. Однако потенциальным за-

казчикам не следует забывать, что и советский рынок, и продукция фирмы Seiko Epson — это далеко не одни принтеры.



ВЫ СМОЖЕТЕ разрабатывать более сложные программы за более короткое время

Clipper®

Nantucket

Вы, наверное, уже знаете, что мощную универсальную систему программирования **Clipper 5.0** и комплект, состоящий из **Clipper Summer'87** и **Tools One**, с документацией на русском языке можно приобрести **за рубли**. Напоминаем Вам также, что легальным пользователям Clipper предоставляется целый **комплекс высококачественных услуг**, включая подписку на журнал Nantucket News, издаваемый для профессиональных программистов на русском языке. Возможно Вас заинтересует информация о том, что сегодня в связи с выходом на рынок нового продукта **Clipper 5.0** фирма Nantucket расширяет свою дилерскую сеть и **открыта для контактов**.

Кроме того, имея большой опыт распространения программной продукции во всем мире, мы готовы оказать Вам помощь в организации сбыта на рынке Вашей программной продукции, разработанной на Clipper.

С вопросами и предложениями обращайтесь в СП "Магнит" - официальному представителю фирмы Nantucket в СССР.

**Наш адрес: 127018, г. Москва
ул. 2-я Ямская, д. 15
Тел: 289-44-77 289-44-83**

Гас Вендитто

Уготована ли OS/2 в 1992 году лишь роль "убийцы" системы QEMM?

Ясно, что настроение у фирмы IBM неважное, и вот почему.

Именно OS/2, а не Windows 3.0 должна была быть в центре внимания в 1991 г.

Именно OS/2, а не Windows должна была расходиться миллионными тиражами, изменяя представления людей о том, как пользоваться персональным компьютером.

И опять-таки OS/2, а не Windows должна была побудить разработчиков программного обеспечения строить свои планы на базе графической операционной системы.

Компания поменьше просто бы сдалась, отказалась от OS/2 и бросила свои силы на поддержку Windows, но IBM не из таких.

По прямому распоряжению председателя Совета директоров IBM Дж. Эйкера приказано "окопаться" и подготовиться к длительным боям за захват территории, на которой пока господствуют пользователи DOS и Windows.

С точки зрения IBM, судьба ее пока явно не балует — Windows постепенно становится монополистом, по мере того как под нее выпускаются все новые приложения и утилиты, а Microsoft успешно кружит головы разработчикам, расписывая на все лады свою новую операционную систему OS/2 NT, которая, по словам представителей фирмы, может решительно всё — от реализации многозадачной DOS и поддержки приложений Windows с использованием 32-разрядных команд до способности работать на самых разнообразных платформах: процессорах серии 386, SPARC-машинах и MIPS-компьютерах (см. рубрику

"Pipeline" PC Magazine, April 16, 1991).

Однако, если агитаторов от Microsoft как следует поприжать, то станет ясно, что все эти прожекты вряд ли осуществляться раньше 1993 г.

Учитывая название майкрософтовской ОС — OS/2 NT (где NT означает New Technology — "новая технология"), фирма IBM прекрасно понимает, что ее собственная OS/2 версии 2.0 — первая DOS-совместимая 32-разрядная операционная система для персональных компьютеров — рискует морально устареть задолго до своего появления на рынке.

Здесь следует, однако, сразу оговориться, что и IBM не выполнила своих обещаний, данных в 1987 г. версии OS/2 выходили в срок, но всегда с какими-нибудь недоработками (без драйверов печати, видеодрайверов, с неоптимизированными программами и т. д.).

Но спустя четыре года IBM наконец обрела уверенность, что очень скоро 32-разрядная версия OS/2 будет в полном порядке.

Версия OS/2 2.0 будет готова лишь к концу года, но IBM уже начала ее де-

монстрацию, с тем чтобы ее окончательно не забыли.

И несмотря на то что все мы уже устали от бесконечных обещаний IBM, демонстрация версии 2.0 показала, что это именно то, о чем мечтает каждый, кто пытается сейчас заставить свой компьютер работать в мультипрограммном режиме, и особенно те, кто на протяжении прошлого года мучился с Windows.

Основные характеристики версии 2.0 уже неоднократно обнародованы. Эта ОС будет работать в 32-разрядной не сегментированной памяти, она будет обеспечивать практически неограниченное число сеансов DOS и поддерживать устанавливаемые файловые системы. Высокопроизводительная файловая система, поставляемая в комплекте с OS/2 2.0, будет иметь более совершенные средства размещения файлов, способные воспринимать длинные имена файлов.

Одно из немногих изменений в конструкции OS/2 заключается в том, что начиная с этого года ее разработчики бьются над тем, чтобы оболочка Windows могла работать в среде этой ОС без какой-либо потери производительности.

В процессе демонстрации было проведено сравнение производительности НИС PageMaker при работе под Windows 3.0 и в среде OS/2 2.0 на компьютерах одной и той же модели; она оказалась абсолютно одинаковой.

А когда в тех же условиях сравнили приложения DOS, то в среде OS/2 2.0 они работали значительно быстрее.

Одним из главных преимуществ OS/2 2.0 является наличие версии DOS, написанной под 32-разрядную модель; все приложения работают в своем собственном пространстве памяти и не могут вывести из строя всю систему. Кроме того, поскольку OS/2 обеспечивает буферизацию печати, работу с расширенной памятью и сетевой доступ, высвобождается оперативная память, которая может быть использована приложениями DOS (см. табл.).

Фирма IBM считает, что ее операционная система — это то, что нужно пользователю уже сегодня. Эта система представляется как интегратор, т. е. средство, обеспечивающее более высокую производительность при работе приложений DOS и Windows.

Говоря о своей ОС, фирма открыто заявляет, что эта система будет обеспечивать именно такое мультипрограммирование и именно такое использование памяти, которые сегодня пытаются (и зачастую неудачно) реализовать системы QEMM, 386MAX и Headroom.

А поскольку стоимость OS/2 будет составлять всего 100 долл., можно надеяться, что будет продано не всего лишь 600 000 экземпляров системы, как на сегодняшний день, а зна-

Фирма IBM утверждает, что в среде OS/2 2.0 у DOS будет больше оперативной памяти, чем в среде Windows 3.0

	OS/2 2.0	Стандартная версия Windows 3.0	Расширенная версия Windows 3.0
Объем ОЗУ, выделяемый приложениям DOS (включая средства поддержки расширенной памяти и мыши)	633К	571К	569К
Объем ОЗУ, выделяемый приложениям DOS (включая средства поддержки ЛВС)	633К	491К	489К
Выделяемая расширенная память (EMS 4.0)	32МВ	Нет	16МВ

В среде OS/2 2.0 приложениям DOS автоматически предоставляется поддержка ЛВС и мыши, в результате чего при работе с DOS значительно экономится оперативная память. Показанные справа оценки были получены для DOS версии 5.0: в случае же DOS 3.3 и 4.0 объемы ОЗУ, выделяемые приложениям при работе в среде Windows будут значительно меньше.

чительно больше.

Новая версия OS/2 значительно отличается от той, которая была разработана в 1987 г., — настолько, что IBM даже подумывает о другом названии. Заказчиков даже просят присылать предложения на этот счет, пока продукт находится в стадии разработки.

Судья отменяет свое решение; авторские права фирмы Ashton-Tate на СУБД dBase подтверждены

С фирмы Ashton-Tate полностью снято обвинение в "незаконном" присвоении права на СУБД dBase.

После того как все пользователи СУБД dBase были повержены в шок решением федерального судьи о том, что авторские права фирмы Ashton-Tate на эту СУБД не имеют законной силы, судья отменил свое решение.

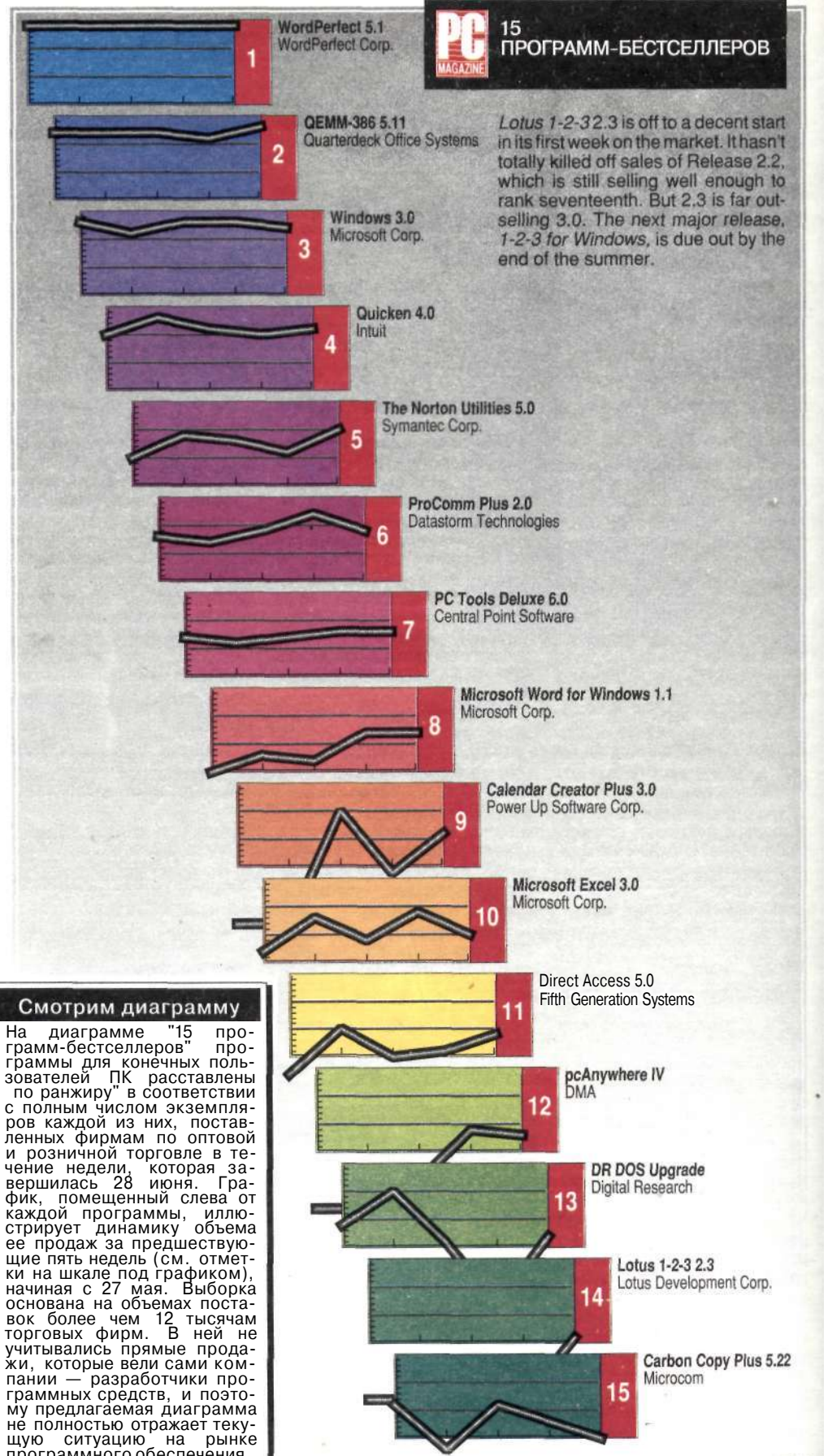
Решение, а затем его отмена были объявлены в окружном суде Лос-Анджелеса, где слушалось дело о нарушении фирмой Fox Software авторских прав фирмы Ashton-Tate на СУБД dBase.

Судья отменил свое решение после того, как Управление регистрации авторских прав США передало в суд официальное уведомление о том, что авторские права на это программное изделие фирме Ashton-Tate выданы в установленном порядке.

Компания AT&T продает часть акций своего филиала Unix Lab, одним из совладельцев которого становится фирма Novell

Пытаясь уже давно доказать мировому сообществу, что операционная система UNIX — это не исключительно ее собственность, компания AT&T продала более 20% акций своего филиала Unix Systems Laboratory, занимающегося разработкой многопользовательских операционных систем.

Предоставившейся тем самым возможностью про-



Смотрим диаграмму

На диаграмме "15 программ-бестселлеров" программы для конечных пользователей ПК расставлены по ранжиру" в соответствии с полным числом экземпляров каждой из них, поставленных фирмам по оптовой и розничной торговле в течение недели, которая завершилась 28 июня. График, помещенный слева от каждой программы, иллюстрирует динамику объема ее продаж за предшествующие пять недель (см. отметки на шкале под графиком), начиная с 27 мая. Выборка основана на объемах поставок более чем 12 тысячам торговых фирм. В ней не учитывались прямые продажи, которые вела сама компания — разработчики программных средств, и поэтому предлагаемая диаграмма не полностью отражает текущую ситуацию на рынке программного обеспечения.

Lotus 1-2-3 2.3 is off to a decent start in its first week on the market. It hasn't totally killed off sales of Release 2.2, which is still selling well enough to rank seventeenth. But 2.3 is far out-selling 3.0. The next major release, 1-2-3 for Windows, is due out by the end of the summer.

* Диаграмма "15 программ-бестселлеров" будет публиковаться в каждом номере PC Magazine/USSR.

никнуть в секреты разработки UNIX поспешили воспользоваться одиннадцать фирм. Большинство из теперешних акционеров Unix Lab, такие компании, как Motorola и Sun, явно заинтересованы в том, чтобы узнать все тонкости технологии UNIX.

Кроме того, среди этих одиннадцати есть одна фирма, которая и сама преуспела в создании своих собственных многопользовательских ОС, — это Novell.

Компания AT&T планирует сохранить по крайней мере 60% акций Unix Lab и не будет делать публичную распродажу. Все нынешние совладельцы Unix Lab не афишируют своего участия в этом предприятии.

Среди акционеров и такие известные компании, как NEC, Oki, Fujitsu, Amdahl и Toshiba.

(начало см. стр. 33)

Для манипуляций с пиктограммами можно использовать манипулятор "мышь" или функциональные клавиши и клавиши управления курсором. При этом работа с мышью идет по нетрадиционным правилам. В обычном случае нажатие одной клавиши манипулятора означает выбор, а двух — выполнение, тогда как в экранной контуре левая клавиша служит для выбора, а правая — для выполнения.

Система LANstep отличается особой простотой постановки и расширения. Для ее эксплуатации не требуется специалист по ЛВС, причем допускается расширение для одновременного обслуживания до 128 пользователей.

Разделение, или коллективное ("широковещательное" — в терминологии системы LANstep) использование прикладных программ, также не представляет никаких сложностей. Прикладные программы ставятся на сервер, а пользователь получает доступ к файлам и программам, указывая соответствующие параметры настройки. Пользователь запускает прикладные программы, выбирая их из списка либо указывая пиктограммы включенные в экранную "контур".

Предлагаемая ОС поддерживает все сетевые интерфейсные платы, соответствующие требованиям спецификации интерфейса сетевых драйверов — стандарту NDIS. В руководстве по эксплуатации системы содержатся указания для работы с такими сетями, как ARCnet, Ethernet и Token-Ring (TRN).

Если говорить о сравнительных показателях быстродействия, то система находится в своем классе где-то посередине. Испытания на быстродействие проводились с использованием схемных плат сети Ethernet, а также плат кольцевой сети с маркерным доступом (TRN). Система LANstep с сервером 386 продемонстрировала пропускную способность 0,679 кбит/с

Поправка к информации относительно планов фирмы WordStar приобрести текстовый процессор под Windows

Недавно в этой колонке новостей было сообщено о том, что фирма WordStar якобы купила у компании NBI пакет Legacy — текстовый процессор, предназначенный для работы в среде Windows. Эта информация, как выяснилось, не соответствует действительности.

Пакет куплен лишь на время, пока фирмой не завершена разработка своего собственного текстового редактора под Windows. Владельцем Legacy по-прежнему остается компания NBI.

Между тем, WordStar приобрела пакет LifeTree, который будет использоваться в качестве текстового процессора в изделиях этой фирмы.

— это несколько меньше, чем аналогичный показатель сети LANtastic на базе DOS (1,27 кбит/с), и значительно меньше пропускной способности, которую в аналогичной конфигурации имеет ОС Advanced NetWare (2,5 кбит/с).

И все же для таких повседневных сетевых работ, как выполнение прикладных программ при помощи сервера и вывод результатов на разделяемые принтеры, подобное быстродействие, по-видимому, вполне приемлемо.

Если говорить о стоимости, то предлагаемая ОС может успешно конкурировать с другими. Лицензия на 5-пользовательскую конфигурацию LANstep стоит 595 долл., что соответствует около 119 долл. в расчете на одну станцию. Поскольку расширение еще на пять станций обходится всего в 95 долл., то версию системы на 10 рабочих станций можно приобрести по цене 99 долл./станция.

Система ReadyLINK (см. рубрику "First Looks" в PC Magazine, February 26, 1991) в аналогичной конфигурации стоит несколько меньше 130 долл./станция, а программные средства LANtastic продаются так же по цене 99 долл./станция.

То что разработчики LANstep решили не ориентироваться на DOS и реализовали свой собственный оригинальный интерфейс с пользователем, имеет и достоинства, и недостатки. Предлагаемая ОС обеспечивает повышенную защиту информации, эффективное управление станциями и достаточно простое расширение сети. В то же время система предъявляет довольно высокие требования к аппаратным средствам. Поэтому в ситуации, когда недорогие ОС ЛВС на базе DOS становятся все более совершенными и популярными, фирме Hayes придется немало потрудиться, чтобы сформировать, так сказать, экологическую нишу для своей ОС.

Всесоюзная выставка с международным участием в рамках I съезда всесоюзного общества информатики и вычислительной техники

Информатика и вычислительная техника.

Ленинград, выставочный комплекс "Ленэкспо" в Гавани,
2 — 7 декабря 1991 г.

На выставке представлены разделы:

- Средства и системы связи и передачи информации.
- Аудиовизуальное оборудование и системы.
- Локальные и распределенные системы и сети ЭВМ.
- Периферийное оборудование и средства расширения ЭВМ.
- Системное и функциональное программное обеспечение.
- Общее и отраслевое прикладное программное обеспечение.
- Базы данных и знаний.
- Контторское и копировально-множительное оборудование.
- Обслуживание вычислительной техники, вспомогательные средства и принадлежности.

Прием заявок до 1 ноября.

Приглашаем также принять участие в формируемых выставках 1992 года:

Международная выставка под эгидой ЮНЕСКО "Автоматизированные информационные системы муниципального управления",
14 — 18 июля 1992 г.

Всесоюзная выставка с международным участием "Информатика и вычислительная техника - 92",
16 — 22 декабря 1992 г.

Подробная информация,
заявки на участие:
199106, Ленинград, а/я 806,
"Выставка".
Тел. (812) 550-78-57
(812) 356-35-56

IQ Electronics

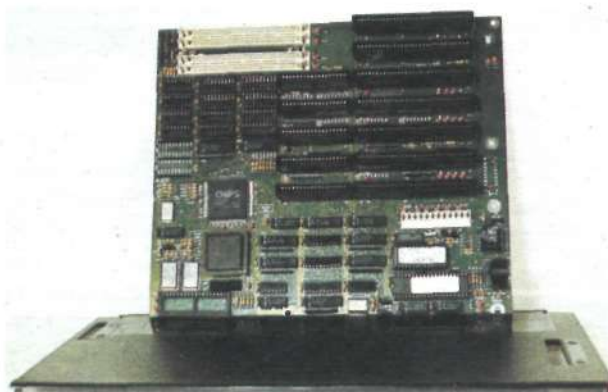
ПО НЕВЫСОКИМ ЦЕНАМ

ПРЕДЛАГАЕТ:



Базовую плату AT-286 12 или 16 МГц

Базовая плата для PC AT реализована на высокоинтегрированном однокристальном приборе 82C235 семейства SCAT компании Chips & Technologies Inc. (Сан-Хосе, шт. Калифорния)



Характеристика

- Центральный процессор Intel 80286, 12 или 16 МГц без состояний ожидания;
- Возможности организации ОЗУ:
- 1 Мбайт на ИС в корпусах типа DIP;
- 4 Мбайт на модулях типа SIMM
- Панель под математический сопроцессор Intel 80287
- Система BIOS AMI SCAT с встроенными утилитами, 64 Кбайт
- Теневая память
- Управление памятью до 16 Мбайт и поддержка LIM EMS 4.0

Условия поставки

- Цены CIF Москва
- Поставка в течение месяца
- Гарантия на два года и техническая поддержка в сервисном центре
- Замена при обнаружении дефектов в течение 48 часов
- Для малых партий поставка со склада в Москве



Компоненты для IBM-совместимых персональных компьютеров

- Базовые платы AT-286, AT-386SX, AT-386DX, AT-486
- Видеоконтроллеры и видеомониторы
- Контроллеры ввода-вывода
- Клавиатуры, корпуса и блоки питания
- Диски и дисковые контроллеры
- Сопроцессоры
- Сетевые платы

Krohstrasse 2, D-5000,
Köln, Germany

Tel. +(49)-221-516061

Fax. +(49)-221-529725

IQ Electronics
103460, Москва, К-460,

Тел. (095)536-14-00

Факс (095)536-14-00

Эндрю Шулман

Оболочка Windows 3.0 фирмы Microsoft может быть весьма удобной в эксплуатации — после того как кто-то поставил ее на машину и настроил. Если, однако, этот кто-то — вы сами, то это совсем другое дело, точнее целых три дела.

Вам как программисту потребуются ответы на следующие вопросы. Какие утилиты DOS можно вызывать из среды Windows? Смогут ли Windows работать с вашим сетевым принтером? Каким образом вызываемые одновременно утилиты DOS будут взаимодействовать с резидентными программами (TSR), которые были установлены до обращения к Windows? Какие прикладные программы расширенной DOS можно будет вызывать? Как будет Windows взаимодействовать с такими средствами управления памятью, как QEMM-386 и 386MAX?

Ответы на подобные вопросы обычно зависят от того, в каком режиме Windows вы работаете. Практически любое утверждение относительно возможностей системы Windows 3.0 имеет смысл только в том случае, если оно относится к конкретному режиму — реальному (Real), стандартному (Standard) или расширенному процессора 386 (386 Enhanced). Очень часто бывает так, что два специалиста, оживленно обсуждающие вопрос о том, каким образом оболочка Windows ведет себя в определенной ситуации, после часа дебатов выясняют, что говорят, по существу, о совершенно различных режимах этой системы.

В настоящей статье мы подробно рассмотрим все три режима подсистемы Windows, сосредоточив особое внимание на самом важном вопросе, интересующем пользователей и разработчиков, которых сковывает ограничение объема памяти (640 Кбайт), заложенное в операционной системе DOS: какую память можно реально получить в свое распоряжение для размещения программ? Чтобы проанализировать сравнительные достоинства и недостатки разных режимов Windows, мы воспользуемся рядом небольших тестовых программ на языке Си, обеспечивающих выделение максимального объема памяти для размещения программ в вашей машине. Эти тест-программы можно либо вызвать через сеть PC MagNet, либо скомпилировать из исходного текста, приводимого в сопровождающую данную статью листингах.

Поможет ли Windows преодолеть 640-Кбайт барьер системы MS-DOS? Как мы увидим, ответ на этот вопрос определяется не только тем, в каком

Windows 3.0: хорошая память и отличная работоспособность



режиме работает Windows, но и зависит от того, идет ли речь о Windows-программе или о программе, ориентированной на DOS, а также (в последнем случае) от типа DOS-программы. Будет показано, что фактически в оболочке Windows предусмотрено даже более трех режимов!

Мы увидим также, каким образом реальный, стандартный и расширенный режимы подсистемы Windows 3.0 связаны с так называемым реальным 16-разрядным защищенным, реальным 32-разрядным защищенным и виртуальным 86 (V86) режимами микропроцессоров компании Intel. А к моменту, когда мы завершим рассмотрение этих вопросов, станет ясно, что при всем кажущемся хаосе режимов оболочки Windows, они все же подчиняются определенным закономерностям.

ФАКТЫ КАК ОНИ ЕСТЬ

Голые факты выявить совсем несложно. Реальный режим подсистемы Windows 3.0 требует, чтобы персональный компьютер содержал центральный процессор 8086 или 8088 (или более мощный) фирмы Intel (или совместимый с ним) и 640 Кбайт обычной памяти. Стандартный режим требует микропроцессора 80286 (или более мощного) и минимум 1 Мбайт оперативной памяти (640-Кбайт обычная память плюс 384-Кбайт расширение). А для расширенного режима требуется процессор 80386 или 80486 и минимум 2 Мбайт памяти.

Когда из командной строки DOS вы введете WIN, подсистема Windows автоматически попытается выбрать "максимальный" режим, возможный для вашей конкретной конфигурации аппаратных средств. К сожалению, единственный способ определить, какой именно режим выбран, состоит в том, чтобы при помощи меню Help менеджера программ (PM) обратиться

к диалоговому окну About. (Программисты не преминут заметить, что в оболочке Windows предусмотрена функция GetWinFlagsO, которая позволяет определять текущий режим из работающей под управлением Windows программы.) Вы можете попытаться выбрать режим для Windows "силой", используя при вызове из DOS команды WIN/R, WIN/S или WIN/3.

Поскольку в стандартном режиме используются некоторые особенности процессора Intel 80286, этот режим предоставляет дополнительные удобства, отсутствующие в реальном режиме. Аналогичным образом, в расширенном режиме используются некоторые характеристики, присущие процессору Intel 80386, и благодаря этому предоставляются дополнительные возможности по отношению к стандартному или реальному режимам. Архитектурные характеристики каждого из режимов подсистемы Windows сведены в таблице на рис. 1.

Смысловое значение и практические достоинства всех этих характеристик мы поясним ниже. Однако сразу укажем главное:

- реальный режим полезен только для выполнения старых прикладных программ Windows 2.x, которые не были перетранслированы для версии Windows 3.0. Лично я считаю, что реальный режим Windows 3.0 скорее всего через некоторое время вообще отомрет;
- стандартный режим рассчитан на выполнение Windows-программ на машинах с микропроцессором 286, и в этом режиме программы Windows 3.0 могут преодолеть 640-Кбайт барьер DOS. К сожалению, стандартный режим не обеспечивает такой же возможности работы с основной памятью большей емкости для DOS-программ. Стандартный (и реальный) режим, однако, предусматривает для DOS-программ. Стандартный (и реальный) режим, однако, предусматривает для DOS-программ возможность переключения с задачи на задачу;
- расширенный режим обладает многими достоинствами с точки зрения выполнения программ и Windows, и DOS. Для этого режима графический интерфейс



СОВМЕСТНОЕ ПРЕДПРИЯТИЕ "УНИТЕХ"

Московский государственный университет
им. М. В. Ломоносова
OWT GmbH



Производит подбор по желанию заказчика и продажу по низким ценам компьютеров и оргтехники лучших зарубежных фирм.

Поставляет оборудование и программное обеспечение для локальных сетей ПЭВМ, являясь официальным дистрибутором фирмы NOVELL.

Осуществляет гарантийное и сервисное обслуживание вычислительной техники, компьютерные услуги, консультации.

Продает готовые программные средства для персональных ЭВМ IBM/PC, адаптирует по требованиям заказчика и обучает работе с ними.

119899, Москва,
Ленинские горы, МГУ, НИВЦ

Телефон:
(095) 939-5069/1619/2344

Телефакс:
(095) 938-2136

VCPI обеспечивает преемственность пакетов прикладных программ

Эндрю Шулман

Сегодня, когда компания Microsoft обеспечила возможность выполнения программ защищенного режима под управлением DOS, значение некогда игравших ключевую роль Спецификации дополнительной отображаемой памяти EMS и Спецификации расширенной памяти XMS должно уменьшиться. С появлением средств поддержки защищенного режима следует ожидать, что важнейшие программы для персональных компьютеров будут, вероятнее всего, либо выполняться в защищенном режиме под управлением Windows, либо использовать тот или иной расширитель DOS. Если в свое время спецификации EMS и XMS были необходимы как обходные решения проблемы увеличения объема памяти, то в дальнейшем они фактически окажутся ненужными.

Но что же делать с множеством существующих пакетов прикладных программ ПК, требующих EMS или XMS? Для системы Windows здесь нет серьезной проблемы, поскольку предусмотренные в ней реальный и стандартный режимы совместимы с имеющимися средствами управления памятью. Только расширенный режим подсистемы Windows не совместим с многими существующими средствами управления памятью. Однако, с одной стороны, имеющиеся модернизированные менеджеры памяти (например, QEMM-386 5.11 фирмы Quarterdeck Office Systems), совместимые с расширенным режимом, с другой стороны, расширенный режим предусматривает собственный встроенный менеджер дополнительной памяти. В расширенном режиме емкость как дополнительно отображаемой, так и расширенной памяти может задаваться при помощи редактора файлов PIF (PIF Editor) индивидуально для каждой прикладной программы.

Тем не менее есть и другие проблемы. Например, у пользователей оболочки Windows постоянно возникают проблемы совместимости, когда они пытаются выполнять прикладные программы расширенной DOS в расширенном режиме.

Главным средством для разрешения этой проблемы является VCPI — виртуальный интерфейс управляющей программы, разработанный компаниями Quarterdeck и Phar Lap Software. Поскольку расширители DOS защищенного режима строятся в предположении, что программа будет запускаться в реальном режиме, необходимо какое-то специальное средство, позволяющее расширителям DOS работать, когда машина находится в режиме Virtual 86 (V86).

До того как интерфейс DPMI был включен в состав пакета Windows 3.0, таким специальным средством был интерфейс VCPI. Это, по существу, набор

функций, который служит для расширения применяемой в EMS команды прерывания INT 67h. Интерфейс DPMI во многих отношениях представляет собой существенный шаг вперед по сравнению с VCPI, однако следует учитывать, что существует огромное количество прикладных программ, которым для работы в режиме V86 требуется VCPI; такие программы ни в коем случае нельзя терять, пока разработчики и пользователи не перейдут на DPMI (если это произойдет). Поставщикам гораздо проще реализовать VCPI, чем DPMI, поэтому уже сейчас имеется немало прикладных программ, совместимых с VCPI. В расширенном режиме DOS-программы запускаются в режиме V86, но интерфейс с VCPI не предусматривается.

Когда прикладные программы будут переработаны с учетом соглашений DPMI, все проблемы "VCPI - Windows" исчезнут как дым

Если вы когда-либо пытались запустить систему АвтоКАД/386 в расширенном режиме, то вы уже сталкивались с подобной проблемой. Эта система была разработана с использованием расширителя 386:DOS-Extender фирмы Phar Lap, в котором пока еще не реализована поддержка DPMI. Как только АвтоКАД/386 или какая-либо другая программа пытается проверить, имеется ли VCPI (вызывая прерывание INT 67h AX=DE00h), средства расширенного режима системы Windows прерывают ее выполнение с выдачей сообщения следующего содержания:

Ошибка: вы попытались выполнить прикладную программу защищенного режима в расширенном 386 режиме. Чтобы выполнить эту прикладную программу, нужно выйти и запустить систему Windows при помощи команды WIN/S или WIN/R.

Это препятствие можно обойти, если указать VCPI-Warning=false в секции [386enh] файла SYSTEM.INI. Однако программа-нарушитель выведет все же свое собственное сообщение об ошибке, которое в переводе гласит:

"Микропроцессор 386 в настоящий момент работает в виртуальном режиме 8086 под управлением другой программы. Вам необходимо отключить эту другую программу, чтобы использовать расширитель 386:DOS-Extender для работы в защищенном режиме".

К сожалению, данное указание спорно. Прикладные DOS-программы

защищенного режима, не совместимые с DPMI, необходимо выполнять либо в реальном, либо в стандартном режиме. Единственный способ обойти эту проблему заключается в том, чтобы кто-нибудь написал драйвер виртуального устройства системы Windows, поддерживающей расширения VCPI для спецификации EMS.

Существует и еще одна проблема. Для стандартного и расширенного режимов Windows требуется драйвер XMS — обычно это HIMEM.SYS. Система Windows не будет запускаться в стандартном или расширенном режиме, если подобного драйвера нет. Однако, точно так же как многие прикладные программы расширенной DOS пока еще не совместимы с DPMI, немало программ не совместимо и с XMS. Расширитель 386:DOS-Extender 3.0 поддерживает спецификацию XMS, однако может потребоваться определенное время, прежде чем такие прикладные программы, как АвтоКАД/386, окажутся переработанными с ориентацией на эту новую версию. (Вы можете определить, какую версию расширителя 386:DOS-Extender использует прикладная программа, если перед ее запуском введете команду SET DOSX= -BANNER.)

В результате, даже если вы не работаете в среде Windows, то, когда у вас установлен драйвер HIMEM.SYS, многие прикладные программы DOS защищенного режима не будут иметь достаточного количества расширенной памяти для своей работы. Драйвер HIMEM.SYS охватывает всю расширенную память, а подобные прикладные программы "не знают", каким образом использовать XMS.

Эта проблема имеет по крайней мере частичное решение. Вместо HIMEM.SYS можно применять усовершенствованные версии таких менеджеров памяти, как QEMM-386 5.11. Этот менеджер памяти организует использование блоков одного и того же общего пространства памяти как для XMS, так и для EMS. Таким образом, QEMM-386 5.11 может обеспечить предоставление памяти EMS, соответствующей требованиям VCPI, когда это требуется системе АвтоКАД/386, и памяти XMS, если это необходимо системе Windows 3.0.

После того как прикладные программы DOS защищенного режима и менеджеры памяти будут модернизированы с учетом требований DPMI (в качестве двух первых примеров здесь можно назвать пакет Lotus 1-2-3 версии 3.1 и расширитель 286:DOS-Extender фирмы Phar Lap) или когда кто-либо реализует расширенные возможности поддержки VCPI в системе Windows, все указанные проблемы исчезнут.



Подобные возможности мульти-программирования распространяются на все три режима Windows. В реальном и стандартном режимах число копий программы MEM, которые вы можете запустить одновременно, ограничивается только емкостью имеющейся дисковой памяти. В этих режимах подсистема Windows ведет себя по отношению к DOS-программам как переключатель задач (в действительности по своим функциям она очень похожа на оболочку DOS Shell будущей версии DOS 5.0). Когда вы выходите из DOS-программы — например, по команде Ctrl-Esc, — система Windows сохраняет ее в файле подкачки — WOAxxxx.TMP. (Аббревиатура WOA означает WINOLDAP — старая прикладная программа системы Windows; DOS-программы, не ориентированные на систему Windows, считаются старыми прикладными программами.)

В расширенном режиме система Windows не просто осуществляет для DOS-программ переключение задач, а организует многозадачный режим их выполнения. Программы не выводятся на диск в целом. Вместо этого обмен частями программы с диском осуществляется страницами по 4 Кбайт, как это предусматривается управлением виртуальной памятью для микропроцессора 386. Такой более сложный способ организации виртуальной памяти означает, что хотя расширенный режим предусматривает возможность выполнять несколько DOS-программ одновременно (что эффективно, в частности, при фоновом выполнении коммуникационных программ), он фактически позволяет осуществлять одновременный запуск меньшего числа DOS-программ, чем реальный или стандартный режим. Увы, за все приходится расплачиваться...

Подведем итоги: программа MEM позволяет четко продемонстрировать 640-Кбайт барьер, свойственный MS-DOS. Когда выполняется эта программа, она получает менее 640 Кбайт памяти. Реальный и стандартный режимы системы Windows обеспечивают переключенные задачи, а расширенный режим — многозадачность, но, хотя в этом режиме могут работать несколько экземпляров одной и той же программы одновременно, на каждый экземпляр распространяется предел 640 Кбайт. Тот факт, что программе MEM удается в лучшем случае получить всего 560 Кбайт в компьютере с 2-Мбайт основной памятью, может вызвать настоящий шок, однако сегодня большинство пользователей ПК воспринимают эту довольно-таки неестественную ситуацию как некую неизбежность.

А теперь мы покажем, что это вовсе не так.

ПРЕОДОЛЕНИЕ 640-КБАЙТ БАРЬЕРА

Следующая программа, WINMEM, выполняет практически те же функции, что и MEM, но является, в отличие от нее, не малой программой DOS, а малой программой Windows для захвата всей имеющейся памяти. Фотографии экрана дисплея, показанные на рис. 4-6, позволяют более полно представить себе общую картину, и позднее мы обратимся к одной важной детали, которую они содержат. Пока, однако, отметим, что программа, выполняемая на одной и той же машине с микропроцессором 80386 и 2-Мбайт памятью, получает в реальном, стандартном и расширенном режимах абсолютно разное число байтов памяти:

Режим	Число байтов
WIN/R	229376
WIN/S	1245184
WIN/3	3080192

В реальном режиме программе WINMEM удастся получить даже меньше памяти, чем программе MEM системы DOS. Таким образом, если говорить об управлении памятью, реальный режим Windows ничего нового пользователю не дает.

В стандартном режиме программа WINMEM "захватывает" такое число байтов памяти, которое мы — впервые за время наших экспериментов — можем признать вполне осмысленным. Наша начальная емкость памяти составляет 2 Мбайт, а программе WINMEM выделяется 1,3 Мбайт; очевидно, что какое-то место занимает сама эта программа. Как бы то ни было, барьер 640 Кбайт взят.

Наконец, в расширенном режиме... Минуточку! Если в нашей машине всего 2 Мбайт памяти, то каким же образом программе WINMEM удалось занять 3 Мбайт? Если вы еще раз взгляните на список архитектурных характеристик, приведенных на рис. 1, то вы заметите, что в расширенном режиме "предусматривается организация виртуальной памяти". Теперь вы понимаете, что это значит. В расширенном режиме подсистема Windows может использовать ваш жесткий диск как расширение оперативной памяти. А поскольку мы используем ту же копию WINMEM.EXE и в реальном, и в стандартном режимах, очевидно, что эта виртуальная память выделяется без всяких "ухищрений" со стороны программы. (Примечание: если вы выполняете программу WINMEM в расширенном режиме, а она не получает большего объема памяти, чем наличная оперативная, удалите некоторые файлы с жесткого диска и запустите программу еще раз.)

Теперь мы можем понять, почему оболочка Windows 3.0 компании Microsoft пользуется таким успехом. Это обусловлено не столько тем, что она по своим внешним признакам выглядит гораздо лучше, чем Windows 1.x и 2.x (что также способствовало ее успеху), сколько тем, что в стандартном и расширенном режимах она предоставляет пользователю значительно больше памяти и, следовательно, обеспечивает гораздо более высокую производительность. Благодаря этому пользователи ПК впервые получают возможность работать с подсистемой Windows как с практическим эффективным инструментальным комплексом, а не просто относиться к ней, как к некоему любопытному эксперименту.

Программисты, возможно, усмехнулись, когда я охарактеризовывал WINMEM как небольшую Windows-программу. Вы, конечно, слышали, что "в среде Windows, чтобы только напечатать "Hello world" ("Здравствуй, мир!)", требуется программа длиной приблизительно 100 строк". Однако это не совсем так. Для простых экспериментов, малых утилит и всегда популярного примера печати "hello world" программист может воспользоваться функцией Message Box () оболочки Windows вместо того, чтобы идти по пути применения полной процедуры системы Windows, предусматривающей регистрацию класса окна, создание окна и обработку сообщений.

Функция MessageBoxO позволяет работать с информацией объемом до полного экрана, что делает ее идеальной для составления простых утилит Windows и проведения экспериментов типа нашего примера программы WINMEM. Используя MessageBoxO, мы даже можем создавать какую-либо функцию типа стандартной функции печати printf () языка Си.

Программа WINMEM занимает всего 75 строк, но отличается ненасытным аппетитом в отношении памяти. В ней в цикле вызываются встроенные функции Windows GlobalAlocO и GlobalLockO. Цикл завершается только после того, как эти функции перестанут выполняться, т.е. когда память окажется исчерпанной. При выходе из цикла программа печатает число выделенных байтов и выдает некоторую другую информацию (и "дезинформацию" — такую, как собственное число имеющихся байтов GetFreeSpaceO подсистемы Windows). Исходный текст программы WINMEM.C с инструкциями по компиляции приведен на рис. 3.

Программисты отметят также, что сама программа WINMEM.EXE занимает всего 5 Кбайт — меньше, чем соответствующая программа MEM.EXE системы DOS. Што же время она обладает более широкими возможностями, и (естественно) ее выходные характеристики в гораздо большей степени удовлетворяют

ПОЛИГЛОТ

пользователя. Это объясняется тем, что в подсистеме Windows предусмотрена динамическая компоновка связей, и исполнимые коды функций, используемых в программе WINMEM, — таких как Message Box (), GlobalAlloc и GetFreeSpaceO, — не копируются в каждую программу, где эти функции используются. Указанные функции не входят и в состав известного Комплекта средств разработки программ (SDK), поставляемого для работы с Windows. При этом в каждом экземпляре оболочки Windows, которую покупатель может приобрести в готовом виде у местной торговой компании, содержится полный интерфейс прикладных программ (API) Windows. Даже функция wvsprintfO, используемая в программе WINMEM, уже есть в среде Windows. Программа типа WINMEM содержит динамические обращения к подпрограммам, но не имеет их в своем составе. По

этой причине сам загрузочный файл довольно мал.

РЕЖИМЫ МИКРОПРОЦЕССОРОВ INTEL

Программа WINMEM демонстрирует достаточно высокие результаты в стандартном режиме и просто отличные — в расширенном (хотя весьма посредственные в реальном режиме), потому что фактически стандартный и расширенный режимы позволяют программам системы Windows использовать еще один режим, а именно защищенный режим микропроцессоров 80286, 80386 и 80486.

Если можно сказать, что у самой оболочки Windows слишком много режимов, то семейство микропроцессоров Intel 80x86 ей в этом отношении не уступает. Микропроцессоры 80286, 80386 и 80486 имеют так на-

зываемый режим совместимости, когда эти усовершенствованные центральные процессоры, по существу, эмулируют прежние процессоры 8088, на основе которых строились первые персональные компьютеры корпорации IBM — PC и PC/XT. Поскольку адреса, используемые в программе для работы с ее собственным кодом и данными в таком режиме совместимости, прямо соответствуют истинным адресам, выдаваемым на системную шину, рассматриваемый режим и называется реальным.

В настоящее время большинство машин с системой DOS (даже с самыми мощными микропроцессорами 80486) обычно работает в этом "старомодном" режиме совместимости с 8088. Машины в таком режиме представляют собой, по сути, компьютеры XT с повышенным быстродействием. Поскольку микропроцессор 8088 рассчитан на адресное пространство всего в 1 Мбайт, процессор 80286, 80386

WINMEM.C	ПОЛНЫЙ ЛИСТИНГ
<pre>WINMEM.C - Сколько памяти можем мы получить под управлением Windows? cl -o -AS -Oais -Zpe winmem.c link /align: 16 winmem,, /nod:slibcew libw.win.def rcwinmem.exe ; WIN.DEF - общий файл определений Windows EXETYPE WINDOWS STUB 'WINSTUB.EXE' CODE PRELOAD MOVEABLE DISCARDABLE DATA PRELOAD MOVEABLE MULTIPLE HEAPSIZE 10240 STACKSIZE 5120 Copyright (c) 1991 Ziff Communications Co. PC Magazine * Andrew Schulman V #include<stdlib.h> #include<stdarg.h> #include<string> #include <ctype.h> #include <time.h> #include<dos.h> #include "windows.h" /* printf в информационном окне Windows */ int megprintf(char *fmt, ...) { static char s[1024]; int len; va1st marker; va_start(marker, fmt); len = wvsprintf(s, fmt, marker); va_end(marker); MessageBox(NULL, s, "Тест PC Magazine для распределения памяти в системе Windows", MB_OK); return len; } /* Выдача имени текущего режима Windows */ char far *win_mode(void) { unsigned long flags = GetWinFlagsO; return (flags & WF_ENHANCED ? "расширенный" flags & WF_STANDARD ? "стандартный": /* по умолчанию */ "реальный"); int PASCAL WinMainfHANDLEhInstance, HANDLEhPrevInstance, LPSTR lpszCmdLine, int nCmdShow) { unsigned long blksize = 32L < 10L; /* 32K на блок */ unsigned long freespace, largest, allocated, save; time t1, t2; char far *fp; /* эти параметры выбираются до начала выделения памяти */ freespace = GetFreeSpace(O); largest = GlobalCompact(-1); /* захват максимально возможного объема памяти */ time(&t1); allocated = 0; while (fp = GlobalLock(globalAlloc(GMEN MOVEABLE, blksize))) { *fp = V; fp[blksize-1] = 'y'; allocated += blksize; save = fp; Yield(); /* предоставление возможности запуска другой программы V } time(&t2); /* вывод результатов */ msgprintf("Текущий режим %s\ \n%lu байтов имеется\ \n%lu байтов в самом большом блокеX \n% Занято %lu байтов за %lu секунд\ ЧпПримеруказателя: %04X:%04X\n", win_mode(), freespace, largest, allocated, t2 - t1, FP_SEG(save)<FP_OFF(save)); /* Занятая память будет автоматически освобождена при выходе из программы */ return O; } </pre>	

Рис. 3: WINMEM. С демонстрирует, что программам подсистемы Windows может выделяться память самого различного объема.

или 80486, работающий в реальном режиме, также способен адресовать только 1 Мбайт — независимо от того, какой объем имеет оперативная память машины. Реальный режим DOS-машины "замечателен" тем, что фактически игнорирует всю добавочную память, которую вы устанавливаете в компьютер.

В действительности, DOS резервирует определенные диапазоны адресов в рамках 1-Мбайт адресного пространства для использования в качестве памяти видеоизображения и для других важных функций. Поэтому программа, работающая в реальном режиме DOS, фактически получает в свое распоряжение пространство менее 1 Мбайт, выделяемое для прикладных программ, а, если говорить точнее, то максимум 640 Кбайт. Теперь мы понимаем, кто в этом виноват — 640-Кбайт барьер обуславливается реальным режимом DOS.

Истинно собственные режимы ЦП 80286 (и более мощных процессоров Intel) называются защищенными. В этих режимах процессоры ведут себя не так, как быстродействующие ЦП 8088, а в программах не используются реальные (абсолютные) адреса. В защищенном режиме программы обращаются к памяти опосредованно, через таблицы дескрипторов, работу с которыми осуществляет такая операционная среда защищенного режима, как Windows.

Эта подсистема является "защищенной" в том смысле, что процессор контролирует обращение к памяти со стороны программы при помощи таблиц дескрипторов. Хотя доступ к памяти в данном случае оказывается "менее прямым", чем в реальном режиме, он осуществляется в основном "прозрачно", незаметным для программ образом, причем позволяет выделять гораздо больший объем памяти. Ограничение в 1 Мбайт, свойственное реальному режиму, и обусловленный этим 640-Кбайт барьер для памяти DOS просто исчезают.

Использование защищенного режима в стандартном и расширенном режимах — самое важное из достоинств, присущих только подсистеме Windows 3.0. То обстоятельство, что оболочка Windows служит расширителем DOS с защищенным режимом, разрешает наконец проблему использования DOS на машинах с микропроцессорами 80286, 80386 и 80486. Вычислительная мощность таких машин просто-напросто слишком велика для этой "старой" операционной системы; при отсутствии программной системы типа Windows 3.0, реализующей преимущества защищенного режима, значительная часть вычислительной мощности, по существу, пропадает впустую.

Однако, если уж защищенный режим является "естественным" режимом микропроцессоров 80286, 80386



и 80486, то почему тогда потребовалось целых три версии системы Windows, чтобы реализовать кажущиеся очевидными возможности?

Причина проста — необходимость обеспечения совместимости. Как только мы преобразовали программу для работы в защищенном режиме, мы сразу же "отсекли" возможность ее использования на компьютерах IBM PC и XT, а ведь в эксплуатации пока еще находится множество таких машин. Только совсем недавно компания Microsoft решила, что уже можно было бы отказаться от сохранения совместимости с микропроцессором 8088 в новых программах, ориентированных на Windows. Несмотря на это, Microsoft все же перестраховалась (что породило серьезные не-

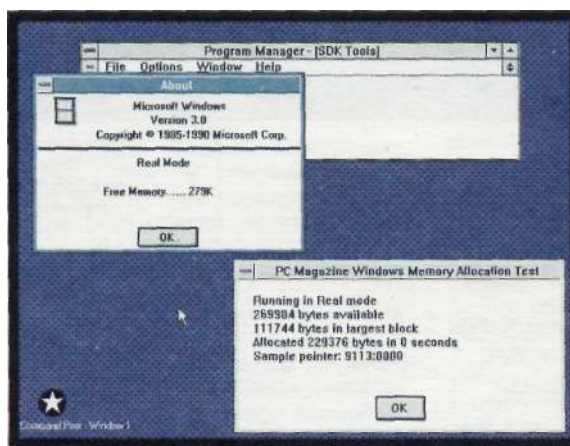


Рис. 4: Если WINMEM выполняется в реальном режиме Windows 3.0 (WIN/R), то объем выделяемой памяти оказывается даже меньше, чем для обычной прикладной программы DOS.

доразумения) и реализовала в подсистеме Windows режим, предусматривающий использование реального режима процессора.

Реальный режим оболочки Windows со временем, по-видимому, отомрет за ненадобностью. Если не считать этого атавизма, Windows-программы сейчас выполняются в защищенном режиме. Здесь, однако, необходимо говорить более конкретно, так как процессоры Intel имеют несколько защищенных режимов:

- 16-разрядный защищенный режим, впервые введенный в микропроцессоре 80286. Этот режим имеется также в микропроцессорах 80386 и 80486. Как и в слу-

чае реального режима, здесь 16 бит (2 байта) определяют максимальный размер сегмента — 216 байт, или 64 Кбайт. Но в отличие от реального режима программа, выполняемая в 16-разрядном защищенном режиме, гипотетически может выделять 16 384 таких сегмента, т.е. 1К Мбайт. В стандартном режиме Windows, программы этой подсистемы и сама она работают в 16-разрядном защищенном режиме.

- 32-разрядный защищенный режим, впервые введенный в микропроцессоре 80386. Этот режим имеется также в микропроцессоре 80486. Здесь в программах используются полные 32-бит (4-байт) регистры процессора. Число битов по-прежнему определяет максимальный размер сегмента, который теперь увеличивается до 232 байт, или 4 Гбайт. Поскольку это возможный размер одного сегмента, программам необходим только один сегмент — данное обстоятельство, по существу, равносильно тому, что программисту вообще не приходится беспокоиться о сегментации. Такую модель организации памяти иногда называют несегментированной (flat model).

В расширенном режиме отдельные программы подсистемы Windows могут работать в 32-разрядном защищенном режиме с использованием библиотеки динамических связей (DLL) WINMEM32. Однако, к сожалению, библиотека WINMEM32.DLL по-видимому, слишком сложна, чтобы ею могли пользоваться "простые смертные". В расширенном режиме элементы самой оболочки Windows выполняются в 32-разрядном защищенном режиме. Так называемые программы-драйверы виртуальных устройств (VxD), используемые в расширенном режиме, — это 32-разрядные программы несегментированной модели. (В отличие от других драйверов устройств, которые создаются главным образом для обеспечения аппаратной независимости систем, драйверы VxD служат для организации последовательного координированного доступа к устройствам от нескольких виртуальных машин. Благодаря подобным драйверам у каждой виртуальной машины "создается иллюзия", будто она обладает собственными клавиатурой, сопро-

цессором, контроллером прямого доступа к памяти и т.д.).

- Наконец, виртуальный 86 (V86) режим был также введен в микропроцессоре 80386 и сохраняется в микропроцессоре 80486. Режим V86 — это, по сути, 1-Мбайт защищенный режим. Программа, работающая в режиме V86, "считает", что она работает в реальном режиме, хотя в действительности ею "дирижирует" управляющая программа V86.

В расширенном режиме отдельные программы подсистемы Windows могут работать в 32-разрядном защищенном режиме с использованием библиотеки динамических связей (DLL) WINMEM32. Однако, к сожалению, библиотека WINMEM32.DLL, по-видимому, слишком сложна, чтобы ею могли пользоваться "простые смертные". В расширенном режиме элементы самой оболочки Windows выполняются в 32-разрядном защищенном режиме. Так называемые программы-драйверы виртуальных устройств (VxD), используемые в расширенном режиме, — это 32-разрядные программы несегментированной модели. (В отличие от других драйверов устройств, которые создаются главным образом для обеспечения аппаратной независимости систем, драйверы VxD служат для организации последовательного координированного доступа к устройствам от нескольких виртуальных машин. Благодаря подобным драйверам у каждой виртуальной машины "создается иллюзия", будто она обладает собственными клавиатурой, процессором, контроллером прямого доступа к памяти и т.д.)

В расширенном режиме DOS-программы фактически выполняются в режиме V86; оболочка Windows 3.0 в расширенном режиме — это управляющая программа V86 (так же как и программы-менеджеры дополнительной отображаемой памяти QEMM-386 и 386MAX). В расширенном режиме DOS-программы могут использовать интерфейс защищенного режима DOS (DPMI) для переключения в 16-разрядный или 32-разрядный защищенный режим; такую возможность мы подробно рассмотрим ниже.

К сожалению, DOS-программы не могут обращаться к DPMI в стандартном режиме. Они могут, однако, использовать любой расширитель DOS, даже не совместимый с DPMI, если только он совместим со спецификацией расширенной памяти XMS (чтобы настроить расширитель DOS на работу с расширенной памятью XMS, обычно необходимо бывает изменить параметры XMS в файле с расширением PIF подсистемы Windows). В качестве примеров здесь можно назвать расширители DOS фирм Phar

Lap, Rational Systems и Ergo Computing. Как это ни прискорбно, многие прикладные программы расширенной DOS не могут работать в расширенном режиме (см. врезку "VCPI обеспечивает преемственность пакетов прикладных программ").

ПРОГРАММА WINMEM

После конспективного описания режимов микропроцессоров Intel давайте вспомним, что мы все-таки программисты, и более подробно рассмотрим программу WINMEM и ее "поведение" во всех трех режимах работы Windows.

Возвращаясь к листингу (рис. 3), отметим, что в программе WINMEM используются приемы формирования указателей языка Си ("*fp" и "fp[]") с целью проверки того, имеется ли абсолютный указатель, содержащий непосредственно используемый адрес для обращения к истинной, прямо адресуемой памяти. В конце концов, мы должны выполнять одну и ту же программу WINMEM.EXE и в расширенном режиме, и в реальном режиме. Тот факт, что в расширенном режиме мы будем использовать виртуальную память (т.е. псевдооперативную память), расположенную на жестком диске), для программы должно оставаться незаметным. Во всех трех режимах адрес, выдаваемый функцией GlobalLockO оболочки Windows, — это всегда указатель в формате "сегментхмещение" микропроцессоров Intel, обеспечивающий операции чтения или записи.

Однако поле xxxx указателя xxxx:uuu в защищенном режиме имеет не то смысловое значение, что в реальном режиме. И именно здесь ключ к пониманию того, почему защищенный режим позволяет работать с большей памятью, чем реальный.

Программа WINMEM выводит не только число байтов памяти, которые ей удалось "занять", но и адрес "своего" последнего блока памяти. В реальном режиме программа WINMEM выводит адрес 9113:0000, а в стандартном и расширенном режимах — адреса типа 1065:0000 или 0DA5:0000.

Значение 9113 — адрес типичного параграфа сегмента для программы реального режима. Он соответствует абсолютному адресу ячейки памяти 9113h. В реальном режиме многие другие комбинации "сегментхмеще-

ние" (например, 9110:0030 или 9100:0130) указывают на ту же ячейку памяти. Чтобы перейти от указателя xxxx:uuu реального режима к абсолютному адресу ячейки памяти, процессор просто умножает xxxx на 16, а затем добавляет uuу. Существуют 216, или 65 536 (64K), различных значений поля xxxx; умножение этого максимального числа на 16 дает 1 048 576 — т.е. 1 Мбайт. Поскольку процессор в реальном режиме интерпретирует указатели именно таким образом, пользователь может работать только с 1 Мбайт памятью —* если даже в машине установлена память емкостью 2, 7 или 10 Мбайт.

Реальный режим имеет тот очевидный недостаток, что у машины появляется эффект "ограничения поля зрения", но в то же время и весьма серьезные достоинства — он делает исключительно простым прямой доступ к адресам памяти по абсолютным адресам. Если вам нужно обратиться к ячейке с адресом 400h, то вы можете использовать для этого ряд различных указателей, в том числе 0000:0400 и 0040:0000.

Рассмотрим теперь значения указателей для защищенного режима — 4065:0000. Напомним, что это было поле xxxx типичного указателя xxxx:uuу, выданное функцией GlobalLockO в стандартном и расширенном режимах работы Windows. Если бы это были адреса реального режима, они указывали бы на ячейки памяти с абсолютными адресами 10650h и 0DA50h, — слишком малые адреса для памяти, выделяемой в системе Windows! Когда программисты персональных компьютеров в первый раз

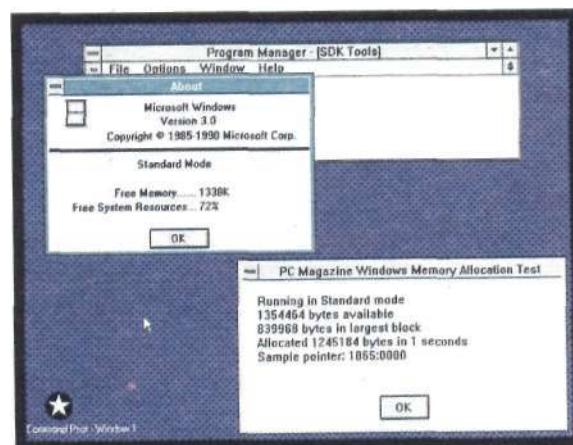


Рис. 5: Когда WINMEM выполняется в стандартном режиме Windows 3.0 (WIN/S), объем выделяемой памяти соответствует фактической 2-Мбайт емкости основной памяти, установленной в машине.



видят такие значения указателей, они обычно думают, что произошла какая-то ошибка — это не могут быть действительные адреса, возвращаемые функцией GlobalLockO или mallocO.

Очень эффективен защищенный режим! В этом режиме поле xxxx указателя xxxхгууу называется полем селектора, а не сегмента. Кроме того, оно не имеет никакого отношения к абсолютному адресу ячейки памяти. Чему же в данном случае соответствует адрес защищенного режима? Если в программе WINMEM выполняется операция *fp = 'x', а значение fp есть 0DA5:0000, то что это означает с точки зрения значения абсолютного, физического адреса ячейки памяти? И каким образом все это связать с тем фактом, что в случае, когда наша программа получает такие "интересные" указатели, она, по-видимому, приобретает способность захватывать гораздо больше памяти? В конечном итоге именно данное обстоятельство можно считать наиболее важным.

Селектор защищенного режима (поле xxxx адреса xxxх:уууу, для которого программа защищенного режима может выполнять операции записи/чтения и который может служить выполняемой командой) — это индекс для обращения к таблице дескрипторов. Дескриптор содержит базовый адрес, размер и признаки прав доступа для блока памяти. Каждый раз, когда программа использует один из сегментных регистров — CS, DS, ES или SS, — процессор на основе информации дескриптора определяет, разрешить или запретить фактический доступ к памяти.

Предыдущий абзац содержит приблизительно полдюжины упрощенных положений, однако вполне может играть роль элементарного описания защищенного режима процессоров семейства 80x86. Главный момент здесь заключается в том, что формула реального режима

абсолютный адрес ячейки памяти =
= (сегмент x 16) + смещение

заменяется, по существу, на следующую формулу:

абсолютный адрес ячейки памяти =
= таблица дескрипторов [селектор]. база + смещение

Этим объясняется, почему в защищенном режиме мы в принципе можем выделять гораздо больше памяти. Здесь нас ограничивает только число возможных дескрипторов, умножаемое на максимальный размер блока, представленного дескриптором, а жесткое правило определения адресов реального режима по формуле 64К x 16 не действует.

Это краткое описание объясняет также, почему наличие защищенного режима существенно облегчает жизнь

программиста, который пользуется функциями управления памятью Windows. В защищенном режиме базовый адрес для указателя 0DA5:0000 может иметь любое значение, которое содержится в дескрипторе 0DA5. Если подсистеме Windows необходимо выполнить перемещение этого сегмента по памяти (что, возможно, делалось, когда мы распределяли всю имеющуюся память), она может просто обновить дескриптор, в то время как программа WINMEM оставляет селектор 0DA5 неизменным. Благодаря защищенному режиму все реализуемые в оболочке Windows приемы управления памятью стали, наконец, "невидимыми" для программ. При этом флаг GMEM_MOVEABLE для функции GlobalAllocO и GlobalLockO в защищенном режиме практически теряет смысл.

Поскольку фактические форматы дескрипторов, таблиц дескрипторов и селекторов в 16- и 32-разрядном защищенном режиме широко известны — они опубликованы во многих других источниках, — я не буду их здесь приводить. Если читателя это интересует, он может обратиться к следующей литературе: Ray Duncan et al., Extending DOS, Reading, Mass.: Addison-Wesley, 1990; Rakesh K. Agarwal, 80x86 Architecture and Programming, Volume II, Architecture Reference, Englewood Cliffs, N.J.: Prentice Hall, 1991; Jeff Prosis, "Segmented Memory", PC Magazine, March 26, 1991, pp. 395-399.

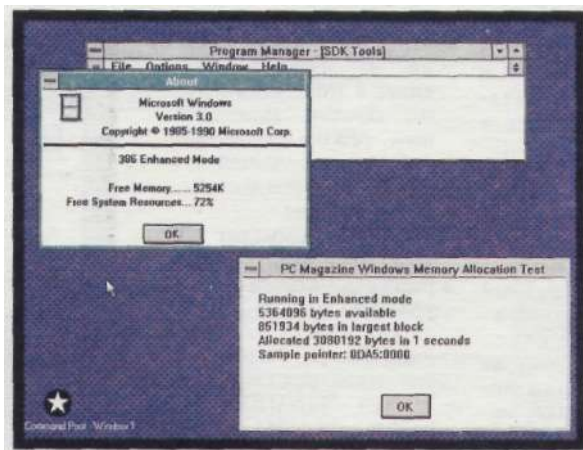


Рис.6. Когда WINMEM выполняется в расширенном режиме Windows 3.0 (WIN/3), она выделяет дополнительную виртуальную память, используя для этого жесткий диск компьютера.

ПРЕОДОЛЕНИЕ 640-КБАЙТ БАРЬЕРА ДЛЯ ПРИКЛАДНЫХ ПРОГРАММ СИСТЕМЫ DOS

Пример WINMEM показывает, почему при составлении программ важно ориентироваться на оболочку Windows 3.0: ее стандартный и расширенный режимы дают возможность использовать гораздо больше памяти. Многие программисты, которые не проявляют ни малейшего интереса к графическому интерфейсу пользователя подсистемы Windows, захотят, по-видимому, перенести свои программы в эту среду только для того, чтобы иметь возможность преодолеть 640-Кбайт барьер.

Но существует ли хоть какая-нибудь возможность предоставить подобное преимущество для DOS-программ, не ориентированных на Windows, но выполняемых в ее среде? Неужели действительно необходимо покупать оболочку Windows целиком только для того, чтобы получить больше памяти?

Нет, это необязательно. Несколько раз в настоящей статье мы упоминали DPMI — интерфейс защищенного режима DOS, который Рей Дункан описал в своей рубрике Power Programming в номерах PC Magazine от 12 февраля и 26 февраля 1991 г. Здесь я покажу, каким образом DOS-программа, работающая в среде Windows, может благодаря DPMI преодолеть 640-Кбайт ограничение, а в следующей статье рубрики "Маленькие хитрости" мы расскажем, каким образом программы Windows могут использовать DPMI для устранения некоторых упущений, характерных для интерфейса API системы Windows.

DPMIMEM.EXE — программа DOS, использующая сервер DPMI в расширенном режиме Windows для переключения процессора 80386 или 80486 в защищенный режим. Если программы Windows могут использовать DPMI либо в стандартном, либо в расширенном режиме, то DOS-программы (по крайней мере в среде Windows 3.0) — только в расширенном режиме. Когда DPMIMEM выполняется в реальном или стандартном режиме или же вне среды

Windows, она выдает сообщение об ошибке. А в расширенном режиме для программы DPMIMEM, которая начала свое существование как обычная DOS-программа, выдается примерно следующая информация:

DS=1321 CS=11D3 в реальном режиме

DPMIMEM.C

ПОЛНЫЙ ЛИСТИНГ

```

/*
DPMIMEM.C – Демонстрирует возможности DPMI и Windows 3.0 при
работе с расширителем DOS

Microsoft C: cl -AS dpmimem.c
Turbo C: tcc -ms dpmimem.c

Copyright (c) 1991 Ziff Communications Co.
PC Magazine * Andrew Schulman
*/
#ifdef _TURBOC_
#pragma inline
#define _asm asm
#define _dos_allocmem(x,y) (allocmem(x,y) == -1)
#else
#define MK_FP(seg, ofs) \
((void far *) (((unsigned long) (seg) << 16) | (ofs)))
#endif

#include <stdlib.h>
#include <stdarg.h>
#include <stdio.h>
#include <time.h>
#include <dos.h>
typedef enum { FALSE, TRUE } BOOL;

/* Вызов функции завершения процесса MS-DOS к кодом возврата
(INT 21h AH=4Ch) либо в реальном, либо в защищенном режиме */
void dos_exit(unsigned err)
{
    (asm mov al, err
    _asm mov ah, 4Ch
    _asm int 21h
    )
}

/* Вызов функции освобождения кванта времени Windows
(INT 2Fh AX=1680h) */
void win_yield(void)
{
    _asm mov ax, 1680h
    _asm int 2Fh
}

/* Вызов функции проверки установления расширенного режима
Windows (INT 2Fh AX=1600h) */
BOOL WIN_3enh(void)
{
    unsigned char vers;
    _asm mov ax, 1600h
    _asm int 2Fh
    _asm mov vers, al
    return (vers && ((vers == 0x00 || vers == 1 || vers == 0xFF)));
}

void fail(char *s) { puts(s); dos_exit(1); }
/* Вызов функции определения режима из DPMI (INT 2Fh AX=1686h),
чтобы проверить, находимся ли мы *уже* в защищенном режиме
под управлением DPMI */
BOOL dpmi_present(void)
{
    unsigned _ax;
    _asm mov ax, 1686h
    _asm int 2Fh
    _asm mov _ax, ax
    return (!_ax);
}

/* Вызов функции DPMI для обращения на точку входа переключателя
из реального в защищенный режим (INT 2Fh AX=1687h), чтобы оп-
ределить, имеется ли DPMI, и, если имеется, перейти в защи-
щенный режим путем вызова функции-переключателя */
BOOL dpmi_init(void)
{
    unsigned rds, rcs;
    void (far *dpmi)();
    unsigned hostdata_seg, hostdata_para, dpmi_flags;
    _asm mov ax, 1687h // проверка наличия DPMI
    _asm int 2Fh
    _asm and ax, ax
    _asm jnz nodpmi // если (AX=0), то DPMI имеется
    _asm mov dpmi_flags, bx
    _asm mov hostdata_para, si // параметры локальных данных DPMI
    _asm mov dpmi, di
    _asm mov dpmi+2, es
    // точка входа переключателя защищенного режима DPMI
    _asm jmp short gotdpmi
nodpmi:
    return FALSE;
gotdpmi:
    if (_dos_allocmem(hostdata_para, &hostdata_seg) != 0)
        fail("невозможно выделить память")

    dpmi_flags &= -1; /* это программа 16-разрядного защищенного
режима */

    _asm mov rds, ds
    _asm mov rcs, cs
    printf("DS=%04X CS=%04X в реальном режиме\n", rds, rcs);
    /* вход в защищенный режим */
    _asm mov ax, hostdata_seg
    _asm mov es, ax
    _asm mov ax, dpmi_flags
    (*dpmi)();
    /* теперь в защищенном режиме */
    _asm mov rds, ds
    _asm mov rcs, cs
    printf("DS=%04X CS=%04X в реальном режиме\n", rds, rcs);
    return TRUE;
}

/* Вызов функции выделения блоков памяти MS-DOS (INT 21h
AH=48h), в защищенном режиме */
void far *my_malloc(unsigned bytes)
{
    unsigned seg;
    return (_dos_allocmem(bytes >> 4, &seg) ? 0 : MK_FP(seg, 0));
}

#define SIZE 20480
main()
{
    unsigned long bytes = 0;
    char far *fp;
    time_t t1, t2;

    if (! dpmi_present())
        if (! dpmi_init())
            fail("Для этой программы требуется DPMI");

    /* теперь в защищенном режиме */
    if (! win_3enh())
        puts("Выполняется под управлением DPMI, но не в расширен-
ном режиме Windows 3.x");

    /* цикл распределения памяти */
    time(&t1);
    while (fp = my_malloc(SIZE))
    {
        *fp = 'x';
        fp[SIZE-1] = 'y';
        bytes += SIZE;
        win_yield();
    }
    time(&t2);
    printf("Выделено %lu байтов за %lu секунд\n", bytes, t2 - t1);
    printf("Чтобы освободить память, нажмите ENTER...");
    fflush(stdout);
    getchar();
    dos_exit(0);
}

```

Рис.7: DOS-программы могут преодолеть 640-Кбайт ограничение, если будут переключаться в защищенный режим и использовать сервер DPMI в рамках подсистемы Windows 3.0.



С первыми двумя строчками этой информации мы разберемся несколько позже. А сначала я хотел бы обратить ваше внимание на последнюю строку, обычная DOS-программа, скомпилированная при помощи системы Microsoft C или Turbo C, захватила память 1,2 Мбайт — в два с лишним раза больше, чем показывает программа MEM. Правда, для этого нам пришлось выполнять эту программу в расширенном режиме Windows, однако сама DPMIMEM не является Windows-программой.

Как и в случае наших предыдущих примеров программ MEM и WINMEM, память, охватываемая программой DPMIMEM, допускает прямой доступ. В отличие от памяти с переключением блоков, описанной в Спецификации дополнительной отображаемой памяти EMS, здесь вся выделяемая память в любой момент доступна программе, использующей DPMI. Аналогичным образом, в противоположность Спецификации расширенной памяти XMS, память, находящаяся за пределами 1 Мбайт, практически ничем не отличается от обычной выделяемой памяти: программе DPMIMEM не требуется куда-либо копировать содержимое этой памяти, чтобы использовать его. Основное и существенное различие между выделением памяти в защищенном режиме и применением таких известных "обходных" способов управления памятью, как EMS и XMS, заключается в том, что в защищенном режиме доступ ко всей памяти осуществляется на равных основаниях. Благодаря DPMI различия между обычной, расширяемой и дополнительной отображаемой памятью просто исчезают.

Как конкретно DOS-программа использует DPMI, чтобы преодолеть ограничение 640 Кбайт? Программистов особенно заинтересует полный листинг программы DPMIMEM.C, показанный на рис. 7. После компиляции средствами Microsoft C или Turbo C программа DPMIMEM может выполняться в среде Windows. Сервер DPMI в расширенном режиме Windows поддерживает прикладные DOS-программы как 16-, так и 32-разрядного защищенного режима. Программа DPMIMEM.C является 16-разрядной.

Работа программы DPMIMEM.C начинается с того, что вызывается функция `dpmi_init()`, которая определяет, доступны ли средства DPMI. Если они доступны, программа использует их для переключения в защищенный режим.

Рассмотрение функции `dpmi_init()` показывает, что программа, которая "хочет" использовать DPMI, начинается с прерывания INT 2Fh

AX=1687h. Если в системе имеется DPMI, эта функция выдает несколько строк информации, в том числе адрес функции для переключателя из реального в защищенный режим DPMI. Вызывая затем данную функцию, программа переключается в защищенный режим.

В этом есть нечто мистическое. В DPMIMEM мы воспроизводим значения регистров DS и CS до и после вызова функции переключения режима DPMI. Перед вызовом функции переключения режима программа DPMIMEM выполняется в реальном режиме, а сегментные регистры DS и CS содержат значения 1321h и HD3h. А когда функция переключения режима выдает информацию, оказывается, что содержимое регистров DS и CS внезапно изменилось: в этих регистрах теперь находятся значения сегментов защищенного режима 008Dh и 0095h. Это что же, содержимое сегментных регистров изменяется по ходу выполнения программы?!

Операции переключения режима в составленной на языке Си функции `dpmi_init()` выполняются при помощи встроенных ассемблерных кодов; это выглядит примерно следующим образом:

```
void(far *dpmi)();
asm mov ax,1687h
asm int 2Fh
//...
asm mov dpmi,di
asm mov dpmi+2,es
//...
(*dpmi)();
//Теперь мы находимся в защищенном режиме!
```

Когда программа оказывается в защищенном режиме, она может воспользоваться либо другими служебными средствами DPMI (к которым можно обратиться через прерывание 31h), либо средствами расширителя DOS. Расширитель DOS — программа, которая дает возможность применять в защищенном режиме прерывание DOS 21h и средства BIOS (прерывание 10h и т.д.). Одной из главных особенностей стандартного и расширенного режимов подсистемы Windows является то, что они построены с ориентацией на расширители DOS. В расширенном режиме такой расширитель DOS может использоваться также прикладными программами, не ориентированными на Windows.

После переключения в защищенный режим в программе DPMIMEM.C мы используем расширитель DOS оболочки Windows для распределения памяти. Для этого просто вызывается функция выделения блоков памяти MS-DOS (прерывание INT 21h AX=48h). Хотя данная функция никогда не предоставляет более 640

Кбайт для DOS реального режима, если расширитель DOS содержит функцию защищенного режима с таким же интерфейсом, то это обеспечивает возможность доступа ко многим мегабайтам памяти.

Поскольку расширитель DOS обеспечивает прерывание INT 21h AX=48h в защищенном режиме, мы можем воспользоваться либо функцией `_dos_alloctem0`, предусмотренной в Microsoft C, либо функцией `alloctem0` компилятора Turbo C, несмотря на то что в обоих случаях предполагается вызов соответствующих функций в реальном режиме. Это возможно потому, что главная цель расширителя DOS заключается в том, чтобы обеспечить "прозрачное" выполнение функций DOS и BIOS в защищенном режиме.

Важную роль играют и две другие функции, вызываемые в программе DPMIMEM.C. Еще перед вызовом `dpmi_init()` программа DPMIMEM вызывает функцию `dpmi_present()`. Эта функция, в свою очередь, обращается к прерыванию INT 2Fh AX=1686h, которое информирует нас о том, работает ли уже программа в защищенном режиме с интерфейсом DPMI. Например, функция `dpmi_present()` примет значение TRUE (истина), если ее вызвать из какой-либо программы Windows, выполняемой в стандартном или расширенном режиме. Оператор вызова здесь включен в программу DPMIMEM, поскольку она может запускаться как программа защищенного, а не реального режима. (Более подробно об этом будет рассказано ниже.)

В рамках цикла резервирования памяти программы DPMIMEM мы вызываем функцию `win_yield()`, которая, в свою очередь, вызывает прерывание INT 2Fh AX=1680h. Расширенный режим подсистемы Windows обеспечивает для DOS-программ многозадачное выполнение с управлением от планировщика, вызов этой функции предназначен для организации работы планировщика.

Все эти вызовы прерываний 2Fh являются частью программного интерфейса, который предусматривается в оболочке Windows для программ, не ориентированных на эту подсистему. Прерывания INT 2Fh AH=16h и AH=17h используются для обращения, например, к средствам DPMI, к буферу clipboard системы Windows, к многозадачному планировщику Windows и т.д. Это серьезная тема, требующая отдельного обсуждения, и о данных вопросах мы, возможно, будем говорить в рубрике "Маленькие хитрости" позднее.

ПРОБЛЕМЫ DPMIMEM

Не следует считать, что DOS-программы, использующие DPMI для преодоления свойственного DOS ограничения 640 Кбайт, делают это без всяких проблем; приведем

несколько важных замечаний по данному поводу.

Во-первых, не следует забывать, что по крайней мере в настоящее время программа DPMIMEM требует для своего выполнения расширенного режима Windows. В стандартном режиме доступ к DPMI предоставляется только программам Windows, а не DOS-программам. Дело в том, что стандартный режим предусматривает вызов прерываний INT 31h, которые может использовать программа, когда она работает в защищенном режиме, но не предусматривает прерывания INT 2Fh AX=1687h, необходимого для первоначального переключения в защищенный режим.

Программы Windows могут инициировать прерывания INT 31h и работать с расширителем DOS оболочки Windows, поскольку эти программы уже выполняются в защищенном режиме — им не требуется вызывать функцию переключения режима DPMI. Как указывалось выше, вызов прерывания INT 2Fh AX=1686h выполняется успешно, когда он происходит из программы Windows, работающей либо в стандартном, либо в расширенном режиме; это прерывание говорит о том, что программа уже работает в защищенном режиме. Однако DOS-программа не может переключиться в защищенный режим, поскольку для нее прерывание INT 2Fh AX=1687h не предусматривается.

Тем самым демонстрируется смешанная природа стандартного режима системы Windows. С одной стороны, программы Windows работают в защищенном режиме, что справедливо и для расширенного режима Windows. С другой стороны, так же как и в реальном режиме Windows, программы DOS в стандартном режиме не получают доступа к таким специальным служебным средствам, как DPMI. Таким образом, стандартный режим фактически представляет собой два различных режима: для DOS-программ он напоминает реальный режим, а для программ Windows — расширенный режим.

Кроме того, что программа DPMIMEM привязана к расширенному режиму, с ней возникает и ряд других проблем. Во-первых, эта программа обязательно должна компилироваться для "малой модели памяти" (Small). Поскольку вызов функции переключения режима DPMI — (*dmpi) () на рис. 7 — меняет содержимое сегментных регистров программы без какого бы то ни было уведомления, библиотечные функции, такие, как функция печати printfO, будут правильно работать только в том случае, если они "не полагаются" на сегментные регистры. Малая модель компилятора — единственное средство, при помощи которого этого можно добиться. Нарочно не придумаешь:

программу, предназначенную для работы с гигантскими объемами памяти, требуется компилировать как работающую с малой моделью памяти.

Если же программу DPMIMEM скомпилировать не средствами малой модели (cl -AS в Microsoft C или tcc -ms в Turbo C), то программа будет выработать признак нарушения защиты. Затем система Windows выдает свой "Приз за достижения в программировании" — на экране воспроизводится диалоговое окно с большим знаком Stop и сообщением "This application has violated system integrity and will be terminated" ("Данная прикладная про-

DPMI — это интерфейс исключительно низкого уровня, он совсем не похож на спецификации EMS или XMS, с которыми программисты и пользователи DOS хорошо знакомы.

грамма нарушила целостность системы и выполняться не будет").

Далее — и опять-таки по причине того, что сегментные регистры программы связаны с переключением режима по вызову (*dmpi) (), — мы не можем для отладки программы DPMIMEM воспользоваться пакетами CodeView фирмы Microsoft или Turbo Debugger фирмы Borland. Как только мы "пройдем" через вызов (*dmpi)(), отладчик зависнет. А поскольку мы работаем в расширенном режиме Windows, мы можем прекратить сеанс DOS по команде Settings/Terminate... и начать снова. Однако, хотя возможность прекратить сеанс DOS в случае ошибки или зависания — очень важное и удобное свойство режима V86, мы все же не можем воспользоваться стандартным отладчиком DOS при работе с DPMIMEM.

Мы указали еще не на все проблемы, возникающие с программой DPMIMEM; например, существует и такая проблема — в некоторых случаях команда Ctrl-C может инициировать выдачу сообщения о нарушении защиты.

Почему же программа DPMIMEM такая "сырая"? Почему существует так много ограничений на использование DPMI? Причина в том, что первоначально интерфейс DPMI предназначался не для прикладных программ типа DPMIMEM, а для таких расширителей DOS, как 386:DOS-Extender и 268:DOS-Extender фирмы Phar Lap, DOS/16M компании Rational Systems, OS/286 и OS/386 фирмы Ergo Computing, а

также расширителей DOS самой системы Windows. DPMI — это интерфейс исключительно низкого уровня, он практически ничем не похож в данном на смысле спецификации EMS или XMS, с которыми программисты и пользователи DOS хорошо знакомы. Комитет, который разрабатывал интерфейс DPMI, предполагал, что служебные средства таких "серверов" DPMI, как Windows, будут непосредственно использоваться только расширителями DOS, а функции более высокого уровня возьмут на себя прикладные программы.

Чтобы применять DPMI без всех этих ограничений, вам необходим расширитель DOS, предусматривающий загрузку DOS-программ защищенного режима. А DPMIMEM начинается как программа реального режима, а затем переключается в защищенный режим. Более рациональным решением проблемы могло бы стать создание программы, которая взяла бы на себя все функции по переключению режимов, ввела такой программный переключатель в защищенный режим, а уже затем загружать другие программы. При этом последние сразу начнут выполняться в защищенном режиме, что существенно упростит ситуацию.

Например, такие совместимые с DPMI расширители DOS, как 286:DOS-Extender фирмы Phar Lap или DOS/16M фирмы Rational Systems, дадут возможность программам DPMIMEM обращаться ко всей стандартной библиотеке Си при любой модели организации памяти, позволят выполнять отладку на уровне исходного кода и т.д. Чтобы подготовить программу DPMIMEM для работы с 286 DOS-Extender, следует использовать переключатель -Lp (связь защищенная) Microsoft C, который обычно применяется при построении прикладных программ, ориентированных на операционную систему OS/2. Затем вы вводите

```
шп286 dpmimem
```

В качестве расширителя DOS программа RUN286 осуществляет все функции по переключению режимов. Теперь, когда программа DPMIMEM запускается, она сразу же начинает работать в защищенном режиме, так что вызов dpmi_present() в DPMIMEM.C дает положительный результат. При этом dpmi_init() не вызывается. Мы можем воспользоваться для отладки на уровне исходного кода средствами Microsoft CodeView для защищенного режима:

```
cl -Lp -Zi dpmimem.c -link
slibre.lib
run286\c600\bin\c600\cvp dpmimem
```

Следует отметить, что CVP — это фактически CodeView для OS/2, но RUN286 позволяет выполнять эту



программу, а также многие другие программы OS/2 символического режима под управлением MS-DOS. При работе в среде Windows в расширенном режиме она использует средства DPMI. Возможность выполнять программы OS/2 под управлением DOS — показательный пример эффективности DPMI.

DPMI будет использоваться преимущественно поставщиками расширителей DOS для построения таких программных продуктов, как 286:DOS-Extender. Однако очевидно, что некоторые разработчики предпочтут применять DPMI непосредственно, в частности в программах, которые имеют относительно небольшие объемы исходного кода, но при этом требуют огромной памяти — по крайней мере в тех случаях, когда программу вполне целесообразно выполнять в расширенном режиме Windows.

Есть еще одно связанное с DPMI обстоятельство, которое следует знать разработчикам. Мы видели, насколько эффективен защищенный режим с точки зрения предоставления программам больших объемов памяти.

Однако защищенный режим имеет и свои недостатки. Как я указывал выше, в защищенном режиме отсутствует связь между селекторами, которыми манипулирует программа, и абсолютными адресами ячеек памяти. В то же время во многих уже существующих программах для персональных компьютеров используются прямые манипуляции с адресами памяти для непосредственного вывода информации на экран, чтения данных из области BIOS, обращения к недокументированным структурам данных DOS, работы с устройствами, отображаемыми в основной памяти и т.д. В защищенном режиме осуществлять подобные операции становится весьма сложно.

В связи с этим возникает особая проблема для программ Windows. Поскольку программы Windows в стандартном и расширенном режимах (а пользоваться Windows целесообразно только в этих режимах) теперь используют защищенный режим, у разработчиков таких программ появля-

ется серьезное препятствие, когда им необходимо манипулировать непосредственно абсолютными адресами ячеек памяти. Даже попытка просто прочитать адрес — например 0040:006C, — приводит к немедленной выдаче сообщения о нарушении защиты. Даже "большой" интерфейс API системы Windows не предусматривает, по-видимому, никаких функций для обращения к абсолютным адресам при работе в защищенном режиме. А это означает, что возникают дополнительные проблемы.

В следующей статье рубрики "Маленькие хитрости" я расскажу о том, каким образом программы Windows могут использовать DPMI для решения подобных проблем. DPMI обеспечит все возможности доступа к ячейкам памяти машины при работе в стандартном и расширенном режимах. Благодаря этому интерфейс DPMI, первоначально предназначенный для небольшой группы фирм с тем, чтобы помочь им в разработке расширителей DOS, в дальнейшем окажется полезным гораздо более обширному контингенту программистов ПК.

НОВОСТИ

Phar Lap Software обеспечивает использование расширенной памяти

Выпущена версия 3.0 пакета 386/DOS Extender Software Development Kit (SDK), в состав которого входят ассемблер, программа сборки, модуль расширения DOS (Run 386) и отладчик для работы в защищенном режиме. Дополнительно можно приобрести модуль управления виртуальной памятью Virtual Memory Manager (VMM). Новая версия пакета совместима со стандартом XMS, требует меньше места в оперативной памяти и позволяет непосредственно загружать файлы данных в память без дополнительных указателей или обращений.

По утверждениям представителей фирмы, прикладные программы, подготовленные при помощи новой версии пакета, будут полностью совместимы с MS-DOS 5.0, а также с оболочкой Windows в реальном и стандартном режимах. Совместимость с расширенным режимом Windows/386 будет обеспечена в последующих версиях.

Цена по каталогу для полного пакета - 495 долл., при переходе от предыдущих версий — 150 долл.; модуль управления виртуальной памятью стоит 295 долл., при переходе от прежних версий — 25 долл.

Требуемые ресурсы: компьютер, совместимый с IBM PC/AT-386, 512 Кбайт оперативной памяти, операционная система MS-DOS 3.0 или более новая версия.

PC Magazine, June 11, 1991, p. 62

AT@T представляет свой новый блокнотный компьютер

В состав нового блокнотного компьютера Safari NSX/20 фирмы AT@T входит модем на 2400 бод, соответствующий рекомендациям V.22bis и V.42 МККТТ и предусматривающий сжатие данных по стандарту MNP5. Компьютер комплектуется операционной системой DOS 4.01, Windows 3.0, пакетом электронной почты AT@T Mail и справочником по графической системе, Tour Guide.

Компьютер, построенный на базе микропроцессора i386SX/20, имеет габариты 4,6x30,5x24,15 см и массу 2,6 кг. С комплектом батарей его масса увеличивается до 3,32 кг. Длительность непрерывной работы на батареях 6 часов.

Фирма предлагает на выбор два жестких диска емкостью 40 или 80 Мбайт. Системное ОЗУ может быть расширено с 2 до 8 Мбайт. 82-клавишная клавиатура включает 12 функциональных клавиш. Компьютер комплектуется 89-мм НГМД и двухкнопочной мышью. 25-см ЖК-монитор на супертвист-эффекте с задней подсветкой поддерживает режим VGA.

Safari NSX/20 оборудован одним параллельным и двумя последовательными портами, внешним портом расширения шины, многофункциональным портом для подключения мыши, клавиатуры или цифровой клавиатуры, а также двумя разъемами встроенного модема (RJ-11).

Цена по каталогу: Safari NSX/20 с 40-Мбайт жестким диском — 5399 долл., с 80-Мбайт жестким диском - 6399 долл.

Ручной цветной однопроходный сканер CS-4096

Ручной сканер CS-4096 компании Migraph поставляется в комплекте с программным обеспечением для сканирования, редактирования и обработки изображений в среде Windows 3.0. Программное обеспечение включает в свой состав утилиту фирмы Migraph для сканирования цветных изображений, пакет Picture Publisher Plus фирмы Astral Development и пакет ImagePrep компании Computer Presentation.

Однопроходный сканер фирмы Migraph может работать в трех режимах — цветном, полутонном со сглаживанием и штриховом, — кроме того, программное обеспечение дает возможность интерпретировать цвета как 256 градаций серого.

Программное обеспечение позволяет хранить изображения, и черно-белые и цветные (4096 цветов), в файлах формата PCX. В пакете Picture Publisher Plus 2.5 для редактирования 24-рядных изображений предусмотрено использование видеоподсистем VGA или SVGA. Пакет ImagePrep 3.1 обеспечивает перевод цветного изображения в полутонное с 256 градациями серого, а также производит сжатие файлов с коэффициентом 20:1.

Цена по каталогу: цветной сканер CS-4096 - 895 долл.

Требуемые ресурсы: 2 Мбайт расширенной или отображаемой памяти, видеоподсистема VGA, 16-разрядное гнездо расширения, Windows 3.0.

Эндрю Шулман

Ранее мы уже останавливались на преимуществах, которые получают программисты, использующие предоставляемые средой MS Windows возможности МП i80286 — i80486 в защищенном режиме (или, более точно, режиме защиты ресурсов).

В этом режиме программ, написанным для работы под управлением DOS, становятся доступны мегабайты памяти и реальная многозадачность, а программисты, пишущие для Windows, обретают простой и надежный инструмент управления системными ресурсами.

Словом, с точки зрения программирования Windows — это не что иное, как работающая в защищенном режиме надстройка над DOS.

Учитывая лидирующее положение фирмы Microsoft на программном рынке и растущую популярность Windows 3.0, можно с большой долей уверенности предположить, что в ближайшие годы именно DOS в защищенном режиме станет наиболее распространенной операционной средой.

Однако недостатки защищенного режима — это, как обычно, продолжение достоинств. Механизм защиты, разработанный фирмой Intel, предотвращает нарушения в работе операционной системы из-за ошибок в прикладных программах. Но для этого пришлось пожертвовать возможностью свободно обращаться к произвольным адресам в памяти. Таким образом, доступ к исполнимым кодам или данным за пределами вашей программы оказывается "запрещен".

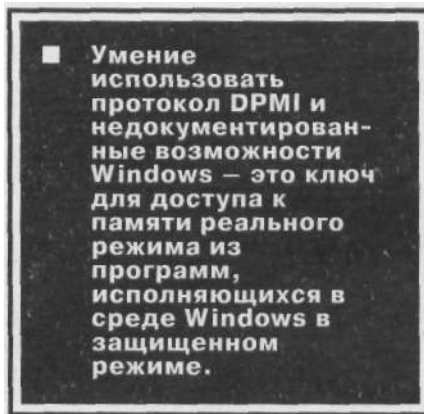
Тут, правда, возникает законный вопрос: если большинство прикладных программ для Windows будут работать в защищенном режиме, то каким же образом они будут обмениваться с драйверами устройств, резидентными программами и другими поистине драгоценными ресурсами, "обитающими" в реальном режиме в первом мегабайте памяти?

В этой статье мы поговорим о подводных камнях, которые подстерегают программистов, разрабатывающих прикладные системы DOS для работы в защищенном режиме Windows 3.0. Будет показано, что решение этой непростой задачи состоит в умелом использовании интерфейса защищенного режима с DOS (DOS Protected Mode Interface — DPMI), а также многих недокументированных средств интерфейса прикладных программ Windows (Windows API).

ПЕРВЫЕ ТРУДНОСТИ

Совершенно естественно, что вывод программы в открытый океан прак-

WINDOWS 3.0 В ЗАЩИЩЕННОМ РЕЖИМЕ — ДОСТИГАЯ НЕВОЗМОЖНОГО



тически неограниченного пространства адресов и реальной многозадачности, защищенный режим просто обязан установить свои "правила судопроизводства", которые были просто излишни в маленькой 640-Кбайт бухте DOS в реальном режиме. Переводя основные из этих правил на язык этикета, их можно было бы сформулировать примерно так: "не встревай в чужой разговор", "не лезь своей вилкой в общую тарелку" и, наконец, "не чавкай во время еды". В нашем случае это означает, что программы никоим образом (чтение, запись, исполнение машинных инструкций) не должны обращаться к памяти, не относящейся к их собственному адресному пространству. Кроме этого, запрещено записывать что-либо в адреса, где находятся исполнимые коды или наоборот пытаться "исполнять" данные. Программы в защищенном режиме могут отводить для себя многие мегабайты, но не могут использовать ни байта памяти, за пределами той, которая им отведена.

Однако программистам хорошо известно, что многие прикладные программы для ПК непременно должны иметь доступ к машинным инструкциям или данным, заведомо находящимся "не в их епархии". Так, основные переменные DOS хранятся в сегменте данных BIOS, находящемся по адресу 400h (который в реальном режиме работы МП может быть представлен, к примеру, в виде 0000:0400 или 0040:0000). Излишне пояснять, что такие адреса не могут "принадлежать" прикладной программе.

Предположим, что вы пытаетесь из программы, работающей в защищенном режиме в среде Windows, прочитать что-нибудь из этой обла-

сти данных, используя простейший фрагмент на языке Си:

```
unsigned long far *pticks= (unsigned
long far *) 0x46C;
unsigned long ticks= *pticks;
```

Все это прекрасно работает в реальном режиме, но увя — при работе в защищенном режиме переменной ticks так никогда и не будет присвоено значение, соответствующее содержанию системного таймера, которое хранится в области данных в BIOS по адресу 046Ch. Зато Windows выдаст неутешительное сообщение:

UNRECOVERABLE APPLICATION ERROR
Terminating current application.
(**"НЕИСПРАВНАЯ ОШИБКА В ПРИКЛАДНОЙ ПРОГРАММЕ.**
Исполнение текущей программы прекращено".)

Правда, если запуск этой программы производился при помощи отладчика (например, CodeView для Windows или Turbo Debugger для Windows), то ее исполнение на этом не завершится: вы получите от отладчика диагностическое сообщение "Trap 13 (ODh) — General Protection Fault" ("Автоматическое прерывание 13 (ODh) — попытка нарушения защиты"). Если удастся как-нибудь выкарабкаться из этой малоприятной ситуации, то процесс отладки можно будет возобновить. Но даже в этом случае продвинуться дальше злополучной строки скорее всего не удастся. Что ж, удовлетворимся тем, что отладчик сообщил нам, в чем проблема.

В самом деле, для разработчика программ сообщение отладчика "Попытка нарушения защиты" гораздо более информативно, чем невнятное сообщение Windows об ошибке. Состояние "Попытка нарушения защиты" (ПНЗ) генерируется микропроцессорами фирмы Intel, как только какие-либо машинные инструкции посягают на "правила этикета" в защищенном режиме. Операционная среда, как правило, устанавливает свой обработчик прерывания ODh (INT ODh), отвечающего за обработку этой нештатной для защищенного режима ситуации. В случае с Windows обработка INT ODh сводится к тому, чтобы "отказать невеже от дома": исполнение некорректной программы прекращается и в инфор-

мационном окне появляется сообщение о критической ошибке. Но сделать сколько-нибудь далеко идущие выводы из этого сообщения невозможно.

Итак, мы убедились, что как только программа DOS, исполняющаяся в защищенном режиме, попытается получить доступ к "чужому" адресу (например, 0040:0000), пользуясь соглашениями реального режима, МП генерирует ситуацию ошибки ПНЗ, что обычно приводит к аварийному завершению. Как же в таком случае ваши программы, написанные для Windows в защищенном режиме, смогут получить доступ к ресурсам, использованным в реальном режиме? Ведь кроме области данных BIOS вам, например, может потребоваться общение с драйвером ЛВС, базой данных, резидентными программами. Наконец, как добраться до недокументированных структур данных, с которыми так часто работают прикладные программы, описываемые в нашем журнале?

Все это далеко не праздные вопросы, поскольку потребуются не один год, пока все драйверы, резидентные программы, а также устройства, управление которыми осуществляется через обмены с адресами памяти, будут составлены так, что мы сможем получить к ним прямой доступ в защищенном режиме. Разработчики программ не могут сидеть сложа руки в ожидании светлого будущего. Им уже сейчас просто необходимо из написанных для защищенного режима программ иметь доступ к абсолютным адресам памяти и функциям, доступным в реальном режиме. К счастью, и это будет показано ниже, упомянутые и другие системные ресурсы низкого уровня все же доступны в защищенном режиме. Излишне упоминать о том, что программист сможет "удержаться на уровне", только зная, как все это делается.

ПЕРЕБИРАЕМ ЦЕПОЧКУ ДРАЙВЕРОВ УСТРОЙСТВ

Посмотрим, как работает обычная программа DOS, которая просто сообщает об именах и адресах драйверов устройств, загруженных в системе. В комплект поставки пакетов Turbo C++ и Borland C++ входит такая программа — TDDEV; полная информация о драйверах может быть получена также с помощью утилиты Manifest фирмы Quarterdeck Office Systems.

Конечно, программа типа TDDEV может быть запущена в Windows. Но все же хотелось бы иметь в своем распоряжении нечто, способное взаимодействовать с пользователем, а не просто выводить информацию на экран дисплея. Со временем вы смогли бы расширить ее возможности так, чтобы, выбрав из меню с помощью клавиатуры или мыши имя интересующего вас драйвера, можно было

КОДЫ УСТРОЙСТВ В РЕАЛЬНОМ РЕЖИМЕ

ФРАГМЕНТ ЛИСТИНГА

```
ListOfLists far *doslist;
DeviceDriver far *dd;
//...
dd = &doslist->nul;
for(;;)
{
    printf("%Fp\t", dd);
    if (dd->attr & CHAR_DEV)
        printf("%.8Fs\n", dd->u.name);
    else
        printf("Блочн. устр.: %u бит\n", dd->u.blk_cnt);
    dd = dd->next;
    if(FP_OFF(dd->next) == -1)
        break;
}
```

Рис. 1: Си - программа для вывода списка устройств DOS в реальном режиме.

получить о нем более подробную информацию. Оболочка Windows открывает перед своими пользователями практически неограниченные перспективы. Но пока что удовлетворимся неинтерактивной программой, лишь изменив ее так, чтобы она исполнялась Windows не в специальном окне DOS, а как стандартный процесс.

Приступая к осуществлению задуманного, следует поразмыслить над тем, как построить программу для Windows таким образом, чтобы ее исходный текст был по своей структуре как можно ближе к исходному тексту аналогичной программы, ориентированной на исполнение в DOS. Например, было бы совсем неплохо использовать в ней часто популярную в языке Си функцию printf() и избежать (по крайней мере пока) сложностей характерного для Windows программирования, управляемого событиями. (Эти вопросы подробнее рассмотрены во врезке "Printf() для Windows".)

Наша программа должна будет использовать стандартный и расширенный режимы работы Windows. А это, в свою очередь, означает, что она будет исполняться в защищенном режиме микропроцессоров i80286, i80386 или i80486. Программа DEV будет перебирать так называемую цепочку драйверов системы, выводя на экран имя и адрес каждого устройства, обнаруженного в связанном списке. Трудность здесь состоит в том, что цепочка драйверов устройств DOS представляет из себя структуру данных, находящуюся в обычной памяти и рассчитанную на использование в реальном режиме. Мы должны будем каким-то образом получить возможность доступа к этой структуре из Windows в защищенном режиме. В изложенном выше примере мы уже убедились, что

обычно подобная попытка приводит к аварийному завершению программы.

Получить информацию о драйверах устройств в программе DOS, исполняющейся в реальном режиме, достаточно просто. Сначала вы вызываете недокументированную функцию DOS 52h (INT 21h AH=52h) и получаете (в формате "сегментхещение") "дальний" указатель на таблицу внутренних переменных DOS, которую иногда называют списком списков. Поскольку в этой таблице содержится заголовок устройства NUL, он служит указателем на начало цепочки устройств. Термин "цепочка" употребляется потому, что заголовок каждого устройства содержит адрес следующего. Именно такая организация данных и называется связанным списком.

Заголовок NUL находится в реальном режиме по адресу 0000:1228. Он содержит адрес следующего устройства, которым, может быть, скажем SmartDrive (SMARTAAR). Указатель — это просто четырехбайтовое число, 0000:8B80, являющееся адресом для SMARTAAR. Под заголовком SMARTAAR, в свою очередь, может быть записан, например, указатель, 0D34:0000 на устройство Microsoft Mouse (MSSMOUSE). Цепочка продолжается аналогичным образом, пока не доходит до устройства, для которого в поле "адрес следующего драйвера" (на месте смещения) содержится -1 (FFFFh). По соглашениям DOS это признак того, что достигнут конец цепочки драйверов.

В процессе просмотра цепочки драйверов программа выдает информацию об их адресах и именах (для символьных устройств) или количестве логических единиц (для блочных устройств). Все эти сведения содержатся в заголовках драйверов. Цикл на языке Си, реализующий

ющий рассмотренный нами алгоритм, ориентированный на работу в реальном режиме DOS, показан на рис. 1.

Но как же получить доступ к адресам в реальном режиме из Windows, не вызвав аварийного завершения?

КОЕ-ЧТО О DPMI

Как я уже упоминал ранее, используя интерфейс DOS с защищенным режимом (DPMI) и некоторые недокументированные функции Windows, программы, исполняющиеся в этой многозадачной оболочке в защищенном режиме, получают возможность, не нарушая системных соглашений, непосредственно манипулировать ресурсами системы. В защищенном режиме DPMI пользуются прерыванием 31h для вызова целого ряда функций, позволяющих в реальном режиме исполнять машинные коды, отображать адреса памяти на адреса защищенного режима, перехватывать прерывания реального режима и т.п.

Следует заметить, что, в отличие от стандартов EMS и XMS, интерфейс DPMI первоначально не предназначался для широкого применения. По замыслу разработчиков, функции DPMI должны были найти применение в системных, а отнюдь не в пользовательских программах. Фирмы, совместными усилиями которых был создан DPMI (Microsoft, Intel, Lotus, Phar Lap, Quarterdeck, Rational Systems, IBM, Borland и др.), полагали, что именно они составят основную пользовательскую базу нового программного изделия.

DPMI задумывался с целью обеспечить мирное сосуществование надстроек над DOS, работающих в защищенном режиме администраторов отображаемой памяти, с различными многозадачными системами, базирующимися на DOS. Как и ранее, в случае использования стандарта VCPi (Virtual Control Program Interface — программный интерфейс виртуального управления, разработанный фирмами Quarterdeck и Phar Lap), от конечных пользователей и подавляющего большинства программистов требовалось только знать, поддерживают ли данный стандарт те или иные программы. Их не должно было интересовать подробное описание функций самого VCPi или DPMI.

Такова теория. Однако на практике разработчики программ для Windows обнаружили значительные недостатки в интерфейсе прикладных программ Windows. Среди более чем пятисот функций, перечисленных в объемистом Справочнике программиста для Windows, нет ни одной, которая позволяла бы, например, воспользоваться в DOS каким-либо драйвером устройства. Вам даже не удастся выяснить положение этого драйвера в памяти.

А вот DPMI не только позволяет использовать системные ресурсы низ-

кого уровня программам, исполняющимся в защищенном режиме, но и обеспечивает доступ к ним из среды Windows. Вот почему использование DPMI и некоторых полезных, но не документированных функций и возможностей Windows — это поистине отмычка ко всем потайным дверям Windows 3.0.

РАБОТА С DPMI

Перед тем как непосредственно применить функции DPMI в программе DEV, рассмотрим более понятный пример их использования в программе DPMIINFO, рассчитанной для работы в Windows. DPMIINFO — это небольшая утилита, сообщающая некоторые интересные подробности о режиме работы DPMI. Эта информация позволяет уяснить различия между стандартным и улучшенным режимами работы Windows. Несмотря на обсуждение этих различий в предыдущей статье, они, тем не менее, заслуживают еще некоторого внимания.

В расширенном режиме, ориентированном на МП i80386, поддерживаются виртуальная память (подкачка сегментов с диска) и работа программ для DOS и Windows, использующих 32-разрядную адресацию. Стандартный режим, ориентированный на МП i80286, не обеспечивает поддержки виртуальной памяти и разрешает лишь 16-разрядную адресацию. Кроме того, в расширенном режиме прерывания, произошедшие в защищенном режиме (например, INT 21h), "отражаются" (передаются) системе DOS, работающей в режиме виртуального процессора 8086 (Virtual 86 - V86). А в стандартном режиме происходящие в защищенном режиме прерывания передаются DOS, работающей в обычном реальном режиме.

Обратимся к исходному тексту DPMIINFO.C, показанному на рис. 2. Из него следует, что перед вызовом через прерывание INT 31 h функций DPMI должны быть выполнены определенные проверки.

Сначала необходимо установить, исполняется ли программа в защищенном режиме. Для этого используется функция Windows API GetWinFlagsO. Затем с помощью функции DPMI Mode Detection call (определение режима работы DPMI - INT 2Fh AX=1686h) осуществляется проверка на наличие DPMI. (Обратите внимание, что эта не обсуждавшаяся ранее функция INT 2Fh AX=1687h, применяется для определения точки входа в программу, переключающую МП в защищенный режим.) Выполнение INT 2Fh AX=1686h производится для того, чтобы узнать, не находимся ли мы в защищенном режиме сразу же после

запуска программы под управлением DPMI.

Программы, ориентированные на Windows, автоматически загружаются для исполнения в защищенном режиме и, следовательно, не нуждаются в вызове INT 2F/1686 — очень важное обстоятельство, если учесть, что упомянутая функция пока не поддерживается в стандартном режиме Windows. Кстати, этим же объясняется и тот факт, что программы DOS, исполняющиеся в стандартном режиме, в отличие от программ Windows, не имеют доступа к DPMI.

Установив, что мы находимся в защищенном режиме под управлением DPMI, получаем возможность обращения к функциям INT 31h, которые поддерживаются только в защищенном режиме. Затем вызываем функцию Get Version DPMI (определение версии DPMI - INT 31h AX=0400h).

Полное описание функций dpmi_present() и dpmi_version(), используемых в DPMIINFO.C, наряду с описаниями других относящихся к DPMI функций, приводятся в листингах DPMI.H и DPMI.C (рис. 3 и 4, соответственно).

Обратите внимание, что в DPMI.C для вызова INT 2Fh и INT 31h применено прямое обращение к ассемблеру из языка Си. Поскольку такая возможность обеспечивается трансляторами как Borland C++, так и Microsoft C 6.0, в программу не придется вносить никаких изменений, связанных с типом используемого компилятора. Избегайте использования имеющихся в Си функций int86() и int86x(), поскольку при обращении через их посредство ко многим из обработчиков прерываний происходит генерация состояния ошибки, чего мы как раз и пытаемся избежать, используя DPMI!

В дополнение к функциям, приводимым в DPMI.H и DPMI.C, вы можете воспользоваться исходными текстами из уже имеющихся на программном рынке библиотек DPMI. Например, одна из таких библиотек, SoftDPMI, продается фирмой SoftWorks International (тел. 404-876-6115).

ДОСТУП К ФУНКЦИЯМ РЕАЛЬНОГО РЕЖИМА

Теперь мы готовы к рассмотрению того, как возможности DPMI могут быть использованы в программе DEV. В ее исходном тексте (рис. 5) некоторые фрагменты ограничены пометками

```
#ifdef WINDOWS
```

означающими, что они будут использоваться при построении программы лишь при добавлении -DWINDOWS к командной строке компилятора.

Первый такой условно компилируемый фрагмент появляется в функции `get_doslist()`. Эта функция, как уже упоминалось ранее, вызывает недокументированную функцию операционной системы `INT 21h AH=52h`, возвращающую дальний указатель на таблицу внутренних переменных операционной системы.

При компиляции для Windows `get_doslist()` вызывает функцию `dpmi_rmode_intr()`, объявленную в `DPMI.H` (рис. 3), исходный текст которой находится в файле `DPMI.C` (рис. 4). `Dpmi_rmode_intr()` обеспечивает обращение из языка Си к функции имитации прерывания реального режима `DPMI` (`Simulate Real Mode Interrupt, INT 31h AX=0300h`) — одной из наиболее важных функций интерфейса между реальным и защищенным режимами, обеспечиваемых протоколом `DPMI`.

Но почему бы нам просто не использовать вызов `INT 21h AH=52h` с помощью функции `intdosx0` языка Си?

Ответ все тот же: версия `DEV.C`, ориентированная на работу под управлением Windows, исполняется в защищенном режиме, а `INT 21h AH=52h` — это функция `MS-DOS`, рассчитанная на работу в реальном режиме.

Поскольку Windows — не что иное, как "настройка" над `DOS`, эта

многозадачная среда просто обязана поддерживать вызовы функций `INT 21h` в защищенном режиме, например, для того, чтобы открывать файлы, менять текущий подкаталог, резервировать память и т.д. Однако, поскольку функция `DOS 52h` является недокументированной, нельзя гарантировать, что Windows обеспечивает правильный доступ к ней в защищенном режиме.

Именно поэтому мы должны вызывать ее через обращение к функции `DPMI` "имитация прерывания реального режима". Как видно из исходного текста функции `get_doslist()` на рис. 5, в структуре `RMODE_CALL` собираются необходимые для предстоящего вызова значения 32-разрядных регистров МП, а затем следует обращение к `dpmi_rmode_intr()`, которая играет здесь роль имеющейся в `Microsoft C` и других компиляторах функции `int86()`.

Мы рассматриваем работу программы `DEV` не просто для того, чтобы продемонстрировать использование вызовов `DOS` из программной среды Windows. В самом деле, обращение к функции `52h DOS` может поддерживаться Windows в защищенном режиме. Более того, настройка над `DOS` в защищенном режиме в со-

стоянии обеспечить прямой доступ и к другим недокументированным функциям `DOS`. Например, программа `386 DOS-Extender 3.0`, разработанная фирмой `Phar Lap`, обеспечивая в защищенном режиме вызов многих недокументированных функций `DOS`, позволяет отказаться от использования имитации прерывания реального режима в `DPMI`.

Тем не менее всегда будут существовать функции `DOS`, прямой доступ к которым из программ-настроек над этой операционной системой будет невозможен. Мы рассмотрели вызов `INT 21h AH=52h` с помощью `DPMI` в качестве выхода из подобной ситуации. Итак, если нельзя непосредственно обратиться к обработчику какого-либо прерывания в защищенном режиме Windows, следует сделать это косвенно, используя `INT 31h AX=0300h`, быть может через функцию языка Си, подобную `dpmi_rmode_intr()`, как было продемонстрировано выше. В `DPMI` также имеются аналогичные функции для вызова работающих в реальном режиме процедур, которые используют передачу параметров через стек.

ОБРАЩЕНИЕ К ПАМЯТИ РЕАЛЬНОГО РЕЖИМА

Функция `get_doslist()` возвращает дальний указатель на структуру, в

DPMIINFO.C	ПОЛНЫЙ ЛИСТИНГ
<pre> /* DPMIINFO.C — программа для работы в Windows, сообщающая информация о режиме работы DPMI Copyright (c) 1991 Ziff Communication Co. PC Magazine * Andrew Shulman Borland C++ 2.0: bcc -W -2 dpmiinfo.c printf.c dpmi.c rc dpmiinfo.exe Microsoft C 6.0: cl -c -As -G2sw -Oais -Zpe dpmiinfo.c printf.c dpmi.c link /align:16 dpmiinfo printf dpmi, dpmiinfo, /nod sllbcw libw,win.def rc dpmiinfo.exe */ #include <stdlib.h> #include <stdarg.h> #include <string.h> #include <dos.h> #include <windows.h> #include "printf.h" #include "dpmi.h" #define FAIL(s) MessageBox(NULL, s, "Информация о DPMI", MB_OK) int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpszCmdLine, int nCmdShow) { unsigned long win_flags; unsigned flags; </pre>	<pre> unsigned maj, min; unsigned proc; if (! ((win_flags = GetWinFlags()) & WF_PMODE)) return FAIL("Для этой программы требуется стандартный или\ расширенный режим Windows "); if (! dpmi_present()) return FAIL("DPMI отсутствует"); dpmi_version(&maj, &min, &flags, &proc); open_display("Информация DPMI"); if (win_flags & WF_STANDARD) printf("DPMI %x.%x\n", maj, min); // из-за ошибки в // Windows здесь применен формат вывода %x else printf("DPMI %d.%d\n", maj, min); printf("Windows %s режим на процессоре 80%d86\n", (win_flags & WF_ENHANCED) ? "расширенный" : (win_flags & WF_STANDARD) ? "стандартный" : "Не другой?!", proc); // процессор: 2=286, 3=386, 4=486 printf("реализация DPMI для процессора %d\n", (flags & 1) ? 386 : 286); printf("Прерывания отражаются в %s режим\n", (flags & 2) ? "реальный" : "V86"); printf("виртуальная память %s поддерживается\n", (flags & 4) ? " " : "не "); show_display(); return 0; } </pre>

Рис. 2: Текст программы `DPMIINFO`, обеспечивающей вывод списка устройств при работе в среде `WINDOWS`.

```

DPMI.H                                ПОЛНЫЙ ЛИСТИНГ

/* DPMI.H */
#pragma pack

typedef struct {
    unsigned long edi, esi, ebp, reserved, ebx, edx, ecx, eax;
    unsigned flags, es, ds, fs, gs, ip, cs, sp, ss;
} RMODE_CALL;

/* структура дескриптора защищенного режима */
typedef struct {
    unsigned limit, addrlo;
    unsigned char addr_hi, access, reserved, addrxhi;
} DESCRIPTOR;

/* Функции DPMI низкого уровня V
BOOL dpmi_present(void);
void dpmi_version(unsigned *pmaaj, unsigned *pmin,
    unsigned *pflags, unsigned *pproc);
BOOL dpmi_rmode_intr(unsigned intno, unsigned flags,
    unsigned copywords, RMODE_CALL far *rmode__call);
unsigned dpmi_sel(void);
BOOL dpmi_set_descriptor(unsigned pmodesel, DESCRIPTOR far *d);
BOOL dpmi_get_descriptor(unsigned pmodesel, DESCRIPTOR far *d);
BOOL dpmisel freefun unsigned pmodesel);

/* Функции DPMI более высокого уровня */
unsigned DosAllocRealSeg( DWORD bytes, unsigned *ppara, unsigned *psel);
unsigned DosFreeRealSeg(unsigned sel);
unsigned DosMapRealSeg(unsigned rmpara, DWORD size, unsigned far *psel)
unsigned DosFreeSeg(unsigned sel);
void far *DosProtToReal(void far *prot);

void far *map_real(void far *rptr, unsigned long size);
void free_mapped_seg(void far *fp);
unsigned getjnappped(void);
unsigned verw(unsigned sel);

/* Недокументированные функции Windows V
extern DWORD FAR PASCAL GetSelectorBase(unsigned sel);
extern DWORD FAR PASCAL GetSelectorLimit(unsigned sel);
extern void FAR PASCAL SetSelectorBase(unsigned sel, DWORD base);
extern void FAR PASCAL SetSelectorLimit(unsigned sel, DWORD base);

```

Рис. 3: Файл - заголовок для интерфейса DPMI.

которой содержатся внутренние переменные DOS. Но какого типа этот указатель? В программах, исполняющихся в защищенном режиме, когда операционная система ориентирована на реальный режим (как в случае DOS), используются указатели двух типов: указатели реального режима (УРР) и указатели режима защиты ресурсов (УРЗР). В случае `get_doslist()` вызов `INT 21h AH=52h` возвращает в регистрах `ES:BX` указателя реального режима, даже если обращение к этому обработчику происходило через функцию `dpmi_rmode_intr()`.

Теперь нам хотелось бы получить доступ к блоку памяти, соответствующему полученному указателю. Для этого в следующем компилируемом только для работы под Windows фрагменте DEV.C вызывается функция `map_real()`. Она получает на входе в качестве параметров описание блока памяти реального режима (размер, адрес в реальном режиме), а

возвращает соответствующий этому блоку УРЗР.

Если вашей программе, ориентированной для работы под управлением Windows необходимо получить доступ к области памяти в реальном режиме, можете смело использовать функцию `map_real()` с последующим обращением после каждого ее вызова к функции `free_mapped_seg()`. (Полные описания `map_real()` и `free_mapped_seg()` содержатся в файлах `DPMI.H` и `DPMI.C`.) При этом программисту совсем не обязательно вникать в тонкости DPMI, скрывающиеся за этими вызовами. Тем не менее, наиболее дотошные читатели, наверное, уже заинтересовались тем, как работает функция `map_real()`.

КАК РАБОТАЕТ `M1AP_REAL()`

Функция `map_real()` построена на базе функции `DosMapRealSegO`, которая, в свою очередь, использует вы-

зовы нескольких функций интерфейса между языком Си и DPMI.

Такая "многослойная" структура вызова обусловлена низкоуровневым характером протокола DPMI, который, если вы помните, первоначально предназначался лишь для обеспечения совместимости между программами небольшой группы фирм, поставляющих использующие защищенный режим надстройки над MS-DOS. Поэтому полезно надстроить над нижним уровнем DPMI еще несколько "этажей" функций более высокого уровня.

Программа DEV.C при вызове `map_real()` использует следующие функции DPMI низкого уровня:

- произвести выборку дескриптора из таблицы локальных дескрипторов (`INT 31h AX=0000h`; вызывается из `dpmi_sel()`);
- найти дескриптор (`INT 31h AX=000Bh`; вызывается из `dpmi_get_descriptor()`);
- изменить дескриптор (`INT 31h AX=000Ch`; вызывается из `dpmi_set_descriptor()`);

Но постойте — какую роль играют таблицы локальных дескрипторов (ТЛД) и сами дескрипторы при доступе из защищенного режима к абсолютным адресам в памяти? Вы, вероятно, помните, что часть `xxxx` произвольного адреса `xxxx:yyyy` в защищенном режиме называется селектором. Селектор, в свою очередь, — это индекс, по которому производится выборка содержимого из так называемой таблицы дескрипторов, что в корне отличается от системы формирования адреса в реальном режиме, где указатель типа `xxxx:yyyy` (например, `1234:0005`) — это всего лишь несколько искаженный способ записи для соответствующего ему вполне определенного абсолютного адреса в реальном режиме (в нашем примере — `12345h`).

Таким образом, в защищенном режиме используется косвенная адресация памяти. Поэтому абсолютный адрес в памяти, определяемый УРЗР `1234:0005`, целиком зависит от базового адреса, который содержится в дескрипторе, соответствующем селектору `1234h`. Это значит, что между значениями дескриптора и селектора не существует никакой прямой связи. Все программы, исполняющиеся под управлением Windows 3.0, используют общую ТЛД, в которой содержатся необходимые ссылки на используемые адреса в памяти.

Косвенная адресация памяти представляет собой часть механизма, используемого в микропроцессорах фирмы Intel для поддержки защищенного режима. Поскольку в защищенном режиме селекторы используются лишь для выборки значений из ТЛД, мы лишены возможности прямого обращения к адресу `12345h` простым

```

/*
DPMI.C содержит функции DPMI, некоторые аналогичные им по
назначению недокументированные функции Windows, а также
несколько "этажей надстроек" для удобства обращения к более
низким уровням

Copyright (c) 1991 Ziff Communication Co.
PC Magazine * Andrew Shulman
*/

#include <windows.h>
#include <dos.h>
#include "dpmi.h"

#define MAKEP(seg ofs) ((void tar*) MAKELONG(ofs), (seg))

BOOL dpmi_present(void)
{
    _asm mov ax, 1686h
    _asm int 2fh
    _asm not ax
}

void dpmi_version(unsigned *pmaj, unsigned *pmin,
unsigned "pflags, unsigned *pproc)
{
    unsigned char maj, min, proc;
    unsigned flags;
    _asm (
        mov ax, 0400h
        int 31h
        mov maj, ah
        mov min, al
        mov flags, bx
        mov proc, cl
    )
    *pmaj = maj;
    *pmin = min;
    *pflags = flags;
    *pproc = proc;
}

/* Пользуется прерыванием реального режима из защищенного режима */
BOOL dpmi_rmode_intr(unsigned intno, unsigned flags,
unsigned copywords, RMODE_CALL far *rmode_call)
{
    if (flags) intno |= 0x100;
    _asm {
        push di
        mov ax, 0300h //имитировать прерывание реального режима
        mov bx, word ptr intno //номер прерывания, флаги
        mov ex, word ptr copywords //сколько слов памяти
        // копировать из стека защищенного режима в стек реального
        les di, dword ptr rmode_call // в ES:DI - адрес
        // структуры вызова (реальный режим)
        int 31h //вызов DPMI
        jc error
        mov ax, 1 //возвращаем TRUE
        jmp short done
    }
error:
    _asm xor ax, ax //возвращаем FALSE
done:
    _asm pop di
}

/* Резервирует один селектор защищенного режима в ТЛД */
unsigned dpmisel(void)
{
    _asm (
        xor ax, ax //Резервировать дескрипторы ТЛД
        mov ex, 1 //резервировать один дескриптор
        int 31h //вызов DPMI
        jc error
        jmp short done //в AX - новый селектор ТЛД
    )
error:
    _asm xor ax, ax //не получилось!
done;;
}

BOOL dpmi_set_descriptor(unsigned pmodesel, DESCRIPTOR far *d)
{
    _asm {
        push di
        mov ax, 000ch // Изменить дескриптор
        mov bx, word ptr pmodesel //селектор защищенного режима
        les di, dword ptr d //дескриптор
        int 31h //вызов DPMI
        jc error
        mov ax, 1 //возвращаем TRUE
        jmp short done
    }
error:
    _asm xor ax, ax //возвращаем FALSE
done;
    _asm pop di
}

BOOL dpmi_get_descriptor(unsigned pmodesel, DESCRIPTOR far *d)
{
    _asm {
        push di
        mov ax, 000bh //найти дескриптор
        mov bx, word ptr pmodesel //селектор защищенного режима
        les di, dword ptr d //дескриптор
        int 31h //вызов DPMI
        jc error
        mov ax, 1 //возвращаем TRUE
        jmp short done
    }
error:
    _asm xor ax, ax //возвращаем FALSE
done:
    _asm pop di
}

BOOL dpmi_sel_free(unsigned pmodesel)
{
    _asm {
        mov ax, 0001h //освободить дескриптор ТЛД
        mov bx, word ptr pmodesel //селектор, который
        //необходимо освободить
        int 31h //вызов DPMI
        jc error
        mov ax, 1 //возвращаем TRUE
        jmp short done
    }
error:
    _asm xor ax, ax //возвращаем FALSE
done;;
}

/* функции более высокого уровня */
unsigned DosAllocRealSeg(DWORD bytes, unsigned *ppara, unsigned
*psel)
{
    DWORD dw = GlobalDosAlloc(bytes);
    if (dw == NULL)
        return 8; //недостаточно памяти
    *ppara = HIWORD(dw);
    *psel = LOWORD(dw);
    return 0;
}

unsigned DosFreeRealSeg(unsigned sel)

```

Рис. 4: Исходный текст DPMI.EXE.

```

{
    return (GlobalDosFree(sel) != NULL);
}

unsigned DosMapRealSeg(unsigned rmpara, DWORD size, unsigned far
*psel)
{
    DESCRIPTOR d;
    unsigned long addr;
    unsigned sel = dpmi_sel();
    if (!sel)
        return 8; //недостаточно памяти
    /* проверка достоверности селектора */
    if (!verw(FP_SEG(psel)))
        return 490; //недействительный селектор
    /* найти дескриптор для любого сегмента данных */
    dpmi_get_descriptor(FP_SEG(psel), &d);
    d.limit = (unsigned) size - 1;
    addr = ((unsigned long)(rmpara) << 4L;
    d.addrlo = (unsigned) addr;
    d.addr_hi = (unsigned char) (addr > 16);
    d.reserved = d.addr_xhi = 0;
    dpmi_set_descriptor(sel, &d);
    *psel = sel;
    return 0; //успех
}

unsigned DosFreeSeg(unsigned sel)
{
    return ! dpmi_sel_free(sel);
}

void far *DosProtToReal(void far *prot)
{
    unsigned long base = GetSelectorBase(FP_SEG(prot));
    //используем недокументированную функцию Windows
    if (base > 0xFFFFL)
        return NULL; //нет доступа в реальный режим
    else
        return MAKEP(base > 4, (base & 0x0F) + FP_OFF(prot));
}

unsigned __0000H = 0; //недокументированный селектор Windows
unsigned mapped = 0; //для отслеживания количества занятых селекторов

unsigned get_mapped(void) {return mapped; }

void far *map_real(void far *rptr, unsigned long size)
{
    unsigned seg, ofs, sel;

    if (!__0000H) //единовременная инициализация:
        //определяем адрес __0000H
        __0000H = LOWORD(GetProcAddress(GetModuleHandle("Kernel"),
        "_0000H"));

    seg = FP_SEG(rptr);
    ofs = FP_OFF(rptr);
    if ((seg < 0x1000) && ((ofs + size) < 0xFFFF))
        return MAKEP(__0000H, (seg << 4) + ofs);
    if (DosMapRealSeg(seg, size + ofs, &sel) != 0)
        return 0;
    mapped++;
    return MAKEP(sel, ofs);
}

void free_mapped_seg(void far *fp)
{
    unsigned sel = FP_SEG(fp);
    if (sel == __0000H)
        return;
    if (DosFreeSeg(sel) == 0)
        mapped--;
}

/* для проверки достоверности указателей используется инструкция
VERW процессоров фирмы Intel
*/
unsigned verw(unsigned sel)
{
    _asm mov ax, 1;
    _asm verw sel;
    _asm je short ok;
    _asm xor ax, ax;
ok:;
}

```

формированием указателя типа 1234:0005. По той же причине нельзя прочитать содержимое ячейки 046Ch, пользуясь указателем 0040:006C или 0000:0046C. Техническая реализация механизма косвенной адресации памяти, конечно, гораздо сложнее, чем здесь описано. Если вы интересуетесь подробностями, обратитесь к книге Ray Duncan et al., *Extending DOS* (Reading, Mass.: Addison Wesley, 1990) или к статье Jeff Prosser, "Segmented Memory", *PC Magazine*, March 26, 1991.

К счастью, косвенная адресация памяти не только ставит перед программистами, работающими для Windows, сложные проблемы, но и помогает их разрешению. Вернемся теперь к нашему самому первому примеру. Для выборки из памяти содержимого адреса 46Ch вовсе не обязательно знать значение указателя в форме xxxx:yyyy. Зато если в защищенном режиме известен дескриптор, базовый адрес которого 400h, можно использовать соответствующую

Все программы, исполняющиеся в среде Windows 3.0, используют общую таблицу дескрипторов, в которой содержатся необходимые ссылки на используемые адреса в памяти.

ший селектор, помня, что значение самого селектора не имеет никакого отношения к числу 40h или 400h. Отыскав нужный селектор, вы мо-

жете сохранить его в какой-нибудь переменной — например, `tu40`, `bios_seg` или `0040H`.

Итак, чтобы получить доступ к какому-либо абсолютному адресу памяти, необходимо найти селектор, соответствующий которому дескриптор имеет базовый адрес, предел и право доступа такие же, как у затребованного вами адреса реального режима. Эту задачу как раз и решает DPMI.

В программе DPMI.C функция `DosMapRealSeg` сначала вызывает `dpmi_sel()`, которая резервирует дескриптор в ТЛД и передает в вызывающую программу соответствующий селектор. Затем происходит обращение к функции `dpmi_get_descriptor()` для определения дескриптора произвольного сегмента данных в вашей программе. Полученный дескриптор применяется в качестве шаблона в дальнейших поисках. (На самом деле, используется лишь значение в поле "право доступа" — прим. перев.) Затем мы изменяем это описание дескриптора так, чтобы он пра-

```

/*
DEV.C — отображает информацию о цепочке драйверов MS-DOS

Copyright (c) 1991 Ziff Communication Co.
PC Magazine * Andrew Shulman

real mode:
  Borland C++ 2.0: bcc dev.c
  Microsoft C 6.0: cl dev.c

Borland C++ 2.0 (DPMI 'должен' компилироваться с флажком -B):
  bcc -W DWINDOWS -2 -B dev.c printf.c dpmi.c
  re dev.exe

Microsoft C 6.0:
  cl -c -As -G2sw -Oais -Zpe -W3 -DWINDOWS dev.c printf.c dpmi.c
  link /align:16 dev dpmi printf,dev, /nod slibcew libw.win.def
  re dev.exe

WIN.DEF:
; WIN.DEF — общий формат .DEF-файла для Windows
EXETYPE      WINDOWS
STUB         'WINSTUB.EXE'
CODE        PRELOAD MOVEABLE DISCARDABLE
DATA        PRELOAD MOVEABLE MULTIPLE
HEAPSIZE    10240
STACKSIZE   5120
V

#include<stdlib.h>
#include<stdio.h>
#include<string.h>
#include<dos.h>
#ifdef WINDOWS
#include<windows.h>
#include "printf.h"
#include "dpmi.h"
#endif

#ifdef WINDOWS
char *app = "Проходим по цепочке устройств DOS";
#define puts(s) MessageBox(NULL, s, app, MB_OK)
#endif
#define fail(s) return puts(s)

#ifndef MK_FP
#define MK_FP(seg, ofs) \
  ((void far *) (((unsigned long) (seg) « 16) | (ofs)))
#endif

/* некоторые биты атрибутов устройств */
#define CHAR_DEV    (1 « 15)
#define INT29      (1 « 4)
#define IS_CLOCK   (1 « 3)
#define IS_NUL     (1 « 2)

#pragma pack(1)

typedef struct DeviceDriver {
  struct DeviceDriver far *next;
  unsigned attr;
  unsigned strategy;
  unsigned intr;
  union {
    unsigned char name[8];
    unsigned char blk_cnt;
  } u;
} DeviceDriver;

typedef struct {
  unsigned char misc[8];
  DeviceDriver far *clock;

  DeviceDriver far *con;
  unsigned char misc2[18];
  DeviceDriver nul; /* это не указатель */
  // ...
} ListOfLists; // для DOS версий 3.1 и следующих

ListOfLists far *get_doslist(void)
(
#ifdef WINDOWS
  RMODE_CALL r;
  memset(&r, 0, sizeof(r));
  r.eax = 0x5200;
  return (dpmi_rmode_intr(0x21, 0, 0, &r)) ? MK_FP(r.es, r.ebx) : 0;
#else
  union REGS r;
  struct SREGS s;
  segread(&s);
  s.es = r.x.bx = 0;
  r.h.ah = 0x52;
  intdosx(&r, &r, &s);
  return MK_FP(s.es, r.x.bx);
#endif
)

#ifdef WINDOWS
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
  LPSTR lpszCmdLine, int nCmdShow)
#else
int main(int argc, char*argv[])
#endif
{
  ListOfLists far *doslist;
  DeviceDriver far *dd;
#ifdef WINDOWS
  DeviceDriver far *next;
  int mapped;
#endif

#ifdef WINDOWS
  if (! (GetWinFlagsO & WF_PMODE))
    fail("f\fnfl этой программы требуется стандартный или
    реальный режим Windows");

  if (! dpmi_present())
    fail("Для этой программы требуется прерывание INT 31h,
    обслуживаемое DPMI");
#endif

  if (! (doslist = get_doslist()))
    fail("Функция 52h для INT 21h не поддерживается");

#ifdef WINDOWS
  /* Найдем указатель защищенного режима на таблицу
  внутренних переменных DOS */
  doslist = map_real(doslist, sizeof(ListOfLists));

  open_display(app);
#endif

  /* Следующий фрагмент осуществляет двойную проверку
  правильности информации, полученной об устройствах.
  NUL - это часть DOSLIST, а не указатель; он не нуждается в
  отображении из реального режима.
  V
  if (_fmemcmp(doslist->nul.u.name, "NUL", 8) != 0)
    fail("Неверное имя NUL");
  if (! (doslist->nul.attr & IS_NUL))
    fail("Неверный атрибут NUL");

  #if def WINDOWS
    dd = map_real(doslist->con, sizeof(DeviceDriver));
  #endif
}

```

Рис. 5: Перебор цепочки устройств в защищенном режиме.

DEV.C

ЧАСТЬ 2

```

#else
    dd = doslist->con;
#endif

if (_fmemcmp(dd->u.name, "CON ", 8) != 0)
    fail("НеВерное имя CON");
if (! (dd->attr & CHAR_DEV))
    fail("НеВерHbitи атрибут CON");

#ifdef WINDOWS
free_mapped_seg(dd);
#endif

#ifdef WINDOWS
dd = map_real(doslist->clock, sizeof(DeviceDriver));
#else
dd = doslist->clock;
#endif
if (_fmemorrtt(dd->u.name, "CLOCKS ", 8) != 0)
    fail("неверное имя CLOCKS");
if (! (dd->attr & CHAR_DEV))
    fail("неверный атрибут CLOCKS");

#ifdef WINDOWS
free_mapped_seg(dd);
#endif

/* распечатка цепочки драйверов устройств */
dd = &doslist->nul;

#ifdef WINDOWS
for (;;)
(
    printf("%Fp ", DosProtToReal(dd));
    // вывод адреса реального режима
    if (dd->attr & CHAR_DEV)
        printf("%.8Fs\n", dd->u.name);

        printf("Block dev: %u unit(s)\n", dd->u.blk_cnt);
        next = dd->next;
        /* при первом проходе цикла следующая строка освободит
        селектор doslist */
        free_mapped_seg(dd);
        /* освобождаем сегмент реального
        режима */
        if (FP_OFF(next) == 0xFFFF)
            /* есть следующий драйвер?
            break;
            // если нет - выход из цикла
            dd = map_real(next, sizeof(DeviceDriver));
            // получаем указатель реального режима
            YieldO;
            /* поскольку программа не имеет цикла обработки сообще-
            ний Windows, следует отдавать часть времени другим запу-
            щенным задачам */
        }
    #else
    do {
        printf("%Fp\t", dd);
        if (dd->attr & CHAR_DEV)
            printf("%.8Fs\n", dd->u.name);
        else
            printf("Bno4H. устр.: %u бит\n", dd->u.blk_cnt);
        dd = dd->next;
        ! while (FP_OFF(dd->next) != 0xFFFF);
    }
    #endif

    #if def WINDOWS
    if (mapped = getmappedO)
        printf("осталось %u отображенных селекторов! \r\n",
            mapped);
        show_display();
        return mapped; /* 0 указывает на успех */
    #else
    return 0;
    #endif
}

```

DEV.C

ФРАГМЕНТ

```

ListOfLists far *doslist;
DeviceDriver far *dd;

/* распечатка цепочки драйверов устройств DOS */

dd = &doslist->nul;
for(;;)
!
    printf("%Fp ", DosProtToReal(dd)); // вывод адреса реального режима
    if (dd->attr & CHAR_DEV)
        printf("%.8Fs\n", dd->u.name);
    else
        printf("Bno4H. устр.: %u бит\n", dd->u.blkcnt);
    next = dd->next;
    /* нашли следующий указатель
    /* при первом проходе цикла следующая строка освободит
    селектор doslist */
    free_mapped_seg(dd);
    if (FP_OFF(next) == -1)
        /* освобождаем сегмент PP
        // есть следующий драйвер?
        break;
        // если нет — выход из цикла
    dd = map_real(next, sizeof(DeviceDriver)); // получаем указатель реального режима
    YieldO;
    // поскольку программа не имеет цикла
    // обработки сообщений Windows, завершим ее асинхронно

```

Рис. 6: Цикл перебора цепочки устройств в программе DEV.C.

вильно описывал необходимый нам базовый адрес и размер (например, размер заголовка драйвера устройства). Теперь передаем это описание как параметр функции `dpmi_set_descriptor()`. Готово! Возвращаемый селектор теперь соответству-

ет запрошенной нами области в памяти реального режима.

Как я упоминал ранее, функцией `map_real()` можно пользоваться и не вникая в детали ее работы; но если

уж вы с этим разобрались, дальнейшее освоение работы с DPMI пойдет значительно быстрее.

ОГРАНИЧЕНИЯ ПО РЕСУРСАМ

Возвращаясь теперь к исходному тексту DEV.C (рис. 5), замечаем, что функция `map_real()` вызывается несколько раз в цикле просмотра цепочки драйверов DOS. Этот цикл, являющийся центральной частью программы DEV, показан отдельно на рис. 6 для удобства сравнения с аналогичным фрагментом, ориентированным для работы в реальном режиме (рис 1).

После каждого вызова `map_real()` происходит обращение к описываемой в DPMI.C функции `free_mapped_seg O` для освобождения зарезервированных селекторов. Это совсем не лишнее, так как дескрипторы и соответствующие им селекторы являются ограниченными системными ресурсами. Вспомним, что все программы, исполняющиеся под управлением Windows 3.0, используют общую ТЛД, которая вмещает не более 8192 дескрипторов.

Освобождение селекторов настолько важно, что функции `map_real()` и `free_mapped_seg O` отслеживают, сколько их зарезервировано в данный

UNDOCSEL.C

ПОЛНЫЙ ЛИСТИНГ

```

/*
UNDOCSEL.C — иллюстрирует применение недокументированных функций
Windows для работы с селекторами

Microsoft C 6.0:
cl -c -As -G2sw -Oais -Zpe undocsel.c printf.c
link /align: 16 undocsel printf, undocsel, /nodslibcew libw, win.def
re undocsel.exe

Copyright (c) 1991 Ziff Communication Co.
PC Magazine * Andrew Shulman
*/

#include <dos.h>
#include <windows.h>
#include "printf.h"

/* Недокументированные функции Windows */
extern DWORD FAR PASCAL GetSelectorBase(unsigned sel);
extern DWORD FAR PASCAL GetSelectorLimit(unsigned sel);
extern void FAR PASCAL SetSelectorBase(unsigned sel, DWORD base);
extern void FAR PASCAL SetSelectorLimit(unsigned sel, DWORD base);

#define FAIL(s) MessageBox(NULL, s, "Информация DPMI", MB_OK)

int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
LPSTR lpszCmdLine, int nCmdShow)
{
    unsigned sel, __ds;
    unsigned long far *pticks;
    if (! (GetWinFlagsO & WF_PMODE))
        return FAIL("fiffl этой программы требуется стандартный или расширенный режим
Windows");
    open_display("Тест недокументированной функции селектора");
    __asm mov __ds, ds;
    sel = AllocSelector(__ds); // копируем DS
    printf("sel=%04X\n", sel);
    SetSelectorBase(sel, 0x400); // область данных BIOS
    SetSelectorLimit(sel, 0xFFFF); // не более 64 Кбайт
    FP_SEG(pticks) = sel;
    FP_OFF(pticks) = 0x6c; // адрес значения системного таймера
    printf("база=%081x предел=%081x тиков=%081X\n",
        GetSelectorBase(sel), GetSelectorLimit(sel), *pticks);
    show_display();
    FreeSelector(sel);
}

```

Рис. 7: Исходный текст UNDOCSEL для вывода "тиков" таймера с использованием недокументированной функции WINDOWS.

момент нашей программой. Перед завершением DEV.C проверяется, все ли использовавшиеся селекторы освобождены, поскольку многозадачная оболочка Windows 3.0 автоматически этого не делает.

ИСПОЛЬЗУЕМ НЕДОКУМЕНТИРОВАННЫЕ ВОЗМОЖНОСТИ WINDOWS

Если вы еще не забыли первоначально поставленную задачу, программа DEV должна сообщать не только имя каждого драйвера устройства, но и его адрес. Хотя функция `map_real()` и возвращает используемый затем в DEV.C адрес защищенного режима, это совсем не та информация, которую должен получить конечный пользователь. Действительно, как мы видели, этот адрес имеет значе-

ние, которое никак не может быть непосредственно преобразовано в искомый адрес реального режима.

Существуют несколько выходов из этой ситуации. Поскольку адрес драйвера в реальном режиме передается функции `map_real()` в качестве первого параметра, его можно просто сохранить на будущее в какой-нибудь переменной.

Но вместо этого программа DEV.C пользуется вызовом описываемой в DPMI.C функции `DosProtToReaK()`. Получая на входе в качестве параметра УРЗР, она пытается преобразовать его в соответствующий УРР. Именно пытается, так как если базовый адрес УРЗР переходит за лежащую на границе 1 Мбайт "линию Мажино", такое преобразование невозможно.

При попытке осуществить вышеупомянутое преобразование функция `DosProtToReaK()` обращается к недокументированной функции `Windows GetSelectorBaseO` (для той же цели можно было бы использовать и документированную функцию `DPMI dpmi_get_descriptor()`).

Если выясняется, что базовый адрес передаваемого в `DosProtToReaK()` УРЗР не превосходит значения `FFFFFh`, он суммируется с селектором смещения данного УРЗР. Это и дает на выходе функции нужный нам УРР.

Прототипы `GetSelectorBaseO` и некоторых других недокументированных функций Windows содержатся в файле `DPMI.H` (рис. 3). На самом деле, используя приводимые в нем функции `SetSelectorBaseO` и `SetSelectorLimitO` в сочетании с документированной функцией `Windows AllocSelectorO`, можно было бы вообще отказаться от услуг DPMI.

Применяя недокументированные функции, следует помнить, что нет никакой гарантии их наличия в последующих версиях Windows или даже того, что они будут правильно работать в настоящей редакции Windows 3.0. Очевидно, поэтому они и названы недокументированными!

Преимущество этих функций состоит в том, что общение с ними более удобно, чем прямое обращение к DPMI через INT 31 п. Кроме того, по крайней мере в моей практике, в Windows 3.0 недокументированные функции всегда работают правильно.

Маленькая программа UNDOCSEL.C (рис. 7) показывает, как с помощью недокументированных функций Windows прочитать содержимое системного таймера из адреса `46Ch` в области данных BIOS (`УРР = 0040.-006C`).

Функция `map_real()` также использует одно из недокументированных средств Windows: селектор `__0000H`, который жестко связан с областью памяти, начинающейся с нулевого абсолютного адреса и имеющей размер 64 Кбайт. Поскольку именно там обычно располагается большинство драйверов устройств DOS, упомянутый селектор окажется весьма полезен для программы DEV.

В руководстве по программированию для Windows, выпущенном фирмой Microsoft, описаны некоторые глобальные селекторы, такие, как `_B000H` и `_B800H`. Эти селекторы применяются в программах, использующих для работы весь экран дисплея и осуществляющих прямую запись в видеобuffer (например, Turbo Debugger для Windows или новая версия `CodeView` для Windows). Тем не менее, о полезном селекторе `__0000H` в документации не сказано ни слова. Кроме того, там утверждается, что глобальные селекторы доступны лишь в программах, написанных на

Printf () для Windows

Первая проблема, с которой вы сталкиваетесь, перенося свою программу в среду MS Windows, - это невозможность применения привычных функций вывода: printf() в языке Си и Write() в языке Turbo Pascal. Взамен придется обращаться к функциям Windows TextOutO или DrawText(). Кроме этого, придется полностью изменить структуру вашей программы, чтобы она реагировала на специальные события, происходящие в операционной среде Windows.

Неужели все это так уж необходимо? Известное утверждение о том, что требуется не менее ста строк текста на языке Си для вывода на экран сообщения "Hello world!" ("Здравствуй, мир!"), часто повторяют даже адепты Windows (причем некоторые считают это несомненным достоинством!). Подобные сложности создают ошибочное впечатление, что даже простейшая программа, состоящая из одной строчки, будучи перенесенной в Windows, должна быть обязательно снабжена обработчиком сообщений WndProc и содержать обращения к функциям RegisterClassO и CreateWindow().

Это заблуждение психологически мешает новичкам начать программировать для Windows, а для опытных разработчиков — составлять небольшие прикладные программы и проводить эксперименты "для себя", которые так необходимы в повседневной работе программиста.

Действительно, если вы будете использовать трансляторы Microsoft C или Borland C++, то обнаружите, что из их библиотек для Windows "изгнаны" стандартные функции ввода/вывода (stdio), в том числе printf(), puts() и gets(). Поэтому попытка использовать их в программе для Windows приведет к выдаче программой-компоновщиком сообщения "unresolved

external" ("неразрешимая внешняя ссылка").

Но в функциях стандартного ввода/вывода нет ничего магического, и, следовательно, они могут быть сконструированы на базе Windows API. Например, если вы пользуетесь пакетом Turbo Pascal для Windows (TPW), то простой замены оператора "Uses Crt" на "Uses WinCrt" оказывается достаточно для переноса программы из среды DOS в среду Windows, так как теперь любое обращение к оператору Write() приведет к выдаче в окно Windows. Очевидно, что аналогичный метод мог бы использоваться фирмой Borland и в языках Си и Си++, но этого почему-то сделано не было. В любом случае можно относительно легко написать специальный модуль, позволяющий составлять программы для Windows, не обращая напрямую к функциям API.

Способность функции Windows MessageBoxO формировать для вывода целый экран текста делает ее полезной для небольших прикладных программ, подобных DEV. Другой подход состоит в применении в качестве средства доступа к дисплею поставленной с Windows 3.0 программы Notepad ("блокнот"), использующей возможности многозадачности и взаимодействия между процессами. Очевидно, что MessageBoxO и Notepad могут послужить основой для создания функций вывода, подобных printf().

Исходный текст DEV.C (рис. 5 в основной статье) выглядит

```

PRINTF.H
ПОЛНЫЙ ЛИСТИНГ

/* PRINTF.H */

BOOL open_display(char *appname);
BOOL show_display(void);
int printf(const char *fmt, ...);

```

Рис. А: Файл - заголовок для версии PRINTFf) под Windows

гораздо проще стандартной программы для Windows. В нем не происходит ни регистрации класса, ни вызова функции CreateWindow(), ни обработки сообщений. Зато программа пользуется "запрещенной" функцией printf (). При компиляции для Windows, однако используется новая версия этой функции, полностью описываемая в файлах PRINTF.H и PRINTF.C (рис. А и В). В новой версии происходит накопление текста, а затем вызов функции show_display(), которая, в свою очередь, использует MessageBoxO.

Если объем текстовой информации не позволяет обработать ее используя MessageBoxO, в PRINTF.C предусмотрен запуск (посредством функции WinExecO) программы Notepad. Затем процессу Notepad, с помощью сообщения WM_SETTEXT и функции SendMessageO, посылается выводимый текст. Возможность пересылать текст из одной программы Windows в другую доказывает, что сообщения в этой многозадачной оболочке — это не только способ описания вызова для отдельных функций, но и реальная форма взаимодействия между процессами.

В самой среде Windows имеются некоторые средства, помогающие созданию нужной нам функции printf(). Так, библиотека динамической компоновки (БДК) USER.EXE содержит функции wprintf() и vprintf(). Подобно функции vsprintfO стандарта ANSI Си, wprintf() может применяться для создания функций, получающих, как и printf(), переменное число параметров. Поскольку функция vprintf() находится в БДК, ее использование позволяет уменьшить размер прикладной программы.

К сожалению, wprintf() не поддерживает некоторые полезные форматы вывода, например, %Pp. К тому же, на-

```

/* PRINTF.C — для упрощения вывода в небольших программах для
Windows;
использует функции Windows MessageBoxO или
WinExec()/SendMessage().

Copyright (c) 1991 Ziff Communication Co.
PC Magazine * Andrew Shulman
V

#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <windows.h>
#include "printf.h"

#define BUF_SIZE          1024

static char *str, *app;
static unsigned cap, len;
static int lines;

BOOL open_display(char *appname)
{
    app = appname;
    cap = 128;
    if (! (str = malloc(cap)))
        return FALSE;
    *str = lines = len = 0;
    return TRUE;
}

/* Максимальное число строк, которое будет содержать окно
сообщений */
static int max_lines(void)
{
    TEXTMETRIC tm;
    HWND hWnd = GetActiveWindow();
    HDC hDC = GetWindowDC(hWnd);
    if (hDC == NULL)
        return 0;
    GetTextMetrics(hDC, &tm);
    ReleaseDC(hWnd, hDC);
    return (GetSystemMetrics(SM_CYFULLSCREEN) /
            (tm.tmHeight + tm.tmExternalLeading)) - 5;
}

BOOL show_display(void)
{
    if (lines <= max_lines())
        MessageBox(NULL, str, app, MB_OK);
    else
        notepad(str);
    free(str);
    return TRUE;
}

static BOOL append(char *s2)
{
    char *s3;
    if (((len += strlen(s2)) < cap) && strcat(str, s2))
        return TRUE;
    cap = len + 128;
    if (! (s3 = malloc(cap)))
        return FALSE;
    strcpy(s3, str);
    Strcat(s3, s2);
    free(str);
    str = s3;
    return TRUE;
}

int nlinesfchar *s2)
{
    int c, n = 0;
    while (c = *s2++)
        if (c == '\n')
            n++;
    return n;
}

int printfconst char *fmt, ...)
{
    static char s2[BUF_SIZE];
    int len;
    va_list marker;
    va_start(marker, fmt);
    len = vsprintf(s2, fmt, marker);
    lines += nlines(s2);
    va_end(marker);
    append(s2);
    return len;
}

BOOL notepad(char far *s)
{
    HWND notepad;
    HWND edit_ctrl;
    if (WinExec("notepad.exe", SW_SHOWNORMAL) < 32)
        return FALSE;
    notepad = FindWindow(NULL, "Notepad - untitled");
    edit_ctrl = GetFocusO;
    SendMessage(notepad, WM_SETTEXT, 0, (char far *) app);
    SendMessage(edit_ctrl, WM_SETTEXT, 0, (char far *) s);
    return TRUE;
}

#ifdef TESTING
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
LPSTR lpszCmdLine, int nCmdShow)
{
    int i;
    open_display("GetSystemMetrics");
    for (i = 0; i < 37; i++)
        {
            printf("%d\t%d\r\n", i, GetSystemMetrics(i));
            YieldO;
        }
    show_display();
}
#endif

```

Рис^ В: Версия PRINTF.C для среды Windows

ходясь в БДК, она нуждается в передаче строк посредством дальних указателей. Поэтому в PRINTF.C я ограничился применением "обычной" функции vsprintf().

Теперь, перенося программу DEV.C в операционную среду

Windows, можно просто считать, что вы используете функцию под привычным названием printf (), которая, хотя получает на входе те же, что и ее, "обыкновенная" предшественница, параметры и возвращает то же

(обычно игнорируемое) значение, ухитряется каким-то образом осуществлять вывод текстовой информации средствами Windows.

Итак, писать простые программы для Windows все-таки возможно!

TRUENAME.C

ПОЛНЫЙ ЛИСТИНГ

```

/* truename.c */

#include <windows.h>
#include <stdlib.h>
#include <ctype.h>
#include "printf.h"
#include "dpmi.h"

#define fail(s) { MessageBox(NULL, s, "TRUENAME", MB_OK); exit(d); }

#define MAKEP(seg,ofs) ((void far*) MAKELONG((ofs), (seg)))

char far *truename(char far *s, char far *d)
{
    RMODE_CALL r;
    unsigned para, sel;
    char far *s2;

    /* INT 21h AH=60h "не любит" пробелов в начале или в конце строки */
    while (isspace(*s)) s++;
    s2 = s;
    while (*s2) s2++;
    s2--;
    while (isspace(*s2)) *s-- = 0;

    if (DosAllocRealSeg(256, &para, &sel) != 0)
        fail("Захват памяти реального режима невозможно");
    memset(&r, 0, sizeof(r));
    r.eax = 0x6000;
    r.ds = r.es = para;
    r.esi = 0;
    r.edi = 128;
    lstrcpy(MAKEP(sel, 0), s);
    if (! dpmi_rmode_intr(0x21, 0, 0, &r))
        fail("Не удалось войти в реальный режим DPMI не сработало");
    lstrcpy(d, (r.flags & 1) ? "Недопустимый" : (char far *)MAKEP(sel, 128));
    if (DosFreeRealSeg(sel) != 0)
        fail("Невозможно освободить память реального режима");
    return d;
}

int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
LPSTR lpszCmdShow, int nCmdShow)
{
    char buf[128];
    if (!(winflags = GetWinFlagsO) & WF_PMODE)
        return PA1Ц"Для этой программы требуется стандартный или
расширенный режим Windows");
    if (!(lpszCmdLine && "lpszCmdLine))
        fail("синтаксис: имя <имя_пути>");
    sprintf(buf, "TRUENAME %Fs", lpszCmdLine);
    opendisplay(buf);
    printf("%Fs\n", truename(lpszCmdLine, buf));
    show_display();
}

```

Рис. 8: Программа TRUENAME.C демонстрирует передачу буфера в обычной памяти прерываниям реального режима.

языке ассемблера. В исходном тексте функции `map_real()` на примере работы с `__0000h` продемонстрирован один из способов такого доступа из программы на языке Си. Аналогичным образом можно пользоваться и другими глобальными селекторами. Следует, однако, помнить, что в Windows имеются "ложные друзья программиста" — например недокументированный селектор `__0040h`, который отображает в защищенном режиме отнюдь не область данных

BIOS, а всего лишь `2FFh` байтов, начиная с адреса `0h`. Другой недокументированный селектор — `__ROMBIOS` — это просто "псевдоним" селектора `__F000h`.

WINDOWS И DPMI

Мы уже видели, что использовать системные ресурсы реального режима из исполняющихся в стандартном или расширенном режиме Windows программ можно или пользуясь про-

токолом DPMI, или обращаясь к недокументированным функциям самой этой оболочки. Теперь разберемся в сложных взаимоотношениях между протоколами DPMI и Windows API.

Понятие DPMI не упоминается в документации к Windows. Более того, там почти ничего не сказано ни о защищенном режиме, ни о том, что при работе Windows в стандартном или улучшенном режиме эта программная оболочка представляет из себя надстройку над DOS. На самом деле, в течение многих лет фирма Microsoft применяла программную эмуляцию защищенного режима в Windows, требуя от разработчиков больших усилий для ее поддержания. Теперь, когда в Windows 3.0 программы действительно исполняются в защищенном режиме, им почти не приходится брать на себя управление системными ресурсами. Но и этот существенный шаг вперед дипломатично назван в документации "усовершенствованным управлением памятью".

DPMI единственный раз прямо упомянут фирмой Microsoft в пятистраничном документе, озаглавленном "Поддержка многозадачной средой Windows INT 21h и NetBIOS для протокола DPMI" (этот документ включен в комплект замечаний о программировании для Windows под номером 050-030-313). Согласно содержащейся там весьма туманной информации, только некоторые из возможностей DPMI могут использоваться программы под управлением Windows.

В число поддерживаемых включены функции отыскания и изменения векторов прерываний реального режима, генерация прерываний в реальном режиме и исполнение машинных инструкций в реальном режиме.

Например, применение `INT 31h AX=0300h` (имитация прерывания реального режима) совместно с вызовом функции `Windows GlobalDosAUocO` позволяет пользоваться прерываниями, не поддерживаемыми в реальном режиме (см. короткую программу TRUENAME.C в конце этой статьи).

Если верить фирме Microsoft, программам для Windows не потребуются никакие функции DPMI, кроме перечисленных в документации, которая утверждает, что ядро (Kernel) операционной среды Windows содержит все необходимое для управления памятью.

К сожалению, многие разработчики обнаружили, что даже сочетая возможности ядра Windows с перечисленными фирмой Microsoft функциями DPMI, не удается получить все необходимое для работы в реальном режиме с резидентными программами, драйверами ЛВС и т.д.

Проблема здесь все та же: как из программы, исполняющейся в защищенном режиме, получить доступ к

абсолютному адресу реального режима, будь то область данных BIOS или начало цепочки драйверов устройств DOS. Даже при использовании недокументированного `___0000h` — и документированного `___A000h` селекторов недоступными остаются адреса, лежащие между `10000h` и `A0000h`.

Мы видели, что для преодоления этой трудности нужно либо применить недокументированные функции `Windows SetSelectorBase0` и `SetSelectorLimit0`, либо напрямую пользоваться протоколом DPMI. Возможность `Windows API` отображать произвольный UPP в адресное пространство программы, исполняющейся в защищенном режиме под управлением `Windows`, в документации не отражена. Поэтому при программировании для `Windows` протокол DPMI будет играть куда более заметную роль, чем та, которая отводится ему в кратком описании фирмы `Microsoft`.

ФУНКЦИЯ GLOBALDOSALLOCO

Эта функция будет крайне важна для разработчиков, нуждающихся в доступе к обычной памяти, а также в работе с резидентными программами и драйверами устройств. `GlobalDosAUoc0` резервирует область в обычной памяти, возвращая не только его параграф в реальном режиме, но и селектор защищенного режима.

Предположим, что вы пишете фрагмент программы поддержки ЛВС под `Windows`, используя, к примеру, стандарт `NetBIOS` или `Novell IPX/SPX`. Кроме обращения к функции `dpmi_rmode_intr()` (или прямого вызова функции имитации прерывания реального режима DPMI — `INT 31h AX=0300h`) придется зарезервировать буферы, необходимые для работы с драйвером ЛВС.

Например, обработчик прерывания `INT 5Ch`, которое используется для общения с `NetBIOS`, ожидает получить на входе в `ES:BX` указатель на блок контроля ЛВС (`network control block, NCB`). При обращении к `INT 5Ch` из защищенного режима через посредство DPMI вы все равно должны убедиться в том, что (а) `NCB` находится в обычной памяти, и (б) `ES:BX` имеет соответствующий UPP. С другой стороны, манипулировать этим `NCB` придется с помощью UPP3P.

Теперь понятно, почему `GlobalDosAUoc0`, резервирующая область в обычной памяти и возвращающая одновременно и ее параграф в реальном режиме, и UPP3P, — это как раз та функция, которая совместно с вызовом `dpmi_rmode_intr()` должна использоваться для работы с драйвером ЛВС. Поскольку мне показался не слишком удобным способ, при помощи которого `GlobalDosAUoc0` возвращает значения, я написал на ее основе свою функцию `DosAllocRealSeg0`, имя которой, как

и имя `DosMapRealSeg0`, позаимствовано из стандарта `Phar Lap API`.

Функция `DosAllocRealSeg0`, вызываемая, в свою очередь, `GlobalDosAUoc0` и `dpmi_rmode_intr()`, применена в программе `TRUENAME.C` (рис. 8) — последней из программ, обсуждаемых в этой статье. Так как многие читатели не имеют в своем распоряжении `NetBIOS`, я снова использовал в `TRUENAME.C` недокументированный вызов DOS, который, во-первых, доступен на каждом компьютере, работающем под управлением DOS, а во-вторых, не поддерживается напрямую операционной средой `Windows` в защищенном режиме.

Программа `TRUENAME` получает из командной строки (которая в `Windows` может вводиться с помощью команды `Run` администратора программ `PM`) путевое имя. Затем, используя вызов `INT 21h AH=60h`, она сообщает истинное путевое имя, раскрывая "псевдонимы", вводимые такими программами DOS, как `ASSIGN`, `SUBST`, а также драйверами ЛВС. К примеру, на моем персональном компьютере накопитель `F:` на самом деле находится на рабочей станции типа `Sun SPARCstation`, поэтому при указании в командной строке `F:` программа `TRUENAME` выдаст полное имя

`\\EXPORT\DOS\A_ANDREW.`

`TRUENAME.C` — хорошая иллюстрация совместного использования функций `GlobalDosAUoc0` и `dpmi_rmode_intr()`, потому что функция `DOS 60h` предполагает, что в качестве параметра будет задано исходное путевое имя в `DS:SI` (например, `"F:"`), а в `ES:DI` — указатель на 128-байт буфер, куда будет помещена строка с истинным именем (в моем примере — `\\EXPORT\DOS\A_ANDREW.`). Перед вызовом функции `DOS` программа должна зарезервировать обычную память под буфер передачи, скопировать в него командную строку, поместить адрес реального режима этого буфера в `DS:SI`, а в `ES:DI` поместить адрес также предварительно зарезервированного буфера для приема информации. После возврата из `dpmi_rmode_intr()` (в которой происходит обращение к `INT 21h AH=60h`) остается лишь скопировать результат из буфера приема, используя UPP3P на этот буфер.

УСПЕШНЫЙ КОМПРОМИСС

Существует множество способов использования протокола DPMI программами, исполняющимися в среде `Windows`. Не рассмотренные в этой статье функции DPMI позволяют программам из защищенного режима перехватывать векторы прерываний реального режима, расширяют воз-

можности отладки (обеспечивая доступ к Глобальной Таблице Дескрипторов (ГТД)), а кроме того, позволяют обрабатывать некоторые особые состояния МП (в частности, состоящие ошибки по нарушению защиты).

Возможность самому обрабатывать ошибочное состояние типа ПНЗ достаточно интересна. Действительно, ранее я рассказывал о сообщениях "Попытка нарушения защиты" и "Критическая ошибка в программе" как о неизменных реалиях защищенного режима. На самом деле ошибочное состояние ПНЗ — это всего-навсего прерывание `ODh`. Отличая особые состояния МП от прерываний, DPMI имеет функции для обработки ошибочных состояний, подобных ПНЗ.

Установив с помощью функций DPMI "Изменить вектор обработчика особых состояний МП" (`INT 31h AX=0203h`) и "Найти вектор обработчика особых состояний МП" (`INT 31h AX=0202h`) свой обработчик ПНЗ, вы сможете избежать раздражающего сообщения `Windows`, которое мы видели в начале статьи. Ваш обработчик может вежливо сообщить конечному пользователю о том, что из-за ошибки в исполняющейся программе ей отказано в системных ресурсах. Это все же лучше, чем внезапно ошеломить пользователя сообщением "Критическая ошибка в программе"!

После описания сложностей работы в защищенном режиме `Windows` уместно в заключение напомнить о грандиозных преимуществах, открывающихся перед программами после разрушения "берлинской стены" на границе первого мегабайта памяти. Эти преимущества намного перевешивают некоторые неудобства, связанные с неожиданной невозможностью достичь адреса `400h` с помощью указателя, подобного `0000:0400`. Как бы то ни было, в этой статье показано, что вы все же можете получить доступ к адресу `400h`, но не посредством UPP, а с помощью UPP3P, возвращаемого функцией `map_real()`. При этом `map_real0` внутри себя может использовать вызовы DPMI, в возможном сочетании с обращениями к недокументированным функциям `Windows`.

Проблема доступа из защищенного режима к системным ресурсам низкого уровня отражает переход программирования для ПЭВМ на новую, более высокую ступень. Мы преодолели барьер в 640 Кбайт, но при этом не лишились возможности использовать накопленное программное обеспечение, ориентированное на работу под управлением DOS. Решение этой проблемы, продемонстрированное в этой и предыдущих статьях, представляет собой успешный компромисс между DOS и защищенным режимом. Все больше программ исполняются в защищенном режиме, но при этом не наносится ущерб рынку программных изделий для DOS. И это прекрасно!

Программирование с человеческим лицом

Рубен Герр

АНИМИЗМ — ...представление о том, что за каждым предметом природы скрывается невидимый дух, который управляет предметом.

"Краткий философский словарь", Госполитиздат, S.I., 1954

Можно считать, что это пережитки каменного века в сознании. Можно рассуждать о дефиците человеческого общения. Можно сказать, что это просто способ выразиться. Явление существует. Моряки "одушевляют" свои корабли, шоферы говорят о своих машинах как о живых существах — не исключение и компьютеры. Здесь даже внешние атрибуты совпадают: есть материальная оболочка (кремний, пластик, стекло) и есть нечто неосознаваемое, но наличествующее — программы (правда, увы, отнюдь не бессмертные...). Помните, давно ли вы обращались к коллеге со словами типа: "Не понимаю, чего от меня хочет?" или возгласами "ну давай же!" поторапливали медлительную программу?

Но душу в компьютер вложил отнюдь не безгрешный Творец. Когда-то на заре компьютеризации автору этих строк пришлось слышать примерно такие рассуждения: "Я предпочитаю работать с программами, составленными А., они так *t £* — работают! Программы Б., конечно, дают результаты быстрее, но он стремится использовать все ресурсы машины, и на его программах компьютер чаще сбивается". (Компьютер, о

котором шла речь, назывался БЭСМ-4.) Немало перфоленты утело с тех пор в мусорные корзины, изменилось многое, но не все. По-прежнему мы говорим, что компьютер или программа как-то "себя ведет", по-прежнему неопытные программисты перезапускают программы по нескольку раз в тщетной надежде на то, что причина неверной работы — сбой, вместо того чтобы попытаться найти свою очередную ошибку. А компьютеры по-прежнему делают только то, что предписано программами.

Поводом для вышеприведенных рассуждений послужило одно крупное разочарование. Многие советские пользователи персональных компьютеров с нетерпением ожидали визита в нашу страну Питера Нортон, легендарного программиста, автора без преувеличения самой популярной в СССР программы — Norton Commander. И свершилось! На выставке Комтек — 91 мы наконец-то увидели этого человека. Были выступления на пресс-конференциях, были интервью, были неформальные встречи с программистами. Но что-то во всем этом поражало: как-то не соответствовало поведение Нортон сложившимся у нас представлениям — не так и не о том говорят программисты. И вот что в конце концов выяснилось. Питер Нортон уже давно не занимается активным программированием. Причина его славы в том, что он реализовал одну превосходную идею — "персонализировать" программу, постараться сделать так, чтобы пользователи чувст-

вовали, что за программой стоит живой человек, который думает и заботится о них. И если в программе что-то неверно, это тоже так по-человечески. А основной автор программы на самом деле некий Джон Соша, о котором кроме имени практически ничего не известно.

Оказалось, такой подход действительно ведет к успеху — несмотря на то, что по данным сравнительных анализов оболочка Norton Commander хотя и входит в список лидеров, но отнюдь не возглавляет его, именно эта программа *de facto* завоевала "приз пользовательских симпатий". Справедливости ради следует отметить, что фирма Symantec, которая не так давно поглотила фирму Peter Norton Computing, постаралась сохранить традицию. В Москве было объявлено, что со всех продаж своих изделий в СССР фирма будет отчислять определенные суммы на приобретение одно-

разовых шприцев для одного из фондов борьбы со СПИДом.

Обидно, конечно, расставаться с легендой, но и на пользу. Во взаимоотношениях человека с компьютером прояснился еще один аспект. Если борьба за материализм в компьютерной области заведомо обречена на поражение, почему бы не воспользоваться нормальными человеческими слабостями? Вместе с тем мало дать программе хорошее имя. Программа должна действительно учитывать реальные потребности и по возможности "защищать" человека от ошибок (вспомним назойливые на первый взгляд предупреждения программы Norton о том, что вы вот-вот уничтожите файл).

Пользователь компьютера совсем не всегда таков, каким его представляет себе автор программы. Именно поэтому отнюдь не редкость ситуация, когда удобную в работе и



действительно нужную программу составляет не профессионал или группа профессионалов, а рядовой пользователь, который поначалу делал "инструмент" для себя.

Автор программы, о которой ниже пойдет речь (по известным одному ему причинам он попросил не называть свою фамилию), по образованию химик. Между прочим, из химиков очень часто получаются прекрасные программисты. Быть может, дело в стиле мышления? Потребность в программе была продиктована типично советским явлением — дефицитом. В данном случае это был дефицит дискет. Если в других странах цена дискеты сравнима с ценой пачки сигарет, и человек может в любой момент купить ее "за углом", а фирмы снабжают своих сотрудников дискетами так же, как, например, бумагой и карандашами, у нас пока все не так. Следует учесть и тот факт, что приобретенные официально программы "у них" поставляются в снабженных отчетливыми надписями папках с документацией, которые можно красиво расставить по полкам. А в каком виде распространяются программы у нас, можно не рассказывать...

Но необходимость откладывать "в долгий ящик" и редко используемые программы, и документы, к которым когда-нибудь может возникнуть потребность вернуться, и какие-то иные данные никогда не отпадает. Для этих целей естественно воспользоваться программами архивации, которые сокращают объем хранимой на диске информации в 1,5 — 2 раза, но когда дискет с архивами не один десяток и на каждой до сотни файлов, отыскание нужной информации само по себе становится работой. Можно, конечно, завести каталог традиционного вида, на бумаге, но зачем, если эту работу можно тоже возложить на компьютер? Так на свет появилась первая версия программы "Дискотека",

которая первоначально была составлена (содрогнитесь, профессионалы!) на языке GWBasic.

Вот так "погибают" лучшие люди — автор "Дискотеки" волей-неволей начал интересоваться всеми смежными вопросами: как происходит архивация, как из программы перемещаться по каталогам и подкаталогам, можно ли заносить "свою" информацию в сектор начальной загрузки дискеты. Те, на ком автор "опробовал" свое создание, дали ему множество полезных советов. Увы, совсем не редкость, когда в ответ на какие-то соображения по улучшению программы автор начинает пренебрежительно разъяснять, как можно добиться нужного результата уже имеющимися средствами, или почему то, чего потребовал пользователь, на самом деле ему совсем не нужно. Автор "Дискотеки" в этом смысле составляет приятное исключение.

Что же такое "Дискотека"? Посмотрим, как работает программа с точки зрения пользователя (а кто мы с вами такие?). Сначала самые обычные операции: упаковка (с помощью той программы-архиватора, которую вы предпочитаете) и перенос архива или архивов на дискету. Самое интересное начинается дальше. Установив в накопитель дискету с архивом, вызываем "Дискотеку". Программа автоматически занесет всю информацию о новой дискете в свои файлы, присвоит дискете номер, запишет его на саму эту дискету и напомнит пользователю, чтобы он не забыл нанести этот номер и на ярлык. Если дискета уже зарегистрирована как архивная, но вы изменили ее содержание, программа распознает это и выполнит все необходимые операции. На жестком диске хранятся каталоги дискет, архивов, файлов в архивах. Очень удобно организован переход с уровня на уровень каталогов: если, например, вы подвели курсор к названию некоторого файла в

каталоге файлов, а затем перешли на уровень архива дискет, курсор окажется против номера той дискеты, на которой хранится ваш файл. Каталоги дискет и архивов можно снабжать комментариями. При просмотре каталогов предусмотрены все варианты сортировки, которые показали нужными пользователям: по именам, по размеру, по дате. Есть, разумеется, и функция быстрого поиска по шаблону. Одно из последних нововведений: программа позволяет искать дискету с заданным объемом свободного места. Кроме того, предусмотрена функция выявления дубликатов — хранящихся в разных архивах одинаковых файлов. Словом, это превосходный пример программы для пользователей

Когда писалась эта заметка, "Дискотека" распространялась среди друзей и знакомых свободно. Автор поначалу и не намеревался заниматься коммерцией, но жизнь заставила. Во-первых, с расширением круга пользователей он постепенно из хозяина своего изделия стал превращаться в его раба — ведь всякая сколько-нибудь серьезная программа нуждается в сопровождении. Во-вторых, все время появляются новые архиваторы, с ними нужно разбираться, подключать их к программе, а это новая работа. Поэтому теперь коммерческим распространением и сопровождением программы "Дискотека" занимается группа "Неософт" при кооперативном объединении "Никана", контактные телефоны (095)135-64-93; (095)135-93-31.

Изделия научно-технической и коммерческой фирмы *SoftScribe, Ltd.* сохраняют ваши усилия и время, если вы создаете информационные системы для IBM-совместимых персональных компьютеров на основе Clipper Summer'87 или Clipper 5.0

Вашими лучшими помощниками станут:

- Fast Pro** — технология и инструментальный пакет для быстрой разработки систем;
- Fast Graph** — библиотека функций деловой графики для Clipper и языка программирования Си;
- Fast Tools** — совместимый с Clipper 5.0 и обладающий более широкими возможностями аналог библиотек Clipper Tools и Nantucket Tools Two;
- Fast Install** — универсальный инсталлятор для установки программных продуктов с возможностью защиты от несанкционированного копирования библиотек объектных модулей и загрузочных файлов;
- MAГ** — библиотека экранного редактора баз данных.

Для получения справок или каталога продуктов обращайтесь непосредственно в фирму *SoftScribe, Ltd.*!

Адрес: СССР, 198328, Ленинград, а/я 76
Телефон: (812) 210-4601
Телекс: 121345
Факс: (812) 272-7268



Билл Макроун

Забудьте о существовании DOS. Забудьте OS/2. Забудьте о LAN Manager. Забудьте NetWare и Unix. Но не забывайте о Windows. Вот чего потребует от вас Microsoft на ближайшие месяцы и годы. Далеко идущие планы, утвержденные на состоявшемся в начале этого года совещании по стратегии развития фирмы, одновременно тревожат, обнадеживают и вдохновляют.

Меня беспокоит факт бесперерывного списания OS/2, которая по существу стала собственной операционной системой фирмы IBM, используемой для обеспечения связи клиент — сервер и микро- — большой компьютер. Я убедился в том, что Microsoft творчески подходит к будущему DOS, чего нельзя было сказать еще год назад, когда предполагалось, что новые версии этой операционной системы будут представлять собой не более чем освобожденные от недочетов и ошибок копии текущей. Меня затронули за живое дерзкие планы, характерные для современной политики фирмы Microsoft.

Мы знаем, что нас ожидает в ближайшем будущем: DOS 5 и Windows 3.1 увидят свет совсем скоро, новая же версия OS/2 — никогда.

Точнее это не совсем так. Дело в том, что новая версия OS/2 будет лишь отдаленно напоминать своих предшественниц. OS/2 3.0 будет носить черты ядра "New Technology" (NT): полная поддержка Windows, мобильность, распределенная обработка и поддержка POSIX (стандартизированная версия Unix). Группу разработчиков возглавляет Дейв Катлер, человек, который в прошлом отвечал за операционную систему VMS в фирме DEC.

Система OS/2 3.0 будет маленьким ядром, с многочисленными функциями интерфейса прикладных программ. Эта система на 99% будет составлена на Си, что обеспечит ее мобильность. Архитектура системы NT должна соответствовать стандарту C2 США и обеспечивать уровень безопасности В в многопользовательском режиме работы. Этого достаточно. Кроме того, будет обеспечена строгая защита от ошибок с регулярным резервированием содержимого дисковой памяти, а также с самовосстанавливающейся файловой системой, на основе принципа транзакций. Будут предусмотрены встроенные средства сетевой поддержки. Это означает, что вам не потребуется файловый сервер, и вы сможете просто соединить 2 или более машины, чтобы они сами "нашли друг друга" и начали обмен данными.

Выполнение задач будет распределенным, причем физическое расположение не будет играть роли ни для программ, ни для данных. NT — это также симметричная мультипроцессорная система, а это означает, что процессы (thread) будут выполняться одновременно. В соответствии с заявлениями представителей Microsoft, система уже отлажена на машинах с процессорами 386/486, ведутся работы по ее переносу на RISC-процессоры разных типов. Учитывая заметную в последнее время активность фирмы Compaq, можно сделать вывод, что один из них — процессор фирмы MIPS.

Если считать микропроцессор мини-компьютером 90-х, то OS/2 3.0 — это его операционная система, по крайней мере на это надеется Microsoft. Из предварительного описания Microsoft следует, что OS/2 3.0 будет королевой операционных систем. Охватив в высокопроизводительном ядре возможности персональных компьютеров, рабочих станций, серверов, сетевых станций и станций, базирующихся на Unix, она, похоже, станет операционной системой, от которой не сможет отказаться никто.

СОЕДИНЕНИЕ DOS И NOVELL

Фирма Microsoft исходит из того, что сетевая модель клиент-сервер не имеет будущего, в лучшем случае она будет служить для взаимобмен в рамках модели равный к равному.

Если фирме удастся этого добиться, не останется никаких оснований для покупки NetWare, LAN Manager или любой другой операционной системы для обслуживания модели клиент-сервер. Такой взгляд распространяется и на DOS, и вот почему: в среднем сеть Novell имеет лишь от 7 до 10



ILLUSTRATION: KEN CONDON

пользователей. Среди читателей PC Magazine средняя цифра достигает 13, но и это ничуть не похоже на локальные сети с двумя — тремя сотнями узлов, которые пытаются внедрить производители сетевого оборудования.

(окончание см. стр. 74)

Новая ОС короля

Компания Microsoft более не ограничивается господством в мире операционных систем. Она хочет прибрать к рукам все: персональные компьютеры, серверы, мощные рабочие станции, портативные модели и даже то, что еще не изобретено.



Вы, конечно, догадались, что речь идет о семействе самых популярных в мире персональных компьютеров Macintosh фирмы Apple.

Знаете ли Вы что-нибудь о компьютерах? Умеете ли на них работать? Владете ли английским языком?

Компьютеру Macintosh все равно, как Вы ответили на эти вопросы. Он Ваш. Он придуман и сделан специально для Вас. Он „свой в доску“, недаром все, кто работает на нем, зовут его по-приятельски — „Мак“.

Мак — самый демократичный!

Мак — самый умный!

Мак — самый добрый!

Он обязательно станет для Вас и самым любимым компьютером! Пальцы, хоть однажды коснувшиеся клавиатуры Мака, не смогут больше притронуться ни к одной другой. Даже если Вы новичок, не отчаивайтесь: с Маком Вы подружитесь в 3 раза быстрее, чем с любым другим компьютером.

Каждый представитель знаменитого семейства Macintosh обладает яркой индивидуальностью, от „карманных“ Маков до самых мощных, которые „заткнут за пояс“ даже AT 486. Хотите убедиться? Обращайтесь в СП ИНТЕРМИКРО по адресу:

107066, Москва, ул. Нижняя Красносельская, 39. Телекс: 411035 FOTON SU.
Телефакс: (095) 200-22-38. Телефоны: (095) 267-32-10, (095) 261-04-47.

По этим же телефонам Вы узнаете адрес Вашего регионального дилера.



Джим Симур

За последнее время в мире Macintosh появилось немало интересных разработок. Некоторые из них — возможно, совсем не те, о которых вы подумали, — могут серьезно сказаться и на тех из нас, кто работает в среде DOS.

Наиболее заметным событием был успех нового изделия фирмы Apple из семейства Macintosh — Mac Classic, появившегося в продаже перед самым Рождеством. Точные цифры назвать трудно, но Classic пользовался сногшибательным успехом. По цене 995 и 1495 долл., за модели только с гибкими и с жестким диском соответственно, Classic завоевал неудержимую популярность, которая так долго ускользала от фирмы Apple.

Ажиотаж вокруг компьютера Classic, безусловно, на несколько кварталов стабилизирует финансовое положение Apple, однако длительная жизнеспособность фирмы на рынке персональных компьютеров все еще вызывает определенные сомнения. Более 60 миллионов компьютеров PC против 4 миллионов машин Macintosh дают кварталов стабильно IBM-совместимым компьютерам, что не может не сказаться на перспективном планировании деятельности разработчиков программ.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Вопрос о том, какую из платформ поддерживать, чрезвычайно актуален для разработчиков программного обеспечения. В течение нескольких последних лет они получали двойную выгоду, маневрируя среди возможностей DOS, Macintosh, Xenix/Unix, OS/2 и OS/2 Presentation Manager. И, конечно, существует более конкретный выбор в рамках этих крупных категорий: например рынок Windows под DOS и рынок Unix, который, как утверждают специалисты, неуклонно движется ко всеобщей совместимости (т.е. к работе одной версии программы на всех компьютерах для Unix), но никак к ней не придет.

Это трудная задача. Возможности для разработок всегда ограничены. Ни одна из известных мне компаний не располагает достаточным штатом программистов для того, чтобы представить свои изделия на всех существующих рынках.

Таким образом, остается лишь опуститься до простого гадания, какие цветы расцветут, а какие увянут на корню. При

этом ошибка может быть чревата весьма серьезными последствиями.

Спросите, например, у Фреда Гиббонса из Software Publishing Corporation. В прошлом году, в разгар шумной рекламной кампании, предвещающей выход версии Windows 3.0, Фред с крайним недовольством говорил о намерениях Microsoft. Он отмечал, что Microsoft обвела и его, и множество других фирм вокруг пальца, убеждая их не обращать внимания на Windows и разрабатывать высококлассное программное обеспечение для OS/2. Но теперь, похоже, что Microsoft начинает делать ставку на Windows. Так оно и произошло.

Или спросите у Джима Мэнджи из фирмы Lotus. Он потратил целое состояние на разработку 1-2-3/G для OS/2 PM. Программа вышла, но тут же практически исчезла. Никто не пользовался OS/2 PM и соответственно никого не интересовали прикладные программы, работающие с ней.

В этой ситуации проще отказаться от поддержки платформы Macintosh. Зачем, шить одежду для призрака?

Таким образом вывод можно сформулировать так: необходимо иметь версии ваших программ и для DOS, и для Windows; за Unix — будущее; неплохо сделать ставку на Microsoft "NT" (New Technology); старая же OS/2 с оболочкой PM уходит в безвозвратное прошлое.

Ну а как быть с семейством Macintosh? Забудьте о нем. Даже разработчики программного обеспечения для Macintosh в последнее время испытывают тревогу. Рассказы о ведущих программистах, работавших на рынок Macintosh и переметнувшихся в лагерь DOS/Windows, еженедельно потрясают Купертино. И уж совсем невероятно то, что Claris, дочерняя фирма компании Apple, занимающаяся разработкой программных изделий, предлагает свои услуги в программировании для Windows.

В религиозной войне легко ошибиться в предсказаниях, получая правильные факты, но делая неверные выводы. То же случилось и в священной (если такие вообще бывают) войне разработчиков программного обеспечения для Macintosh и PC.

Наиболее точную информацию об этом я получил несколько недель назад, когда разговаривал с авторами одной широко известной программы, которая первоначально появилась на машинах Macintosh, но получила распространение и принесла доходы своим хозяевам в версии для Windows. Я спросил, планируют ли они на время заморозить производство версии для Macintosh, сосредоточив свое внимание на совершенствовании версии для PC.

Вести из мира Macintosh

Невидимая рука рынка заставляет всех пользователей PC призадуматься над новыми достижениями в лагере Macintosh.

КОРОЛИ, КАПУСТА И...
КОМПЬЮТЕРЫ

Их ответ меня несколько не удивил: "Нам нужна продукция для Macintosh, чтобы стимулировать продажи версии для Windows". До чего же это верно!

ДРУГАЯ ПЛАТФОРМА

В наш век всеобщего взаимодействия, при общей тенденции к широкой поддержке множества платформ, фирмы — производители программных изделий, которые хотят предложить качественную и пользующуюся длительным успехом продукцию, приходят к пониманию того, что у них нет иного выбора, как поддержка лагеря Macintosh, — даже в том случае, если это не принесет им желаемого дохода.

Корпорации-покупатели дают понять, что при выборе Windows в качестве операционной среды и использовании какой-либо прикладной программы как принятого во всей компании стандарта им необходимо, чтобы

данная программа могла служить и пользователям, стоящим на другой платформе. И, безусловно, речь в первую очередь идет о пользователях Macintosh. Таким образом, даже небольшое число пользователей не может быть оставлено без внимания.

Компании Aldus и Microsoft прекрасно понимают это требование. То же можно сказать и о WordPerfect Software Corp. Не отстают и Lotus. (Как вы думаете, почему они возобновили работу над 1-2-3/Mac? Конечно, не потому, что рынок Macintosh сулил им большие прибыли.)

Другие разработчики пытаются доказать, что их это несколько не пугает, и бросаются смелыми заявлениями о том, что лишь временно игнорируют существование пользователей Macintosh. Однако на самом деле

и им не чужды переживания на этот счет и опасения, что они пиллят сук, на котором сидят.

Таким образом, в данной ситуации для фирмы Apple главная новость — не неожиданные доходы от продаж модели Classic, а невидимая рука рынка — внимание его участников.

Что же это значит для вас, пользователей PC, которым Macintosh глубоко безразличен. С одной стороны, теперь вы можете сказать производителям программных изделий, чья продукция кажется вам надежной, о том, что вас интересуют и версии для Macintosh. Это нужно для того, чтобы их со временем не снесло волной всеобщего взаимодействия.

С другой стороны, мы должны следить за тем, что в целом случается в мире персональных компьютеров, ведь события, происходящие в нем, и свойственные ему тенденции значительно влияют на наш уютный мир DOS.

(начало см. стр 71)

Принимая во внимание сложность NetWare 286 и NetWare 386, соединение в сеть такого небольшого числа машин напоминает стрельбу из пушки по воробьям. Простая сеть модели равный к равному, как, например, LANtastic, более чем удобна и экономнее расходует системные ресурсы, "забирая" лишь 10 Кбайт памяти у рядовой машины и 40 Кбайт у машины, играющей роль сервера. Примите в расчет и будущие версии DOS, так как версия 6.0, возможно, будет иметь встроенную сетевую поддержку модели равный к равному.

Положение фирмы Novell было бы рискованным, если бы у нее был серьезный конкурент. Производители локальных сетей низкого класса просто не в состоянии устанавливать промышленные стандарты. В течение нескольких лет Microsoft пыталась "выбиться в люди" с системой LAN Manager. IBM чересчур замкнута в рамках своей модели, чтобы устанавливать стандарты на сетевую поддержку персональных компьютеров, а Banyan просто не принимает во всем этом участия. Однако Novell постоянно соблазняет та же пресловутая цифра, вызывающая довольную улыбку на лице Билла Гейтса: 60 миллионов компьютеров, работающих

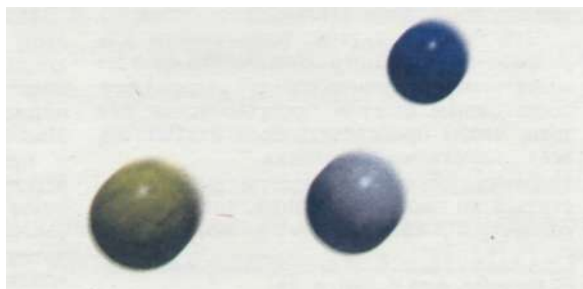
с DOS. Число клиентов всегда будет превышать число серверов, и, если архитектура модели клиент-сервер отстанет от модели равный к равному, Novell со временем растает как леденец. Поэтому происходит зондирование почвы на предмет интереса к предложенной Novell совместимой с DOS операционной системе для персональных компьютеров. Резонно предположить, что сетевая поддержка будет встроенной и скорее всего будет строиться по принципу равный к равному. Пока реакция была незначительной, тем не менее это реально.

На месте президента фирмы Novell Рея Нурда я бы завязал деловые контакты с фирмой Digital Research. Существует мнение, что во многих отношениях DR-DOS превосходит MS-DOS (правда, о Microsoft никак нельзя сказать, что эта фирма стоит на месте). Я бы построил сети для своих клиентов по принципу равный к равному, уделив особое внимание файловым серверам NetWare, входящим в ту же сеть. Торговля пойдет неважно, так как Microsoft имеет огромное преимущество в отношении продаж на уровне системотехнических фирм. Цены должны быть очень низкими, ниже 100 долл.

А пока Microsoft ждет своего часа, чтобы выступить с новой версией

DOS, снабженной устанавливаемой файловой системой и вышеупомянутой сетевой поддержкой модели равный к равному, а также с 32-разрядной версией Windows. Она будет дополнена такими возможностями OS/2, как вытесняющая многозадачность и многочисленные конкурентные процессы. Кроме того, здесь будет несементированное 2-Гбайт адресное пространство, а преимущества архитектуры 80386 будут реализованы в полной мере. Каждая задача занимает свой защищенный сегмент памяти. Каждая задача DOS также помещается в отдельном пространстве памяти. Таким образом, система должна быть "ошибкоустойчивой" и не столь остро реагировать на "проколы" в прикладных программах, как Windows 3.0. В действительности, единственные прикладные программы, которые будут делить общее место в памяти, — это существующие уже сегодня 16-разрядные программы под Windows. Где-нибудь в 1992 г. мы увидим Windows 4.0.

Никто всерьез не подвергает сомнению первенство Microsoft в области персональных компьютеров, однако, планы фирмы в отношении мощных систем высшего класса на редкость смелы. Посмотрим, что из этого получится.





Джон Дворак

Все течет, все изменяется.
Гераклит

Куда подевались любители?

Рост производства персональных компьютеров и быстрое распространение научно-технических достижений оставили любителей не удел.

Когда микрокомпьютерная революция только начиналась, в первых рядах шли любители, т.е. люди, считавшие, что компьютеры — это лишь модное развлечение. Именно они стали ядром новой популяции, именно их искания рождали новые направления и двигали вперед события. Более того, кто как не любители породили идею настольных компьютерных систем: ведь это им был присущ сугубо индивидуальный стиль работы, это они ликовали при каждом снижении цен, да и просто своим существованием привлекали внимание к проблеме. В наши дни не существовало бы журналов, подобных PC Magazine, да и вообще компьютерного рынка в его современном виде, если бы тогда, в 70-е годы, эти люди активно не пропагандировали компьютеры как вид хобби.

Но что же в конце концов случилось с любителями? Неужели это было лишь кратковременное увлечение, модная новинка и все?

Осталось лишь несколько пионеров, которые и по сей день считают компьютеры своим хобби. Некоторые сделали компьютеры своей профессией, стали скучными бизнесменами. Кое-кто вовсе отказался от своего увлечения. Где сейчас Дон Тарбелл, первый человек, выпустивший в продажу контроллер жесткого диска? Где Брюс Силе? А ведь в 70-е годы он был одним из первых продавцов плат расширения памяти!

Впервые я обратил внимание на исчезновение любителей, когда несколько лет назад заметил, что выражение "компьютерные вдовы" как-то вышло из употребления. Похоже, эта беда больше не обрушивается на наших женщин.

Можно отметить четыре причины отмирания компьютерного хобби:

1) Бизнес открыл для себя компьютеры. Как только американские бизнесмены узнали о существовании компьютера как выгодного инструмента для организации бизнеса, поставщики, выпускавшие продукцию для любителей, осознали возможность получения сверхприбылей за счет бизнесменов. Они отвернулись от любителей. Журналы, до этого момента ориенти-

ровавшиеся на любителей, выбрали тот же путь.

2) Машины стали слишком сложными. Трудно назвать хобби сборку компьютера, когда все "упростилось" до одной кремниевой микросхемы, выполняющей все функции. Хотя компьютер состоит более чем из одной микросхемы, появление СБИС свело число микросхем к минимуму. Более того, современная технология лишает любителя даже удовольствия самому припасть что-нибудь.

3) Темпы перемен были слишком высокими. Только избранные могут угнаться за ежегодными переменами и соответственно ростом затрат на такое хобби. Это нельзя сравнить с коллекционированием старых автомобилей и редких почтовых марок. В большинстве хобби всегда можно сделать передышку. В увлечении же компьютерами быстрый прогресс и вытекающие отсюда затраты на то, чтобы постоянно идти в ногу со временем, неприемлемы для большинства пользователей. В результате любители ограничились общением со старыми, но еще работающими машинами, которые уже не имеет смысла совершенствовать.

4) Сейчас компьютеры не увлечение, а работа. Радость открытий прошла. Любители черпали жизненные силы из программирования на Бейсике. Они играли со всеми новыми общедоступными программами. Теперь, если вы программируете, то обязательно на Си++ или объектно-ориентированном Паскале, а освоение этих языков требует кропотливой работы. Более того, на нас свалилось такое разнообразие условно-бесплатных и просто бесплатных программ, что сделать что-то новое и достойное внимания забавляясь стало просто невозможно. Компьютерное хобби теперь невозможно совмещать с какой-либо другой работой.

Однако в старых компьютерах и сейчас есть нечто захватывающее, и я полагаю, что хобби, о котором идет речь, возродится в новой форме, как это случилось с коллекционированием автомобилей. О старых компьютерах уже вспоминают. Недавно я был в Музее истории США в Вашингтоне, и представьте мое удивление, когда среди экспозиции я увидел старые компьютеры.

Конечно же те любители компьютеров, которых мы знали в 70-е годы, остались в прошлом. Но мы не должны забывать об их вкладе в общее дело. Если вдруг они, подобно коллекционерам старых колымаг, объявятся снова, отнесите к ним по-хорошему и отдайте машины, собирающие пыль в чулане.



Уильям Ф. Закман

Какая из операционных систем станет победительницей в 90-х годах? С таким же успехом можно задать вопрос, какая из бейсбольных команд выиграет мировое первенство в 1991 г. Действительно, пока еще невозможно назвать ту команду, которая займет первое место, но в то же время мы знаем наверняка, что тот или иной спортивный клуб обязательно победит в чемпионате. Проводя аналогию, можно утверждать, что среди таких программных систем, как Windows, OS/2, Unix и др., одна ОС в конечном счете добьется триумфа. Представляется вполне очевидным, что в 90-х годах одна из соперничающих операционных систем займет те ведущие позиции, которые в 80-е годы были за MS-DOS.

Однако, сколь бы убедительным ни казалось подобное предположение, это не единственное возможное развитие событий,

и более того, по-видимому, даже не наиболее вероятный поворот. Не исключено, что сама идея превосходства одной ОС над другими является ошибочной. А если так, то попытки определить победителя не только преждевременны, но и просто ошибочны. Если в будущем нас ждет "плюрализм" операционных систем, то требовать от всех сотрудников той или иной организации приверженности одной ОС было бы неправильно.

Перспектива лидерства какой-то одной ОС привлекает, конечно, простотой ситуации. Такое положение вызовет особый энтузиазм у вечно загруженных и задержанных сотрудников служб технической поддержки, которые, понятно, предпочтут иметь дело с одной, а не с несколькими ОС одновременно.

Тем не менее вовсе не очевидно, что подход "один размер для всех", ведущий к выбору в качестве стандартной — для конкретного предприятия — какой-либо одной операционной системы, в наибольшей степени отвечает его интересам. Ориентация на использование нескольких ОС не только лучше согласуется с реалиями рынка, но и позволяет максимально полно удовлетворить индивидуальные запросы служащих предприятия.

Действительно, вполне вероятно, что в большинстве организаций, независимо от их размера, достойное место займут практически все (а может быть, и все) ОС, борющихся за лидерство в 90-х годах. DOS, различные версии "DOS plus", Windows, OS/2 и Unix, а также операционная среда компьютера Macintosh, все эти системы в ближайшее десятилетие так или иначе

найдут применение — по крайней мере в большинстве достаточно крупных организаций. Однако и малый бизнес выиграет от ориентации более чем на одну операционную систему. Каждая ОС имеет свои достоинства и недостатки и каждая может вносить свой вклад, решая те или иные задачи на предприятии.

Несмотря на заметно растущий в последнее время интерес к новым операционным средам, DOS еще долго будет играть важную роль — может быть, на протяжении всех 90-х годов. Начать с того, что фирма Microsoft продолжает совершенствовать MS-DOS. Подобные усовершенствования, наряду с нынешними и будущими версиями "DOS plus" других изготовителей, такими, как DR-DOS компании Digital Research, надолго продлят жизнь DOS и ее "производных".

Частично причина подобной "жизнестойкости" DOS заключается в том, что значительную часть парка персональных компьютеров составляют устаревшие системы, быстрое развитие которых, а также емкость основной и массовой памяти не позволяют установить более современные ОС, предъявляющие повышенные требования к ресурсам компьютера. Однако и среди владельцев более современных компьютеров немало людей, которые предпочитают графическому интерфейсу более быстродействующие программы, рассчитанные на символьный интерфейс.

Большой успех выпал на долю системы Windows после появления ее версии 3.0 — число ее приверженцев значительно возросло. Тем не менее есть пользователи, предпочитающие знакомую и "быструю" DOS, а есть и такие, которым требуются средства, превышающие возможности Windows, поэтому они присматриваются к OS/2 или Unix.

Система OS/2 с появлением версии 2.0 достигла той точки в своем развитии, когда она поддерживает широкий спектр систем и периферийных устройств и, что более важно, позволяет наиболее эффективно выполнять одновременно несколько DOS-ориентированных прикладных программ. Кроме того, OS/2 поддерживает сложные сетевые программы и быстродействующие коммуникационные системы, в то время как оболочка Windows, опирающаяся на DOS, на это не способна. Долгое время ОС Unix была основной операционной системой для высокопроизводительных графических станций. И это действительно единственный достойный выбор, если необходима широко распространенная, стандартная многопользовательская ОС, которая была бы альтернативой операционным системам, предназначенным для конкретных моделей больших ЭВМ и мини-компьютеров. Сегодня Unix становится достойным соперником другим ОС и в области компьютерных систем для учреждений!

(окончание см. стр 97)

Первое место присуждается ...

В мире персональных компьютеров найдется место для многих операционных систем.

15 реляционных баз данных:

Простота доступа, мощные программные средства



С

истемы управления базами данных (СУБД) можно с уверенностью считать наиболее стратегически важными среди программных пакетов для деловых приложений. Приходилось ли вам слышать о "жизненно необходимом" для выполнения какой-либо работы текстовом процессоре? Пакеты СУБД для настольных компьютеров сейчас выходят на передний план среди систем обработки данных для учреждений, обеспечивая возможности, предоставлявшиеся прежде лишь пакетами для мини- и больших компьютеров. В то же время конечные пользователи требуют простоты доступа к данным, а разработчики хотят получить богатую возможностями программную среду для создания коммерческих прикладных систем.

Изготовители современных СУБД с большим апломбом претендуют на выполнение этих требований, однако, как вы увидите, ни одна система полностью не удовлетворяет все потребности пользователей. Данная апробация пятнадцати реляционных СУБД — не сопоставление характеристик и функций, а руководство для выбора пакета, сочетающего в себе все необходимые возможности для разработки ваших собственных баз данных и сегодня, и завтра.

Дэвид Кэлман

Какие пакеты реляционных СУБД обеспечивают наилучшее сочетание возможностей программирования и простоты использования? Чтобы это выяснить, лаборатория журнала PC Magazine подвергла 15 пакетов реляционных СУБД серии тестов.

III

КРИТЕРИИ ОТБОРА

Начиная эту апробацию, мы имели список из почти полусотни пакетов многотабличных и реляционных СУБД, из которого выделили пятнадцать, стоимостью от 99 до 1000 долл. Мы отбирали пакеты реляционных баз данных, предназначенные как преимущественно для конечных пользователей, так и для разработчиков.

Для включения в число тестируемых пакет должен был претерпеть значительные улучшения за последние восемнадцать месяцев и обеспечивать интерфейс с конечным пользователем, не требующий программирования запросов и отчетов. Более десятка пакетов, таких как D the data language и KnowledgeMan/2, требуют от конечного пользователя грамотного употребления соответствующего языка программирования и поэтому здесь не рассматриваются. Тем не менее от всех пакетов требовалось наличие языка программирования для профессиональной разработки прикладных систем. (Это требование исключило из рассмотрения популярный пакет Alpha Four фирмы Alpha Software.)

Два пакета, dBXL фирмы WordTech и DataFlex фирмы DataFlex, запрошенные для апробации, фирмы-изготовители нам в срок не передали. Два других пакета, Magic PC фирмы Aker Corp. и The O'Hanlon Database Solution фирмы O'Hanlon Computer Systems, оказались не в состоянии пройти до конца полный набор тестов на 10 000 записей и были сняты с тестирования.

В этом обзоре не рассматриваются пакеты обслуживания баз данных (БД-серверы), такие как SQL Server, OS/2 EE Database Manager и SQLBase Server фирмы Gupta, или обслуживания пользователей (клиент-серверы), например, SQL Windows той же фирмы. Мы рассматриваем, однако, несколько пакетов, которые могут использоваться и в качестве внешних интерфейсов для БД-серверов, когда в этом возникает необходимость. Такие пакеты, как Advanced Revelation, DataEase, dBASE IV, Omnis5 и Paradox, обеспечивают доступ к данным БД-сервера, незаметно для пользователя подменяя команды структурного языка запросов SQL на свои собственные или допуская прямое употребление SQL. Чтобы использовать эти пакеты для доступа к данным БД-сервера, как правило, необходимы факультативные модули.

ОБРАТИМСЯ В ПРОШЛОЕ, ЗАГЛЯНЕМ В БУДУЩЕЕ

В последнем из опубликованных в журнале PC Magazine обзоров программируемых СУБД мы выбрали для рубрики "Редактор советует" пакеты FoxBase Plus 2.0, Paradox 2.0 и R:BASE for DOS. Обнаружилась забавная деталь: с тех пор номера вер-

сии этих пакетов воспроизводили номера очередных серий кинокартины "Роки". Однако, в отличие от фильмов, новые версии действительно становились все лучше.

Упомянем среди новинок версию 1.1 пакета dBASE IV фирмы Ashton-Tate, которая заметно превосходит предыдущую версию. Эта версия, освобожденная от ошибок, имеющая новую систему управления памятью, встроенную дисковую кэш-память и множество новых возможностей, вновь подтверждает крепкие позиции фирмы Ashton-Tate на рынке баз данных для ПК. (См. "dBASE IV, Version 1.1: A New Beginning", PC Magazine, January 29, 1991.) Вскоре появится новая версия пакета FoxPro 2.0 фирмы Fox Software, но мы, к сожалению, не успели включить ее в данный обзор. Предварительную информацию о ней мы даем во врезке "FoxPro 2.0: новые средства, новый подход".

В реляционных базах данных доступ организован не на физическом, а на логическом уровне.

Пакеты Superbase 4 и Omnis 5 работают под управлением оболочки Microsoft Windows 3.0. Эти системы представляют первое поколение удобных графических СУБД для PC. СУБД, работающие в среде Windows, используют такие преимущества этой подсистемы, как возможность объединения рисунков и текста, выполнения функций из библиотек динамической компоновки DLL, совместного с другими прикладными программами использования данных с помощью динамического обмена данными DDE. Придется подождать, пока быстроедействие ПК станет достаточным для того, чтобы удовлетворить одновременно требованиям обработки данных и графических пользовательских интерфейсов (ГПИ). Но так как быстроедействие ПК неуклонно возрастает, воцарение ГПИ — только вопрос времени.

РЕЛЯЦИОННОСТЬ

При обсуждении баз данных невозможно обойти понятие уровня реляционности — степени соответствия пакета реляционной модели данных

Э.Ф.Кодда. Реляционная модель, теоретической основой которой является логика предикатов первого порядка и теория отношений, предлагает 333 характеристики и определяет специальные правила, согласно которым базу данных можно квалифицировать как "достаточно реляционную" или "полностью реляционную". Эти определения описаны в книге E.F.Codd, The Relational Model for Database Management, Version 2 (Addison Wesley). Вопрос о реляционности™ вызывает споры, так как ни один коммерческий пакет СУБД для ПК нельзя квалифицировать как полностью реляционный, хотя во многих случаях изготовители утверждают, что это так. Конечно, это помогает продавать пакеты, однако вводит в заблуждение пользователей. Изготовители СУБД для ПК злоупотребляют термином "реляционная" для описания любых систем, которые работают с данными табличной структуры. Это, по-видимому, продиктовано стремлением подчеркнуть отличие программных изделий этого типа от менее мощных пакетов, работающих с неструктурированными файлами.

Конечно, ни один пакет не удовлетворяет полностью характеристикам, предложенным Коддом, но реляционная СУБД должна подчиняться по крайней мере некоторым принципиальным требованиям его модели. Реляционная база данных должна обеспечивать доступ к данным не на физическом, а на логическом уровне, т.е. пользователь или программист выбирает данные по именам и значениям без явного перемещения по записям. Не пользователь (программист), а система выбирает наилучший способ выполнения запроса (так называемый "непроцедурный" метод). В реляционной базе данных при выполнении каждого запроса образуется совокупность из одной или нескольких записей. Эта совокупность — фактически сложный тип данных (называемый отношением), над которым определен полный набор реляционных операций. И суть дела в том, что некоторые многотабличные базы данных являются реляционными, а некоторые — нет.

Как реляционная, так и простая многотабличная СУБД имеют преимущества перед пакетами с неструктурированными файлами вроде Q&A фирмы Symantec, так как позволяют потребителям эффективно решать проблемы управления данными сложной структуры. В неструктурированной СУБД данные помещаются в одну таблицу или в произвольно сгруппированные таблицы. Такая система оказывается довольно негибкой, поскольку в каждой записи должно быть фиксированное число элементов строки. Кроме того, если, скажем, информация об одной и той

БАЗЫ ДАННЫХ

же фирме-поставщике встречается более одного раза, вам приходится каждый раз вводить ее заново, расходуя свое время и место на диске.

Преимущество многотабличного подхода — в его гибкости. Вы можете иметь столько элементов строки, сколько захотите, можете добавлять отчеты или процессы, пользуясь таблицей фирм-поставщиков, не меняя ее структуры. Неструктурированные базы данных хуже приспособлены к изменениям из-за того, что из них нелегко извлекать дискретные наборы осмысленных данных.

Еще одно преимущество многотабличной базы данных — простота корректировки. Например, если вы изменили адрес, он изменится во всех точках базы, связанных с данной. Поменяв адрес в одной записи неструктурированной базы и обратившись к базе позднее, вы не узнаете, какая информация соответствует истине.

Достоинство неструктурированного подхода в его простоте. Он работает очень похоже на существующие в реальном мире шкафы с папками, с которыми мы обращаемся с такой легкостью. Но, как и в случае шкафов, чем больше информации, тем меньше порядка. При работе с многотабличной базой данных добавление информации никакой дезорганизации не вносит.

Среди многотабличных СУБД имеются системы с различной степенью реляционное™. Более реляционные СУБД предусматривают полностью непроцедурный доступ к данным, используя языки SQL, QBE (query-by-example, запрос на примере) или эквивалентные непроцедурные команды. Напомним, что непроцедурный метод означает, что наилучший способ доступа к данным выбирает система, а не программист или пользователь. К таким СУБД относятся Ingres, R:BASE и Paradox. Другие системы, например dBASE IV, содержат SQL и QBE в качестве элементов расширения процедурного интерфейса, но для многих операций требуют программирования. Некоторые системы, такие как FoxPro и Advanced Revelation, предоставляют лишь немногие реляционные возможности — непроцедурные генераторы отчетов и инструментальные средства создания запросов, а для всех остальных операций требуют процедурного программирования. Вообще говоря, чем больше вы любите разрабатывать собственные методы доступа к файлам, тем меньше удовлетворения получите от работы с непроцедурными системами. Непроцедурные СУБД выигрывают в производительности, а языки программирования процедурных систем обеспечивают им большую гибкость.

АДАПТАЦИЯ К "ИЗМЕЛЬЧАНИЮ"

Несмотря на то, что реляционные СУБД, представленные сегодня на рынке, богаче возможностями, произ-

водительнее и мощнее, чем три года назад, по-настоящему изменилось не их программное обеспечение, а отношение к ним в организациях.

Движение корпораций сверху вниз по пути внедрения баз данных для ПК продолжается. Отделы информационного обеспечения настойчиво вытесняют традиционные системы на мини- и больших компьютерах информационными системами на базе ЛВС (наш внештатный редактор Уильям Ф. Закман называет этот процесс "измельчением"). В результате современные СУБД для ПК начали приспосабливаться ко многим требованиям систем коллективной обработки данных. Эти требования — защита данных, контролируемый доступ к данным с учетом секретности, обработка транзакций (OLTP, on-line transaction processing) и неявный ("прозрачный") доступ к данным главного компьютера — не "ма-

Удобный
пользовательский
интерфейс снижает
затраты на
обучение и
вероятность ошибок

ленькие хитрости". Наиболее точным ответом на эти требования стало широкое внедрение SQL, субязыка баз данных, отвечающего стандарту Американского национального института стандартов (ANSI). Широко распространенный на мини- и больших компьютерах, SQL служит "интернациональным языком" для работы с распределенными базами данных. Пакеты, ориентированные на ПК, например DataEase, dBASE IV, Ingres, Paradox и R:BASE, позволяют теперь разрабатывать прикладные системы, в явном виде содержащие операторы SQL для манипулирования данными БД-сервера.

Другие возможности СУБД, которые всегда требовались информационным отделам корпораций, — обработка транзакций, подключение к БД-серверу и стандартизация пользовательского интерфейса — имеются, по крайней мере частично, в пакетах, изготовители которых приняли стандарт общения с пользователем CUA (Common User Access) из стратегии архитектуры прикладных систем SAA (System Application Architecture) фирмы IBM. Обслуживание сотен рабочих станций в крупной фирме требу-

ет наличия универсального удобного пользовательского интерфейса, чтобы снизить расходы на обучение и свести к минимуму ошибки пользователей. Пакеты, придерживающиеся близкой к стандарту CUA системы "выпадающих" меню или работающих в среде Windows (Superbase 4), обладают этим преимуществом.

АРХИТЕКТУРА ИНФОРМАЦИИ

Если считать, что в ближайшем будущем с большими СУБД будут работать в ЛВС (охватывающие в том числе и мини- и большие компьютеры) с помощью таких БД-серверов, как Oracle Server, SQLBase, SQL Server и OS/2 EE Database Manager, что же остается на долю "чистокровных" пакетов для ПК? Ответ таков: работа с прикладными системами на вашем рабочем столе, где они и родились. Вы можете продолжать пользоваться СУБД, которая вас устраивает, пока будете присматриваться к развитию клиент-серверов и пока можете быть уверены, что она соответствует всем вашим потребностям.

Главная цель реляционной модели — обеспечить доступ к информации в базе данных на логическом уровне, т.е. независимо от ее физического представления. Это означает, что пользователи и прикладные системы могут взаимодействовать с БД через команды SQL, не обращая внимания на структуру данных. Различные прикладные системы могут обращаться к одним и тем же данным независимо, поэтому у таких СУБД для ПК, как Advanced Revelation с подсистемой Environmental Bonding, DataEase SQL, Paradox с SQL Link и dBASE IV Server Edition, появляется возможность пользоваться одними и теми же базами данных.

Независимость от физических данных означает также, что по мере роста баз данных их можно переносить на более крупные вычислительные установки, не переписывая прикладных систем. Такая "масштабируемость" сокращает расходы на инструментальные средства и изготовление заказных прикладных систем.

БЕСКОНЕЧНОЕ РАЗНООБРАЗИЕ

СУБД для ПК бывают "всех форм и размеров". Например, БД-серверы для ЛВС, подобные системам SQL Server и OS/2 EE Database Manager, специализируются на администрировании и защите данных для групп разработчиков. Пакеты для разработки прикладных систем, такие как Advanced Revelation, Clarion Professional Developer, FoxPro и dBASE IV, специализируются на инструментальных средствах и языках для создания прикладных систем. Полнофункциональные реляционные

РЕДАКТОР СОВЕТУЕТ

- **FoxPro**
- **Paradox**

Как показывают наши рейтинги соответствия поставленной задаче, не так-то просто найти программируемую многотабличную СУБД, которая одновременно удовлетворяла бы требованиям разработчиков прикладных систем и конечных пользователей. Наиболее мощные пакеты для разработчиков иногда оказываются "недружественными" к конечным пользователям, а пакеты с наилучшим пользовательским интерфейсом проигрывают в возможностях средств программирования.

Тем не менее "золотая середина" есть, и около нее мы обнаруживаем два пакета.

FoxPro рекомендуется для всех разработчиков баз данных. Его средства работы с конечным пользователем должны еще пройти определенный путь в сторону "дружественности" (выходящая в ближайшее время версия 2 должна продвинуться в этом направлении), однако это выдающийся пакет с поразительным быстродействием. За этим быстродействием несомненно кроется применение самой современной технологии программирования. Более ранняя версия FoxPro уже попадала в рубрику "Редактор советует" в обзоре Project Database III журнала PC Magazine; с тех пор этот пакет не переставал совершенствоваться. При цене 795 долл. FoxPro — замечательный пакет почти во всех отношениях, особенно по быстродействию; он финишировал первым почти во всех тестах на производительность.

Пакет Paradox фирмы Borland стоимостью 795 долл., также попавший в рубрику "Редактор советует" в предыдущем обзоре, вновь одержал победу (на этот раз версия 3.5). Пакет, построенный на базе метафоры таблицы, отличающей его от большинства других СУБД, требует определенного времени, чтобы научиться с ним работать, но тщательно спроектированное рабочее пространство делает его одновременно "дружественным" и многофункциональным. Paradox заполняет пропасть между разработчиками прикладных систем

и конечными пользователями благодаря мощному набору инструментальных средств программирования, включающему полнофункциональный язык PAL (Paradox Application Language). Его производительность, хоть и не столь ошеломляющая, как у FoxPro, вполне достаточна.

Хорошего отзыва заслужили еще три пакета, в том числе DataEase стоимостью 795 долл. Хоть он и не стал победителем по скорости, некоторая медлительность пакета с лихвой перекрывается той легкостью разработки и обслуживания баз данных, которую обеспечивает его применение.

В числе заслуживающих упоминания и респектабельный пакет R:BASE фирмы Microgim. Интуитивно понятный интерфейс, превосходный генератор прикладных систем (Application Generator) и связи с dBASE делают его серьезным конкурентом. Версия 3.1 пакета стоимостью 795 долл. имеет удобный язык программирования, но отсутствие инструментальных средств не позволяет рекомендовать его профессиональным программистам. И все-таки это прекрасная СУБД для непрограммирующих пользователей, желающих создавать персональные или "домашние" прикладные системы. Вы можете "врасти" в этот пакет (возможно, начав с выпускаемого фирмой Microgim более простого варианта — Personal RrBASE), используя преимущества его более мощных функций, насколько позволяют ваш опыт и потребности.

У "фанатов" системы Microsoft Windows пока не очень-то большой выбор в этой категории программной продукции, но пакет Superbase 4 стоимостью 595 долл., на удивление быстрый, несмотря на работу в среде Windows, наводит на определенные размышления. Это единственный вариант, если вам уже сейчас нужна графическая СУБД, а не пакет, работающий в текстовом режиме и запускаемый из Windows. Меню и руководства нуждаются в доработке (что обещано в следующей версии), однако интеграция графических изображений, обеспечиваемая пакетом, и его в определенной степени объектно-ориентированный стиль убеждают в том, что у графических СУБД есть будущее.

Среди пакетов для разработчиков прикладных СУБД неплохо смотрятся Advanced Revelation и Ingres. Оба пакета содержат мощные инструментальные средства программирования и удобные связи с более крупными распределенными системами. Потери в "дружественности" к конечному пользователю они компенсируют своим потенциалом в области разработки.

Те, кто пользуется СУБД от случая к случаю и не программирует, могут спросить: а нужна ли нам вообще программируемая многотабличная СУБД? Может быть и нет. Если вы не ожидаете, что ваши потребности в СУБД выйдут далеко за пределы сортировки листов рассылки или простейших запросов, применение любого из рассмотренных здесь пакетов будет для вас стрельбой из пушки по воробьям. Возможно, окажется, что хороший пакет с неструктурированными файлами, скажем Reflex фирмы Borland, Q&A фирмы Symantec или PC-File фирмы Buttonware, вполне удовлетворяет всем вашим требованиям. Имея цену | от 125 до 350 долл., неструктурированные пакеты, как правило, дешевле реляционных и в большинстве случаев намного проще в освоении и в работе.

Остался один важный вопрос: а как же лидер рынка — пакет dBASE IV стоимостью 795 долл.? Очевидно, I что версия 1.1 — серьезное улучшение по сравнению с версией 1.0, но говоря уже о dBASE III Plus, I и всем приверженцам dBASE, которые еще не перешли на новую версию, следует это сделать. Но пакет по-прежнему проигрывает в ряде отношений. Имеются проблемы с реализацией языков запросов QBE и SQL, требует переработки "центр управления" (Control Center). Производительность пакета также не мешало бы повысить.

✉ — "PCMagazine/USSR:

В СССР пакет Paradox можно приобрести за рубли например в СП "Параграф" (контактный телефон (095) 200-25-66 или в СП "Диалог" (контактный телефон (095) 320-32-11. Насколько нам известно, пакет FoxPro за рубли пока не продается; за СКВ его можно приобрести через СП "Интермикро" (контактный телефон (095) 267-32-10.

СУБД, скажем, Paradox и R:BASE, выделяются своими интерактивными возможностями и встроенными инструментальными средствами для создания прикладных систем. Графические СУБД, например Omnix 5 и Superbase 4, предоставляют средства для построения прикладных систем с ГПИ использующих масштабируемые шрифты и изображения. Кроме того, такие системы, как DataEase, Paradox, Advanced Revelation и dBASE IV, могут работать и в качестве внешних интерфейсов БД-серверов для четко контролируемого распределения коллективных данных.

Пользуясь традиционным термином СУБД, можно провести "декомпозицию" всей области управления базами данных на следующие категории, указывающие, какому пользователю адресованы пакеты и каково их основное назначение:

- программируемые пакеты разработки прикладных систем для профессиональных разработчиков баз данных;
- интерактивные средства разработки и администрирования данных для профессионалов-компьютерщиков;
- интерактивные средства обработки данных для конечных пользователей;
- инструментальные средства разработки прикладных систем для конечных пользователей;
- внешние интерфейсы БД-серверов;
- БД-серверы;
- вариации и комбинации перечисленного выше.

О том, к какой категории относится та или иная СУБД, по надписям на упаковке обычно узнать невозможно, разве что это БД-сервер или его внешний интерфейс. Поэтому два на первый взгляд не отличимых пакета, могут предназначаться для совершенно различных классов пользователей. Изготовитель может утверждать, что он создал абсолютно универсальную систему, однако почти наверняка она по-разному приспособлена к различным областям обработки данных. Ваша задача состоит в том, чтобы определить, к какому типу пользователей вы относитесь. Отнюдь не редкость, когда люди приобретают значительно более совершенную систему, чем та, которая им реально необходима, и напрасно тратят деньги и время.

Авторского права нет, но переписывать нельзя

Уинн Л.Рош

Взрывы бомб никогда не проходят бесследно, и силу их ударов редко можно предвидеть. Взрывная волна от обыкновенного двухстраничного приговора окружного судьи Соединенных Штатов Терри Дж.Хэттера, объявленного 11 декабря 1990 года, до сих пор гуляет среди изготовителей СУБД, как будто это был ядерный взрыв.

"Суд находит, что фирма Ashton-Tate, подавая заявку на авторские права на свою оригинальную продукцию, систематически утаивала важную информацию от Управления охраны авторских прав Соединенных Штатов, — гласил приговор. — Поэтому суд считает авторские права фирмы Ashton-Tate на ее серию компьютерных программ dBASE недействительными в связи с незаконными действиями".

Эти строгие слова стали кульминацией судебного процесса, начавшегося 18 ноября 1988 года, когда фирма Ashton-Tate возбудила иск против фирм Fox Software и Santa Cruz Operation, обвиняя их в нарушении авторских прав и нечестной конкуренции.

Ashton-Tate утверждала, что пакет FoxBase фирмы Fox Software скопирован с ее семейства dBASE. Fox Software выдвинула в свою защиту несколько положений, среди которых самым сильным было, по-видимому, то, что фирма Ashton-Tate участвовала (и даже в числе инициаторов) в разработке FoxBase, а следовательно, косвенным образом санкционировала выпуск пакета.

Перед тем, как дело было передано в суд, фирма Fox Software выставляла встречный иск. Fox Software утверждала, что семейство dBASE было создано на базе общедоступной программы для большой ЭВМ под названием Jet Propulsion Laboratory Data Management and Information Systems (система обработки данных и информационная система Лаборатории реактивного движения) — факт, который фирма Ashton-Tate утаивала от Управления охраны авторских прав, пока Fox Software не выступила со своим ходатайством.

При рассмотрении встречного иска суд принял во внимание все имеющиеся в деле свидетельства — ходатайства, письменные и устные показания и другие документы — чтобы определить, может ли противная сторона, т.е. Ashton-Tate, выиграть процесс, даже в том случае, если закон и доказательства будут в ее пользу. Идея заключалась в том, что бессмысленно продолжать судебное разбирательство, ни одна из сторон не имеет шансов на победу.

КТО ВЫИГРАЛ?

Фирмы Ashton-Tate и Fox Software несколько по-разному расценивают решение суда. Fox Software считает, что она выиграла, война закончена и фирме Ashton-Tate стоит обратить внимание на качество своей продукции вместо того, чтобы "душить" конкурента, опираясь на закон об авторских правах. Ashton-Tate намеревается обжаловать приговор, полагая, что судья не понял закона об авторском праве и вместо

него руководствовался концепцией патентного законодательства.

Хотя кардинальный вопрос в этом деле уже решен, судебный процесс еще не закончился. Разбор второстепенных вопросов и рассмотрение апелляций будут тянуться годами. Приговор может быть отменен, либо фирму Ashton-Tate могут обязать выплатить фирме Fox Software миллионы долларов на основании встречного иска, обвиняющего Ashton-Tate в нечестном ведении дел.

На сегодня, однако, важнее всего то, что авторские права фирмы Ashton-Tate признаны недействительными. Но какие бы далеко идущие последствия ни имело это решение, оно не означает, что можно свободно переписывать последнюю версию dBASE IV, как если бы она была общедоступным программным обеспечением. Это мнение разделяет даже Fox Software. Семейство пакетов dBASE по-прежнему защищают закон о торговых знаках и лицензии на программное обеспечение, а документация на dBASE остается защищенной авторским правом.

Но приговор — хорошая весть для вас и изготовителей баз данных в том смысле, что он рассеивает туман, начавшие было сгущаться вокруг языка и структуры файла dBASE. Сообщество пользователей dBASE может больше не беспокоиться о праве свободного использования, без сомнения, самого популярного в мире языка СУБД. Он предоставляет разработчикам СУБД открытый стандарт, которому можно следовать, не боясь судебных преследований в будущем.

КАК МЫ ТЕСТИРОВАЛИ СУБД

В этом обзоре мы не определяли степень реляционное™ СУБД, поскольку ни один из пакетов не отвечает ортодоксальному определению этого термина, данному Коддом. Вместо этого мы оценивали реляционную функциональность изделий — их способность выполнять набор операций и запросов к базе данных, которой должны обладать все реляционные системы. Наш тест, подмножество стандартного масштабируемого и переносимого теста AS3AP (ANSI SQL Standard Scalable and Portable Benchmark Test), обеспечивает исчерпывающее тестирование производительности СУБД в широком диапазоне программных изделий и платформ.

Хотя тесты составлены на языке SQL, они могут выполняться и над системами, не "знающими" этого языка. Более того, фирмы, которые подключили к кодированию спецификаций наших тестов для своих пакетов собственных технических экспертов, использовали для достижения тех же результатов свои оптимальные методы.

НЕОБХОДИМА ВАША ОЦЕНКА

Правильный выбор СУБД предполагает тщательное планирование. Каждый потенциальный покупатель должен рассмотреть вопросы совместимости с ПК и ЛВС, возможности, простоту применения, простоту разработки, качество сопровождения со стороны изготовителя. Более крупные организации должны также обратить внимание на возможность взаимодействия с системами на мини- и больших компьютерах, совместимость с сетевыми БД-серверами, вопросы секретности и политику фирменного сопровождения.

В данной апробации вы найдете большую часть исходной информации для принятия квалифицированного решения, включая обширные таблицы характеристик и рейтинги соответствия поставленной задаче — однако поиск СУБД вам следует начинать с изучения ваших собственных потребностей и возможностей. Что вам нужно — пакет для разра-

ботки прикладных систем, адресованный профессионалам-компьютерщикам, или интерактивная система обработки данных для конечных пользователей? Кто вы — администратор базы данных или конечный пользователь с некоторыми навыками программирования?

FoxPro
Ричард Хейл Шоу

Если вы ищете СУБД, сочетающую массу возможностей и средств для разработчика со скоростью и простотой доступа, необходимыми для конечного пользователя, пакет, который совместим с форматом файлов и языком программирования БД, ставшими промышленным стандартом, не "застряв" при этом на уровне технологии середины 80-х, как пакет FoxPro, версия 1.02 — это как раз то, что вам требуется.

Эта СУБД фирмы Fox Software стоимостью 795 долл. превзошла dBASE и лидирует в своей области по новизне технических решений, воз-

FoxPro 2.0: новые средства, новый подход

Сол Рикьярди

Одним из важнейших событий 1991 года в области СУБД станет появление версии 2.0 пакета FoxPro. Пакет обещает превзойти нынешнюю версию и еще раз подтвердить высочайший класс программистов фирмы Fox Software. Хотя еще рано делать какие-либо окончательные заключения, первый взгляд на бета-версию пакета убедил нас в том, что FoxPro 2.0 станет серьезнейшим конкурентом для других СУБД.

Технология Рашмора

Одна из наиболее заметных особенностей пакета FoxPro 2.0 — новая технология оптимизации запросов Рашмора. Рашмор запатентовал способ резкого ускорения доступа к индексированным данным для некоторых "оптимизируемых" запросов. По утверждениям фирмы Fox Software, технология обеспечивает ускорение доступа к "степеням важности" данных при определенных условиях. Фирма продемонстрировала, что некоторые операции с использованием этой технологии выполняются в тысячи раз быстрее.

Метод Рашмора автоматически работает с имеющимися прикладными системами и индексными файлами, однако для максимального использования его преимуществ потребуются забыть некоторые приемы улучшения характеристик систем. Например, в нынешних прикладных системах более медленные

команды часто заменяют более длинными последовательностями других команд, дающими те же результаты, но быстрее. При использовании технологии Рашмора в этом больше нет необходимости — более того, этого не следует делать, поскольку метод Рашмора позволяет оптимизировать простые команды.

Похоже, что и операции, не использующие технологии Рашмора, в FoxPro 2.0 выполняются достаточно быстро. В порядке эксперимента мы прогнали нашу бета-версию FoxPro 2.0 через свои новые тесты производительности реляционных СУБД. Бета-версия прошла всю серию без ошибок и на 21% быстрее, чем FoxPro 1.02. (Хотя это и впечатляет, тот факт, что тестировалась бета-версия, делает эти результаты бесполезными, разве что любопытными. И тем не менее, все говорит за то, что пакет FoxPro 2.0 будет очень быстродействующим.)

Пользователи языка SQL, несомненно, оценят реализацию оператора SELECT. В FoxPro 2.0 команда SELECT полностью интегрирована в командный язык и доступна в любой момент — не требуется никаких переключений



Рис. 1: "Строитель экранов" (Screen Builder) пакета FoxPro 2.0 позволяет конструировать экраны с "кнопками" и прокручиваемыми областями редактирования текста. Предусмотрена работа с минимизированными окнами и взаимоотношения типа "один к многим" с использованием новой команды SET SKIP TO.

режимов. Поскольку оператор SELECT может выполнять операции, которые требуют нескольких обычных команд FoxPro, его использование предпочтительно в ситуациях, когда работает технология Рашмора, тогда эта команда будет оптимизирована. Новый экран реляционного запроса на примере RQBE (Relational query-by-example)

ТОЛЬКО ФАКТЫ

Изделие: FoxPro, версия 1.02
Фирма: Fox Software Inc., 134 W. South Boundry, Perrysburg, OH 43251; 419-874-0162.
Рекомендуемая цена: 795 долл.
Требуемые ресурсы: 420 Кбайт ОЗУ, DOS 2.0 или более поздняя версия.

Высокопроизводительная, совместимая с dBASE СУБД с элегантным интерфейсом и мощными программными средствами для разработчика. Хотя, быть может, и не слишком "дружелюбный" к конечному пользователю, этот пакет — "рай" для разработчика.

возможностям и быстрдействию — сочетание, благодаря которому предыдущая версия пакета попала в рубрику "Редактор советует". Пользовательский интерфейс FoxPro, рассчитанный на текстовый режим работы дисплея с применением мыши, близок к стандарту CUA: он ис-

БАЗЫ ДАННЫХ

пользует выпадающие меню, накладывающиеся диалоговые окна и списки, а также линейки прокрутки. Удобный экранный калькулятор, календарь и даже таблица кодов ASCII дополняют стандартные меню.

Если вам потребовалось, работая в FoxPro, открыть таблицу и просмотреть ее содержимое, забудьте про команды USE и BROWSE: подведите курсор к пункту "открыть" (Open) меню "файл" (File), нажмите клавишу на мыши, выберите нужное имя файла из появившегося списка, а затем подведите курсор к пункту "просмотр" (Browse) меню "редактирование" (Edit) и еще раз нажмите клавишу. Таблица, которую вы хотите просмотреть, появится на экране в перемещаемом окне переменного размера, которое может двигаться по экрану с захватывающей дух скоростью. Если вы предпочитаете работать с командной строкой, окно "команда" (Command) позволяет вво-

дить команды языка dBASE точно так же, как и прежде, через строку с приглашением-точкой.

РАЙ ДЛЯ РАЗРАБОТЧИКОВ

Современный интерфейс — не единственное достоинство пакета FoxPro: разработчики найдут в нем множество замечательных возможностей и средств. По сравнению с предшественником, пакетом FxBase Plus, в FoxPro диалект языка dBASE расширен более чем на 200 команд, введены такие нетривиальные возможности, как поля комментариев (memo) переменной длины. Имеются функции для выполнения операций поиска и замены в текстовых полях, а также операций ввода-вывода нижнего уровня для файлов DOS. Первый из клонов dBASE, в котором появились функции пользователя (User Defined Functions, UDF), FoxPro дает возможность передавать им параметры как по значению, так и по ссылке.

Все преимущества интерфейса FoxPro доступны для использования

позволяет вводить информацию, по которой FoxPro генерирует оператор SELECT.

Открытая архитектура
 Разработчиков систем на базе FoxPro порадует новая открытая архитектура пакета. Новый внешний интерфейс прикладных систем (API) FoxPro 2.0 обеспечивает доступ к большинству внутренних функций пакета. Пользуясь языком Си или ассемблером, разработчики смогут создавать внешние библиотеки процедур, которые динамически загружаются и освобождаются по мере необходимости. При помощи API программисты смогут обращаться к базе данных, манипулировать памятью, выполнять операторы и выражения FoxPro, пользоваться подпрограммами управления окнами и дисплеем пакета. API обеспечит доступ к циклу ожидания событий FoxPro, давая возможность программистам обрабатывать события FoxPro и устанавливать собственные обработчики событий.

Вы сможете также использовать в своих программах расширенный интерфейс FoxPro — например, зависимые экраны "клавиши", окна, области редактирования текста, которые чаще можно встретить в графических пакетах. Все это доступно в FoxPro через расширенную команду @ GET и новую команду @ EDIT.

Инструментальные средства для разработчика

Конструирование экранов, использующих все эти расширения, упростит новый объектно-ориентированный "строитель экранов" (Screen Builder) — часть "тройки" инструментальных средств, в которую входят также "строитель меню" (Menu Builder) и "менеджер проекта" (Project Manager). Menu Builder позволит вам конструировать, проверять и генерировать код для меню, тогда как Project Manager будет выполнять почти все функции программы MAKE для программистов. Вам будет предоставлена возможность перечислить экраны, меню, программы, библиотеки и практически все необходимое для прикладной системы. Затем Project Manager построит прикладную систему.

Одной из функций подсистемы Project Manager станет изготовление полностью независимых EXE-файлов. Он сможет создавать прикладные системы типа .FXP, самостоятельные EXE-файлы или серии исполнимых файлов, использующих центральную библиотеку FoxPro.

В FoxPro 2.0 появятся также новые составные индексы, использу-



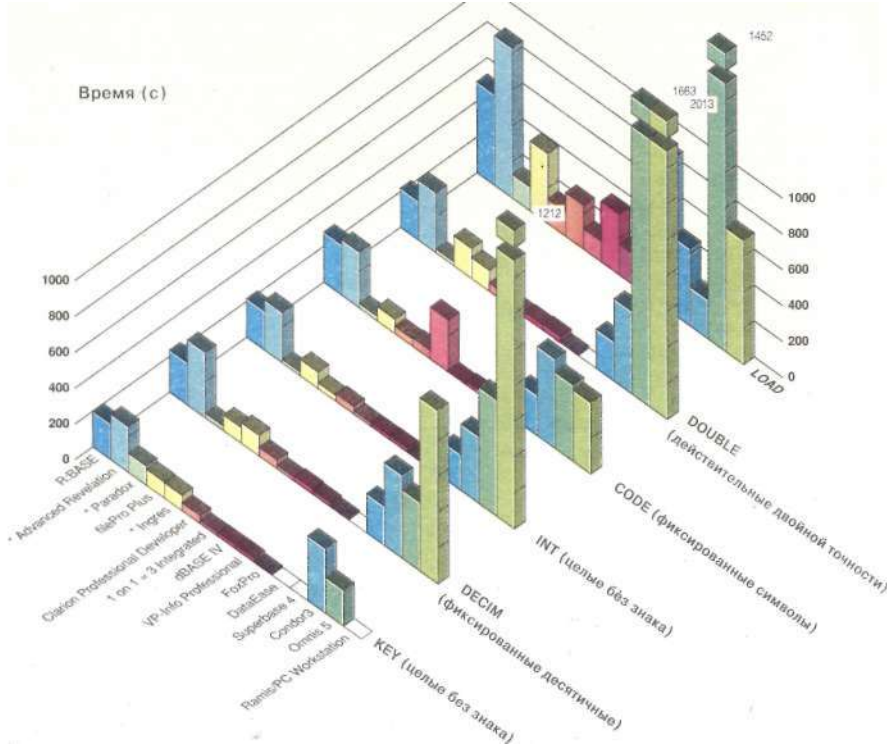
Рис. 2.: Пакет FoxPro имеет многооконный интерфейс с использованием мыши, близкий к стандарту CUA. Он позволяет использовать вместо традиционной командной строки "выпадающие" меню и "выскакивающие" диалоговые окна, однако в окне Command можно по-прежнему вводить команды в ответ на приглашение-точку.

ющие запатентованный метод сжатия, сокращающий размер индексного файла, специальные версии, работающие в расширенных режимах процессоров 386/486, встроенная обработка шаблонов. Короче говоря, в новой версии FoxPro есть на что посмотреть.

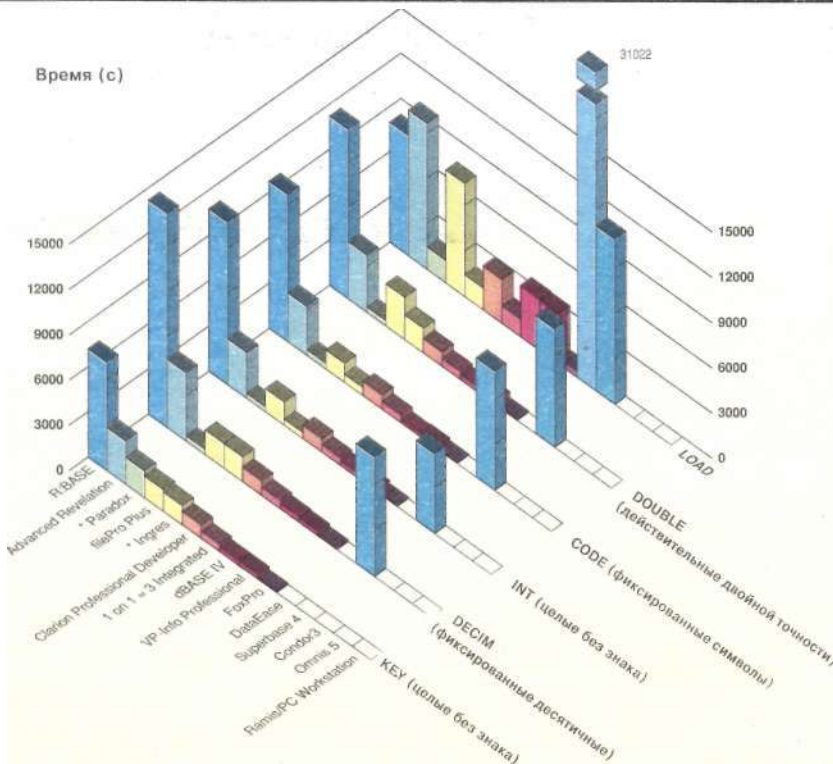


РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ: ПРОИЗВОДИТЕЛЬНОСТЬ

ЗАГРУЗКА И ИНДЕКСИРОВАНИЕ (10 тыс. записей)



ЗАГРУЗКА И ИНДЕКСИРОВАНИЕ (100 тыс. записей)



в прикладных системах. Есть, например, команды и функции, позволяющие вам создавать многооконные структуры. С помощью командного предложения IN WINDOW ("в окне") можно установить отношения подчиненности между двумя окнами и ограничить порожденное окно границами породившего. Если вы перемещаете, убираете с экрана или закрываете породившее окно, то же самое происходит и с порожденным. Тем самым открывается возможность создания более глубоких, более интуитивно понятных интерфейсов прикладных систем при меньшем объеме программирования. Благодаря свободному доступу к значениям, хранящимся в калькуляторе или календаре, их можно без ограничений использовать в разрабатываемых базах данных.

Пользуясь FoxPro, нет ничего проще, чем создать новую прикладную систему. Для этого существуют три дополнительные программы: FoxView для конструирования экранов и генерирования программного кода прикладной системы, FoxCode для создания прикладных систем с помощью языка шаблонов и FoxDoc для документирования текстов программ и обработки перекрестных ссылок. Язык шаблонов программы FoxCode позволяет создавать шаблоны многократного применения, которые избавляют вас от повторяющихся деталей программирования прикладной системы. FoxDoc формирует таблицы перекрестных ссылок в вашей программе, показывает древовидную диаграмму ее структуры и ведет словарь данных. FoxPro создает программный код для каждого пункта меню и показывает этот код в окне в нижней части экрана. Вам предоставляется возможность просмотреть, отредактировать код и немедленно проверить, что получилось. Существуют также дополнительные окна для запуска отладчика и "вырезания-вклеивания" кусков исходного кода между частями программы или из окон подсказки в вашу программу.

ОГРОМНАЯ ПРОИЗВОДИТЕЛЬНОСТЬ

Пакет FoxPro — чемпион по производительности. Он занял первые места в малых и больших тестах загрузки (Load) и индексирования (Index), тестах проекта (Project), отчета (Report) и вывода (Output Overhead). Кроме того, он финишировал в первой пятёрке на тестах связывания (Join) и выбора (Select).

FoxPro — один из немногих среди рассмотренных здесь пакетов (исключая Paradox и R:BASE), который

БАЗЫ ДАННЫХ

не ограничивается использованием только обычной памяти DOS. Он может пользоваться отображаемой памятью (EMS) стандарта LIM 4.0 и содержит программу FOXQ.EXE, которая использует верхние адреса памяти, доступные через драйвер QEMM-386, не затрагивая графической памяти адаптеров EGA/VGA. Пакет резервирует первые 64 Кбайт отображаемой памяти для работы с окнами, меню, переменными и программами, а оставшуюся использует в качестве дисковой кэш-памяти. Принимая во внимание, что сам FoxPro занимает только 384 Кбайт (хотя для загрузки требует 420 Кбайт), для ваших прикладных систем и сетевых драйверов остается достаточно много места.

ОЖИДАЕМЫЕ УЛУЧШЕНИЯ

Каковы же недостатки пакета? Он не позволяет создавать EXE-файлы DOS и не имеет интерфейса прикладных программ, который обеспечивал бы вызовы из программ на других языках (например Си). По-прежнему отсутствуют средства работы с SQL или какой-либо другой теоретико-множественный подход к управлению базой данных. Интерфейс FoxPro также не свободен от ограничений. Он слишком ориентирован на команды языка dBASE: USE, BROWSE, APPEND, COPY... Как конечный пользова-

тель, вы можете и не знать (да и не хотите знать), что означают эти команды. Быть может, вам нужно всего лишь открыть таблицу и изменить ее содержание.

И наконец, неприятно, что в FoxPro не реализована многоиндексная схема, подобная файлам .MDX dBASE. В конце концов, эта технология существует с момента первого появления dBASE IV в конце 1988 года. Правда, новая версия 2.0 FoxPro к моменту подготовки к печати этого обзора проходила бета-тестирование, и она, возможно, ответит на многие из высказанных претензий (см. врезку "FoxPro 2.0: новые средства, новый подход").

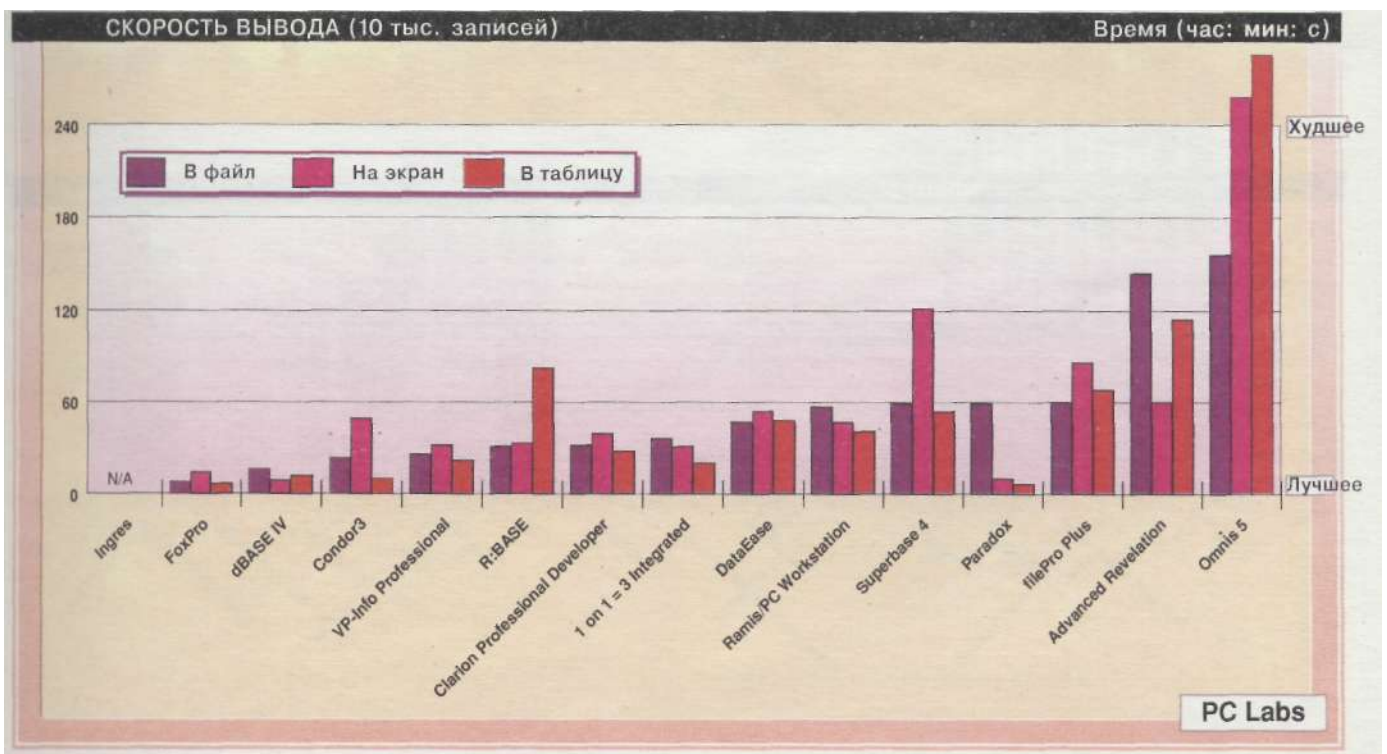
Если вам никогда не приходилось работать с FoxPro, для вас заготовлено несколько приятных сюрпризов. Один из самых приятных — предлагаемая фирмой Fox Software гарантия с возвратом денег: вам разрешается в течение 30 дней пробовать пакет, а затем вернуть свой экземп-

ляр, если система не оправдала ваших ожиданий. Но не беспокойтесь. Это вряд ли случится.

ID	Last	First	State
1	Haughton	Robert	WA
2	Hertzig	Susantha	NY
3	Hilicky	Courtney	PA
4	Horch	Jennifer	WA
5	Bain	Elizabeth	CA
6	Ratz	Gary	CA
7	Kristoff	Imy	VA
8	Lane	Gina	CA
9	Levin	Jennifer	TN
10	Lord	Josie	NY

Рис. 3: В пакете Paradox вы создаете запрос, предоставляя "галочки" в скелете таблицы. Этот запрос выдает записи, в которых фамилия начинается с букв от "H" до "L". Записи, удовлетворяющие запросу, появляются в таблице ответа под запросом.

PC LABS РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ: ПРОИЗВОДИТЕЛЬНОСТЬ



Paradox

Альфред Пур

Когда пакет Paradox впервые появился в 1985 году, он ознаменовал новый подход к управлению базами данных на ПК. Нынешнее изделие фирмы Borland International, версия 3.5 — по-прежнему неординарная и мощная система обработки данных. "СУБД для думающего пользователя", Paradox остается одной из тех редких программ, которые обращены в одинаковой степени и к начинающим, и к квалифицированным пользователям.

Своим успехом Paradox, по крайней мере отчасти, обязан принятому здесь способу представления задач работы с БД. Вместо того, чтобы начинать с традиционного изображения отдельных полей и записей, как это сделано в dBASE и подобных ей системах, разработчики пакета Paradox воспользовались более простым и понятным, на их взгляд, представлени-

ем. Вместо файлов и отношений программа использует "таблицы", которые отображаются на экране в виде строк и столбцов, подобно представлению базы данных в виде динамической таблицы в системе Lotus 1-2-3.

Сходство на этом не кончается. Как только вы вводите запрос, записи, отвечающие критериям выбора, "выпадают" в нижнюю часть экрана, образуя временную таблицу под названием "ANSWER" ("ОТВЕТ"). В дальнейшем эту таблицу можно использовать в качестве источника информации для отчетов или сохранить ее под другим именем для последующей обработки.

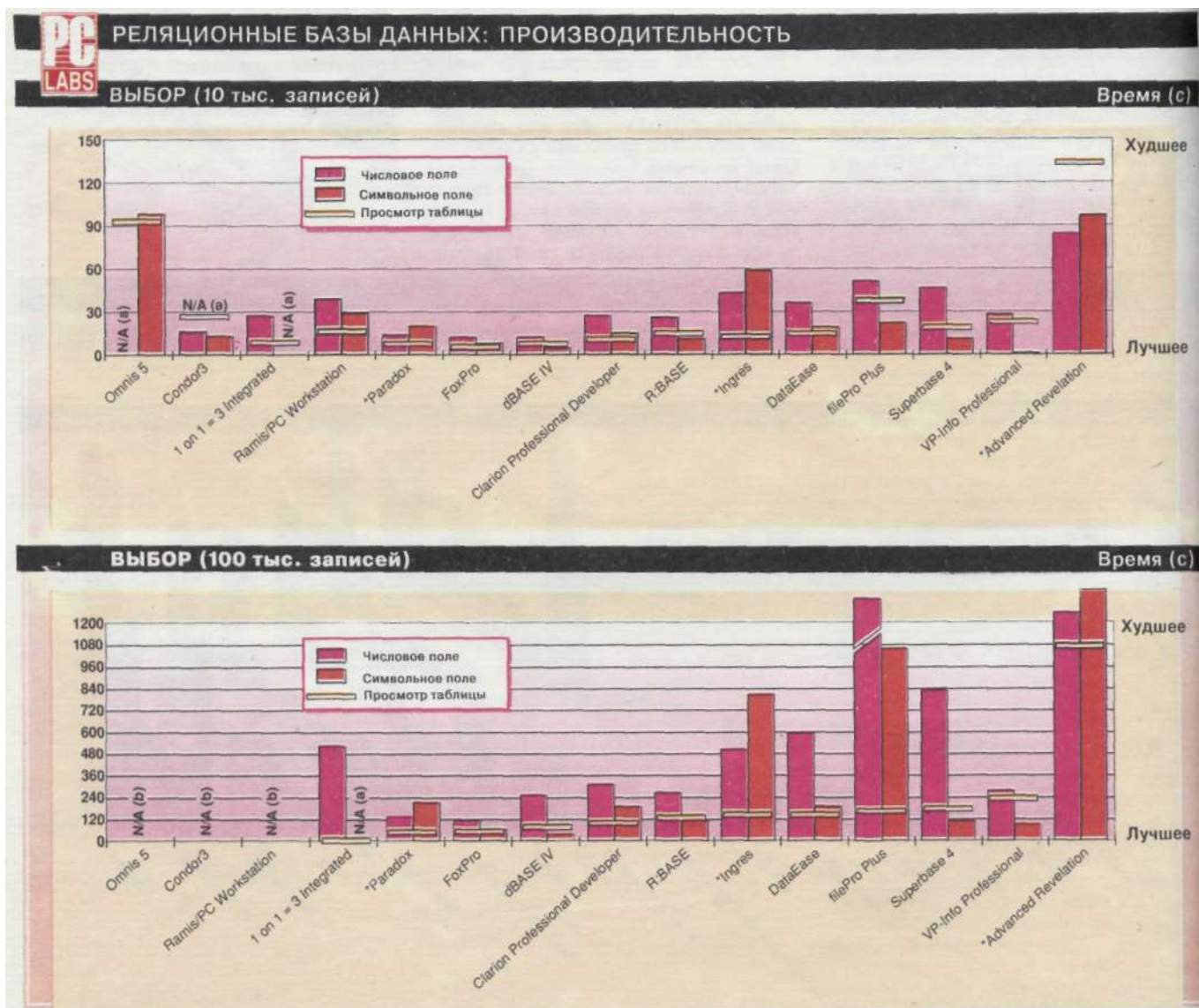
Операции задания запроса столь же наглядны и просты, как и все остальные режимы работы пакета Paradox. В нем используется собственная версия "запроса на примере"

(Query by Example). Вы можете вводить в столбцах таблицы условные операторы для задания критериев выбора. "Галочки" показывают, какие поля должны появиться в ответе. Вы можете даже создать "на лету" новые поля, задав выражения с содержимым полей и константами.

А если вам понадобилось связать две или более таблицы, Paradox прямо "просияет" — вы только введете одинаковые значения в колонки, которые хотите объединить, а остальное сделает программа. Нет никакой необходимости в сложном программировании, предварительном задании отношений или индексировании полей.

ИНСТРУМЕНТ ДЛЯ ПРОФЕССИОНАЛОВ И ЛЮБИТЕЛЕЙ

Фирма Borland стремилась к тому, чтобы пользователи любой квалификации могли извлекать информацию



БАЗЫ ДАННЫХ

из базы данных Paradox быстро и безболезненно. Поскольку ответы появляются в новой таблице, исключается опасность порчи исходных данных. Процесс, аналогичный запросу, позволяет вам выполнять глобальные исправления и удаления, однако, помня о безопасности, Paradox сохраняет исходные копии всех измененных или удаленных записей во временной таблице, поэтому вы быстро можете отменить внесенные изменения, если ваш запрос почему-либо вызвал не ту, реакцию, которую вы ожидали.

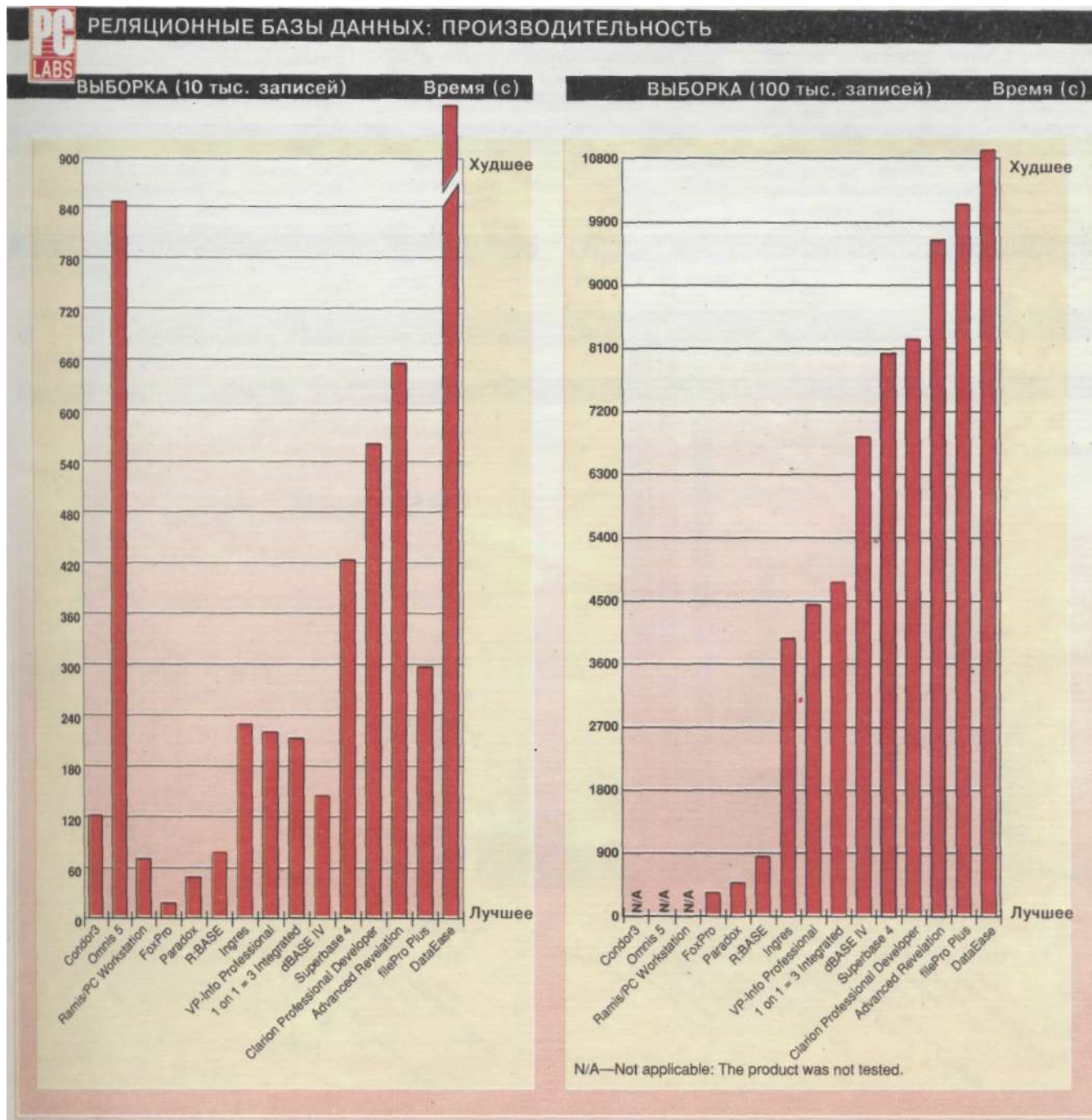
Легко доступен также широкий выбор средств организации ввода данных, в том числе проверка попа-

дания в диапазон, проверка допустимости значения путем просмотра других таблиц, шаблоны для форматирования ввода.

Если вам не нравится табличное представление, можно создать формы для ввода данных. Эта функция заметно улучшена по сравнению с предыдущими версиями пакета, и теперь можно даже создавать формы, показывающие записи из двух таблиц. Например, вы можете создать форму счета, содержащую в заголовке информацию о покупате-

ле, а затем на том же экране, в виде многострочной формы, информацию об отдельных закупках. С помощью функциональной клавиши можно переходить от одной области к другой.

Подсистема вывода информации пакета Paradox имеет несколько сильных сторон. Можно создать "мгновенный отчет", а можно воспользоваться средствами формирования индивидуального отчета. Программа базируется на "диапазонном" подходе к заданию формы отчета, причем каждый "диапазон" представляет уровень группировки. Не нравится простой список названий и чисел? Хорошо, а как насчет графи-



БАЗЫ ДАННЫХ

ка? Paradox может создавать на основе ваших данных самые разнообразные графики, в том числе столбиковые, колончатые, линейные и круговые диаграммы.

СЦЕНАРИИ, ПРОГРАММИРОВАНИЕ, PAL

Даже такая простая в пользовании программа может показаться надоедливой, если приходится снова и снова вводить одни и те же команды. Paradox предоставляет несколько способов программирования, различающиеся степенью сложности.

Программы СУБД Paradox называются сценариями (script), и простейший способ создания такой программы — ее "запись". Выберите пункт меню "начать запись" ("BeginRecord"), и программа будет запоминать все нажатия на клавиши,

пока вы не выберете пункт "закончить запись" ("EndRecord"). В результате получается текстовый файл, который можно изменять при помощи редактора сценариев пакета Paradox или любого текстового процессора. Аналогично можно "записать" и "проиграть" запросы.

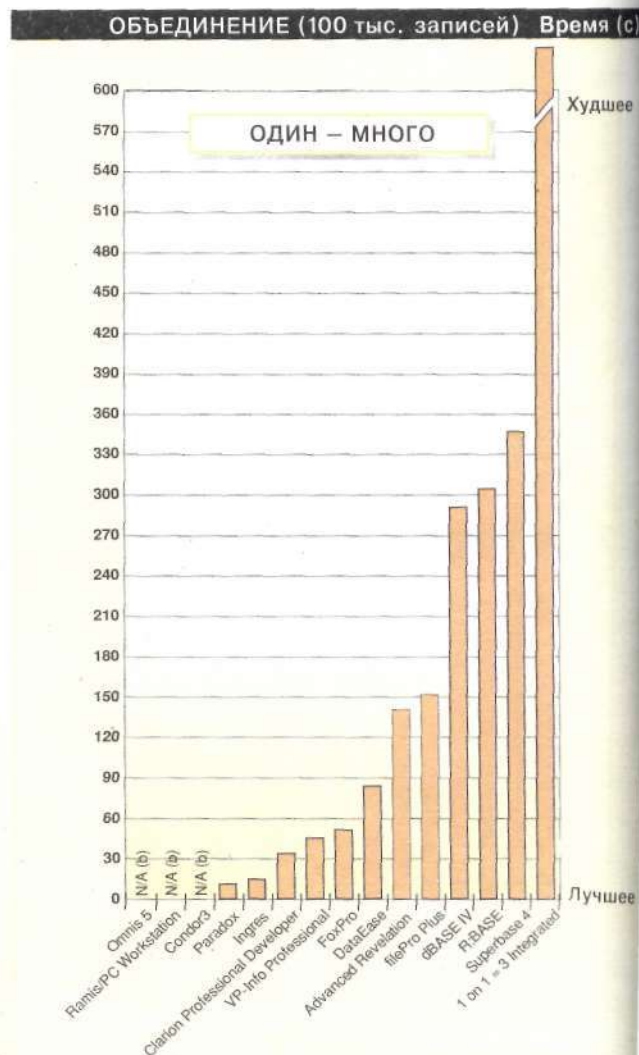
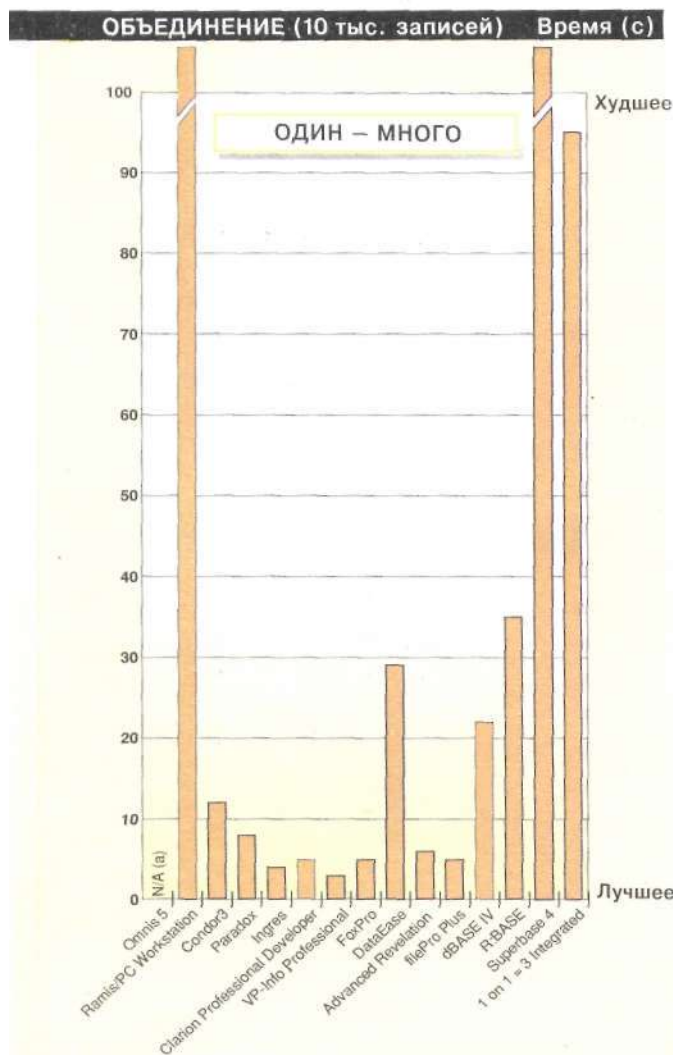
Следующий шаг в программировании — это "персональный программист" (Personal Programmer), который позволяет создавать собственные меню и процедуры, проводящие вас через серию операций СУБД Paradox, не требуя более глубоких знаний о программе. Это, собственно говоря, генератор программ, создающий сценарии, которые можно редактировать обычным образом, внося дальнейшие изменения.

И наконец, вы можете писать собственные процедуры на языке программирования прикладных систем PAL (Paradox Application Language). Это полнофункциональный язык десятками различных команд и вариантов работы. Кроме того, имеется пакет Paradox Engine —библиотек функций Си и процедур Паскаля, которыми можно пользоваться в одно- и многопользовательских прикладных системах.

ПОТРЯСАЮЩАЯ СКОРОСТЬ

Вы, возможно, ожидаете, что столь простая в пользовании программа должна быть медленной. Вероятно, вы думаете, что создание всех этих временных таблиц ответов и использования процедур, которые воспроизводят операции меню, требуют больших затрат времени, но вы ошибаетесь.

РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ: ПРОИЗВОДИТЕЛЬНОСТЬ



ТОЛЬКО ФАКТЫ

Изделие: Paradox, версия 3.5

Фирма: Borland International Inc.,
1800 Green Hills Rd., P.O. Box
660001, Scotts Valley, CA 95067;
408-438-5300.

Рекомендуемая цена: 795 долл.;
Paradox Multi-Pack (пять
пользователей) — 995 долл.;
Paradox Engine — 495 долл.;
Paradox SQL Link — 495 долл.

Требуемые ресурсы: 512 Кбайт,
DOS 2.0 или более поздняя версия.

В СУБД Paradox реализован особый подход к разработке баз данных, однако после того, как вы освоили его структуру, он обеспечивает впечатляющее сочетание простоты, мощности и скорости. Есть несколько "вывертов", но они незаметны на фоне множества привлекательных возможностей, предоставляемых пакетом.

Paradox работает очень быстро. На тестах лаборатории журнала PC Magazine скорость работы пакета была близка к максимальной по многим операциям, даже когда ему приходилось проходить промежуточные этапы, создавая временные таблицы. Наиболее явно это проявилось на тестах с 10 тыс. записей. На тестах по 100 тыс.

БАЗЫ ДАННЫХ

записей быстродействие было несколько дальше от рекордного. Однако даже при очень больших базах данных вы, скорее всего, сочтете быстродействие вполне удовлетворительным. На некоторых сложных связках он обходит даже FoxPro.

РАЗНИЦА В ФИЛОСОФИИ

При работе с программой возникают, однако, некоторые трудности. Философия, на которой основана ее структура, заметно отличается от традиционных СУБД и, возможно, потребует более длительного обучения. К чести фирмы Borland, ее документация — одна из самых привлекательных. Вместо традиционного скоросшивателя вам предлагаются книги в бумажной обложке размером 21,5 x 28 см, страницы которых полны полезных иллюстраций и наглядных примеров.

И все-таки остаются несколько сложных концепций, которые необходимо воспринять. Например, если у вас есть некий часто выполняемый запрос и вы хотите формировать с его помощью отчет определенного вида, придется это делать в несколь-

ко этапов. Сначала вы должны создать пустую таблицу такой же структуры, как получающаяся в результате выполнения запроса таблица ANSWER, и задать формат отчета для этой таблицы. Затем пишется сценарий, который открывает исходную таблицу, выполняет запрос, записывает таблицу ANSWER под другим именем, копирует формат отчета из пустой таблицы в переименованную таблицу ANSWER, а затем выводит отчет. Это не так уж трудно после того, как вы все это освоили, но требует определенной практики.

У пакета Paradox есть и многопользовательские средства, причем для работы в этом режиме используется тот же самый пакет, что и для одного пользователя. Необходимо либо иметь по одной копии на каждого пользователя, либо приобрести "пухлую" упаковку для установки в сети, но, по крайней мере, не приходится покупать другую версию.

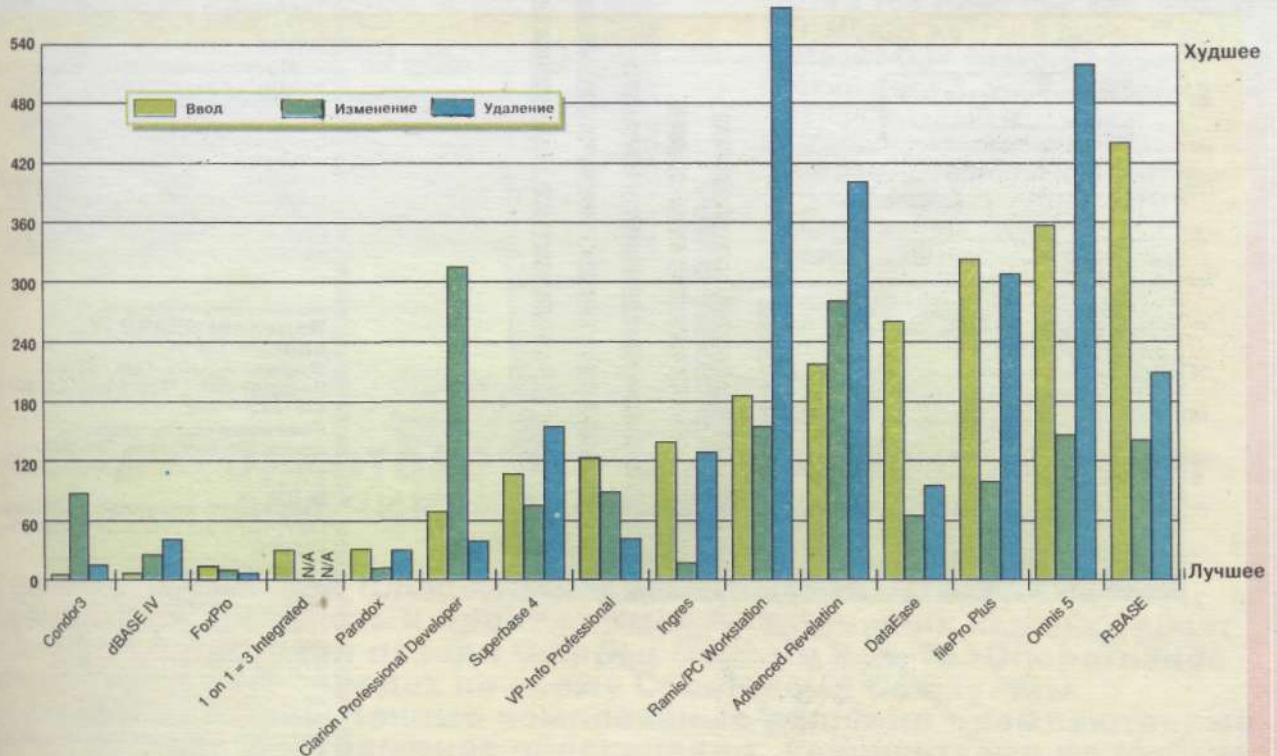
Чтобы освоить Paradox, требуется некоторое время, но затраченные усилия будут вознаграждены. Протрив свою "дорогу" по меню и структуре запросов, вы увидите, что Paradox — быстродействующая, мощная и простая в применении СУБД.



РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ: ПРОИЗВОДИТЕЛЬНОСТЬ

МОДИФИКАЦИЯ (10 тыс. записей)

Время (с)



N/A—Not applicable: Because of coding errors, we were not able to obtain a valid result.

БАЗЫ ДАННЫХ

ТОЛЬКО ФАКТЫ

Изделие: Advanced Revelation, версия 2.03
Фирма: Revelation Technologies Inc., 2 Park Ave., New York, NY 10016; 800-262-4747, 212-689-1000
Рекомендуемая цена: 995 долл.
Требуемые ресурсы: 640 Кбайт ОЗУ, DOS 3.1 или более поздняя версия.

Мощная система для разработки баз данных, ориентированная на профессиональных программистов. Пакет позволяет работать с четырьмя языками: R/BASIC, R/LIST, TCL и SQL и обращаться к символьным файлам и файлам dBASE III через так называемые "связки рабочей среды". За особую плату можно приобрести специальные "связки" для обращения к серверам SQL.

ТОЛЬКО ФАКТЫ

Изделие: Clarion Professional Developer, версия 2.1
Фирма: Clarion Software, 150 E. Sample Rd., Pompano Beach, FL 33064; 305-785-4555.

Рекомендуемая цена: 845 долл.
Требуемые ресурсы: 512 Кбайт ОЗУ, DOS 2.0 или более поздняя версия

СУБД, ориентированная на профессиональных и эпизодических разработчиков. Средства для разработки автономных исполнимых модулей делают эту систему особенно привлекательной для эпизодических разработчиков. Конечным пользователям баз данных приобретать эту СУБД не стоит.

ТОЛЬКО ФАКТЫ

Изделие: Condor3
Фирма: Condor DBMS Services Inc. 5209 North Clark St., Chicago, IL 60640; 312-271-8759.

Рекомендуемая цена: 395 долл.
Требуемые ресурсы: 336 Кбайт ОЗУ, DOS 2.1 или более поздняя версия

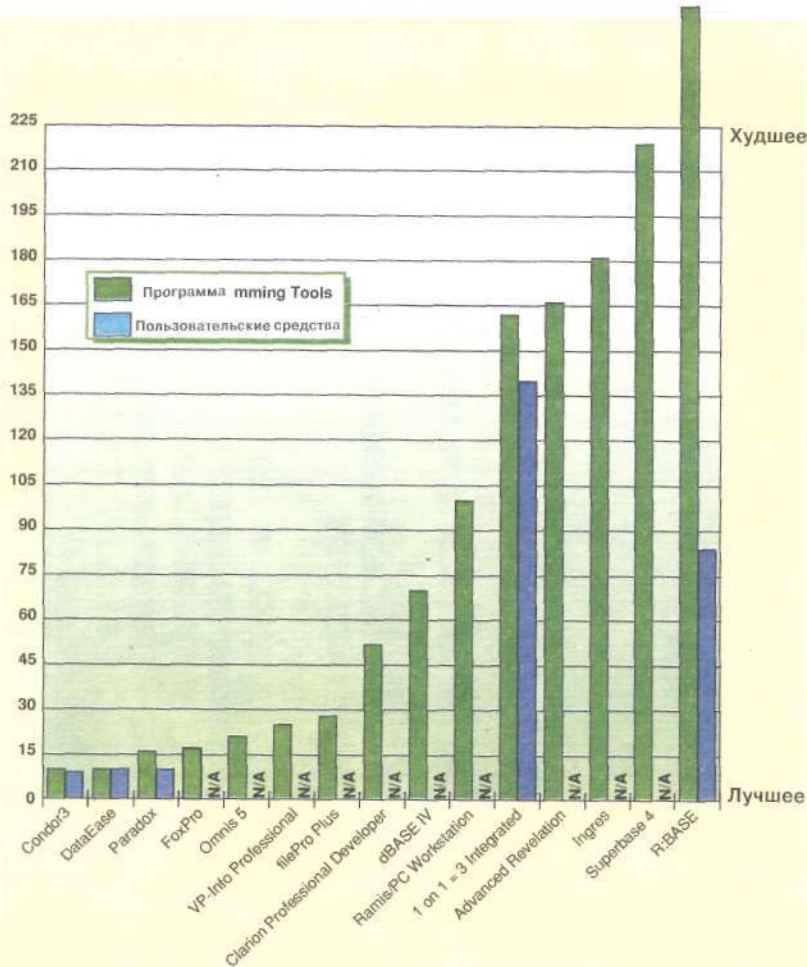
Эта СУБД наиболее эффективно работает с уровня исполнения команд, ее меню представляют собой не что иное как вариант обращения к командам. Вместе с тем язык программирования этой СУБД достаточно прост для освоения. Пользователь может разрабатывать достаточно сложные прикладные системы. Число недостатков следует отнести к предельному числу записей в файле, ограниченное 65 тыс. записей, и неудобную подсистему составления отчетов.



РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ: ПРОИЗВОДИТЕЛЬНОСТЬ

ОТЧЕТ (10 тыс. записей)

Время (с)



Изделие: DataEase, версия 4.2

Фирма: DataEase International Inc., 7 Cambridge Dr., Trumbull, CT 06611; 800-243-5123, 213-374-8000

Рекомендуемая цена: 795 долл.; комплект на три рабочих места 795 долл., комплект на 5 рабочих мест 1095 долл., комплект для разработки прикладных систем DataEase Developer (неограниченное число готовых исполнимых модулей) 795 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, DOS 3.1 или более поздняя версия

И для начинающих, и для опытных разработчиков, в однопользовательских и в многопользовательских системах DataEase позволяет строить сложные базы данных с минимальными затратами времени и сил. Немногие другие пакеты могут сравниться с этим по мощности и простоте в пользовании.

Изделие: dBASE IV, версия 1.1

Фирма: Ashton-Tate, 20101 Hamilton Ave., Torrance, CA 90501; 213-329-9989

Рекомендуемая цена: стандартная версия 795 долл., версия для разработчиков 1295 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, DOS 3.1 или более поздняя версия

Мощная СУБД, ориентированная на конечных пользователей, с комплектом средств для разработчиков. Хотя это наилучшая из всех СУБД фирмы Ashton-Tate, тем из пользователей dBASE IV, кто еще не успел приобрести эту последнюю версию, стоит подумать о другом пакете с более совершенными средствами разработки конечных систем.

САММИТ СИСТЕМС



Summit 286

12 или 16 МГц,
без периодов ожидания
Цветной монитор Super-VGA
(800 x 600 или 1024 x 768 точек)
52-Мбайт жесткий диск (17 мс)
1-Мбайт ОЗУ (макс. 8 Мбайт)
гибкий диск 5,25 дюйма
1 параллельный и
2 последовательных порта
Русифицированная клавиатура
(101 клавиша)
Лицензионная ДОС
2 года гарантии

Summit 386

33 МГц,
без периодов ожидания
Цветной монитор Super VGA
(1024 x 768 точек при 256 цветах)
105-Мбайт жесткий диск (17ms)
4-Мбайт ОЗУ (макс. 32 Мбайт)
гибкий диск 3,5 и 5,25 дюйма
1 параллельный и
2 последовательных порта
Русифицированная клавиатура
(101 клавиша)
Лицензионная ДОС
2 года гарантии

КАЧЕСТВО, КОТОРОЕ ВЫ ОЖИДАЕТЕ ТОЛЬКО ОТ АМЕРИКАНСКОЙ ПРОДУКЦИИ

Американский дизайн плюс лучшие западные компоненты - это высочайшая надежность и производительность наших компьютеров: Summit 286-12 оставил позади Compaq Deskpro 386-16. Оперативное обслуживание в 20 городах по всему Советскому Союзу. Мы предлагаем вам законченные комплексные решения - компьютерные сети, принтеры, программное обеспечение, консультации по любым вопросам.

Минск (0172) 973 119

Москва (095) 329 4655

ТОЛЬКО ФАКТЫ

Изделие: filePro Plus

Фирма: The Small Computer Co. Inc., 41 Saw Mill River Rd., Hawthorne, NY 10532; 800-847-4740, 914-769-3160

Рекомендуемая цена: 699 долл.

Требуемые ресурсы: 512 Кбайт ОЗУ, DOS 2.0 или более поздняя версия

СУБД filePro Plus содержит богатую систему меню, которая позволит вам выполнять большинство операций, не прибегая к программированию. Вместе с тем, пользуясь подобным Бейсику языком программирования этой системы, можно строить достаточно сложные базы данных. Программа позволяет хранить до миллиарда записей в одном файле, работать с многочисленными типами данных и разрабатывать собственный режим ввода информации.

Изделие: Omnis 5, версия 1.2

Фирма: Blyth Software Inc., 1065 E. Hillsdate Blvd., #300, Foster City, CA 94404; 415-571-0222

Рекомендуемая цена: 1000 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, 1,2 Мбайт расширенной памяти, оболочка MS Windows 2.x или 3.0, DOS 3.1 или более поздняя версия, желательно наличие мыши

Эффектное, но не слишком быстродействующее изделие для пользователей Windows, которые не нуждаются в разработке изощренных программ. В СУБД не предусмотрен режим работы через командные строки, так что приходится строить программу через последовательность меню. СУБД обеспечивает сетевой режим работы с одновременным подключением до 451 пользователя с возможностями блокировки файлов и записей и иных средств защиты.

Изделие: R:BASE, версия 3.1

Фирма: Microrim, 15395 S. E. 30th Place, Bellevue, WA 98007; 206-649-9500

Рекомендуемая цена: 795 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, DOS 3.1 или более поздняя версия

СУБД R:BASE имеет удобный в работе пользовательский интерфейс, но явно недостаточные средства программирования. Средства сопряжения с файлами dBASE делают эту систему хорошей альтернативой dBASE IV.

Изделие: 1 on 1 = 3 Integrated, версия 2.0

Фирма: 1 on 1 Computer Solutions Inc., 26 Finchwood Dr., Trumbull, CT 06611; 203-375-0914.

Рекомендуемая цена: 99 долл. в комплекте с программами Switcher, As-Easy-As и PC-Write; в комплекте с программой Switcher и с лицензией на As-Easy-As и PC-Write — 189 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, DOS 2.0 или более поздняя версия

Сохраняя совместимость с СУБД dBASE, 1 on 1 = 3 воспроизводит в несколько улучшенном варианте работу этой системы. СУБД 1 on 1 = 3 проще в использовании и работает несколько быстрее, чем dBASE, при обращении к интерактивным функциям. Самое слабое место этой программы — фирменное сопровождение, что не удивительно, когда речь идет о фирме, состоящей из одного человека, но принимающей во внимание цену, это приобретение — совсем не плохой вариант.

Изделие: Ingres, версия 5.0

Фирма: ASK Computer Systems Inc., Ingres Products Div., 1080 Marina Village Pkwy., Alameda, CA 94501; 415-748-2675

Рекомендуемая цена: 695 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, DOS 2.1 или более поздняя версия

СУБД Ingres обнаружила рекордное быстродействие и имеет неплохие функциональные возможности, но относительно слабые средства оформления. Эта СУБД будет очень хороша для тех, кто намерен хранить информацию в больших и мини-компьютерах, равно как и в микрокомпьютерах. Неизощренный пользовательский интерфейс наводит на мысли о применении системы в чисто деловых приложениях. Отсутствуют средства отладки или редактирования, но предусмотрены средства сопряжения с программами на Си и на языке ассемблера.

Изделие: Ramis/PC Workstation, версия 3.0

Фирма: On-Line Software International, Fort Lee Executive Park Two Executive Dr., Fort Lee, NJ 07024-800-526-0272, 201-592-0009

Рекомендуемая цена: 595 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, DOS 3.0 или более поздняя версия

Простая в использовании база данных с применением оконных меню, предназначенная для обработки запросов и формирования отчетов для конечных пользователей. Отсутствие достаточно развитых средств разработки не позволяет порекомендовать эту СУБД для применения в случаях, когда требуются более или менее богатые возможности.

Изделие: Superbase 4, версия 1.2.

Фирма: Precision Inc., 8404 Sterling St., Irving, TX 75063; 214-929-4888

Рекомендуемая цена: 695 долл.

Требуемые ресурсы: 640 Кбайт ОЗУ, оболочка MS Windows 2.x или 3.0, DOS 3.2 или более поздняя версия

Мощная СУБД для работы под управлением Windows, содержащая многочисленные средства, ориентированные на графический режим работы. Если вы до этих пор работали с ориентированными на текстовый режим базами данных, освоение графики может потребовать некоторого времени.

Изделие: VP-Info Professional, версия 4.23

Фирма: Sub Rosa Publishing Inc., 1170 Sheppard Ave., West, #34, Downsview, Ontario, Canada M3K 2A3; 416-398-8414

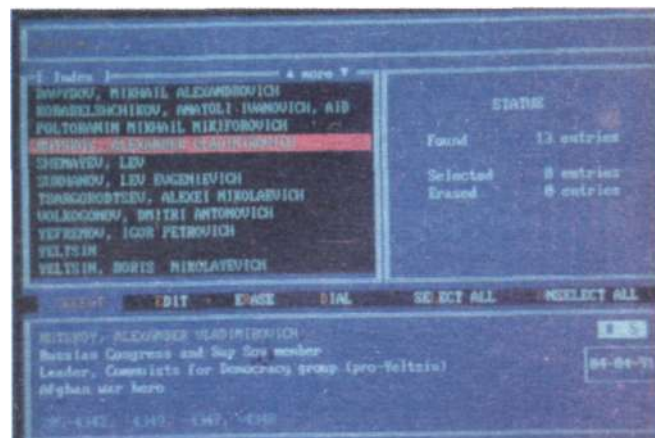
Рекомендуемая цена: 189 долл.

Требуемые ресурсы: 512 Кбайт ОЗУ, DOS 2.0 или более поздняя версия

Для работы с VP-Info не обязательно быть программистом, требуется привычка к вводу информации через командную строку, а не путем заполнения экранных бланков и выбора пунктов меню. В состав пакета входит компилятор с богатыми средствами языка, что позволяет рекомендовать СУБД VP-Info в основном программистам.

Алло! СУБД слушает вас

Юрий Филипчук



Каждому из вас зачастую приходится мучиться, отыскивая нужный телефонный номер по толстенному справочнику или в истрепанной записной книжке. И здесь облегчить вам жизнь (особенно если у вас есть компьютер, подключенный к телефонной линии) сможет новая программа telBASE, созданная московской группой разработчиков ! Brainy Instruments Laboratory (BIL). Она представляет собой специализированную СУБД с фиксированным форматом записи, ориентированную на хранение телефонных номеров и адресов абонентов с ассоциированной краткой "карточкой".

В программе реализованы все основные функции СУБД: манипулирование записями (добавление, удаление и редактирование), а также генерация простых отчетов. Пожалуй, единственное "необычное" качество программы — способность автоматически набирать номера с использованием модема, но и оно не является уникальным, поскольку уже есть ряд коммерческих телефонных СУБД, обладающих той же способностью.

Тем не менее программа отличается рядом свойств, которые позволили нам отнести ее к разряду новинок. Прежде всего, это ее исключительно малый размер (примерно 70 Кбайт) и низкие требования к объему памяти (минимум 85 Кбайт, рекомендуется 128 Кбайт). В наш век графических интерфейсов, когда стандартом становится уже 4 Мбайт, многие фирмы — разработчики

программного обеспечения склонны забывать об экономии компьютерных ресурсов. В то же время программа telBASE, не оставаясь резидентной, может быть вызвана практически из любых современных редакторов текста, электронных таблиц, больших СУБД или других программ, являющихся основными для пользователей различных профессий.

Корреспонденты, секретари, референты оценят (и уже оценили) возможность оперативно получить справку о необходимых в данный момент телефонных номерах, не покидая "любимого" пакета WordPerfect и даже не сохраняя текста на экране.

Программа telBASE отличается очень простым и удобным интерфейсом пользователя. Если в результате запроса найдено несколько записей, пользователь может просматривать их по "оглавлению", состоящему из первых строк каждой записи. Еще одно отличие программы telBASE — отсутствие жесткого деления записи на поля имени, фамилии, организации и т.п. Не нужно также специально указывать ключевые слова, по которым будет осуществляться поиск. Единственное ограничение — адрес должен находиться в пятой строке, а телефонные номера — в шестой. Количество и размер телефонных номеров ограничивается только длиной строки. Разрешаются комментарии типа "раб.", "дом." или "звонить после 12 дня" — программа автоматически выделяет телефонные номера,

подвергая соответствующую строку записи синтаксическому анализу. Далее программа автоматически разбивает запись на слова, отбрасывая при этом так называемый шум (предлоги, союзы...), и каждое из них трактует как ключевое. Таким образом, возможность "потерять" запись в базе данных практически исключена.

Программа telBASE может использоваться в любой локальной сети или на отдельных компьютерах. В последнем случае функция объединения баз данных, позволяющая выявлять и устранять повторы и редактировать различные версии одной и той же записи, служит инструментом для периодического "обновления" баз данных на каждом компьютере.

Достоинства программы telBASE можно перечислять долго, но, к сожалению, имеются и недостатки. Во-первых, программа, поддерживающая как русские, так и английские символы в записях, общается с пользователем исключительно по-английски. Во-вторых, программа использует свой формат базы данных, не совместимый ни с одним стандартом (правда, в будущей версии должна появиться утилита обмена с dBASE). В-третьих, отчеты выводятся на принтер в простом формате ASCII. Этот список можно продолжать, не забывая при этом, что реализация новых функций неизбежно приведет к увеличению объема программы. Остается добавить, что следующую версию этой программы BIL обещает представить в октябре 1991 г.

ПЕРЕНАЗНАЧЕНИЕ И КОНВЕЙЕРНАЯ ОБРАБОТКА

Время от времени мы получаем от наших читателей письма, в которых они упрекают нас в том, что мы мало задумываемся над уровнем их подготовки и знаний. Им кажется, что нас больше заботят потребности опытных пользователей и мы упускаем из виду, что журнал читают и начинающие, и пользователи среднего уровня, которые не всегда могут отличить режим ожидания от состояния прерывания. Вот почему рубрика Наставник обратит ваше внимание на предмет, столь же фундаментальный для DOS, как сама командная строка, но часто неправильно понимаемый и новичками, и корифеями. Это процедуры переназначения и конвейерной обработки.

Возможно, вы встречались с ними раньше. Это длинные командные строки, содержащие команды DOS, знаки "больше" и "меньше", а также, вертикальную черту.

Например, такие команды, как

```
TYPE TEXTFILE.TXT | MORE
```

которая показывает за один раз лишь уместную часть текста файла, а не сплошной текст, который можно просматривать, а также

```
COPY »» F:\ > NUL
```

которая копирует содержимое текущего каталога на диск F: но при этом не дает появиться сообщению "files sorted", которое обычно засоряет экран после исполнения команды. В каждом из этих случаев выходные данные одной команды перехватываются и передаются не на экран, а куда-то еще. В первом случае выходные данные одной команды фактически использовались как исходные для другой. Но это лишь отдельные примеры того, что дают возможности DOS по переназначению источника и адресата. Мы изучим процедуру переназначения в более широком смысле и попытаемся нарисовать более точную картину вариантов ее использования.

Два канала, которые DOS использует для большей части входящей с клавиатуры и выводимой на экран информации, известны как стандартный ввод и стандартный вывод. DOS сохраняет внутреннюю таблицу, в которую записываются отношения между "указателями" файлов, т.е. специфическими номерами, закрепленными за ними системой DOS и

■ Переназначение и конвейерная обработка. Умение осуществлять эти процедуры — первый шаг к лучшему использованию вашей системы.

используемыми программами для обращения к файлам или устройствам, открытым посредством внутренней функции DOS Open File. Указатели файлов 0 и 1 зарезервированы за стандартными вводом и выводом. Эти два указателя автоматически отображаются в драйвере устройства CON (консоль) (или ANSI.SYS, если это "устройство" было установлено директивой DEVICE= в файле CONFIG.SYS). Таким образом, данные, передаваемые на стандартное устройство вывода, обычно отображаются на экране, а данные, считываемые со стандартного ввода, принимаются с клавиатуры. То есть, по сути дела, переназначение осуществляет переадресацию данных, идущих на стандартный вывод и получаемых со стандартного ввода.

DOS обеспечивает три различных операции переназначения:

> операция переназначения выходных данных, которая переназначает вывод результатов исполнения команд с экрана на указанный вами объект;

< операция переназначения исходных данных, которая обеспечивает прием информации не с клавиатуры, а от другого объекта;

| операция конвейерной обработки, которая реализует цепочку последовательных процессов, когда выход одного звена цепочки подается на вход другого.

Эти операции придают системе дополнительную гибкость. Например, используя переназначение выхода, можно занести результаты работы такой команды DOS, как CHKDSK, в файл или перенаправить их на принтер. Исполняя процедуру переназначения исходных данных, вы можете заменить вводимые с клавиатуры команды на хранящиеся в файле. Переназначение выходных данных направляет вывод DOS, направленный на экран, и перенаправляет его в другое место. Переназначение исходных данных позволяет специфицировать для них отличный от клавиатуры источник. Конвейерная обработка соединяет в себе переназначение и исходных, и выходных данных, так чтобы экранный вывод одной команды становился исходными данными для другой.

Когда дешифратор файла **COMMAND.COM** сталкивается с каким-либо символом переназначения в командной строке, он понимает, что вы хотите осуществить переназначение. Поэтому эти символы нельзя использовать в именах файлов или для любых других целей в командной строке, за исключением осуществления процедур переназначения и конвейерной обработки.

Когда в командной строке встречается символ < , DOS закрывает указатель файла, соответствующий стандартному вводу, и открывает для доступа файл или устройство с заданным в строке именем. Далее, когда программа или команда DOS, заданная слева от символа переназначен, читает со стандартного ввода, она как ни в чем ни бывало, принимает исходные данные от файла или устройства, подменяющего консоль. То же самое происходит при обнаружении оператора > . Отличие заключается лишь в том, что закрывается и перемещается указатель файла, по умолчанию закрепленный за стандартным выводом.

ПЕРЕНАЗНАЧЕНИЕ ВЫВОДА

Обычный синтаксис для переназначения выхода данных команды DOS следующий:

```
команда > адресат
```

где команда — команда, выход которой вы хотите переназначить, а адресат — файл или устройство, куда будет передаваться информация, Команда

```
DIR L: > LPT1
```

посылает перечень содержимого диска A: на принтер, имеющий идентификатор LPT1. Если принтер не был переназначен при помощи команд **MODE**, то команда

```
DIR A: < PRN
```

сделает то же самое. На рис. 1 приводится перечень всех драйверов физических устройств, которые DOS инициализирует каждый раз при запуске системы. Вы можете использовать любое из них в качестве направления для переназначения выходных данных. По умолчанию данные, направленные на стандартный вывод, идут на экран через CON, так что нет необходимости переназначать выходные данные на CON. Единственное возможное исключение составляет тот случай, когда стандартные вход и выход были приспосаблированы другому устройству командой **CTPV**. В этом случае вы все же можете по

сылать информацию на экран переназначением выхода на CON.

Устройство NUL дает вам удобную возможность избежать сообщений DOS. Все надписи на экране просто исчезают. Если вы запустили командный файл, в котором несколько команд COPY копируют файлы в оперативную память, и вы не хотите, чтобы на экране появлялось сообщение "files copied", то вы можете переназначить выходную информацию команды COPY в NUL.

Например, команды

```
COPY C:\CAD\EXE F:\ > NUL
COPY C:\CAD\OVL F:\ > NUL
COPY C:\CAD*.DAT F:\ > NUL
```

копируют все файлы с расширениями .EXE, .OVL, .DAT из указанного каталога на диск F:, но делают это не выдавая сообщений, которые обычно сопровождают завершение каждой команды. Однако, обратите внимание на то, что сообщения об ошибках команд DOS все же будут появляться на экране, так как они идут на драйвер устройства CON через еще один специально отведенный для этих сообщений канал.

Если параметр назначения, стоящий справа от знака, не соответствует ни одному из резидентных драйверов DOS или драйверов, установленных в системе через директиву DEVICE=, DOS воспринимает его как имя файла и заносит результаты исполнения команды в этот файл.

Команда

```
DIR A: > DIRLIST.TXT
```

записывает содержимое диска A: в файл DIRLIST.TXT. Если DIRLIST.TXT не существует, DOS создает его. Если файл существует, информация записывается поверх старой, которая будет утрачена. Вы можете заставить DOS просто добавить выходную информацию в конец, удвоив оператор >.

Например, команда

```
DIR A: > DIRLIST.TXT
```

добавляет новую информацию в конец существующего файла. Если DIRLIST.TXT не существует, DOS создаст его и в этом случае.

Важно отметить, что переназначение выходных данных хорошо использовать лишь для перехвата и переназначения вывода, адресуемого на экран. Начинаящие пользователи часто думают, что переназначение может быть использовано для передачи информации в файл вместо печати на принтере. Это невозможно. Такая процедура требует наличия отдельной утилиты, которая составлена специально для перехвата данных, передаваемых через прерывание 17h (прерывание по выходу принтера в

BIOS), как, например, PRN2FILE. Если вы запомните, что DOS дает возможность для переадресации данных, только поступающих на стандартный вывод или приходящих со стандартного входа, то вам всегда будет ясно, применима ли данная процедура в некотором конкретном случае.

ПЕРЕНАЗНАЧЕНИЕ ИСХОДНЫХ ДАННЫХ

Переназначение исходных данных совершенно аналогично переназначению выходных данных, за исключением того, что в этом случае DOS ищет отличный от клавиатуры источник информации. Синтаксис команды по переназначению исходных данных следующий:

команда < источник

где источник — файл или устройство, с которого поступает исходная информация. В качестве идентификатора источника, так же, как и адресата (при переназначении вывода), может использоваться имя файла или название любого физического устройства из перечисленных на рис. 1.

Наиболее часто переназначение исходных данных применяют для передачи данных из текстового файла команде DOS. Например, команда FORMAT делает паузу после форматирования диска и "спрашивает", хотите ли вы отформатировать следующий. Для возвращения в режим ввода команд приходится нажимать N. Это неудобно, если вы хотите форматировать диски через командный файл, который должен исполняться от начала до конца без вмешательства оператора. Чтобы избежать этого, можно создать текстовый файл, содержащий только букву N и символ возврата каретки (назовем его NO.TXT), и задавать форматирование командой

В этом случае, когда форматирование обратится за данными, будет принято N из текстового файла. Вы можете запускать все команды DOS, которые требуют ввода исходных данных с клавиатуры, предварительно помещая клавиатурную команду в текстовый файл и делая переназначение. Например, PC Magazine часто печатает сценарии DEBUG, которые создают исполнимые файлы с расширением .COM. Вместо того, чтобы вводить эти сценарии прямо в DEBUG, гораздо проще ввести те же команды в файл и воспользоваться переназначением исходных данных для передачи их в DEBUG.

Предположим, вы хотите, чтобы следующий сценарий создавал утилиту под названием KBCLEAR.COM, которая очищает буфер клавиатуры от ненужной информации, введенной с нее (такая утилита может пригодиться, например, между программами, запускаемыми из командных файлов, для того чтобы введенные с клавиатуры данные для одной программы не повлияли на исполнение следующей):

```
A 0100
MOVAX.40
MOVDS.AX
MOVAX,[1A]
MOV[1C]AX,
RET

N KBCLEAR.COM
RCX
C
W
Q
```

Вы можете запустить DEBUG и набирать команды в точности, как показано. Но более удобный способ — создать файл с именем KBCLEAR.SCR, который содержит эти команды и вместо приведенных выше команд, набрать

Драйверы системных устройств

Драйвер	Устройство
NUL	Пустое устройство (физический объект отсутствует)
CON	Консольное устройство (экран или клавиатура)
AUX	Вспомогательное устройство (обычно соответствует CON1)
PRN	Принтер (обычно соответствует LPT1)
COMx	Последовательный порт (COM1, COM2, COM3 или COM4)
LPTx	Параллельный порт (LPT1, LPT2, LPT3 или LPT4)
CLOCKS	Часы реального времени

Рис.1: Система DOS устанавливает перечисленные драйверы после каждого запуска, чтобы обеспечить программам связь с экраном, клавиатурой, часами реального времени, последовательными и параллельными портами, а также принтерами. Любое из этих устройств может играть роль источника или адресата информации при переназначении.

```
DEBUG < KBCLEAR.SCR
```

Каков же будет результат? DEBUG исполняет команду и создает файл с расширением .COM. Более того — если во введенной вами команде была ошибка, вам не потребуется "входить" в DEBUG и набирать все сначала. Просто отредактируйте текстовый файл и снова исполните ту же команду DEBUG, чтобы перекомпилировать утилиту.

Можно передавать исходные данные команде DOS и от таких устройств, как последовательные порты (по крайней мере теоретически). Команда

```
DEBUG < COM1
```

заставит DEBUG обратиться к COM1 за командой. Но такая операция используется редко.

КОНВЕЙЕРНАЯ ОБРАБОТКА

Конвейерная обработка соединяет в одной операции переназначение и исходных, и выходных данных, перехватывая выход одной команды и передавая его другой команде в качестве операнда. Синтаксис команды конвейерной обработки следующий:

```
команда J \ команда _2
```

где команда_1 — команда, вывод которой переназначается, а команда_2 — команда, которая получает переназначенный вывод в качестве исходных данных. Типичный пример — строка:

```
DIR | MORE
```

которая показывает содержание текущего каталога частями по одному экранному кадру. Как это происходит? DIR дает весь перечень, но операция конвейерной обработки перехватывает выход из DIR и перенаправляет его в утилиту MORE.COM. (MORE хранится вместе с другими файлами DOS на жестком диске.) MORE воспринимает перечень как исходную информацию и показывает его порциями по одному кадру, всякий раз дожидаясь, пока вы нажмете на клавишу для дальнейшего просмотра. Использование операции конвейерной обработки, таким образом, эквивалентно использованию отдельных операций переназначения выходных и исходных данных. Например, того же эффекта, что и для предыдущей операции, мы могли бы добиться посредством команд:

```
DIR > TMPFILE.TXT  
MORE < TMPFILE.TXT
```

НАСТАВНИК

Но зачем нужны два этапа, когда все можно уместить в один? Конвейерная обработка избавляет от необходимости создавать временный файл, чтобы держать выходные данные команды DIR.

С точки зрения пользователя конвейерная обработка отличается от переназначения выходных и исходных данных тем, что предметом переназначения должно быть имя файла или физическое устройство, в то время как параметр, стоящий справа в командной строке, должен представлять собой название программы. Наиболее часто исходные данные передаются посредством конвейерной обработки фильтровым командам DOS: FIND, MORE и SORT. Фильтрами называ-

Для начинающих пользователей одна из главных загадок работы с DOS — вопрос о том, когда переназначение работает, а когда нет.

ются программы, которые обрабатывают потоки символов, считываемых со стандартного входа, и записывают модифицированные потоки на стандартный выход. Обычно фильтры не могут сами запрашивать исходные данные и обрабатывают исходные данные, поступающие в процессе конвейерной обработки от других команд DOS (FIND представляет собой исключение: в отличие от MORE и SORT, она может запуститься как автономная команда).

Команда

перечисляет файлы текущего каталога в алфавитном порядке по именам, в то время, как команда

```
DIR | SORT /+13
```

дает перечень, рассортированный по объему. (Цифра 13 заставляет SORT организовать перечень на основе символов начиная с 13 позиции строк, т.е. с той, где отображаются объемы файлов.) Вы можете объединить многочисленные операции конвейерной обработки (и переназначения) в одной строке для того, чтобы далее модифицировать команды DOS. Команда

```
DIR | SORT /+13 | MORE
```

дает перечень файлов в порядке увеличения длины и выводит их порциями по одному экранному кадру. Как видно, возможности практически неограничены.

Если вы программируете на Си, ассемблере или любом другом языке, который позволяет читать со стандартного входа и записывать через стандартный выход, то сможете и самостоятельно создавать фильтровые команды. В 13 главе книги Рэя Дункана "Advanced MS-DOS", опубликованной в издательстве Microsoft Press, содержится отличное объяснение того, как работают фильтры и как они написаны.

ПОЧЕМУ ПРОЦЕДУРА ПЕРЕНАЗНАЧЕНИЯ НЕ ВСЕГДА СРАБАТЫВАЕТ

Для начинающих пользователей одна из главных загадок работы с DOS — вопрос о том, когда переназначение работает, а когда нет. Процедура срабатывает с командами DOS и несколькими утилитами командной строки, но не срабатывает при попытке перехвата выходных данных таких прикладных программ, как Lotus 1-2-3 и WordPerfect. Понимание того, как программы DOS выдают информацию на экран и считывают с клавиатуры, поможет вам понять и то, почему операция переназначения в одних случаях срабатывает, а в других нет.

Программы, работающие в DOS, используют несколько способов для вывода на экран. Они могут использовать любой из предоставляемых DOS способов вывода символов на экран, а также подпрограммы, занятые в видео-BIOS, или записывать данные прямо в видеобuffer. Преимущество прямой записи — быстрота, в то время, как BIOS предлагает разумный компромисс между скоростью и простотой программирования. В то же время серьезные программы по выводу символов на экран работают медленно, так что лишь немногие программы, кроме утилит командной строки, используют их. Аналогично большинство программ использует вектор прерывания в BIOS для чтения клавиатуры, а не имеющиеся в DOS сервисные программы клавиатуры. Ключ к предсказанию работы переназначения в знании того, что лишь программы, использующие возможности DOS по осуществлению ввода и вывода данных, могут быть кандидатами на переназначение.

Вот пример, демонстрирующие разницу. Следующий набор команд DEBUG создает маленькую программу HELL01.COM, которая пишет на экране "Привет всем!", используя прерывание DOS 21h и функцию 0Ah для вывода символов на экран:

НАСТАВНИК

```
A 100
JMP010F
DB "Привет всем!"
MOV AH, 9
MOV DX, 0102
INT 21
RET
```

```
NHELLO1.COM
RCX
17
W
Q
```

После того как HELLO 1 скомпилирована, ввод команды

```
HELLO1 > HELLO.TXT
```

будет иметь своим результатом создание текстового файла, содержащего строку "Привет всем". Теперь введите

```
A100
JMP010F
DB "Привет всем!"
CLO
MOVSI.0102
LODSB
CMPAL.24
JZ 011E
MOVAN.0E
INT10
JMP0113
```

```
RET
```

```
N HELLO2.COM
RCX
1F
W
Q
```

Утилита DEBUG по этим командам создаст вторую программу под названием HELLO2.COM. HELLO2 делает то же самое, что и HELLO 1, но несколько по-другому. Вместо использования DOS для вывода на экран текста "Привет всем!", HELLO2 использует вектор прерывания 10h и функцию OEH, которая содержится в BIOS. Теперь посмотрим, что произойдет при попытке переназначения вывода HELLO2 с помощью команды

```
HELLO2 > HELLO.TXT
```

DOS образует новый файл HELLO.TXT, но этот файл будет пустым. Более того, "Привет всем" все равно появится на экране, хотя бы вы попытались переназначить вывод этого текста в файл. Что же происходит? Ответ прост. Так как HELLO2 использует подпрограмму BIOS, а не DOS, DOS даже "не знает" о том,

что строка появляется на экране. А так как система не знает, она не может записать выход в файл.

Последнее, о чем надо сказать, говоря о переназначении — это резидентные программы (TSR). Попытки переназначить вывод резидентных программ редко бывают успешными, так как очень редко резидентные программы используют функции консоли в DOS по выводу на экран. Но существует и более серьезная проблема. В результате переназначения вывода резидентных программ указатель файла фиксируется навсегда, так как DOS возвращает указатель, используемый для переназначения входа и выхода, в исходное состояние лишь после завершения соответствующей программы, а резидентная программа по определению не завершается. Это может повлиять на дальнейшую работу системы, если указатель файла потребуется другой программе, а она не сможет его получить, так как система уже распределила все возможные указатели.

Знание того, как осуществлять операции переназначения и конвейерной обработки — еще один шаг к более эффективному использованию вашей системы. И теперь, когда вы получили такие знания, их применение не покажется вам таким сложным, как могло показаться на первый взгляд.

(начало см. стр 76)

Даже говоря о "засилье" DOS в 80-х годах, не следует забывать об операционной системе компьютера Macintosh фирмы Apple, захватившего не очень большую, но заслуживающую внимания часть рынка ПК. В результате на компьютер Mac приходится доля — и зачастую весьма значительная — парка персональных компьютеров в большинстве крупных организаций и во многих небольших компаниях.

В зависимости от нужд, накопленного программного обеспечения и установленного оборудования того или иного подразделения предприятия наилучшим выбором для него может стать одна или несколько из упомянутых здесь ОС. Попытка навязать один-единственный ответ, исключая другие варианты, редко приводит к хорошим результатам.

Для технического отдела, уже хорошо знакомого с ОС Unix по работе на высокопроизводительных станциях, одна из версий этой операционной системы для компьютера IBM PC и совместимых с ним машин может оказаться наилучшим вариантом при выборе среды для прикладных программ автоматизации учрежденческого труда. Отдел, занятый художественно-оформительскими работами и уже использующий системы на базе

ПК Macintosh, получит максимальный выигрыш, расширяя сферу использования именно этих машин.

Для многих пользователей дальнейшее применение DOS или одного из альтернативных вариантов этой ОС может стать вполне разумным решением на весьма продолжительный срок. Для других же удачной находкой станет Windows или OS/2 — в зависимости от того, чем они занимаются или намерены заниматься в будущем и чему отдадут предпочтение.

Есть еще ряд других, весьма убедительных причин, по которым вовсе не обязательно, чтобы какая-то одна ОС стала лидером, затмив собой конкурентов. Речь идет о растущем числе программных изделий независимых поставщиков, способных работать в нескольких операционных средах, о стандартизации форматов файлов данных, что улучшает их переносимость из одной ОС в другую, и о широком распространении сетевых протоколов, обеспечивающих возможность общения между компьютерами, которые работают под управлением различных операционных систем.

Приведенные здесь доводы, конечно же, не остановят нас, и мы будем продолжать изучать и сравнивать различные операцион-

ные системы, размышляя над их достоинствами и недостатками и пытаясь определить победителей и проигравших. Они далеко не одинаковы по своим возможностям, но тем не менее безусловного лидера среди этих ОС быть не может. В конце концов некоторые системы могут оказаться на обочине магистрального пути развития компьютерной техники.

Нет оснований полагать, что в 90-х годах одна из операционных систем обязательно станет лидером, подобно тому как в мировом чемпионате по бейсболу 1991 г. будет только один победитель. Подобный взгляд открывает перед нами широкий спектр возможностей и, вероятно, позволит избежать необоснованного сужения нашего выбора, когда разумнее опереться на несколько операционных систем.

Для организации (а часто и для отдельного пользователя) смысла в ограничении себя возможностями одной ОС не больше, чем в решении плотника пользоваться только молотком и обходиться без других инструментов, скажем отвертки и пилы. Задача не в том, чтобы найти инструмент, способный делать все, — надо просто выбрать наиболее подходящий инструмент для конкретной работы.

ЗНАКОМСТВО С СИСТЕМОЙ BIOS Джефф Просис

Базовая система ввода-вывода (BIOS — Basic Input Output System) называется так потому, что включает в себя обширный набор программ ввода-вывода, путем вызова которых прикладные программы и операционные системы могут взаимодействовать с различными устройствами, подключенными к ПК. Сам термин BIOS был заимствован, вероятно, из системы CP/M, где программно реализованный модуль с подобным названием выполнял машинно-зависимые функции взаимодействия ОС с аппаратными средствами компьютера. В персональных компьютерах система BIOS хранится в постоянном запоминающем устройстве (ПЗУ, по-английски ROM) и поэтому часто называется ROM BIOS.

Система BIOS реализована в виде одной или нескольких интегральных схем, установленных на системной плате компьютера. Некоторые видеоадаптеры, например EGA и VGA, а также дисковые контроллеры имеют собственную систему BIOS, которая дополняет основную BIOS. Во многих случаях программы, входящие в BIOS плат-адаптеров, используются вместо программ системы BIOS, находящейся на системной плате. Два основных изготовителя персональных компьютеров — компании IBM и Compaq — написали свои собственные варианты BIOS. Большинство менее крупных компаний используют в выпускаемых ими ПК систему BIOS, совместимую с ROM BIOS таких компаний, как Phoenix, Award и AMI. Компания IBM поместила снабженные комментариями листинги исходных текстов программ BIOS для ПК типа IBM PC, XT и AT в технической документации на эти машины. Если последняя имеется в вашем распоряжении и вас не смущает перспектива изучения длинных программ, написанных на языке ассемблера, то вы можете самостоятельно разобраться в коде BIOS и получить представление о том, как BIOS взаимодействует с различными устройствами.

Вызов программ BIOS осуществляется через программные прерывания — команды центрального процессора, с помощью которых выполнение текущей программы прерывается и управление передается программе обработки прерываний, находящейся в другом месте памяти машины. При запуске ПК система BIOS инициализирует в таблице векторов прерываний (занимающей первые 400 байт памяти) определенные векторы, слу-

■ **ЗНАКОМСТВО С СИСТЕМОЙ BIOS:**
Прочтя эту статью, вы узнаете, каким образом прикладные программы и операционные системы взаимодействуют с различными устройствами, подключенными к вашему ПК.

жащие для указания местонахождения различных программ обработки прерываний. Когда DOS или прикладная программа выдает запрос на прерывание, происходит вызов программы обработки прерываний с требуемым вектором. Далее обработчик прерываний выполняет определенную функцию, код которой хранится в регистре AH. В этом случае физический адрес местонахождения программы обработки прерываний не важен: прикладной программе достаточно "знать" код функции и номер прерывания для вызова необходимой функции.

С такими устройствами, как дисплей, клавиатура и принтер, DOS общается исключительно через программы BIOS. Например, для того чтобы вывести символ на экран дисплея, DOS пересылает ASCII-код этого символа драйверу устройства CON ("консоль", т.е. дисплей + клавиатура) с помощью прерывания 29h, иногда называемого Fast Console Output Interrupt (прерывание для быстрого вывода на консоль). Далее для отображения символа на экране драйвер CON вызывает BIOS-функцию Write Teletype — Запись на терминал (прерывание 10h, функция 09h), — находящуюся в системе BIOS видеоадаптера. Если вместо CON используется драйвер ANSI.SYS, то аналогичное действие выполняется с помощью BIOS-функции Write Character and Attribute — Запись символа и атрибута (прерывание 10h, функция 09h).

Аналогичным образом, драйвер PRN использует прерывание 17h — обращение к сервисным функциям BIOS, служащим для организации взаимодействия с принтером, — для вывода символов на принтер. И лишь в одном, крайне редко применяемом режиме — если DOS непосредственно обращается к системе BIOS, а не к одному из своих собственных драйверов устройств, она вызывает BIOS-функцию Scroll Active Video Page Up (Прокрутка активной видеостраницы вверх) для очистки экрана в том случае, когда вы вводите с клавиатуры символы CLS (если только вы не загружали ANSI.SYS; в этой ситуации сброс экрана выполняется путем пересылки Esc-последовательности

сброса экрана в драйвер ANSI.SYS); Прикладные программы ведут себя не столь корректно. Для взаимодействия с аппаратными средствами они могут использовать либо соответствующие функции DOS, либо программы BIOS, либо обращаться к этим устройствам непосредственно — в зависимости от того, что удобнее быстрее. Для обращения к устройствам ввода-вывода большинство прикладных программ использует систему BIOS, а не функции DOS, так как первая, как правило, отличается более мощным набором функций и менее подвержена внешним воздействиям типа нажатия комбинации клавиш Ctrl-C и операторов переадресации потоков ввода-вывода. Но в некоторых случаях недостаточно даже функций BIOS. Например, большинству прикладных программ, обращающихся к последовательным портам, программисты микросхем универсального асинхронного приемопередатчика самостоятельно, поскольку встроенные в DOS и BIOS примитивные функции последовательного ввода-вывода не обеспечивают скорость обмена данными более чем 1200 бит/с]

ФУНКЦИИ BIOS

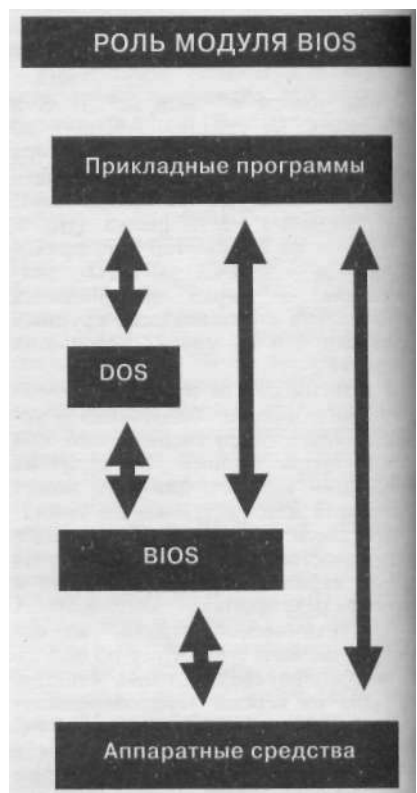


Рис. 1: Модуль BIOS содержит набор программ, которые операционная система и прикладные программы используют для связи с аппаратными элементами.

за исключением нескольких специальных функций сервисные функции ввода-вывода в системе BIOS можно разделить на семь основных групп:

- I управление дисплеем (прерывание 10h)
- управление диском (прерывание 13h)
- управление последовательным портом (прерывание 14h)
- системные функции (прерывание 15h)
- I управление клавиатурой (прерывание 16h)
- I управление параллельным принтерным портом (прерывание 17h)
- запрос/установка времени и даты (прерывание 1Ah)

Перечень функций BIOS, их коды, а также номера прерываний приведены в таблице на рис.1. Здесь перечислены не все функции BIOS — отсутствуют редко используемые функции и функции, имеющиеся только в устаревших компьютерах PC (например, PCjr). Кроме того, не все указанные функции имеются во всех моделях ПК. Наличие или отсутствие той или иной функции в BIOS зависит от нескольких факторов: какой ПК вы используете, каков состав его аппаратных средств, какая компания разработала вашу систему BIOS, когда вы занесли BIOS в ПЗУ. Довольно часто в прикладных программах используется ограниченный набор вызовов BIOS, что гарантирует возможность выполнения таких программ на большем числе типов ПК без каких-либо доработок. Например, лишь немногие программы пользуются функцией Write String (запись последовательности символов) видео- BIOS, несмотря на очевидное удобство ее применения. Дело в том, что эта функция отсутствует в видеоадаптерах, выпускавшихся до появления стандарта EGA, поэтому ее использование делает прикладную программу не совместимой с компьютером, оснащенным CGA- и MDA-видеоадаптерами.

Таблица на рис. 1 дает лишь общее представление о функциях системы BIOS. К сожалению, недостаток места не позволяет рассмотреть их более детально в данной статье. Дополнительную информацию о функциях BIOS можно найти во многих книгах, в том числе в "Advanced MS-DOS" by Ray Duncan, Microsoft Press. Кроме того, можно заказать официальное справочное руководство компании IBM по системе BIOS "Personal System/2 and Personal Computer BIOS Interface Technical Reference" (номер по каталогу 68X2341), для чего достаточно позвонить по телефону 800-IBM-PCТВ. Однако цена этой книги составляет (ох!) 150 долл. Имеется также дополнение к руководству (номер 15F2161) по цене 18 долл.

НЕ ТОЛЬКО ВВОД-ВЫВОД

Система BIOS кроме указанных программ организации взаимодействия с аппаратными средствами на физическом уровне содержит программу тестирования ПК при включении пита-

ния, POST, и начальный загрузчик. POST представляет собой последовательность диагностических подпрограмм, выполняемых при включении ПК; подобная проверка гарантирует, что аппаратная часть компьютера находится в работоспособном состоянии. Начальный загрузчик — это программа, считывающая загрузочный сектор системного диска и пере-

Перечень функций BIOS

ПРЕРЫВАНИЕ 10H: ФУНКЦИИ УПРАВЛЕНИЯ ДИСПЛЕЕМ		ПРЕРЫВАНИЕ 15H: СИСТЕМНЫЕ ФУНКЦИИ	
Функция 00h	Задание видеорежима	Функция 00h	Включение двигателя ленточного накопителя
Функция 01h	Задание типа курсора	Функция 01h	Выключение двигателя ленточного накопителя
Функция 02h	Задание положения курсора	Функция 02h	Чтение блоков данных с ленты
Функция 03h	Чтение положения курсора	Функция 03h	Запись блоков данных на ленту
Функция 04h	Чтение положения светового пера	Функция 4Fh	Перехват данных от клавиатуры
Функция 05h	Выбор активной видеостраницы	Функция 85h	Реакция на нажатие клавиши SysRq
Функция 06h	Прокрутка активной видеостраницы вверх	Функция 86h	Задание времени ожидания реакции от ленточного накопителя
Функция 07h	Прокрутка активной видеостраницы вниз	Функция 87h	Пересылка блока данных из одной области памяти в другую в защищенном режиме
Функция 08h	Чтение символа и атрибута	Функция 88h	Определение размера расширенной памяти
Функция 09h	Запись символа и атрибута	Функция 89h	Переключение ЦП в защищенный режим
Функция 0Ah	Запись только символа	Функция C0h	Определение конфигурации системы
Функция 0Bh	Установка цветовой палитры	Функция C1h	Определение адреса расширенной области данных BIOS
Функция 0Ch	Запись точки (элемента изображения)	Функция C2h	Интерфейс мыши компьютера PS/2
Функция 0Dh	Чтение точки (элемента изображения)	Функция C3h	Запрет или разрешение таймаута от контрольного таймера
Функция 0Eh	Запись на терминал	Функция C4h	Программируемый интерфейс выбора факультативных средств PS/2в
Функция 0Fh	Чтение параметров видеорежима		
Функция 10h	Установка регистров палитры		
Функция 11h	Программирование генератора символов		
Функция 12h	Альтернативный выбор (различные сервисные функции EGA/MCGA/VGA-адаптеров)		
Функция 13h	Запись строки символов		
ПРЕРЫВАНИЕ 13H: ФУНКЦИИ УПРАВЛЕНИЯ ДИСКОМ		ПРЕРЫВАНИЕ 16H: ФУНКЦИИ УПРАВЛЕНИЯ КЛАВИАТУРОЙ	
Функция 00h	Сброс диска	Функция 00h	Чтение клавиатуры
Функция 01h	Чтение состояния последней операции	Функция 01h	Определение состояния буфера клавиатуры
Функция 02h	Чтение сектора или секторов	Функция 02h	Определение состояния регистров клавиатуры
Функция 03h	Запись сектора или секторов	Функция 03h	Задание временных характеристик клавиатуры
Функция 04h	Проверка сектора или секторов	Функция 05h	Запись символа в буфер клавиатуры
Функция 05h	Форматирование дорожки или цилиндра	Функция 10h	Чтение усовершенствованной клавиатуры
Функция 06h	Форматирование дорожки или цилиндра и установка флагов на "плохих" секторах	Функция 11h	Определение состояния буфера усовершенствованной клавиатуры
Функция 07h	Форматирование диска	Функция 12h	Определение состояния регистров (shift) усовершенствованной клавиатуры
Функция 08h	Чтение параметров накопителя		
Функция 09h	Инициализация таблицы параметров ИЖМД		
Функция 0Ch	Установка на дорожку или цилиндр		
Функция 0Dh	Сброс диска		
Функция 10h	Проверка готовности накопителя		
Функция 11h	Повторная калибровка накопителя		
Функция 15h	Чтение информации о типе накопителя		
Функция 19h	Парковка головок		
ПРЕРЫВАНИЕ 14H: ФУНКЦИИ УПРАВЛЕНИЯ ПОСЛЕДОВАТЕЛЬНЫМ ПОРТОМ		ПРЕРЫВАНИЕ 17H: ФУНКЦИИ УПРАВЛЕНИЯ ПАРАЛЛЕЛЬНЫМ ПРИНТЕРНЫМ ПОРТОМ	
Функция 00h	Инициализация последовательного порта	Функция 00h	Печать символа
Функция 01h	Передача символа	Функция 01h	Инициализация параллельного порта
Функция 02h	Прием символа	Функция 02h	Определение состояния параллельного порта
Функция 03h	Определение состояния последовательного порта		
Функция 04h	Инициализация последовательного порта (расширенная)		
Функция 05h	Чтение или установка регистра состояния модема		
		ПРЕРЫВАНИЕ 1Ah: ФУНКЦИИ ЧТЕНИЯ/УСТАНОВКИ ВРЕМЕНИ И ДАТЫ	
		Функция 00h	Чтение системного таймера
		Функция 01h	Установка системного таймера
		Функция 02h	Чтение времени
		Функция 03h	Установка времени
		Функция 04h	Чтение даты
		Функция 05h	Установка даты
		Функция 06h	Установка "будильника"
		Функция 07h	Сброс "будильника"

Рис. 2. К программам BIOS можно обратиться с помощью программных прерываний — команд ЦП, которые передают управление от выполняемой в данный момент программы тому или иному обработчику прерываний.

НАСТАВНИК

дающая управление хранимой в этом секторе программе, которая загружает операционную систему в память.

BIOS также содержит обработчики для отдельных аппаратных прерываний. Например, каждый раз, когда на выходе канала 0 системного таймера появляется тактовый импульс (приблизительно 18 раз в секунду), вырабатывается аппаратное прерывание, вызывающее запуск обработчика прерываний BIOS. Последний обновляет состояние программно реализованных системных часов и выполняет ряд других служебных функций. Другой пример: при нажатии или отпуске клавиши система BIOS регистрирует прерывание, вырабатываемое контроллером клавиатуры, и считывает расширенный код клавиши. После преобразования такого расширенного кода в эквивалентный ASCII-код с учетом нажатия других клавиш (например, Ctrl, Alt или Shift), а также текущего состояния клавиш NumLock и CapsLock обработчик прерываний помещает код клавиши в буфер клавиатуры. Система BIOS также содержит программу, которая выдает копию экрана на принтер при нажатии клавиши Print Screen.

Еще одна составляющая, о которой нельзя не упомянуть при рассмотрении BIOS, — это таблица параметров жесткого диска, хранимая в ПЗУ. Таблица содержит характеристики (например, число головок, секторов и цилиндров) всех тех моделей накопителей на жестком магнитном диске (НЖМД), которые можно устанавливать в данный ПК. В общем случае установка в компьютер того НЖМД, характеристики которого отсутствуют в таблице, требует замены ROM BIOS или использования специальной утилиты типа Disk Manager фирмы Ontrack.

ОБЛАСТЬ ДАННЫХ BIOS

Область основной памяти размером 256 байт, расположенная непосредственно за таблицей векторов прерываний, — начиная с адреса 400h и кончая 4Ffh — предназначена для использования программами BIOS. Эта часть памяти называется областью данных BIOS. В таблице на рис.2 приведено описание части информации, помещаемой системой BIOS в указанную область памяти. Обращаясь к ней, прикладные программы получают важную информацию о состоянии системы. Например, прочтя байт по адресу 484h, программа может определить, сколько строк текста выведено на экран (если ПК оснащен EGA-, VGA- или XGA-адаптером).

Иногда оказывается полезным вносить изменения в данные, хранимые в этой области памяти. Например, можно очистить буфер клавиатуры, прочтя значение слова по адресу 41Ah и записав его по адресу 41Ch. По указанным адресам хранятся начальный и конечный адреса буфера клавиатуры, и поэтому после записи в них

одного и того же значения BIOS будет полагать, что кодов клавиш, ожидающих обработки (программой обслуживания прерываний), больше нет.

В компьютерах PS/2 для системы BIOS отводится в основной памяти дополнительный участок, называемый Расширенной областью данных BIOS. При включении питания программа автоматического тестирования (POST) компьютера PS/2 определяет объем наличной основной памяти и резервирует ее старшие адреса под сверхоперативную память для системы BIOS. Объем такой области памяти обычно составляет 1 Кбайт, однако разработчики PS/2 предусмотрели возможность ее расширения. Адрес расширенной области данных BIOS определяется путем вызова функции C1h по прерыванию 15h. Первый байт в этой области данных задает размер области в килобайтах; следовательно, таким путем может быть зарезервировано до 255 Кбайт. В компьютерах PS/2 память данных, выделенная для BIOS, "упрятана" так, что остальные компоненты системы "не знают" о ее существовании.

ABIOS

Расширенная область данных для В1 — не единственное новшество, предложенное создателями компьютера PS. Большинство моделей PS/2 на деле держат два вида системы BIOS: CBI — включающую в качестве подмножества первоначальный вариант В1 компании IBM (С означает "совместимость") — и АВ1ОS, или Advan BIOS, поддерживающую многозаданные ОС, например OS/2. Сист. АВ1ОS является аналогом СВ1ОS, ботающим в защищенном режиме) При установке OS/2 на компьютер класса АТ, последняя игнорирует Е и взаимодействует с аппарата средствами напрямую. Однако на и пьютере PS/2 эта операционная система полностью опирается на фут АВ1ОS. Система АВ1ОS (если она : ется) играет на компьютере, рабсшем под управлением OS/2, ту роль, что и обычная система В1С ПК, на котором установлена DOS.

В первой модели IBM PC система BIOS занимала 8 Кбайт; BIOS компьютера PS/2 занимает уже 128 КИ. При подобных темпах роста трудно предсказать, каких размеров достигнет через десять лет!

Область данных BIOS

Адрес	Длина	Описание	Адрес	Длина	Описание
		ОБЛАСТЬ ДАННЫХ ПОСЛЕДОВАТЕЛЬНОГО/ПАРАЛЛЕЛЬНОГО ПОРТОВ	460h	Слово	Тип курсора (начальная/конечная строки разрыва)
400h	4 слова	Базовый адрес ввода-вывода для COM1-COM4	462h	Байт	Активная видеостраница
408h	4 слова	Базовый адрес ввода-вывода для LPT1-LPT4	463h	Слово	Базовый адрес контроллера ЭЛТ
		СМЕШАННАЯ ОБЛАСТЬ ДАННЫХ	465h	Байт	Текущая установка разрешения 3х8
410h	Слово	Флаги оборудования	466h	Байт	Текущая установка разрешения 3х9
413h	Слово	Объем памяти в килобайтах			ОБЛАСТЬ ДАННЫХ СИСТЕМНОГО ТАЙМЕРА
		ОБЛАСТЬ ДАННЫХ КЛАВИАТУРЫ 1	46Ch	Слово	Младшее слово кода текущего таймера
417h	Байт	Флаги состояния регистров (shift) клавиатуры 1	46Eh	Слово	Старшее слово кода текущего таймера
418h	Байт	Флаги состояния регистров (shift) клавиатуры 2	470h	Байт	Индикатор заполнения
419h	Байт	Ввод с дополнительного клавиатурного поля			СИСТЕМНАЯ ОБЛАСТЬ ДАННЫХ 2
41Ah	Слово	Адрес начала буфера клавиатуры	471h	Байт	Состояние клавиши сброса
41Ch	Слово	Адрес конца буфера клавиатуры	472h	Слово	Флаг сброса
41Eh	32 байт	Буфер клавиатуры			ОБЛАСТЬ ДАННЫХ ЖЕСТКОГО ДИСКА
		ОБЛАСТЬ НАКОПИТЕЛЯ НА ГИБКИХ ДИСКАХ	474h	Байт	Состояние последней операции
43Eh	Байт	Состояние повторной калибровки	475h	Байт	Число подключенных дисков
43Fh	Байт	Состояние двигателя			ВРЕМЕНА ОЖИДАНИЯ (ТАЙМАУТЫ)
440h	Байт	Счетчик числа отключений двигателя	478h	4 байт	Значения времен ожидания для LPT1-LPT4
441h	Байт	Состояние последней операции	47Ch	4 байт	Значения времен ожидания для COM1-COM4
442h	7 байт	Байты состояния контроллера			ОБЛАСТЬ ДАННЫХ КЛАВИАТУРЫ 2
		ОБЛАСТЬ ДАННЫХ ВИДЕОАДАПТЕРА 1	480h	Слово	Адрес начала буфера клавиатуры
449h	Байт	Текущий видеорежим	482h	Слово	Адрес конца буфера клавиатуры
44Ah	Слово	Количество столбцов в отображаемом тексте			ОБЛАСТЬ ДАННЫХ ВИДЕОАДАПТЕРА 2
44Ch	Слово	Длина буфера регенерации в байтах	484h	Байт	Количество отображаемых строк текста минус высота символа в развертке
44Eh	Слово	Адрес смещения активной видеостраницы	485h	Слово	Высота символа в развертке
450h	8 слов	Положение курсора (видеостраницы 0-70)	487h	Байт	Параметры видеорежима
			488h	Байт	Параметры видеорежима

Рис. 3. Область данных BIOS занимает 256 байт в ОЗУ — между адресами 400h и 4Ffh.

ТОЛЬКО ВЫСШЕЕ КАЧЕСТВО!

Электронные системы
производства HANTAREX
укомплектованы
периферийным оборудованием
ведущих мировых фирм:

HEWLETT PACKARD
лазерные принтеры, плоттеры, дигитайзеры

BULL
матричные принтеры

HANTAREX
мониторы VGA, Multisync,
мониторы размера 21" для САПР

Системы также оснащаются
ОС UNIX и Novell Netware

Работу издательских систем
обеспечивает программа
Aldus PageMaker

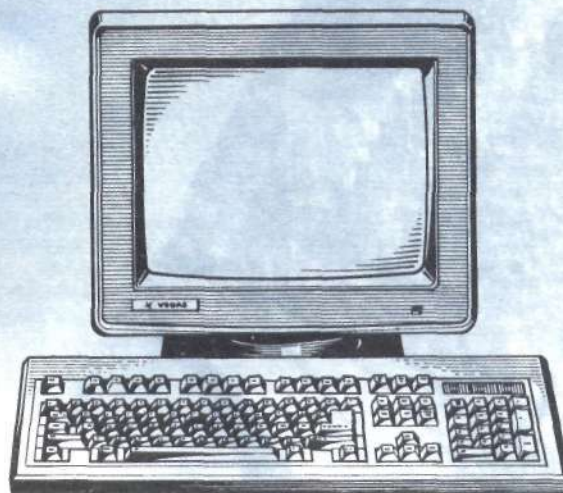
Для автоматизации офисов
предлагается оборудование
известных фирм:

ALCATEL
телексы и
цифровые телефонные станции

CANON
копировальные машины

Цены - самые благоприятные в Европе

Гарантия - два года
с дальнейшим обслуживанием за рубли



HANTAREX

117342, Москва, ул.Обручева 36
Телефон (095) 334-29-74
Телефакс (095) 420-22-50
Телекс 412160 ANТАR SU
Телетайп 11428 АНТАР

**HANTAREX -
ИДЕАЛЬНАЯ ВОЗМОЖНОСТЬ
ОБЕСПЕЧИТЬ ВАШЕ ПРЕДПРИЯТИЕ
ТЕХНИКОЙ ВЫСШЕГО КАЧЕСТВА!**

Рей Дункан

Ежегодная Конференция разработчиков программного обеспечения, финансируемая издательским концерном Miller-Freeman Publications, стала поистине важнейшим событием для серьезных программистов. Это прекрасная возможность увидеть и услышать известных технических писателей, теоретиков и разработчиков — от Адели Голдберг (Adele Goldberg) до Стэна Келли-Бутла (Stan Kelly-Boote) — но это также и очень точный барометр тенденций в индустрии программного обеспечения. Авторы многих выступлений на конференциях 1988 и 1989 годов предсказали скорый конец операционной системы DOS и полагали, что программистской и операционной средой ближайшего будущего станут ОС Unix и OS/2. Однако на конференции 1990 года большинство участников вынуждено было смириться с тем, что система DOS и оболочка Windows продолжают доминировать, а важнейшей новостью конференции стало объявление фирмы Borland о выпуске новой версии компилятора с языка программирования Си++, позволяющей создавать программы для подсистемы Windows.

Новичку в мире Windows может показаться удивительным, как это транслятор Microsoft C и пакет для разработчиков Microsoft Windows Software Development Kit (SDK) в течение столь долгого времени остаются полными монополистами на рынке систем программирования для Windows. В конце концов, если функции Windows рассчитаны на вызов непосредственно из программы на Си, а прикладные программы по определению используются для работы с клавиатурой, мышью и дисплеем вместо библиотечных функций Си системными функциями Windows, почему бы не писать программы для Windows с помощью любого из известных компиляторов языка Си?

Это соображение, представляющееся на первый взгляд вполне очевидным, оборачивается серьезнейшей ошибкой, если принять во внимание два обстоятельства, абсолютно не связанных с языком Си как таковым: макроструктуру исполнимых файлов для среды Windows и наличие специфических соглашений системы управления виртуальной памятью Windows, которые должны обязательно соблюдаться в кодах, вырабатываемых компилятором Си.

EXE-ФАЙЛЫ ДЛЯ WINDOWS

Прежде чем мы заглянем внутрь программ для Windows, рассмотрим строение двух типов программных фай-

Постигая интерфейс прикладных программ оболочки Windows

■ До появления пакета Borland C++ Microsoft C и Microsoft Windows Software Development Kit были полными монополистами на рынке систем программирования для Windows.

лов, различаемых "старой доброй" системой DOS. Простейший из них — COM-файл; он содержит только машинные команды и данные, без всякого заголовка, таблицы переадресации или отладочной информации; его максимальный размер — 64 Кбайт. Загрузчик обращается с COM-программой так: отводит максимально возможный блок свободной памяти, строит префикс программного сегмента (program segment prefix, PSP) в начале блока, считывает

Исторически формат COM-файлов скопирован с формата исполнимых модулей операционной системы CP/M.

COM-файл в память сразу вслед за PSP и передает управление на первый байт программы. Исторически формат COM-файлов скопирован с формата исполнимых модулей операционной системы CP/M фирмы Digital Research для микрокомпьютеров на базе процессоров 8080 и Z-80, поэтому COM-файлы унаследовали от этих модулей присущую им "незащищенность". Присвойте любому t>айлу данных расширение COM, навяжите его имя в командной строке, и DOS преспокойно совершит самоубийство: загрузит эту "программу"

в память, передаст ей управление — и... с треском "рухнет".

Другой, более сложный тип исполнимых файлов DOS — это EXE-файлы. EXE-файл состоит из трех основных компонентов: заголовка файла, таблицы переадресации и собственно кода и данных программы (см. рис. 1). Заголовок не отличается большой сложностью и содержит всего несколько информационных блоков: двухбайтовую сигнатуру MZ (инициалы Марка Збиковски, одного из главных авторов версий DOS от 2.0 до 3.1); размеры заголовка, таблицы переадресации и файла в целом; начальные значения CS:IP и SS:SP в момент запуска программы; контрольную сумму; некоторые указания по распределению памяти для загрузчика DOS. Максимальный размер EXE-файла не ограничен; в него можно включить отладочную информацию, дописав эту информацию в конец файла и оставив без изменения его размер, указанный в заголовке. Загрузчик DOS эту информацию просто проигнорирует.

Формат EXE-файлов — несомненный шаг вперед по сравнению с COM-файлами, однако он вовсе не так хорош, как может показаться на первый взгляд. Главный недостаток EXE-файлов заключается в том, что они не сохраняют информацию об отдельных сегментах программы; когда компоновщик формирует EXE-файл, он собирает все объявленные вами в исходном тексте сегменты в одну громадную "глыбу" с общей таблицей переадресации. Системный загрузчик не в состоянии определить, какая часть программы представляет собой исполнимый код, какая — статические данные (например, строковые и числовые константы), а какая — переменные данные. Из-за этого загрузчик лишен возможности отвести для хранения каждого из этих компонентов отдельные блоки памяти и независимо манипулировать сегментами.

Создавая версию 1.0 подсистемы Windows, разработчики фирмы Microsoft поняли, что ограниченность объема памяти, доступной в реальном режиме в DOS, 640 Кбайт действительно сковывает, и только тщательная сегментация программ может дать надежду на то, что система в целом заработает. Оболочка должна была обладать способностью оставлять в памяти только ту часть кон-

КОМПОНЕНТЫ EXE-файла

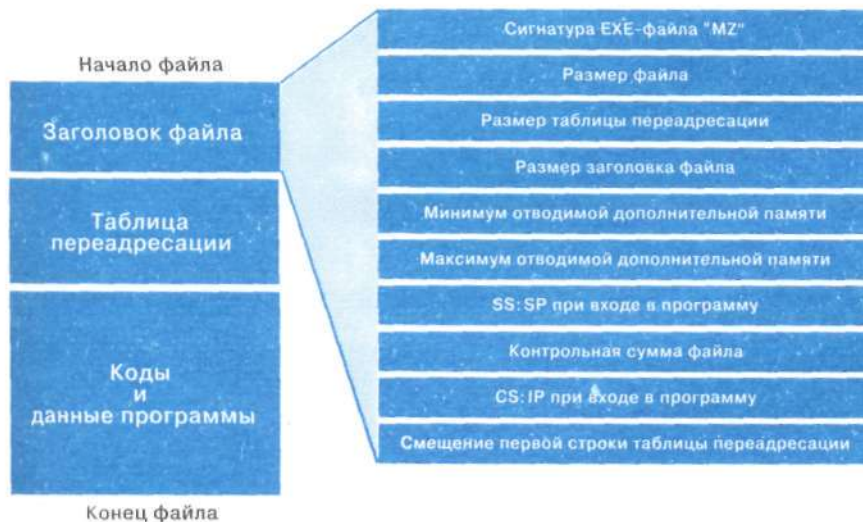


Рис. 1: Три компонента EXE-файла DOS (старого EXE-файла). Файл имеет простую структуру и состоит из заголовка, таблицы переадресации и собственно кода и данных программы.

ретной программы, которая обеспечивает ее дальнейшее исполнение, и уничтожить или выводить на диск фрагменты, не используемые в данный момент. Группа разработчиков Windows приняла решение ввести новую структуру исполнимых файлов, называемых сегментированными, или новыми EXE-файлами, и создать новый загрузчик таких файлов специально для Windows. Во время работы Windows он перехватывает обращения прикладных программ к функции DOS EXEC (функция 4Bh прерывания 21h) и правильно загружает как новые, так и "старые" EXE-файлы.

Новый формат EXE-файлов — это существенное расширение формата DOS (или старого EXE-файла); его структуру можно рассматривать на нескольких уровнях абстракции (рис. 2). На высшем уровне новый EXE-файл состоит из двух частей: старого EXE-файла и нового EXE-файла. "Старая" часть содержит полный заголовок старого EXE-файла, таблицу переадресации и собственно программу, если новый EXE-файл почему-либо будет запущен из командной строки DOS, загрузчик DOS распознает заголовок старого EXE-файла и запустит программу (для Windows-программы эта часть обычно очень короткая — она просто выводит на экран сообщение "This program requires Microsoft Windows" ("Для этой программы требуется среда Microsoft Windows") и возвращает управление DOS). Если новый EXE-

файл запускается через модули управления Program Manager или File Manager оболочки Windows, "старая" часть EXE-файла игнорируется.

Обратившись к "новой" части нового EXE-файла, мы обнаружим, что его элементы распадаются на три четко различающихся класса. Первая группа состоит из заголовка и различных таблиц, определяющих размер, положение и характеристики всех остальных частей файла; они используются системным загрузчиком и "невидимы" для самой программы во время ее исполнения. Вторую группу составляют сегменты кода и данных программы, каждый со своей таблицей переадресации. В третью группу входят так называемые ресурсы. Ресурс — это статическая порция данных, например пиктограмма, курсор, растр, меню или набор символьных строк. Каждый ресурс в файле опознается по имени и типу; он загружается в память системой по требованию прикладной программы.

Формат нового EXE-файла прекрасно выдержал проверку временем и может служить монументом искусству и дару предвидения первых разработчиков Windows 1.0. Версия Windows 3.0, в которой первая редакция претерпела настолько решительную переработку, какую только можно себе представить, потребовала введения в заголовок нового EXE-файла лишь одного дополнительного флага, указывающего, что программа может работать в режиме защиты. Версии

OS/2 и растущее в последнее время число расширителей DOS для 16-разрядных процессоров также пользуются форматом нового EXE-файла с минимальными изменениями.

Пока еще не существует служебной программы, которая позволяла бы непосредственно и эффективно работать с новыми EXE-файлами — например, извлечь какой-либо ресурс из существующего EXE-файла, исправить его и вернуть на место — но, возможно, все увеличивающаяся популярность системы Windows 3.0 заставит кого-нибудь из разработчиков инструментальных средств заняться этим.

ВИРТУАЛЬНАЯ
ПАМЯТЬ В WINDOWS

Как только сложился формат нового EXE-файла, архитекторы Windows 1.0 переключили внимание на создание для прикладных Windows-программ, работающих в реальном режиме, системы виртуальной памяти — другими словами, обеспечение возможности использовать пространство логических адресов, превышающее объем доступной физической памяти. Поскольку они вынуждены были строить свою систему, не опираясь на аппаратные функции, в основу схемы виртуальной памяти положили (а) невытесняющую мультизадачность, при которой прикладная программа уступает управление только в строго определенные моменты своего исполнения, и (б) сложный набор соглашений, интерфейсов и самомодифицирующихся программ, от которых мурашки побежали бы по спине у любого ортодоксального ученого-программиста.

Во-первых, разработчики Windows постановили, что Windows-программы должны составляться только с использованием моделей памяти, имеющих единственный сегмент данных: другими словами, малой (small) и средней (medium) моделей. Если у прикладной программы возникает необходимость использовать данные, не помещающиеся в сегмент размером 64 Кбайт, ей предлагается отводить дополнительную память в процессе исполнения в "глобальной куче" (global heap) подсистемы Windows, инициализируя эту память (при необходимости) ресурсами из собственного EXE-файла или данными, хранящимися в других файлах. Это ограничение позволяет ядру Windows безнаказанно перемещать сегмент данных прикладной программы после ее запуска, подправляя по мере необходимости соответствующий регистр сегмента данных DS, поскольку программе никогда (по крайней мере, теоретически) не может понадобиться менять свой DS "за спиной" системы.

Во-вторых, Windows-программам было запрещено пользоваться для отведения памяти нормальным интер-

ФОРМАТ НОВОГО EXE-ФАЙЛА

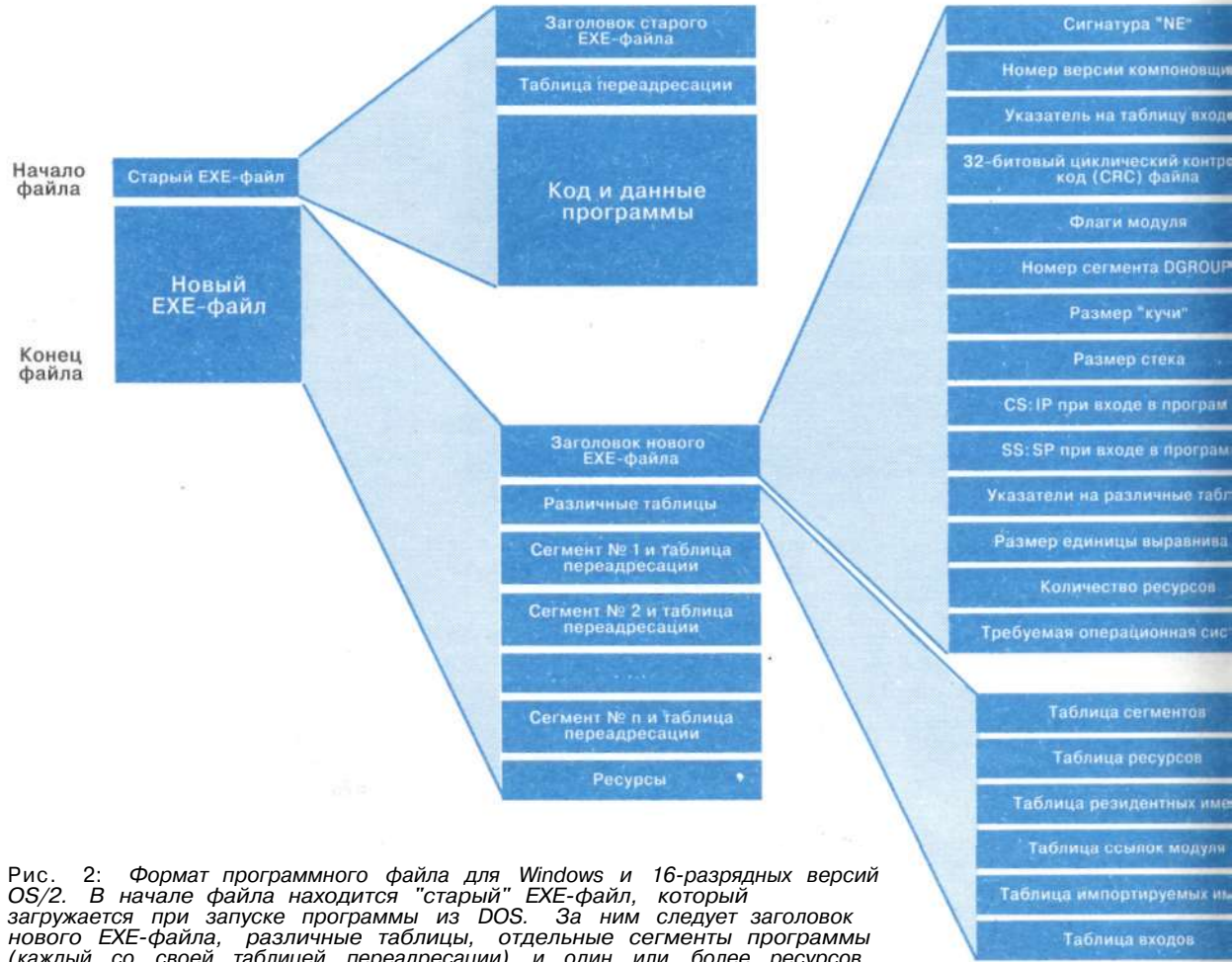


Рис. 2: Формат программного файла для Windows и 16-разрядных версий OS/2. В начале файла находится "старый" EXE-файл, который загружается при запуске программы из DOS. За ним следует заголовок нового EXE-файла, различные таблицы, отдельные сегменты программы (каждый со своей таблицей переадресации) и один или более ресурсов.

фейсом DOS и предписано обращаться для этого к новой системе управления памятью в ядре Windows. Когда программа запрашивает блок памяти через этот интерфейс, Windows возвращает вместо конкретного физического адреса номер блока (handle). В момент, когда программе на самом деле потребуется доступ к памяти, она обращается к еще одной функции Windows, чтобы "запереть" блок памяти и получить его текущий адрес. Ожидается, что после чтения из памяти или записи в нее программа при первой возможности вновь "отпирает" блок, чтобы система могла перемещать его или, при необходимости, сбрасывать на диск. Таким образом в какой-то степени имитируется взаимосвязь между селекторами, дескрипторами и сегментами физической памяти, имеющая место при работе процессоров 80286/386/486 в защищенном режиме.

В-третьих, архитекторы потребовали, чтобы Windows-программа регистрировала все свои процедуры,

которые вызываются из ядра системы и не являются оконными процедурами (т.е. процедуры, не связанные с очередью сообщений Windows), путем вызова системной функции MakeProcInstance. Эта функция создает небольшой фрагмент кода (thunk), на который первым делом передается управление при "дальних" вызовах программы из Windows; этот фрагмент заносит в регистр AX адрес сегмента данных программы и передает управление на ее точку входа. Когда Windows перемещает сегмент кода или данных программы, она компенсирует такое перемещение, модифицируя соответствующим образом инструкции MOV и JMP фрагмента.

В-четвертых (и это наиболее причудливая часть схемы виртуальной памяти Windows), разработчики системы потребовали, чтобы все функции, экспортируемые Windows-программой или библиотекой — функции, точки входа которых допускают связывание по имени во время исполнения

со стороны ядра Windows и прикладных программ — кодировались со специальными последовательностями в начале и в конце — прологом и эпилогом, без сомнения, известными составляющие "обычную" языка Си, обрамляются элементарным прологом и эпилогом (или иными словами — прологом и эпилогом):

```

push bp
mov bp, sp
sub sp, xxx
.
.
add sp, xxx
pop bp
ret
    
```

Стандартный пролог Си формирует кадр стека (stack frame) обеспечивая адресацию стека с помощью регистра BP. Подпрограмма в дальнейшем может адресовать метры, переданные ей через

ложительными смещениями от BP, а локальные переменные (зарезервированные инструкцией `sub sp,xxx`) — отрицательными. В эпилоге память, занятая локальными переменными, освобождается инструкцией `add sp,xxx` (иногда `mov sp,bp`), затем восстанавливается регистр BP и происходит возврат к вызвавшей процедуре. Поскольку язык Си допускает функции с переменным числом параметров, ответственность за освобождение стека от параметров вызванной функции возлагается на вызывающую программу. Пролог и эпилог экспортируемой процедуры в Windows-программе заметно сложнее:

```
push ds
pop ax
nop
inc bp
push bp
mov bp,sp
sub sp,xxx
push ds
mov ds,ax
.
.
.
pop ds
mov sp,bp
pop bp
dec bp
ret yyy
```

Странный код, не так ли? Прежде всего заметьте, что процедура освобождает стек от своих параметров инструкцией `get ууу`, не возлагая эту работу на вызывающую процедуру. Это часть соглашения о дальних вызовах языка Паскаль, которое (несмотря на все его несходство с соглашениями языка Си) получило признание разработчиков первой версии Windows за то, что экономит несколько килобайт памяти в целом по системе. Далее, обратите внимание на таинственное увеличение на единицу содержимого регистра BP перед занесением в стек. Объяснение этого маневра относится к области черной и белой магии, и вам, вероятно, потребуется провести некоторое время за отладкой Windows-программы, чтобы убедиться, что он и в самом деле работает. В немногих словах можно объяснить его следующим образом.

Стек Си-программы состоит из последовательности кадров, которая точно отражает текущее состояние программы: она запускается путем вызова процедуры `main O`, которая в свою очередь вызывает другую процедуру и т.д. Предположим, что стек программы всегда выровнен на слово. Тогда нечетное значение BP в стеке может служить признаком дальнего вызова. Если процедура является внутренней (*internal*) или ближней (*near*), она будет иметь стандартный для Си пролог, занесенное в стек значение BP (которое фактически является указателем на кадр стека ближайшей охватывающей процедуры)

будет четным, а число, "лежащее" на стеке следующим, будет адресом ближнего возврата. Если вход в процедуру осуществляется путем дальнего (*far*) вызова и она имеет специальный пролог стандарта Windows, занесенное в стек значение BP будет нечетным, и лежащий над ним адрес будет дальним указателем (*far pointer*).

А дальше чудеса! Всякий раз, когда ядро Windows перемещает сегмент кода, оно восстанавливает текущее значение регистра BP каждой из прикладных программ и "гуляет" по их стекам от самого внутреннего к самому внешнему кадру, находя и исправляя "по дороге" все дальние указатели, относящиеся к этому кодовому сегменту. Если сегмент кода не перемещен, а удален, адрес возврата меняется на точку входа в процедуру ядра, которая занимается перезагрузкой сегмента.

Схема виртуальной
памяти системы
Windows —
торжество
изобретательности
человеческого
разума.

И последняя интересная для нас часть специального пролога стандарта Windows — кажущееся излишним копирование DS в AX, а затем обратно в DS. Как ни странно, если вы "посмотрите" программой-отладчиком пролог экспортируемой процедуры в Windows-программе, которая уже загружена и работает, вы вообще не найдете в нем этих кодов! Вместо них вы увидите вот что:

```
nop
nop
nop
inc bp
push bp
mov bp,sp
sub sp,xxx
push ds
mov ds,ax
```

Во время компиляции о функции известно только то, что она запускается с помощью дальнего вызова, так как функции помечаются как экспортируемые (или нет) в процессе компоновки. Поэтому компилятор генерирует для всех дальних процедур этот излишний, но безвредный код,

никак не влияющий на содержимое DS. Загрузчик Windows модифицирует пролог каждой экспортируемой функции, заменяя инструкции `push ds` и `pop ax` на инструкции `pop`. Тогда при входе в процедуру из thunk-фрагмента с адресом сегмента данных в AX, в DS попадает нужное значение. Конечно, практически ни в одном из этих "фокусов" нет необходимости, когда Windows-программа работает в режиме защиты, однако прикладным программам и ядру Windows приходится тащить за собой весь этот старый багаж. В целом схема виртуальной памяти системы Windows — это торжество изобретательности человеческого ума, памятник неподатливости реального режима, воздвигнутый с большим размахом, который, похоже, простоят еще несколько лет. Она, несомненно, вызывает ностальгию по добрым старым временам OS/2. Но я увлекся.

А ТЕМ ВРЕМЕНЕМ...

Я начал эту колонку с упоминания о том, что версия 2.0 пакета Borland C++, со своими возможностями создания прикладных программ для Windows, кладет конец монополии фирмы Microsoft на рынке инструментальных средств программирования для Windows. Обзоры расхваливают (и не зря) новую интегрированную среду разработки (*Integrated Development Environment, IDE*) Borland C++, замену запутанных файлов подготовки (*make*) файлами проекта (*project*), возможность предварительной компиляции файлов-заголовков, появление системы управления оверлеями *VROOM*, возможность работы в расширенной памяти и т.д. Но несмотря на то, что Borland C++ — действительно внушительный программный пакет, подтверждающий выход фирмы Borland на передовые позиции в области систем программирования как по стилю, так и по возможностям, вы теперь сможете оценить, что многие остроумные средства пакета бесполезны с точки зрения их применения в качестве системы разработки Windows-программ. Нужно было только обеспечить компиляцию специальных прологов и эпилогов стандарта Windows, создать компоновщик, способный строить новые EXE-файлы, и варианты нескольких несложных инструментальных программ фирмы Microsoft, скажем, редактора пиктограмм и компилятора ресурсов.

Теперь встает вопрос: сможет ли Си++ как язык программирования исполнить свои обещания и помочь нам овладеть интерфейсом прикладных программ (*Application Program Interface, API*) системы Windows во всей его сложности? Некоторые различия между языками Си и Си++ мы рассмотрим в нашей колонке в одном из следующих номеров.

ТРАНСПОРТИРОВКА КОНСТРУКЦИЙ

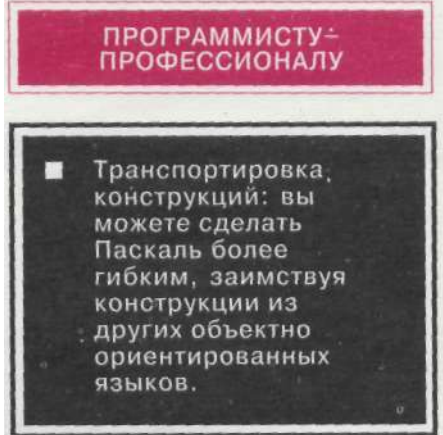
Когда экземпляр объекта передается функции или процедуре в виде параметра оператора VAR, он может лишиться некоторых своих отличительных характеристик. Некоторые поля и правила, связанные с объектом, могут стать недоступными, и при ссылках на эти характеристики будут выдаваться сообщения об ошибках. На самом деле все эти характеристики существуют, и можно попытаться "обмануть" компилятор так, чтобы он стал их использовать, путем создания типов. Но, если при этом тип окажется неверным, программа может повести себя самым непредсказуемым образом.

В языке Turbo Pascal версии 5.5 есть средство для определения фактического типа экземпляра объекта — функция TypeOfO. К сожалению, она работает только с конкретным типом. TypeOfO не сможет распознать потомков данного конкретного типа, несмотря на то что каждый из потомков унаследовал все его поля и правила.

Другие объектно-ориентированные языки программирования могут определять, какой ветви дерева семейства принадлежит тот или иной объект. Например, в языке Smalltalk правило IsKindOf:aClass возвращает значение "истина", если объект является экземпляром класса aClass или любого из его подклассов. Правило IsKindOf и сопутствующее ему правило IsA (см. листинг) — результаты моих усилий по реализации подобных правил, предопределенных в языке Objective-C и рассматриваемых в книге Брэда Дж. Кокса "Объектно-ориентированное программирование: Эволюционный подход" (Brad J. Cox, "Object Oriented Programming: An Evolutionary Approach", Addison-Wesley, 1987). Правило IsA, описанное у Кокса как поле, сравнительно просто: оно возвращает TypeOf(self). Я обычно помещаю его определение на вершине базовой иерархии.

В отличие от него IsKindOf прослеживает снизу вверх дерево иерархии объекта с целью найти его предшественников. При этом необходимо, чтобы в каждом типе-потомке была реализована своя собственная версия виртуального правила IsKindOf. Она связывает каждое правило IsKindOf с предшествующим ему IsKindOf до тех пор, пока не будет обнаружено совпадение функций TypeOfO или пока не будет достигнута вершина дерева, где функция будет иметь значение "истина" только в том случае, если передаваемый указатель соответствует TypeOf(base).

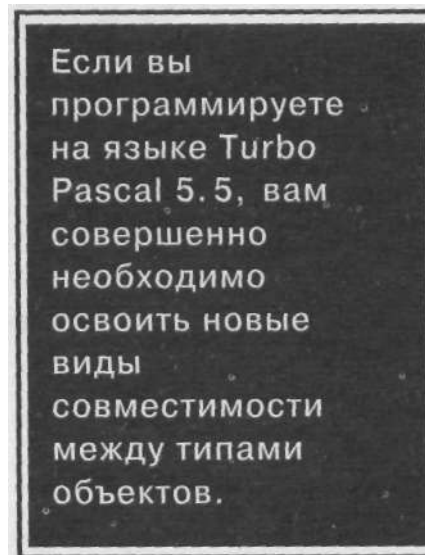
Правило IsKindOf дает возможность использовать корректное описание типа объекта при обращении к полям и правилам, которые не были определены в базовом типе. Пусть,



например, в типе XType, который является потомком базового, вводится определение булевого поля данных F. Тогда это поле будет определено для всех потомков типа XType, но не для потомков других потомков базового типа. Пусть Y — имя рассматриваемого объекта. Тогда, если

Y.IsKindOf (TypeOf (XType))

истинно, то поле F является доступным.



За такую гибкость приходится платить. В процессе сборки TP5.5 удаляет из исполняемого файла все коды и данные, на которые нет ссылок. Но, если для объекта создана таблица VMT (Virtual Method Table), то она будет содержать ссылки на каждое из виртуальных правил. Правило constructo инициализирует VMT, так что, когда программа обращается к конструктору переменной экземпляра, то все виртуальные правила этого типа объекта связываются при сборке в исполняемый файл.

Но и функция TypeOfO обращается к VMT. Правило IsKindOf просматривает VMT каждого объекта, экземпляра которого вы создаете, а также всех предшественников этого объекта. Коды всех виртуальных правил всех этих объектов будут подключены к испол-

няемому файлу, даже если вы объявили или не использовали ни одного из этих предшественников!

Виртуальные правила предшественников часто бывают очень невелики — ненамного больше, чем на BEGIN END;. Так что проблема скорее потенциальная, нежели реальная. Если она окажется более серьезной, то можно применять IsKindOf только к критическим поколениям (т.е. к тем, которые вы на самом деле намерены проверить), а не к каждому отдельному объекту.

Дж. У. Райдер
Сан-Диего, шт. Калифорния

От редакции:

Если вы собираетесь программировать на языке Turbo Pascal 5.5, вам совершенно необходимо освоить новые виды совместимости между типами объектов. Присваивание A := B; обычно допускается только тогда, когда A и B принадлежат к одному и тому же типу, если A и B — объекты, то такое присваивание будет законным и в случае, когда тип объекта B является потомком типа объекта A. У B могут быть поля и правила, отсутствующие у A; они пропадут. То же самое относится к передаче объектов путем ссылки на процедуры или функции — вы не можете передавать любой объект типа-потомка, но если в нем есть поля или правила, которых не было в типе-предшественнике, то они будут недоступны.

Процедура Test (второй листинг) получает на входе параметр типа base, один из которых передается значением, а второй ссылкой (параметр VAR). Когда передаем одну и ту же переменную в обоих параметрах этой процедуры, передаваемый значением параметр, передаваемый значением будет экземпляром типа base. Переменная же, передаваемая ссылкой, сохранит свою истинную структуру как экземпляр типа GrandA.

Самое лучшее решение этих проблем — вообще не пользоваться виртуальными правилами среди разных типов. Если объект имеет какие-то виртуальные правила, то такое присваивание все равно будет недействительным — мы обязаны инициализировать такой объект его конструктором. Лучше использовать указатели на объекты или же передавать параметры VAR. Указатель на виртуальный тип объекта может фактически указывать как на сам этот тип, так и на любой из его потомков, и компилятор сможет автоматически подключить нужное виртуальное правило.

Показанный листинг также демонстрирует возможности использования правила IsKindOf. Здесь иерархия сразу после разветвления в ветви ChildA до объекта: в ветви WhoAreYou, а в ветви — нет. В каждой ветви имеется один потомок — "внуку" переменная P является указателем

ISKIND.PAS

ПОЛНЫЙ ЛИСТИНГ

```

PROGRAM iskind;
TYPE
  Pt = "base;
  GrandAP = "GrandA;
  GrandBP = "GrandB;

  base = object
    CONSTRUCTOR Init;
    DESTRUCTOR Done;
    FUNCTION IsKindOf(T: Pointer): boolean; virtual;
    FUNCTION IsA; pointer;
  END;

  ChildA = object(base);
    CONSTRUCTOR Init;
    FUNCTION IsKindOf(T: Pointer): boolean; virtual;
    PROCEDURE WhoAreYou; virtual;
  END;

  ChildB = object(base);
    CONSTRUCTOR Init;
    FUNCTION IsKindOf(T: Pointer): boolean; virtual;
  END;

  GrandA = object(ChildA);
    CONSTRUCTOR Init;
    FUNCTION IsKindOffT: Pointer): boolean; virtual;
    PROCEDURE WhoAreYou; virtual;
  END;

  GrandB = object(ChildB);
    CONSTRUCTOR Init;
    FUNCTION IsKindOf(T: Pointer): boolean; virtual;
  END;

(**** Все конструкторы и деструктор: ****)
DESTRUCTOR Base. Done; BEGIN END;

CONSTRUCTOR Base. Init; BEGIN END;
CONSTRUCTOR ChildA. Init; BEGIN END;
CONSTRUCTOR ChildB. Init; BEGIN END;
CONSTRUCTOR GrandA. Init; BEGIN END;
CONSTRUCTOR GrandB. Init; BEGIN END;

(**** Правило IsA — используется всеми объектами в иерархии: ****)
FUNCTION base. IsA = pointer; BEGIN IsA = TypeOf(self); END;

(**** Все правила IsKindOf: ****)
FUNCTION base. IsKindOf(T: Pointer): boolean;
BEGIN IsKindOf := (T=TypeOf(base)); END;

FUNCTION ChildA. IsKindOf(T: Pointer): boolean;
BEGIN IsKindOf := (T=TypeOf(ChildA)) OR base.IsKindOf(T); END;

FUNCTION ChildB. IsKindOf(T: Pointer): boolean;
BEGIN IsKindOf := (T=TypeOf(ChildB)) OR base. IsKindOf(T); END;

FUNCTION GrandA. IsKindOf(T: Pointer): boolean;
BEGIN IsKindOf := (T=TypeOf(GrandA)) OR ChildA.IsKindOffT); END;

FUNCTION GrandB. IsKindOffT: Pointer): boolean;
BEGIN IsKindOf := (T=TypeOf(GrandB)) OR ChildB.IsKindOf(T); END;

(**** Правила WhoAreYou для объектов, где они есть: ****)
PROCEDURE ChildA. WhoAreYou; BEGIN WriteLn('Tnn ChildA'); END;

PROCEDURE GrandA. WhoAreYou; BEGIN WriteLn('THn GrandA'); END;

PROCEDURE Test(B1: Base; VAR B2 ; base);
BEGIN
  Write('непеМеНHafl, переданная ссылкой, ВИДИМО, относится');
  IF B1.IsA = TypeOf(base) THEN WriteLn('К базовому типу');
  IF B1. IsA = TypeOf(ChildA) THEN WriteLn('К типу ChildA');
  IF B1. IsA = TypeOf(GrandA) THEN WriteLn('К типу GrandA');
  Write('непеМеНHafl, переданная значением, относится');
  IF B2. IsA = TypeOf(base) THEN WriteLn('К базовому типу');
  IF B2. IsA = TypeOf(ChildA) THEN WriteLn('К типу ChildA');
  IF B2. IsA = TypeOf(GrandA) THEN WriteLn('К типу GrandA');
END;

VAR P : Pt;
BEGIN
  P := New(GrandAP, Init);
  Test(P, P');

  IF P~.IsKindOf(TypeOf(ChildA)) THEN
    BEGIN
      Write('THn, определенный по объекту - ');
      ChildA(p-).WhoAreYou;
    END;
  ELSE WriteLn('06beKT не имеет правила WhoAreYou');
  Dispose (P, done);

  P := New(GrandBP, Init);
  IF P.MsKindOf(TypeOf(ChildA)) THEN
    BEGIN
      Write('CTnn, определенный по объекту - ');
      ChildA(p-). WhoAreYou;
    END;
  ELSE WriteLn('06beKT не имеет правила WhoAreYou');
  Dispose (P, done);

  ( ChildA(P").WhoAreYou;)>(*Эта строка развалит вам всю систему*)

```

Правило *IsKindOf* реализует в языке *Turbo Pascal 5.5* функцию, аналогичную встроенной в *QuickPascal* функции *Member*.

base, но фактически она может указывать на любой тип в этой иерархии. Если экземпляр типа, на который в данный момент указывает переменная P, находится на ветви A, то вы можете вызвать правило *WhoAreYou*, а если он находится на ветви B, то этого сделать нельзя. Используя правило *IsKindOf*, программа проверяет, допустимо ли обращение к правилу.

Обязательно помните о том, что проверка фактического типа объекта — на грани нарушения инкапсуляции. Строго говоря, у вас не должно возникать необходимости определять, к какому типу принадлежит объект. В рассмотренном примере можно бы-

**Обязательно
помните о том, что
проверка
фактического типа
объекта на грани
нарушения
инкапсуляции.**

ло бы решить задачу и путем создания пустого правила *WhoAreYou* в базовом типе. Использование *IsKindOf* — это просто альтернативный вариант.

Пусть те, кто работает с *QuickPascal*, не чувствуют себя обойденными. Хотя у вас нет функции *TypeOf*, но она вам и не нужна. *QuickPascal* содержит функцию *Member*, которая принимает значение "истина", когда переменная экземпляра относится к заданному типу объекта или к любому из его потомков. Это устраняет необходимость задавать в каждом из объектов отдельное правило *IsKindOf*, что весьма удобно.

Нил Дж. Рубенкинг

О вреде языкознания

Рубен Герр

Рассказывают, что во времена борьбы с неграмотностью некто, доказывая, что ученье — свет, приводил примерно следующие аргументы: "Если бы муха умела читать, она поняла бы — садиться на бумажку с надписью "Смерть мухам!" опасно, и осталась бы в живых".

Каждому пользователю персональных компьютеров волей-неволей приходится запоминать десяток — другой английских слов, а многих именно необходимость работать на компьютере побудила изучить этот язык поглубже. Но так ли уж полезно знать смысл всех слов, используемых для обозначения команд и сигналов в схемах? Применение в языках программирования "естественных" слов, которые там означают не совсем то (а может быть и совсем не то), что в живом языке, зачастую только мешают. Это обнаружили наши специалисты по компьютерам, когда в свое время делались попытки "перевести" языки программирования (например, Алгол) на русский, так и до сих пор считают многие англоязычные программисты.

Опытные пользователи ПК знают, что в большинстве IBM-совместимых компьютеров порт принтера однонаправленный: байты информации можно передавать на подключенное к этому порту устройство, но нельзя вводить через него. Те, кому приходилось разбираться с интерфейсом принтера, знают также, что принтер сигнализирует компьютеру о своем состоянии через несколько специализированных линий. Известно, в частности, что сигнал на линии ERROR указывает на какой-то неполадку в работе принтера, сигнал BUSY — на неготовность принтера к приему информации, а сигнал ACK подтверждает прием очередного байта, так что можно передавать следующий.

Не очень давно перед одним из моих знакомых возникла задача подключить некий прибор к IBM-совместимому компьютеру, а я взялся ему помогать. Разделяя распространенное заблуждение, мы сначала подумали, что без порта типа Centronics с полным набором функций нам не обойтись. Купить "настоящую" (импортную) плату — не было валюты, "самодельная" (читай кооперативная),

во-первых, стоит недешево, а во-вторых, по крайней мере в той фирме, куда мы обратились, оказалась слегка "недоделанной": сначала нам ее расхваливали со всех сторон, а когда речь зашла о приобретении, объяснили, что есть одна особенность — перед приемом каждого байта необходимо засылать в регистр данных этой платы все логические единицы. Каждому, кто имел дело с цифровой схмотехникой, ясно, что разработчики просто забыли предусмотреть сигнал сброса. А раз так, кто их знает, что они еще забыли?.. Надеюсь, что наши несостоявшиеся поставщики со временем пересмотрят свои взгляды на готовые изделия, и не стану обнаруживать название кооператива.

А мы продолжили свои изыскания. Вот так жизнь заставила обратить внимание, во-первых, на форму вывода информации из прибора, во-вторых, на то, как работает параллельный порт ПК. Что касается подключаемого к компьютеру объекта, было выяснено, что информация из него поступает, в частности, и на некий аппарат, которому в документации было присвоено гордое название "цифropечатающее устройство". (На самом деле это был ветхозаветный настольный (!) калькулятор "Искра 106Д".) При этом выводимая информация представлялась в двоично-десятичной форме, т.е. по 4 бита на цифру. Вместе с тем регистр состояния принтера (379h) содержит 5 бит информации (если забыть о предписываемых документацией к операционной системе назначениях этих бит). А ведь это в точности то, что требуется, — 4 бита на цифру плюс сигнал стробирования! Попробовали. Хотя и не сразу, но получилось.

А вот почему не сразу. Как, по-вашему, можно передавать 4 бита информации по 5 линиям? Можно использовать стандартный прием с четырьмя информационными линиями и одной линией стробирования, можно, для надежности, использовать пятую линию для контроля на четность (тогда о наличии информации будет свидетельствовать единица на хотя бы одной из линий)... Но есть еще один способ, сочетающий в себе все недостатки двух указанных, и именно этот способ был избран в нашем случае. Если передается любая цифра, кроме нуля, она выставляется на четырех основных линиях. Если передается ноль — передается сигнал по пятой линии. Вот так был организован прием двоично-десятичных данных в "компьютере" Искра-106Д! Кроме того, обнаружилось, что в раз-

ных компьютерах незадействованные биты регистра состояния принтера принимают в некоторых случаях единичные значения, а в других — нулевые. Еще оказалось, что имеются некоторые несоответствия в нумерации выводов соединителей (мне показалось, что проще переделать программу, чем перепаивать провода). Программа была полностью составлена на Си. Все, что потребовалось, это выяснить, какая из функций обеспечивает считывание информации из порта. А уж "обратить" эту функцию программой так, чтобы она реагировала на нужное сочетание бит и преобразовывала двоично-десятичную цифру (например) в соответствующий код ASCII, настолько просто, что об этом даже скучно рассказывать.

Справедливости ради следует отметить, что мы "изобрели велосипед". Внимательные читатели PC Magazine/USSR наверняка заметили что в опубликованной в первом номере статье о программах пересылки файлов ни в одном случае не требовалось, чтобы параллельный порт был оснащен платой Centronics с полным набором функций как вывода, так и ввода информации, а ведь параллельные порты использовались почти всеми программами. Так что описанный способ известен уже дано и многим.

Тем не менее, учитывая, что еще во многих организациях в СССР установлены цифровые приборы, позволяющие выводить информацию в двоично-десятичном формате, мы считали полезным поделиться обретенным знанием. Для тех, кому лень обращаться к технической документации, сообщим номера разрядов в байте по адресу 379h и соответствующие им выходы соединителя.

Разряд	Название сигнала	Выход соединителя
7	-BUSY	11
6	+ACK	10
5	+PE	12
4	+SLCT	13
3	+ERROR	15

Разряды перечислены от старшего к младшему, знак минус перед названием BUSY означает, что соответствующий бит инвертируется; иными словами, когда к соединителю принтера ничего не подключено, т.е. на всех входах единицы, функция inportb(0x379) примет значение 7h или 7Fh, в зависимости от индивидуальных характеристик интерфейса принтера. Профессионалы, составляя TSR-модули, учтите: прерывания генерируются только от сигнала ACK.

Zortech DOS 386 C++: простой и мощный компилятор

Кааре Кристиан

Оригинальная версия нового компилятора Zortech DOS 386 C++ — это первый истинно 32-разрядный компилятор с языков Си и Си++. Он построен на базе также весьма неплохого 16-разрядного компилятора Zortech DOS-C и C++, который был первым компилятором с языка Си++, предназначенным специально для PC. DOS 386 C++ можно приобрести только в предназначенной для разработчиков версии по достаточно высокой цене в 995 долл. В состав пакета входят компилятор, 32-разрядный символьный отладчик, среда программирования Zortech Workbench, вспомогательные утилиты, библиотеки классов Си++, графическая библиотека Zortech Flash и исходные тексты для библиотеки классов Си++ и библиотеки Си. 32-разрядные прикладные программы, которые вырабатывает этот компилятор, будут работать только в системах на базе процессоров 386 и 486, хотя сам компилятор может работать на любой DOS-системе. Вам также потребуется Phar Lap 386/DOS Extender SDK стоимостью 495 долл. Если вы хотите распространять свои 32-разрядные прикладные программы, вам будет необходима утилита Phar Lap BIND 386.

Как и другие стабильные поставщики 32-разрядных компиляторов Си, например MetaWare и Watcom, Zortech утверждает, что DOS 386 C++ совместим с большинством Си-программ, составленных для 16-разрядных Си-компиляторов, ориентированных на DOS. Однако, в то время, как другие 32-разрядные компиляторы вызываются традиционным образом через командную строку, DOS 386 C++ обладает законченной средой для разработки.

DOS 386 C++ достаточно просто установить в систему, он занимает около 6 Мбайт дискового пространства. Вы можете поместить компилятор Си++ в тот же каталог, где находится 16-разрядный DOS-компилятор, и выбирать, какая программа будет сгенерирова-

на, 32- или 16-разрядная, используя ключи командной строки компилятора. Это дает возможность легко создавать одновременно по два варианта прикладных программ. К сожалению, эта важная возможность не очень хорошо описана.

На самом деле документация — самая слабая часть пакета Zortech, так как она не совсем удачно переписана с документации для 16-разрядной версии. В ней содержится слишком мало специальной информации по 32-разрядной версии и слишком много утверждений, касающихся 16-разрядной версии, что сейчас уже не актуально. Примером этому может служить отсутствие информации о модификаторе типа "far" 32-разрядного компилятора, который представляет собой 48-разрядный указатель на адресный сегмент, отличный от текущего, и используется на машинах 386, работающих в защищенном режиме. Другой пример — описание подпрограммы MK_FP, в котором рассказано о манипуляциях над 16-разрядными указателями, не соответствующих работе с 32-разрядными.

Одна из наиболее выигрышных особенностей компилятора Zortech — его глобальный оптимизатор, построенный на базе уже "проверенного в боях" 16-разрядного оптимизатора. 32-разрядный оптимизатор значительно повышает качество конечных программ. Один из тестов обнаружил, что задача, исполнявшаяся 32 секунды при использовании неоптимизированного кода для микропроцессора 386, стала срабатывать за 25 секунд после использования оптимизатора. Та же программа после компиляции и оптимизации 16-разрядным компилятором отработала за 29 секунд. Большинство программ в результате перехода от 16-разрядного варианта к 32-разрядному существенно ускоряются.

Вас может заинтересовать, насколько сложно преобразовать обычную программу на Си или Си++ для работы в 32-разрядной оболочке Phar Lap. Если программа

работала независимо от аппаратной конфигурации PC и от границы в 640 Кбайт DOS, проблем, действительно, не будет. К сожалению, такие программы — редкость. В большинстве программ для DOS приходится предпринимать "обходные маневры" для преодоления ограничений на объем памяти; во многих программах содержатся аппаратно-зависимые фрагменты. В оболочке Phar Lap предусмотрена обработка традиционных (и не совсем традиционных) вызовов системных прерываний 21h, а Zortech предлагает библиотеки для символьного дисплея, графики и мыши.

Однако остается множество обращений к BIOS, таймеру, клавиатуре, сетевым интерфейсам, аппаратным прерываниям и иным объектам, с которыми вам придется "бороться" само-

лютеки. Для получения этой информации нужно обращаться к резидентной системе подсказок фирмы Zortech. Workbench также содержит 2 утилиты для просмотра исходного текста программ, так что получить информацию несложно.

В отличие от среды для разработки программного обеспечения Borland, где имеются встроенный компилятор и простой отладчик, Zortech Workbench — это скорее редактор, расширенный для более удобного доступа к другим средствам отладки программ. Преимущество подхода Zortech заключается в том, что вы всегда используете полную версию компилятора, средств подготовки исполнимых модулей "Make" и отладчика. Преимущества подхода фирмы Borland — это отсутствие задержек при переходе от одной задачи на другую, а также в более ком-

Компилятор Zortech 386 C++ — это хороший пропуск в 32-разрядную среду, особенно если вы уже пользовались предыдущей 16-разрядной версией

стоятельно. Несмотря на то, что соответствующие приемы описаны в документации Phar Lap, к компилятору Zortech лучше обращаться при работе со стандартными операциями. Например, существует около дюжины специфических для Phar Lap подпрограмм, но Zortech не предоставляет для них интерфейс из Си.

Zortech Workbench — альтернатива традиционному способу задания параметров в цепочке редактирования — компиляция — отладка. Workbench — это многооконный редактор, который дает возможность компилировать код, просматривать сообщения компилятора, запускать программу, вызывать отладчик.

Встроенная система подсказок объяснит, как использовать Workbench, но она не содержит информации ни по языку Си, ни по Си++, а также по подпрограммам стандартной биб-

лотеки. Несмотря на то, что соответствующие приемы описаны в документации Phar Lap, к компилятору Zortech лучше обращаться при работе со стандартными операциями.

Символьный отладчик Zortech, легко доступный из среды Workbench, был специально разработан для оболочки Phar Lap. Это простой в использовании многооконный отладчик. Три стандартные панели показывают исходный текст текущей программы, значения глобальных переменных и значения текущих локальных переменных. Можно, конечно, получать и другую информацию.

Компилятор Zortech 386 C++ — это хороший пропуск в 32-разрядную среду, особенно если вы уже пользовались предыдущей 16-разрядной версией. В документации слишком много ошибок, а в программе встречаются плохо отработанные места, характерные для новинок. Но, в целом, это надежный 32-разрядный компилятор, с удобной программной оболочкой, хорошим отладчиком и впечатляющим оптимизатором кода.

Пакет Object Toolkit облегчает объектно-ориентированное программирование

Рик Эйр

Если последние годы вы не провели в спячке, то вам должно быть известно, что мир персональных компьютеров в буквальном смысле влюбился в объектно-ориентированное программирование (ООП). Ну если не в него, то по крайней мере в саму идею. Такие заложенные в ООП принципы, как наследование, инкапсуляция и использование полиморфизмов, позволяют программистам устанавливать некий порядок без введения ограничений.

Как только стало возможным установить такой новый порядок, сразу появились принципиально новые библиотеки. Библиотеки классов, разрабатываемые сторонними фирмами, скоро, по-видимому, придут на смену более традиционным библиотекам функций. Хорошим примером такой библиотеки является пакет разработчика Object Toolkit производства фирмы TechnoJock Software.

Этот пакет предназначен для работы с компилятором Turbo Pascal версии 5.5 (или более поздней) и требует наличия компьютера IBM PC или машины, совместимой с DOS версии 3.0 (или более поздней), и оперативной памяти объемом 640 Кбайт.

В системе Object Toolkit все сосредоточено на ООП и интерфейсе с пользователем. Это — совокупность объектов и правил, имеющих четко выраженную иерархию. С помощью инструментальных средств, заложенных в систему, можно создавать перемещаемые, допускающие прокрутку окна, управлять входными полями, обрабатывать строки, целые числа и числа с плавающей точкой, а также да- ты, что значительно упрощает создание различных



форм. Кроме того, поддерживаются связанные списки, меню, а также ввод с клавиатуры и с помощью мыши.

Все эти средства конструктивно оформлены в виде блоков компилятора Turbo Pascal; в качестве примера можно привести такие из них, как totLOOK, totSYS, totFAST и totWIN. Эти блоки включают типы объектов, которые, в свою очередь, содержат разнообразные правила. При выполнении некоторых из этих блоков автоматически инициализируются определенные связанные конкретизации типов объектов (т. е. объектные переменные). Так, чтобы ввести в разрабатываемую прикладную программу некоторые базовые интерфейсные функции, можно воспользоваться блоком totFAST. При этом среди всего прочего будет инициализироваться функция записи на экран SCREEN, представляющая собой конкретизацию типа ScreenOBJ.

Пакет Object Toolkit отличается не только мощью заложенного в него инструментария, он еще и довольно прост в освоении. Документация на него хорошо организована и написана доступным языком. В нее включено краткое введение в концепции объектов и объектно-ориентированного программирования. К сожалению, это краткое введение может показаться слишком кратким. Однако все, что следует затем, является прекрасным руководством по работе с пакетом, хотя время от времени и могут встретиться нетрадиционные для объектно-ориентированного программирования термины. Например, для обозначения одного и того же понятия авторы руководства используют три термина — "конкретизация объекта", "глобальная конкретизация" и "глобальный объект", а из них в объектно-ориентирован-

ном программировании применяется лишь "конкретизация объекта". Фирма-разработчик пообещала, что в следующей редакции руководства с терминологическими вольностями будет покончено.

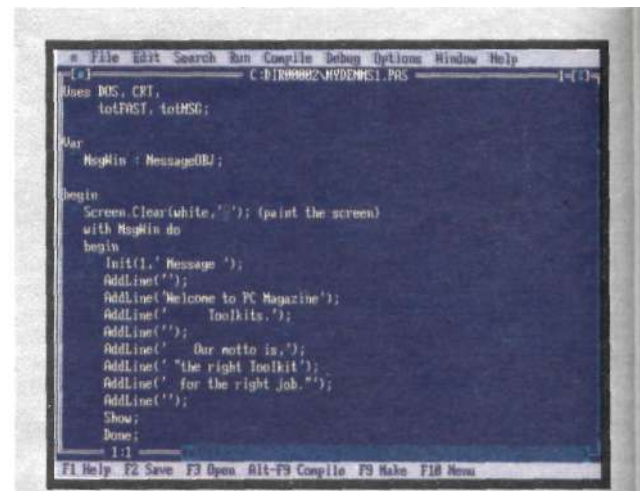
Однако все эти "аномалии" практически не мешают освоению пакета; любой читатель (и специалист по ООП, и новичок) очень скоро "набивает руку" и становится, так сказать, экспертом. (Конечно, чем лучше вы разбираетесь в ООП, тем эффективнее будет ваша работа с пакетом.)

В комплект системы входят одно руководство и две дискеты, которые после распаковки занимают на жестком диске меньше одного мегабайта. Пакет Object Toolkit содержит и некоторые "экзотические" вещи — например наглядное пособие в виде набора карточек, иллюстрирующих все, начиная от иерархии объек-

разработки интерфейсов пользователем не является основным назначением последнего?

Кроме более совершенных по сравнению с Turbo Vision средств ввода и проверки правильности данных в Object Toolkit также реализован "более модульный" подход к программированию. Для него характерна простота выбора и запуск требуемых в процессе работы инструментов. В Turbo Vision, с другой стороны, все объекты связаны друг с другом; это среда, построенная по принципу "один размер подходит всем".

В пакете Turbo Vision также реализован лишь один набор правил разработки, которого необходимо строго придерживаться, причем эти правила выходят за пределы ООП, перекрывая и область программирования с управлением по прерываниям. Основываясь на модели, во многом



Для того чтобы создать диалоговое подокно в тонированном перемещаемом окне, достаточно написать программу длиной всего в несколько строк.

тов и кодов ASCII и кончая составными атрибутами отображения. Отрадно видеть, что иногда можно и отказаться от традиционного стиля подачи материала, изобилующего техническими подробностями и, как результат, — часто весьма сложного для восприятия.

Здесь уместен вопрос: зачем нужен пакет Object Toolkit, когда у вас уже есть хорошо известный Turbo Vision? Разве обеспечение

напоминающей систему программирования Microsoft Windows, этот пакет требует, чтобы для создания текстовых интерфейсов, аналогичных тем, которые реализованы в Windows, обязательно использовалась модель с управлением по прерываниям.

В системе же Object Toolkit принцип управления по прерываниям не используется; более того, не (окончание см. стр. 113)

Коста Родис

ИЗМЕНЕНИЕ НАЧАЛЬНЫХ ПАРАМЕТРОВ ПЕЧАТИ В РЕДАКТОРЕ WORD 5.5

Я работаю с текстовым редактором Microsoft Word. Недавно у меня появилась версия 5.5, и в целом я ею доволен. Но, к сожалению, в этой последней версии редактора отсутствует макрокоманда, которой я регулярно пользовался. Я имею в виду макрокоманду Printer Reset (переопределение параметров принтера), позволяющую устанавливать новые начальные параметры принтера в рабочем окне режимов.

Последняя версия редактора Word не дает возможности переопределять некоторые параметры печати, которые действовали бы по умолчанию. В частности, вы не можете установить число экземпляров иначе, чем непосредственно перед выдачей на принтер. Поэтому приходится всякий раз, перед тем как запустить печать, проверять правильность установки параметров, вместо того чтобы менять их только тогда, когда нужно. Правда, существует способ обойти данное ограничение: можно установить стандартные для вас режимы работы принтера и записать подготовленный для печати документ на диск, используя функцию PRINT File. По завершении сеанса печати редактор Word 5.5 автоматически восстанавливает режим вывода данных на принтер.

Терри Абрахам
Москоу, шт. Айдахо

От редакции:

Некоторые параметры принтера, такие как его тип, разрешающая способность и вид порта, могут быть установлены в режиме Printer Setup (начальная установка параметров принтера). Однако такие характеристики, как диапазон страниц (Page Range) и число экземпляров, можно определять только в процессе конкретного сеанса работы с принтером. Тем не менее отмеченные ограничения можно обойти, выводя данные для печати не сразу на принтер, а сначала записывая их на диск — тут Терри Абрахам совершенно прав. Но, если вы редко меняете параметры печати, то для ускорения работы лучше отказаться от промежуточного вывода на диск и пользоваться макрокомандой, такой, например, как PrinterReset (см. листинг).

Применение макрокоманды Printer Reset, которая устанавливает основные режимы работы принтера, гаран-

■ ИЗМЕНЕНИЕ НАЧАЛЬНЫХ ПАРАМЕТРОВ ПЕЧАТИ В РЕДАКТОРЕ WORD 5.5:

Последняя версия Microsoft Word позволяет переопределять некоторые параметры печати только непосредственно перед выдачей на принтер. Приведена макрокоманда, позволяющая обойти это ограничение.

■ ПОВЫШЕНИЕ ПОЛИГРАФИЧЕСКОГО КАЧЕСТВА ТЕКСТА С ВЕРХНИМИ И НИЖНИМИ ИНДЕКСАМИ:

За счет небольшого изменения макрокоманд простановки верхних и нижних индексов редактора Word for Windows можно избавиться от увеличения интерлиньяжа.

■ Недоработки в оболочке Windows: измерительные линейки и шрифты

тирует наличие нужных вам параметров непосредственно перед выводом данных на печать. В двух словах действия макрокоманды сводятся к следующему: открыть окно параметров печати; установить количество экземпляров, равное 1; присвоить параметру выбора страниц значение "все" (all); удалить прежний список номеров выводимых страниц (если есть); установить тип принтера (HP LaserJet). Если вам нужно задать другие параметры печати, то следует соответствующим образом изменить макрокоманду. Вы можете также расширить возможности макрокоманды, включив туда установку вида порта, разрешающей способности принтера, имени активного каталога и режима черновой или двухпроходной печати. Кроме того, можно указать, нужна ли динамическая загрузка шрифтов или печать итоговых данных и скрытого текста. Для этого нужно всего лишь добавить к тексту макрокоманды соответствующие строки.

Макроязык редактора Word версии 5.5 тот же, что и версии 5.0. Надо только помнить, что спецификации меню стали шире и последовательность нажатия клавиш должна соответствовать новому интерфейсу. Например, вместо того чтобы набирать P для вызова на экран меню режимов вывода на печать, теперь следует набирать Alt-F P. Если вы перед этим некоторое время работали с пакетом Word for Windows, то вы непременно заметите сходство структуры меню этих редакторов, хотя интерфейс в целом и может показаться вам несколько неудобным. Те, кто до этого работал с версией Word 5.0, могут при знакомстве с версией 5.5 найти ее такой же удобной или же почувст-

PRINTRESET

ПОЛНЫЙ ЛИСТИНГ

```
«message Printer Defaults Reset»
«COMMENT Вывод сообщения в нижней части экрана»
«set echo = "OFF"»
«COMMENT Отключение режима "эхо"»
<alt f>p
«COMMENT Переход в меню File и выбор пункта установки режимов печати»
<alt c>1
«COMMENT Подвод курсора к параметру числа экземпляров и ввод значения "1"»
<alt a>
«COMMENT Установка режима печати всех страниц документа»
<altsxdel>
«COMMENT Удаление записанного ранее списка страниц для печати»
<alt r>hplaser
«COMMENT Выбор пунктов Printer Setup в меню и определение типа принтера»
<alt e>continuous<enter>
«COMMENT Установка режима непрерывной подачи бумаги и возврат в главное меню»
<alt f>testprnt<enter>
«COMMENT Запись документа в формате вывода на печать в файл TESTPRNT»
«COMMENT Прежнее содержимое файла утрачивается»
<enter>
«COMMENT Печать/возврат в режим редактирования документа»
```

Эта макрокоманда позволяет быстро устанавливать необходимые вам параметры работы принтера для редактора Microsoft Word.

воват полный дискомфорт. После византийского стиля меню предыдущих версий редактора Word новая организация интерфейса может показаться просто предательством.

Для того чтобы записать макрокоманду, необходимо вызвать меню макрокоманд, выбрать пункт Record, ввести имя макрокоманды и, если хотите, сокращенную комбинацию клавиш, а затем нажать <enter>. Далее по приведенному тексту, нажимайте клавиши, показанные в угловых скобках (), и набирайте на клавиатуре символы, расположенные вне скобок. Так, для записи последовательности alt fr введите букву f, держа при этом нажатой клавишу Alt, а затем отпустите клавишу Alt и введите букву P. Аналогичным образом запишите все строки. Затем нажмите Alt-M (в режиме записи макрокоманды "мышь" не функционирует) и выберите в меню пункт Stop Recorder.

Если вы предпочитаете набирать текст макрокоманды так, как он выглядит на рисунке, то после ввода имени макрокоманды и присвоения ей сокращенной комбинации клавиш выберите в системе меню последовательно пункты Macro Stop Recorder (Alt-M,C). Затем снова вызовите меню макрокоманд Macro и войдите в режим Edit. Выберите имя вашей макрокоманды и опять укажите режим Edit. Теперь вы можете набирать текст макрокоманды в том виде, в котором он представлен на рисунке. Для ввода двойных угловых кавычек, в которые заключаются комментарии, используйте клавиши <[> и <]> в комбинации с <enter>. Завершив процедуру ввода, выделите весь текст и выполните команду удаления или копирования (Alt-E, C). Выберите в меню последовательно пункты Macro, Edit, Define и ответьте "Yes" на вопрос о замене макрокоманды. Закройте окно редактирования макрокоманды и вернитесь в режим работы с документом. Удалите, если нужно, случайно оставленный в тексте документа фрагмент макрокоманды.

К макрокоманде можно обратиться двумя способами: выбрать пункты Macro и Run в меню или нажать присвоенную макрокоманде комбинацию клавиш. Теперь, что бы вы ни печатали, макрокоманда установит указанные вами режимы работы принтера.

Повышение полиграфического качества текста при работе с индексами

Простановка нижних и верхних индексов в редакторе Word for Windows обычно приводит к увеличению расстояния между строками,

НЕ "СЛОВОМ" ЕДИНЫМ

SUPERSCRIPIT

ПОЛНЫЙ ЛИСТИНГ

```
Sub MAIN
FormatDefineStylesStyleName$( )
REM          Определить новую гарнитуру шрифта на основе текущей.
Dim dig As FormatDefineStylesChar
REM          Присвоить текущей гарнитуре имя "dig".
GetCurValues dig
REM          Сохранить параметры переменной "dig" в DialogRecord.
If SuperScript() Then
REM          Если требуется верхний индекс, то...
FormatCharacter .Points = dig.Points, .Position = 0
REM          Отформатировать выбранный текст в соответствии с параметрами,
REM          указанными в "dig".
Else
REM          В противном случае...
FormatCharacter .Points= Val(dig.Points) - 2, .Position = "2pt"
REM          Отформатировать выбранный текст так, чтобы его размер был на
REM          два пункта меньше, чем указано в "dig", а базовая линия
REM          поднялась на два пункта выше прежней.
End If
End Sub
```

Данная макрокоманда позволяет получить более полиграфический текст с верхними индексами, чем стандартная встроенная макрокоманда редактора Word for Windows. Вариант макрокоманды для нижних индексов показан на следующем листинге.

а текст с неравномерными межстрочными интервалами выглядит непрофессионально. Однако расстояние между строками можно регулировать с помощью специальных макрокоманд Superscript и Subscript, представленных на соответствующих листингах. По существу это обычные встроенные макрокоманды редактора Word for Windows, в которые я внес необходимые изменения. Мои макрокоманды сжимают предварительно выделенный текст по вертикали на два пункта. Кроме того, они сокращают высоту верхнего индекса до двух и глубину нижнего индекса до

полтора пунктов относительно базовой линии. Если размеры выделенного текста уже были модифицированы с учетом индексов, то макрокоманды восстанавливают начальные параметры строк и базовой линии.

При помощи-линейки пиктограмм I редактора Word for Windows можно I вызывать только встроенные макрокоманды. Иначе говоря, применение пиктограмм не позволяет изменять принятые по умолчанию параметры изображения текста, содержащего верхние и нижние индексы. Поэтому вы не можете обращаться к но-

SUBSCRIPT

ПОЛНЫЙ ЛИСТИНГ

```
Sub MAIN
REM          Любая макрокоманда должна начинаться с заголовка Sub MAIN.
FormatDefineStyles StyleName$( )
REM          Определить новую гарнитуру шрифта на основе текущей.
Dim dig As FormatDefineStylesChar
REM          Присвоить текущей гарнитуре имя "dig".
GetCurValues dig
REM          Сохранить параметры переменной "dig" в DialogRecord.
If Subscript() Then
REM          Если требуется нижний индекс, то...
FormatCharacter .Points = dig.Points, .Position = 0
REM          Отформатировать выбранный текст в соответствии с параметрами,
REM          указанными в "dig".
Else
REM          В противном случае...
FormatCharacter .Points=Val(dig.Points) - 2, .Position="-1.5pt"
REM          Отформатировать выбранный текст так, чтобы его размер был на
REM          два пункта меньше, чем указано в "dig", а базовая линия
REM          опустилась на полтора пункта ниже прежней.
End If
End Sub
```

Предлагаемые макрокоманды (см. также предыдущий листинг) дают возможность избежать увеличения межстрочных интервалов, которое возникает при работе со стандартными встроенными макрокомандами редактора Word for Windows.

вым макрокомандам посредством пиктограмм.

Р. Шоу
Жёврэн, Швейцария

От редакции:

Достоинства предлагаемого решения легко увидеть на тестовом примере, где одна строка получена с использованием модифицированных макрокоманд, а другая — при помощи встроенных. Строки, выбранные посредством псевдокнопок, не меняют своего размера относительно остального текста, а только опускаются или поднимаются на половину межстрочного интервала, в то время как макрокоманды, модифицированные мистером Шоу, придают документу более "опрятный" в полиграфическом отношении вид за счет уменьшения размера верхних и нижних индексов по сравнению со строчными буквами текста. Еще одно отличие предлагаемых макрокоманд от встроенных состоит в том, что они не увеличивают столь заметно размер строки вверх или вниз относительно базовой линии. Это означает, что верхний индекс только чуть-чуть выступает за самую высокую, а нижний — за самую низкую букву строки.

Если вы загружаете предлагаемые макрокоманды из сети PC MagNet, то их можно легко импортировать в библиотеку макрокоманд Word for Windows. Сначала загрузите каждую макрокоманду как файл в текстовом формате, используя для этого режим Insert File. Выделите в окне текст макрокоманды и выберите в меню пункты Edit и Cut. Затем проверьте верхнюю меню. Если в ней не появилось имени макрокоманды, то выберите пункт Full Menus в режиме View, и слева от окна появится текст макрокоманды. Откройте окно редактирования макрокоманд (Macro, Edit) и выберите из списка пункт Superscript или Subscript. Удалите существующую макрокоманду и запишите на ее место новый вариант.

Если вы будете вводить текст макрокоманды вручную, то просто скопируйте приведенный листинг. Любую строку, начинающуюся со слова REM, можно пропустить, поскольку она представляет собой комментарий к предшествующему оператору. Если вы хотите добавить собственные строки комментария,

(начало см. стр ПО)

требуется даже, чтобы применялся интерфейс с окнами. А для тех из вас, кто выше всего прочего ценит скорость, в систему включены объекты, составленные на ассемблере, который выводит информацию прямо на экран. Это единственная часть пакета, запрограммированная на языке ассемблера.

НЕ "СЛОВОМ" ЕДИНЫМ

Есть ли на свете какой-либо текстовый процессор который может достоверно отображать на экране то, что будет потом напечатано на бумаге?

то начинайте их также со слова REM или апострофа.

А WYSIWYG ЛИ ЭТО?

Что происходит с пакетом Word for Windows? Складывается впечатление, что обещанный режим WYSIWYG работает далеко не всегда.

В описании Windows утверждается, что пользователю доступны десятки шрифтов разных размеров, однако очень немногие из них отображаются на экране в том виде, в каком они будут выглядеть на бумаге. Например, при использовании кегля 6 в строке длиной 6 дюймов (15,24 см) перенос, если верить измерительной линейке на экране, может появиться на отметке 4,5 дюйма (11,43 см). Тем не менее на бумаге будет напечатана одна-единственная строка длиной 6 дюймов. По существу, линейка оказывается бесполезным инструментом при форматировании. Можно ли как-нибудь разрешить эту проблему, и все ли я правильно делаю?

Эдди Николсон
Брукстон, шт. Техас

От редакции:

Есть ли на свете какой-либо текстовый процессор который может достоверно отображать на экране то, что будет потом напечатано на бумаге. Если и есть, то я такого не видел. Основная проблема заключается в несоответствии между разрешающими способностями монитора и конкретного принтера. Поскольку монитор и принтер представляют собой устройства вывода на разные физические носители, добиться абсолютной достоверности отображения шрифтов на экране практиче-

PC Magazine. June 11, 1991, p. 407.

Системе Turbo Vision по плечу самые разнообразные задачи, например создание прототипов. Однако, если вам требуется такое введение в ООП, которое можно "одолеть", не погрязнув в технических подробностях, то вам нужен именно Object Toolkit. Мне, например, удалось всего за несколько минут (написав программу в

ски невозможно. Действительно, даже совершенно плоский экран дисплея, пусть и при самом тщательном подборе цвета, очень мало походит на документ, полученный с помощью лазерного принтера.

Проверяя сообщение нашего читателя, я распечатал образцы различных моноширинных и пропорциональных шрифтов. Для каждого из них по измерительной линейке экрана Word for Windows отмечалось точное местоположение тильды. Те же измерения затем были проделаны с пакетами Ami Pro, WordPerfect 5.1 и Word 5.0.

Очевидно, в случае с Word for Windows мы имеем дело с несовершенством самого пакета. Например, версия 1.0 очень точно позиционирует линейку по отношению к шрифтам гарнитуры Courier. Однако, когда дело доходит до пропорциональных шрифтов, пакет ведет себя из рук вон плохо. Интересно, что последняя, усовершенствованная версия Word for Windows, наоборот, довольно точно измеряет пропорциональные шрифты, в то время как Courier со шкалой не совпадает.

Пакет Ami Pro выводит шрифты на экран в точном соответствии со шкалой табуляций. Сравнивая его с Word for Windows, приходишь к выводу, что проблема последнего заключается отнюдь не в экранных шрифтах. Скорее, как это и предположил наш читатель, шкалу Word for Windows нельзя считать надежным измерительным инструментом.

Проверка пакетов WordPerfect 5.1 и Word 5.0 показала, что если моноширинные тексты на их экранах с какой-то степенью точности соответствуют шкале, то о пропорциональных гарнитурах такого не скажешь. Поскольку обе эти программы не работают в графическом режиме, положение каждого символа приходится сверять по подсказке в нижней строке экрана, которая содержит только текущие координаты курсора. При работе во всех рассмотренных пакетах позиционирование текста (не важно, моноширинного или пропорционального) можно достаточно точно осуществлять, оперируя табуляцией, отступами или используя другие приемы, рассчитанные на абсолютные единицы измерения (как, например, режим Advance Format пакета WordPerfect).

26 строк) создать диалоговое подокно в тонированном перемещаемом окне. Фирма TechnoJock не новичок в Паскале. Ею уже создано несколько библиотек для этого языка.

Цена по каталогу: 89 долл.
Требуемые ресурсы: Turbo Pascal, версия 5.5 или более поздняя, и DOS, версия 3.0 или более поздняя.
TechnoJock Software Inc., P.O. Box 820927, Houston, TX 77282; 713-493-5872.

ПОЛЬЗОВАТЕЛЬ ПОЛЬЗОВАТЕЛЮ

Нил Дж. Рубенкинг

ПЕРЕХОДЫ
МЕЖДУ КАТАЛОГАМИ

На рисунке ниже приведен текст удобного командного файла 2.BAT, с помощью которого можно сразу перейти в любой каталог любого диска. Все, что при этом требуется ввести, — это буква, обозначающая диск, и имена каталогов нужного вам маршрута, разделенные пробелами. Например, написав

```
2 DTURBOC INCLUDE
```

вы попадете в каталог D:\TURBOC\INCLUDE, где бы вы перед этим ни находились. Если нужный каталог находится на том же диске, то имя диска задавать не обязательно. Чтобы выйти в корневой каталог текущего диска, достаточно написать просто 2.

Если вы зададите подкаталог, которого не существует, то программа все же переместит вас по маршруту на максимально возможное расстояние. Пусть, например, вы ввели

```
2ATURBOCFILES
```

причем каталог A:\TURBOC существует, а A:\TURBOC\FILES — нет. Тогда произойдет переключение на каталог A:\TURBOC. Следует отметить, что для правильной работы программы нужно, чтобы у вас не было таких подкаталогов, имена которых совпадали бы с буквенными обозначениями дисков.

Уинстон Ли
Монреаль, пров. Квебек
Канада

От редакции:

Я терпеть не могу вводить имена маршрутов файловой системы. Во-первых, двоеточие нужно вводить при нажатой клавише SHIFT. Во-вторых, несколько раз приходится искать клавишу "\", причем делу отнюдь не помогает, что изготовители клавиатур до сих пор экспериментируют с ее размещением! Так как программа 2.BAT не требует ни той, ни другой из этих клавиш, я с удовольствием включил ее в свой набор командных файлов повседневного использования.

Если у вас больше пяти логических дисков, добавьте новый перечень их буквенных обозначений. Для своей системы я ввел дополнительно диски F: и G:, убрав REM в третьей и пятой строках приводимого текста.

- ПЕРЕХОДЫ МЕЖДУ КАТАЛОГАМИ: Этот удобный командный файл позволяет за один шаг перейти в любой каталог любого диска.
- ЗАПОМИНАНИЕ ИСХОДНОГО ПУТИ: Два экзотических командных файла дают возможность запоминать текущий каталог и возвращаться в него.

Программа 2. BAT "снисходительнее" к пользователю, чем команда CD, которую она заменяет. Если хотя бы один элемент в длинном наименовании маршрута неверен, CD не сработает вообще. В то же время 2.BAT по крайней мере переключится на последний правильный элемент заданного ей списка, причем есть шанс, что получится еще лучше. Пусть, например, у вас есть каталог C:\DATA\LETTERS, а вы по ошибке написали

```
2 C DATA FILES LETTERS
```

Тогда 2.BAT переключится на диск C:, перейдет в каталог DATA и попытается перейти в подкаталог FILES. Такого подкаталога не существует, что покажет сообщение Invalid directory, но 2.BAT на этом не останавливается — она попытается переключиться на подкаталог LETTERS, и это ей удастся! Итак, хотя 2. BAT не может за вас восстановить пропущенное имя подкаталога, она может проигнорировать лишнее.

ЗАПОМИНАНИЕ
ИСХОДНОГО ПУТИ

Утилиты PUSHDIR и POPDIR, опубликованные в PC Magazine (27 мая 1986 г., раздел "Программы и утилиты"), очень хороши, но, будучи резидентными, занимают драгоценное

пространство в оперативной памяти и, кроме того, могут вызвать крах системы в случае конфликта с другими резидентными программами. Я написал вместо них короткие командные файлы, которые, не будучи резидентными, тем не менее позволяют запоминать переходы между подкаталогами и возвращаться обратно при любом желаемом числе уровней вложения.

Используемый метод основан на идее, впервые предложенной Дж. Кевином Уэллсом (PC Magazine, June 26, 1990). Основная программа PUSHD.BAT имеет вид

```
@IF '%1' == 'X' PROMPT @$_:$_ @CD $P
@IF '%1' == ' ' %COMSPEC%/C %0
X > C:\BATCH\POPD.BAT
```

Этот компактный файл запоминает текущие диск и каталог путем создания второго командного файла (POPD.BAT), который при вызове осуществляет возврат к исходному диску и каталогу. Все его команды содержатся в команде PROMPT первой строке.

Первый раз вы запускаете PUSHD без параметров, поэтому исполняться будет вторая строка. В соответствии со значением переменной COMSPEC в командной среде оператор %COMSPEC%/C временно загружает второй экземпляр командного процессора, который исполняет следующую строку файла. Эта строка — имя того же командного файла с параметром X. При повторном исполнении файла с таким параметром изменяется приглашение (PROMPT); затем исполняется пустая строка, находящаяся в конце файла. (Символ \$_ в приглашении соответствует комбинации "возврат каретки — перевод строки", то есть создают пустую строку.) Второй командный процессор, встретив пустую строку в командном файле, должен выдать приглашение, если включен режим ECHO ON (а он включен по умолчанию). Но его выход перенаправлен в файл POPD.BAT, который примет следующий вид:

```
@<drive>:
@CD <subdirectory>
```

2. BAT

ПОЛНЫЙ ЛИСТИНГ

```
@ECHO OFF
FOR %%h IN (a A b B c C d D e E) DO IF (%1) == (%%h) %%h:
REM FOR %%h IN (f F g G) DO IF (%1) == (%%h) %%h:
FOR %%h IN (a A b B c C d D e E) DO IF (%1) == (%%h) SHIFT
REM FOR %%h IN (f F g G) DO IF (%1) == (%%h) SHIFT
FOR %%f IN (\ %1 %2 %3 %4 %5 %6 %7 %8 %9) DO CD %%f
```

Рис.1: S отличие от команды CD ЭТОТ командный файл доставит вас в любой каталог любого диска даже при не совсем корректно заданных параметрах.

а это как раз то, чего мы хотим. Вторичный командный процессор всегда наследует приглашение от первичного, но не наоборот. Поэтому после выхода из вторичного процессора исходное приглашение автоматически восстановится.

Когда вы заканчиваете работу с каким-нибудь прикладным пакетом, часто бывает нужно вернуться на тот диск и в тот каталог, где вы находились до этого, и PUSHHD идеально подходит для такой цели. Предположим, что вы каждый день работаете с WordPerfect. Если PUSHHD находится в одном из подкаталогов текущего маршрута, то вы можете написать следующий командный файл:

```
@echo off
CALL PUSHHD
C:
CD\WP
WP
CALL POPD
```

После выхода из WordPerfect этот файл автоматически вызовет POPD, и вы тут же вернетесь в свой исходный каталог.

В программе PUSHDIRS.BAT (см. листинг) этот метод развит таким образом, чтобы обеспечивать запоминание нескольких путей. Программа основана на том же принципе, но создает последовательно несколько командных файлов, каждый из которых содержит очередную текущую настройку. Она также создает файл POPDIRS.BAT, который выполняет все созданные командные файлы в обратном порядке, возвращая вас к нужному диску и подкаталогу. В POPDIRS имеется строка, передающая управление обратно к PUSHDIRS с целью "очистить стек" — каждый командный файл после его выполнения уничтожается.

Последовательность, в которой все это происходит, задается командами FOR в PUSHDIRS. Эти циклы могут содержать любое желаемое число разных аргументов, лишь бы порядок их следования во втором цикле был обратным по отношению к порядку в первом. Программы PUSHHD И PUSHDIRS можно запускать как из командного файла, так и прямо из командной строки DOS. Удобно держать их в одном из подкаталогов, указанных в операторе PATH вашего файла AUTOEXEC.BAT. Можно сделать даже лучше: скопировать все на электронный диск и установить соответствующий путь.

Арнл П. Рейес
Лос-Аламос, шт. Нью-Мексико

От редакции:

Эти командные файлы — самые загадочные из всех, какие мне попадались,

PUSHDIRS.BAT

ПОЛНЫЙ ЛИСТИНГ

```
@IF '%1'=='X' PROMPT @$N:$_CD $P
@IF '%1'=='X' GOTO X
@ECHO OFF
@IF '%1'=='Z' GOTO Z
SET D=
FOR %%A IN (3 2 1) DO IF NOT EXIST C:\BATCH\%%A SET D=C:\BATCH\%%A
IF NOT '%D%' == " GOTO NotFull
ECHO Извините, список каталогов заполнен!
REM > C:\BATCH\QUIT.BAT
QUIT
:NotFull
%COMSPEC%/C%0X>%D%
ECHO @%0 Z >%D%
GOTO Y
:Z
IF EXIST %D% DEL %D%
FOR %%A IN (1 2 3) DO IF EXIST C:\BATCH\%%A SET D=C:\BATCH\%%A
:Y
COPY %D% C:\BATCH\POPDIRS.BAT >NUL
: X
```

Рис.2: Эта программа создает файлы, которые запоминают и восстанавливают несколько пар диск/каталог.

но они работают! Проследим строка за строкой, что делает PUSHDIRS.BAT.

Первая строка проверяет, не задан ли в командной строке параметр X. Так как мы его не задавали, то ничего не происходит. Вторая строка, наоборот, проверяет условие отсутствия параметра; оно истинно, поэтому выполняется остальная часть этой строки. Переменная среды %COMSPEC% разворачивается в полное имя пути к вашему текущему командному процессору — обычно это COMMAND.COM. Аргумент %0 заменяется именем выполняемого в данный момент командного файла. Таким образом, эта команда вызывает вторую копию командного процессора, который должен еще раз запустить этот же командный файл, но уже с параметром X, причем весь выход перенаправлен в файл POPD.BAT.

При втором исполнении нашего командного файла условие, проверяемое в первой строке, оказывается истинным, поэтому происходит изменение системного приглашения. В строке, описывающей приглашение, \$п обозначает текущий диск, \$р — текущий каталог, а \$_ — новую строку. Таким образом, данная команда формирует системное приглашение из двух строк, в первой из которых стоит буква текущего диска и двоеточие, а во второй — команда переключения на текущий каталог.

Во время второго исполнения этого командного файла командная строка уже не пуста, поэтому вторая строка файла не выполняется. Однако пустая третья строка заставляет DOS выдать системное приглашение. А в этот момент весь выход, в том числе

и системное приглашение, направлен в файл POPD.BAT.

На этом заканчивается повторное обращение к нашему командному файлу, поэтому вторичный командный процессор завершает свою работу. Измененное системное приглашение действовало только на него. Как только он завершил работу, восстановилось то приглашение, которое существовало до этого. В результате создан файл POPD.BAT из двух строк, в первой из которых стоит команда перехода на текущий диск, а во второй — в текущий каталог.

Попробуйте сами проследить таким же образом работу файла PUSHDIRS.BAT. Это не так просто, но, я думаю, что вы по достоинству оцените его изощренную логику. Я добавил в него пару строк для проверки переполнения стека. Если вы получите сообщение, что стек заполнен, то это скорее всего будет говорить о том, что надо ввести дополнительные параметры в два цикла FOR программы PUSHDIRS.BAT. Очистить этот стек можно с помощью следующей программы CLERSTAK.BAT:

```
@ECHO OFF
FOR %%v IN (1 2 3) DO IF EXIST
C:\BATCH\%%v DEL C:\BATCH\%%v
```

Поместите команду вызова этой программы в свой AUTOEXEC.BAT, и тогда каждый день вы будете начинать работу при пустом стеке каталогов. Не забывайте, что, если вы добавите параметры в циклы FOR программы PUSHDIRS.BAT, то нужно это же сделать и в CLERSTAK.BAT.

НАСТРОЙКА AUTOEXEC. BAT

Статья "Выше скорость, меньше размер" (PC Magazine, April 10, 1990) была достаточно информативной, но к ней, по-моему, следует кое-что добавить. Когда вы копируете AUTOEXEC.BAT на виртуальный диск, переименуйте файл (скажем, в REFRESH.BAT), чтобы в дальнейшем не запутаться при его запуске, удалении или при внесении в него изменений.

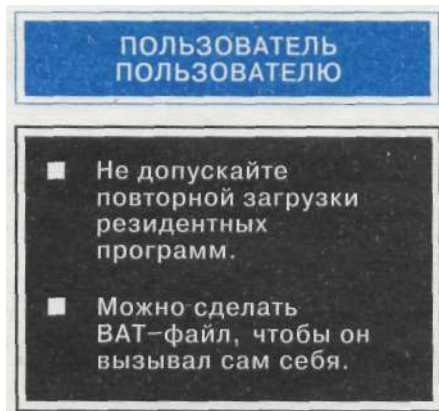
Команды в моем AUTOEXEC.BAT подразделяются на две категории. К первой относятся команды, повторное выполнение которых никак не сказывается на операционной системе компьютера (например, PROMPT и PATH). Вторая категория содержит команды, повторное выполнение которых может оказаться губительным для системы (например, резидентные программы, которые при их повторном запуске вызывают "зависание" компьютера). Поскольку при загрузке операционной системы параметр %0 в файле AUTOEXEC.BAT имеет нулевое значение, а при перезапуске файла параметр %0 получает значение AUTOEXEC, появляется возможность избежать повторной загрузки "опасных" резидентных программ. Обновленный AUTOEXEC. BAT может выглядеть примерно так:

```
IF NOT '%0'==" GOTO again
RAMDISK: /M=100
REM здесь включены команды
REM однократного использования
COPY AUTOEXEC. BAT C: REFRESH. BAT
C: REFRESH
: again
REM здесь включены команды
REM многократного использования
```

Чтобы предостеречь пользователя от повторного вызова AUTOEXEC. BAT вместо REFRESH.BAT, добавьте следующие команды в начало файла AUTOEXEC.BAT:

```
IF NOT '%0'=="AUTOEXEC GOTO bootchk
ECHO Пожалуйста, используйте команду
ECHO REFRESH для восстановления
ECHO начальных параметров системы
TYPE C: QUIT. BAT > C: QUIT. BAT
C: QUIT
: bootchk
```

QUIT.BAT — это командный файл нулевой длины, который останавливает дальнейшее выполнение команд и возвращает вас в DOS. Учтите, что приведенный вариант не обладает необходимой завершенностью: он чувствителен к тому, например, прописными или строчными буквами вы набираете команду в командной строке. Однако с его помощью можно достаточно наглядно



проиллюстрировать используемый технический прием.

Роберт Катлер
Голд Ривер, шт. Калифорния

От редакции:

Большинство резидентных программ достаточно интеллектуальны для того, чтобы при повторных вызовах не загружать себя еще раз, но существуют и такие, которые ведут себя иначе. Если даже при повторном обращении к подобной программе ваша система не "зависнет", лишняя копия вашей программы займет место в оперативной памяти. Чтобы этого не случилось, нужно определенным образом сконфигурировать файл AUTOEXEC.BAT.

В командной строке командного файла может быть до девяти параметров: от %1 до %9. DOS присваивает параметру %0 имя самого bat-файла, причем именно то, которое напечатано в командной строке. Однако начальная загрузка AUTOEXEC.BAT представляет особый случай: здесь имя файла параметру %0 не присваивается.

Хочется сделать несколько замечаний относительно второго отрывка приведенного автором примера. Начнем с того, что начальная проверка чувствительна к регистру символов в командной строке. На самом деле, здесь нет никакой проблемы, поскольку мы можем написать первую строку следующим образом:

```
IF '%0'==" GOTO bootchk
```

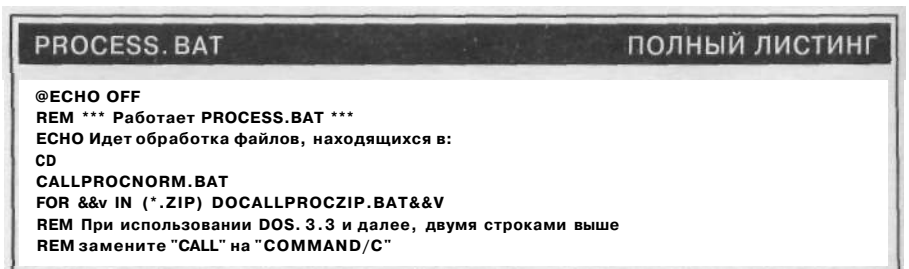


Рис.1: ЭТОТ командный файл входит в тройку программ, исполняющих заданные операции над всеми файлами в каталоге, включая и входящие в состав архива.

Затем автор, что называется "на ходу", создает файл QUIT.BAT (bat-файл нулевой длины). Очевидно, полезно файл QUIT.BAT постоянно "держать" в каталоге своих командных файлов, с тем чтобы использовать его по мере надобности (например, для выхода из работающего bat-файла). На диске QUIT.BAT занимает лишь 32 байта.

РЕКУРСИВНЫЕ КОМАНДНЫЕ ФАЙЛЫ

Как правило, информация, которая приходит по BBS, заархивирована в виде .ZIP-файлов, причем BBS добавляет в каждый из них какую-нибудь рекламу. В такой ситуации неплохо бы иметь возможность удаления рекламы, а также проверки принимаемых файлов на наличие вирусов, прежде чем сбрасывать их на архивные диски.

Для автоматизации этого процесса я написал командный файл, который прекрасно работает с любыми файлами, включая .ZIP, до тех пор, пока в них не обнаружатся вложенные .ZIP-файлы. Единственный способ, который я смог придумать в этой ситуации, — это распаковать все вложенные .ZIP-файлы в отдельный подкаталог, проверить их на наличие вирусов и затем стереть подкаталог, оставив .ZIP-файлы нетронутыми. Хотя такой способ и показал свою состоятельность, приходится выполнять массу лишней работы. Существует ли какой-нибудь другой вариант решения проблемы, позволяющий обойтись без промежуточных операций?

Том Блэк
Макхенри, шт. Иллинойс

От редакции:

Решение проблемы — в создании рекурсивного командного файла (то есть такого, который может неоднократно вызывать самого себя). Я модифицировал присланный автором оригинальный файл и получил для управления .ZIP-файлами с любым уровнем вложенности три командных файла, работающих совместно. Основной файл —

ДОРОГОЙ ЧИТАТЕЛЬ!



Мы рады сообщить, что издательство SK Verlag International, не ограничиваясь выпуском журнала PC Magazine/USSR, приступает к изданию серии русских переводов книг по компьютерной тематике, публикуемых издательством Ziff-Davis Press. В Соединенных Штатах книги этого издательства, составляющие "Библиотеку PC Magazine", неизменно возглавляют списки бестселлеров. В 1992 г. вашему вниманию будут предложены (названия, объемы, время выхода в свет и цены предварительные):

Н. Рубенкинг. Turbo Pascal 6.0: приемы и методы программирования. 25 печ. л.
Цена 20 р.II кв.

Г. Вендитто. Руководство по работе с системой Windows 3.1. 15 печ. л.
Цена 15 р.III кв.

Р. Хаммел. Современные процессоры и сопроцессоры: справочные данные для программистов. В 2-х томах. 40 печ. л.
Цена 45 р.IV кв.

Тираж изданий ограничен, и если вы или ваша организация хотите успеть приобрести эти книги, присылайте заявки в адрес редакции PC Magazine/USSR до 1 марта 1992 г.

НО ЭТО ЕЩЕ НЕ ВСЕ!

Только в 1991 г. в рамках "Библиотеки PC Magazine" выходит более 15 названий книг. Ознакомьтесь с их списком — правда, на английском языке — и пришлите нам ваши пожелания относительно издания книг из этой серии. Ваши письма помогут нам наиболее полно удовлетворить потребность советских пользователей персональных компьютеров в информации. Чем больше "голосов" соберет та или иная книга, тем больше шансов на то, что вы сможете читать ее на русском языке.

1. *Frank J. Derfler Jr.* Guide To Connectivity.
2. *Stephen Cobb.* Guide To 1-2-3 Release 2.3.
3. *Jeff Prosis.* DOS 5 Techniques And Utilities.
4. *Dale Lewallen.* PC Computing Guide To Excel 3.0.
5. *Les Freed/Frank Derfler.* Guide To Using NetWare.
6. *Dvorak/Hummel.* Dvorak's Dirty PC Tricks.
7. *Ed Jones.* Guide To Word 2.0 For Windows.
8. *Paul Bonner.* PC Computing Customizing Windows 3.1.
9. *Preston Gralla.* PC Computing Guide To Shareware.
10. *Bill Howard.* Guide To Notebook And Laptop Computers.
11. *The LeBlond Group.* Guide To QuattroPro 3.0.
12. PC Learning Labs Teaches WordPerfect 5.1.
13. *John C. Dvorak.* Buyer's Guide: PC's, Printers, And Monitors.
14. *E. Michael Lunsford.* Guide To 1-2-3 For Windows
15. PC Learning Labs Teaches 1-2-3 Version 2.3
16. *Karen Acerson.* Guide To WordPerfect For Windows

Десять счастливых, первыми приславшие письма с названиями книг из этого списка, которые будут приняты к изданию, получают по бесплатному экземпляру!

Адрес редакции
117607 Москва, ул. Раменки, д. 12,
редакция PC Magazine/USSR.
Телефон (095) 931 0003

К ЧИТАТЕЛЮ!

1. Как вы относитесь к несанкционированному копированию программ, т.е. использованию программ, которые официально являются предметом продажи, но установлены на компьютере без приобретения — путем списывания чужой копии?

- негативно (несанкционированное копирование программ — нарушение если не закона, то профессиональной морали);
- в принципе несанкционированное копирование программ нежелательно, но в нынешних условиях оно допустимо в ограниченных масштабах;
- безразлично (вас не интересует происхождение программы — легальная или нелегальная копия);
- позитивно (программы могут и должны копироваться свободно);
- затрудняюсь ответить.

2. Для каких категорий специалистов, по вашему мнению, введение правовой охраны программ, запрещающей использование несанкционированных копий программного обеспечения и побуждающей к активному развитию рыночных отношений в отрасли, является полезным и выгодным?

- программисты;
- предприниматели (руководители программистских фирм);
- специалисты по сбыту программ (дилеры, дистрибьютеры);
- лица, предоставляющие сопутствующие услуги (сопровождение, обучение и т.д.);
- пользователи программного обеспечения;
- юристы и специалисты по правовой охране программ.

3. Как вы оцениваете последствия введения правовой охраны программ (формального юридического запрета на использование несанкционированных копий) для нелегального копирования?

- объем несанкционированных копий, возможно, даже увеличится (в связи с независимым увеличением спроса по мере роста парка компьютеров и в связи с устойчивой традицией в стране к свободному копированию программ и отсутствием уважительного отношения к законам);
- объем несанкционированного копирования вообще не будет зависеть от наличия или отсутствия законодательства по правовой охране программ (отношение к свободному копированию определяется не юридическими факторами: цены, уровень сопровождения, профессиональная мораль и т.д.);

- объем нелегального копирования существенно сократится при введении в действие законодательства о правовой охране программ.

4. Каковы, на ваш взгляд, экономические последствия введения правовой охраны программ в ближайшем будущем?

- заметно сузятся возможности пользователей, лишенных права использовать программы по своему усмотрению;
- усложнится жизнь программистов, доступ которых к легальным копиям последних программных продуктов будет ограничен;
- руководителям будет труднее, поскольку придется тратить заметные средства на закупку программного обеспечения;
- увеличится объем производства советских программ;
- увеличится объем продаж западных программ;
- увеличится экспорт советских программ за рубеж;
- улучшится качество сервиса;
- улучшится качество использования программ благодаря наличию фирменной документации и гарантированному сопровождению.

5. Знали ли вы что-нибудь о введении (или перспективах введения) правовой охраны программ в нашей стране?

- да;
- нет;
- затрудняюсь ответить.

6. Допустимо ли, на ваш взгляд, копирование одного легально приобретенного экземпляра программы на другие компьютеры, установленные в той же организации?

- не допустимо;
- допустимо с оговорками;
- допустимо;
- затрудняюсь ответить.

7. Допустимо ли приобретение программных продуктов у "пиратов", распространяющих чужое программное обеспечение без договоренности с обладателями прав и, как правило, дешевле с целью получения выгод?

- не допустимо;
- допустимо с оговорками;
- допустимо;
- затрудняюсь ответить.

Отметьте, пожалуйста, ваши ответы крестами в квадратиках и вышлите анкету в адрес редакции.

PROC NORM.BAT

ПОЛНЫЙ ЛИСТИНГ

```
@ECHO OFF
REM *** Работает PROCNORM.BAT ***
STRIPZIP*.ZIP
REM STRIPZIP удаляет комментарии из ZIP-файлов.
REM Если в STRIPZIP нет необходимости, удалите предыдущую строку.
IF EXISTINTRO.TXTERASEINTRO.TXT
IF EXIST READJst.EXE ERASE READJst.EXE
IF EXIST DO_READ.ME ERASE DO_READ.ME
IF EXIST READJVIE.750 ERASE READJ/IE.750
IF EXIST READ_ME.512 ERASE READ_ME.512
IF EXIST HAPPY.BBS ERASE HAPPY.BBS
IF EXISTHAPPY.EXEERASEHAPPY.EXE
IF EXIST READ_ME.PLS ERASE READ_ME.PLS
IF EXISTUGLY.COMERASEUGLY.COM
IF EXIST GOOD.COM ERASE GOOD.COM
IF EXIST BAD.COM ERASE BAD.COM
IF EX1STWELCOME.TXTERASEWELCOME.TXT
IF EXISTHOWDY.TXTERASEHOWDY.TXT
IF EX1STHOWDY.EXEERASEHOWDY.EXE
IF EXIST ALLREAD.ME ERASE ALLREAD.ME
IF EXIST READ-PLS.EXE ERASE READ-PLS.EXE
REM Поиск всех известных вирусов в основном каталоге
REM Можете использовать любую программу поиска вирусов.
SCAN *.* /a /pomet /x /t
REM Если SCAN вызвал ошибку 1 или 2 уровня,
REM на экран посылается сообщение
IF ERROR LEVEL 1 ECHO Внимание!!! SCAN.EXE обнаружил вирус!!!!
IF ERRORLEVEL 2 ECHO Внимание!!! SCAN.EXE аварийное завершение...!
IF ERRORLEVEL 2 ECHO Продолжайте с осторожностью!!!!
IF ERRORLEVEL 0 GOTO QUITOK
REM Замените символы "G в следующей строке на Ctrl-G
ECHO -G"G"G"G"G"G"G"G"G"G
ECHO Пожалуйста, нажмите Ctrl-Break (вызов прерывания) и разберитесь с вирусом,
ECHO file, затем удалите все подкаталоги TEMPZIP. $$$,
ECHO а также их содержимое, и начинайте снова.
PAUSE
ECHO Сказано, нажмите Ctrl-Break!
REM "C в следующей строке - это Ctrl-C. Имейте в виду, что
REM следующая строка работает только в DOS 3.3 или выше.
ECHO V PAUSE > NUL
:QUITOK
```

Рис.2: PROCNORM. обеспечивает обработку нормальных файлов — в данном случае она проверяет наличие вирусов и уничтожает некоторые файлы-объявления.

PROCESS.BAT — работает предельно просто. Сначала он вызывает PROCNORM.BAT, содержащий все необходимые команды для обработки файлов. Затем для каждого .ZIP-файла текущего каталога вызывается PROCZIP.BAT, который создает временный каталог и размещает там распакованные .ZIP-файлы. Все три BAT-файла должны находиться в каталоге, путь на который указан в AUTOEXEC.BAT.

После того, как PROCZIP завершил свою работу, он снова вызывает файл PROCESS. Как и в самом начале, PROCESS вызывает PROCNORM, который производит проверку на наличие вирусов и удаляет ненужные файлы. Затем PROCESS вызывает PROCZIP для каждого из файлов временного каталога. После завершения всего цикла оригинальный .ZIP-файл уничтожается и создается заново с упаковкой в него содержимого временного каталога. Учтите, что любые структуры каталогов, запи-

санные в .ZIP-файле, после его обработки оказываются утерянными. Перед завершением работы командный файл возвращает вас в исходный каталог и уничтожает временный, к тому времени уже пустой каталог.

НОВОСТИ

Базовая плата
фирмы Orchid

Не успела фирма AMD (Advanced Micro Devices) объявить о выпуске 40-МГц процессора AM386, как сразу несколько компаний приступили к разработке новой продукции, основанной на применении этого кристалла.

Фирма Chips and Technologies предложила 40-МГц версию своего процессора PEAK/DM 386, компания Cube Computer добавила к выпускаемым ею системам се-

PC Magazine, June 11, 1991, p. 62

PROCZIP.BAT
ПОЛНЫЙ ЛИСТИНГ

```
@ECHO OFF
REM *** Работает PROCZIP.BAT ***
MD TEMPZIP. $$$
CD TEMPZIP. $$$
PKUNZIP ..\&1
CALL PROCESS.BAT
REM Если у вас DOS старше версии
REM 3.3, замените "CALL" в строке
REM выше этой, на "COMMAND /C"
DEL ..\&1
PLZIP -m ..\&1 *.*
CD ..
RD TEMPZIP. $$$
```

Рис.3: PROCZIP раскрывает содержимое .ZIP-файла и обрабатывает упакованные файлы, рекурсивно вызывая файл PROCESS.BAT (рис. 1).

Важно иметь в виду, что обработка .ZIP-файла ведется в подкаталоге того каталога, где сам он и размещен. Предположим, вы начали работать в каталоге C:\INCOMING. В процессе обработки первого .ZIP-файла происходит перемещение в каталог C:\INCOMING\TEMPZIP. \$\$\$. Если здесь также встретится .ZIP-файл, обработка его содержимого будет производиться в каталоге C:\INCOMING\TEMPZIP. \$\$\$\TEMPZIP. \$\$\$. Единственное ограничение на количество вложенных .ZIP-файлов налагает максимальная длина пути поиска.

Поскольку весь процесс обработки сосредоточен в файле PROCNORM.BAT, для настройки вам будет достаточно изменить только его содержимое, избегая смены текущего каталога.

Все командные файлы используют команду CALL, которая впервые появилась в DOS 3.3. Если вы работаете с более старой версией DOS, замените каждую команду CALL на COMMAND /C или % COMSPEC% /C.

рии Plus еще и систему 386/40, а фирма Sugh начала работы по созданию соответствующего сопроцессора.

Компания Orchid Technology также включилась в "гонку за лидером", предложив свою новую базовую плату Privilege Gold 386-40C, построенную на кристалле фирмы AMD. Плата содержит 128-Кбайт кэш-память, которую можно нарастить до 256 Кбайт, восемь 16-разрядных гнезд расширения и гнездо для подключения сопроцессора. На плате, которая, кстати, имеет стандартные для AT размеры, установлены диагностические светодиоды.

Кристина Кордова, Нат Зельник



112-34-11, 112-12-44, 112-33-24
ФАКС: 112-13-80

115446, Москва, Коломенский проезд, 1/1

«СЕРВИС ВАНГ» —

ЭТО ГАРАНТИЯ ПОЛНОГО УДОВЛЕТВОРЕНИЯ ЗАКАЗЧИКА!
ЭТО ФОКУСИРОВАНИЕ ИНТЕРЕСОВ НА ПОТРЕБНОСТЯХ ЗАКАЗЧИКА!
ЭТО ЗАЩИТА КАПИТАЛОВЛОЖЕНИЙ ЗАКАЗЧИКА!

«СЕРВИС ВАНГ» —

ЭТО УСЛУГИ ПО:

обслуживанию аппаратной части вашей вычислительной техники;
обслуживанию и поддержке программного обеспечения;
переводу программного обеспечения,
а также
консультации;
обучение пользователей

**РАБОТАТЬ НА КОМПЬЮТЕРАХ, КОТОРЫЕ ОБСЛУЖИВАЕТ НАШ «СЕРВИС ВАНГ»,
НАСТОЛЬКО ЖЕ ПРОСТО, НАСКОЛЬКО ПРОСТО ВКЛЮЧИТЬ СВЕТ!**

COMTEK EXPOSITIONS, INC.:

**YOUR
GATEWAY
TO THE
SOVIET
UNION.**



Comtek '92

The 3rd Annual
Computer Technology
Exposition and
Conference.

April 6-10, 1992
Moscow



CEM '91

The Only Dedicated
Consumer Electronics
Show in the Soviet
Union.

Sept. 23-27, 1991
Moscow



Made In USA

A Unique Event for
U. S. Companies
who Export to the
Soviet Union.

June 15-19, 1992
Moscow

COMTEK
EXPOSITIONS, INC.

134 Old Ridgefield Rd. Tel: 203-834-1122
Wilton, Connecticut Fax: 203-762-0773
USA 06897

©1991 Comtek
Expositions, Inc.



ОПЕРАЦИОННАЯ СИСТЕМА NETWARE ФИРМЫ NOVELL — ЭТО ГАРАНТИЯ НАДЕЖНОСТИ СОЕДИНЕНИЯ ВАШИХ КОМПЬЮТЕРОВ

Фирма Novell — ведущий в мире разработчик программного обеспечения для сетевых серверов. Netware, операционная система для серверов фирмы Novell, позволит объединить все имеющиеся у вас

компьютеры в локальные или территориальные (в масштабах вашего предприятия) сети — независимо от операционной системы или типа компьютеров. Netware версии 2.2 разработана специально для нужд малых предприятий и рабочих групп, например на уровне отдела. Netware версии 3.11 служит основой для создания сетей в масштабе предприятия. Она обеспечивает локальное или территориальное соединение настольных компьютеров руководителей и их подчиненных и является идеальной базой для разработки ваших собственных сетевых приложений.

Адреса наших дилеров в Советском Союзе вы можете узнать у следующих дистрибьютеров фирмы Novell:

СП «Ланит», 113149 Москва,

Сивашская ул., 4-2,

д-р Георгий В. Генс;

тел. (095)119-88-04;

факс (095)119-09-09.

СП «Унитех», 119899 Москва,

Ленинские горы, МГУ, Григорий И.

Элькин; тел. (095)939-50-69;

факс (095)938-21-36.

Фирма «Би-Джи-Эс Москоу»,

Москва, Варшавское шоссе, 21-й

км, гостиница «Русская» мотеля

«Солнечный», 1920, г-н Томас

Богданович; тел. (095)247-16-02;

факс (095)247-10-52.

SOFT-TRONIC

creative computer technology

191180 Ленинград, Наб.

р. Фонтанки, 88.

Мирослав Гржибовски

Директор представительства;

тел. (812)315-92-76, 311-23-25

факс (812)311-01-08,

телекс 121184 SOFT SU

 **NOVELL®**